High-Performance Secure MCUs for Industrial IoT and Metering Applications, Up to 2MB Flash, 512K Dual-Port SRAM, Advanced Crypto and Security Features

PIC32CXMTG

Product Page Links

Description

The Microchip PIC32CXMTG series is a family of system-on-chip solutions well suited for smart energy applications such as communication modules and residential split meter systems. Built around a 32-bit Arm® Cortex®-M4F RISC processor, the devices run at a maximum speed of 200 MHz and embed up to 2 Mbytes of embedded Flash, up to 512 Kbytes of SRAM and 16 Kb of Instruction and 8 Kb of Data Cache/Tightly Coupled Memory (TCM) on-chip.

A QSPI interface with eXecute-in-Place (XiP) coupled with on-the-fly AES128 encryption bridge futureproofs the devices and ensures platform scalability in case of memory extension needs or changing system requirements.

A Secure Boot and secure key storage coupled with advanced cryptographic/Hash hardware accelerators, such as ECC, RSA, AES128/256, and SHA256/512, support the different encryption and authentication protocols.

The series also features a segmented LCD controller, anti-tampers, Floating Point Unit (FPU), MPU, FLEXCOM peripherals supporting I^2C, SPI, UART/USART interfaces, three PWMs for pulse output functions, 12-channel general-purpose 32-bit timers, 12-bit ADC, analog comparators and a battery backed-up RTC running as low as 1 µA.

PIC32CXMTG devices are available in EP-TQFP128-pin exposed pin and VQFN64 packages.

See the table Configuration Summary for device configuration versus packages.

Features

- Core
  - Arm Cortex-M4F running at up to 200 MHz
  - 16 Kbytes ITCM/I-Cache, 8 Kbytes DTCM/D-Cache
  - Memory Protection Unit
  - IEEE® 754-compliant, single-precision Floating Point Unit (FPU), DSP instruction, Thumb®-2 instruction set
  - Up to 2 Mbytes of embedded dual plane/dual boot Flash
    - Built-in ECC, Read-While-Write (RWW) support and Suspend/Resume of write/erase operations
    - Endurance of 100K Write/Erase cycles (1M in qualification)
  - Up to 512 Kbytes of embedded SRAM
- Security and Cryptography
  - Secure Boot
  - Secure key storage and transfer to Advanced Encryption Standard (AES) and AES Bridge (AESB) without processor intervention
  - One-time programmable block for software monotonic counter
  - Security bit to disable debug port access
  - True Random Number Generator (TRNG)
  - Deterministic Random Number Generator (DRNG) support as required by NIST 800-90B

© 2026 Microchip Technology Inc. and its subsidiaries

  - AES: 256-bit, 192-bit, 128-bit key algorithm, compliant with FIPS PUB-197 specifications, ARIA
  - AES128 on-the-fly encryption/decryption bridge for QSPI communications
  - Secure Hash Algorithm (SHA) supporting SHA1, SHA224, SHA256, SHA384, and SHA512 compliant with FIPS PUB-197 specifications, and associated HMAC
  - Classical Public Key Cryptographic Controller (CPKCC) supporting:
    - RSA up to 7168 bits, elliptic curve, DSA, DRNG
  - 768 bits of General Purpose Backup Registers (GPBR) can be defined as write-once
  - Automatic and immediate keys erasing upon detection of Flash erase signal assertion
    - Erase of AES keys, QSPI scrambling keys and GPBR
  - Anti-tamper with time-stamping, configurable and immediate actions upon detection
    - Erase of AES, AESB keys, QSPI scrambling keys and General Purpose Backup registers
  - Integrity Check Module (ICM) based on SHA256, autonomous
- System Controller
  - Power supply
    - Embedded core and LCD voltage regulator for single-supply operation
    - Supply monitors
    - Ultra-low-power Backup mode
  - Clock
    - Programmable PLLs and oscillator clock sources
    - Clock failure detection
    - CPU frequency monitor
    - Main crystal oscillator failure detection
    - 32.768 kHz crystal oscillator frequency monitor and failure detection
    - Independent Dual Watchdog Timer (DWDT)
    - 12 MHz internal RC oscillator
  - Ultra-low power Real-Time Clock (RTC) with Gregorian, Persian calendars, and UTC mode, calibration circuitry to compensate the 32.768 kHz crystal drift with observability of compensated clock on output pin
  - Peripheral and memory-to-memory DMA channels
- Shared Peripherals
  - Segmented LCD Controller (SLCDC)
    - Static, ½, ⅓, and ¼ bias
    - Software-selectable LCD output voltage (contrast)
    - Can be used in Backup mode
- Communication Interfaces
  - FLEXCOMs supporting UART/USART, SPI, TWI/I^2C with FIFOs
  - Metrology-specific multi-channel SPI with legacy support
  - Metrology-specific two-wire UARTs supporting optical transceiver providing an electrically-isolated serial communication with hand-held equipment, such as calibrators, compliant with ANSI-C12.18 or IEC62056-21 norms
- Debugger Development Support
  - Arm CoreSight™ JTAG-DP or SW-DP debug port
  - Wake-up from debug request

© 2026 Microchip Technology Inc. and its subsidiaries

  - Unlimited software breakpoints
  - IEEE 1149.2-compatible (JTAG) boundary scan
- Analog Conversion Block
  - 12-bit ADC module
    - 1 Msps S/H ADC
    - Up to 8 channels
  - Two analog comparators
  - Temperature sensor
  - Closed-Circuit Voltage (CCV) measurement on battery voltage input rail (VBAT)
- Timers/Output Compare/Input Capture
  - 12-channel 32-bit Timer Counters (TC) with Capture, Waveform, Compare and basic PWM modes. Quadrature decoder logic and 2-bit Gray up/down counter
  - 3-channel, 24-bit PWM with complementary outputs, dead-time generator, and external trigger modes
- I/O
  - I/O lines with slew rate control to ease PCB design and EMC compliance, external interrupt capability (edge or level sensitivity), Schmitt trigger, internal pull-up/pull-down, debouncing, glitch filtering, and on-die series resistor termination
- Software and Tools Support
  - Microchip Smart Energy Framework Software Examples
  - Microchip G3/PRIME/IEEE PLC Stacks
  - Microchip MPLAB® X and MPLAB Harmony
  - IAR Embedded WorkBench, Keil® µVision®
- Packages
  - 64-lead VQFN, 7.5 x 7.5 x 0.9 mm, pitch 0.4 mm
  - 128-pin EP-TQFP, 14 x 14 x 1.0 mm, pitch 0.4 mm
- Operating Conditions
  - 2.25V to 3.6V, -40°C to 85°C

  Note: Restrictions on range may apply. Refer to the section Electrical Characteristics.

© 2026 Microchip Technology Inc. and its subsidiaries

Table of Contents

Description 1

Features 1

1. Block Diagram 12

2. Configuration Summary 13

3. Package and Pinout 14
   - 3.1. Pinout Topology 14
   - 3.2. Pinout and Multiplexing 14

4. Power Supply and Power Control 24
   - 4.1. Power Supplies 24
   - 4.2. System State 28
   - 4.3. Active Mode 28
   - 4.4. Low-Power Modes 29

5. Input/Output Lines 34
   - 5.1. General Purpose I/O Lines 34
   - 5.2. TST Pin 34
   - 5.3. NRST Pin 34
   - 5.4. Anti-Tamper (TMPx) Pins 34
   - 5.5. RTCOUTx Pins 34
   - 5.6. Shutdown (SHDN) Pin 34
   - 5.7. Force Wake-up (FWUP) Pin 35
   - 5.8. Flash Memory Hardware ERASE Signal 35

6. Core and Interconnect 37
   - 6.1. Cortex-M4 Processor 37

7. Product Mapping and Peripheral Access 40

8. Memories 44
   - 8.1. Embedded Memories 44
   - 8.2. External Memories 47

9. Safety and Security Features 48
   - 9.1. Design for Safety 48
   - 9.2. Design for Security 48

10. Real-Time Event Management 50
    - 10.1. Real-Time Event Mapping List 50

11. Peripherals 51
    - 11.1. Peripheral Identifiers 51
    - 11.2. Peripheral DMA Controller (PDC) 54

12. Cortex-M4 Processor (Arm) 56
    - 12.1. Description 56

© 2026 Microchip Technology Inc. and its subsidiaries

12.2. Reference Documents....56
12.3. Embedded Characteristics....56

13. Flash Programming, Debug and Test Features....58
13.1. Flash Programming Sequence....58
13.2. Description....58

14. ROM Code and Boot Strategies....66
14.1. Description....66
14.2. Functional Description....69
14.3. Secure Boot Mode....73
14.4. SAM-BA Monitors....77
14.5. FFPI Monitor....81

15. Supply Controller (SUPC)....91
15.1. Description....91
15.2. Embedded Characteristics....91
15.3. Block Diagram....92
15.4. Functional Description....92
15.5. Register Summary....104

16. Reset Controller (RSTC)....127
16.1. Description....127
16.2. Embedded Characteristics....127
16.3. Block Diagram....128
16.4. Functional Description....128
16.5. Register Summary....142

17. System Controller Write Protection (SYSCWP)....148
17.1. Functional Description....148
17.2. Register Summary....150

18. Dual Watchdog Timer (DWDT)....153
18.1. Description....153
18.2. Embedded Characteristics....153
18.3. Block Diagram....154
18.4. Functional Description....154
18.5. Register Summary....157

19. Clock Generator....186
19.1. Description....186
19.2. Embedded Characteristics....186
19.3. Block Diagram....187
19.4. Slow Clock....187
19.5. Main Clock....188
19.6. PLL Controls....191

20. Power Management Controller (PMC)....195
20.1. Description....195
20.2. Embedded Characteristics....195
20.3. Block Diagram....196

© 2026 Microchip Technology Inc. and its subsidiaries

20.4. Processor Clock Controller 196
20.5. Free-running Processor Clock 197
20.6. Peripheral and Generic Clock Controller 197
20.7. SysTick Clock 198
20.8. Programmable Clock Output Controller 198
20.9. Ultra Low Power Modes and Fast Start-up 198
20.10. Asynchronous Partial Wake-Up 200
20.11. Start-up from Embedded Flash 202
20.12. Main Crystal Oscillator Failure Detection 202
20.13. 32.768 kHz Crystal Oscillator Frequency Monitor 203
20.14. MCK0 Frequency Monitor 203
20.15. Recommended Programming Sequence 204
20.16. Clock Switching Details 205
20.17. Register Write Protection 206
20.18. Register Summary 208

21. Parallel Input/Output Controller (PIO) 260
21.1. Description 260
21.2. Embedded Characteristics 260
21.3. Block Diagram 261
21.4. Product Dependencies 262
21.5. Functional Description 262
21.6. I/O Lines Programming Example 273
21.7. Register Summary 276

22. Real-Time Clock (RTC) 321
22.1. Description 321
22.2. Embedded Characteristics 321
22.3. Block Diagram 322
22.4. Product Dependencies 322
22.5. Functional Description 322
22.6. Register Summary 334

23. Real-time Timer (RTT) 368
23.1. Description 368
23.2. Embedded Characteristics 368
23.3. Block Diagram 368
23.4. Functional Description 368
23.5. Register Summary 372

24. General Purpose Backup Registers (GPBR) 379
24.1. Description 379
24.2. Embedded Characteristics 379
24.3. Register Summary 380

25. Special Function Registers (SFR) 383
25.1. Description 383
25.2. Embedded Characteristics 383
25.3. Functional Description 383
25.4. Register Summary 384

© 2026 Microchip Technology Inc. and its subsidiaries

26. Special Function Registers Backup (SFRBU)....391
26.1. Description....391
26.2. Embedded Characteristics....391
26.3. Functional Description....391
26.4. Register Summary....392

27. Bus Matrix (MATRIX)....398
27.1. Description....398
27.2. Embedded Characteristics....401
27.3. Memory Mapping....401
27.4. Special Bus Granting Techniques....401
27.5. No Default Host....402
27.6. Last Access Host....402
27.7. Fixed Default Host....402
27.8. Arbitration....403
27.9. Register Write Protection....405
27.10. Privilege Protection of AHB and APB....406
27.11. Register Summary....414

28. Chip Identifier (CHIPID)....439
28.1. Description....439
28.2. Embedded Characteristics....439
28.3. Register Summary....440

29. Secure Embedded Flash Controller (SEFC)....444
29.1. Description....444
29.2. Embedded Characteristics....444
29.3. Product Dependencies....444
29.4. Functional Description....445
29.5. Register Summary....465

30. Memory to Memory (MEM2MEM)....479
30.1. Description....479
30.2. Embedded Characteristics....479
30.3. Block Diagram....479
30.4. Product Dependencies....479
30.5. Functional Description....479
30.6. Register Write Protection....480
30.7. Register Summary....481

31. Peripheral DMA Controller (PDC)....489
31.1. Description....489
31.2. Embedded Characteristics....489
31.3. Block Diagram....490
31.4. Functional Description....490
31.5. Register Summary....493

32. Cortex-M Cache Controller (CMCC)....507
32.1. Description....507
32.2. Embedded Characteristics....507

© 2026 Microchip Technology Inc. and its subsidiaries

32.3. Block Diagram 508
32.4. Functional Description 508
32.5. Register Summary 510

33. Flexible Serial Communication Controller (FLEXCOM) 524
33.1. Description 524
33.2. Embedded Characteristics 525
33.3. Block Diagram 528
33.4. I/O Lines Description 528
33.5. Product Dependencies 528
33.6. Register Accesses 529
33.7. USART Functional Description 529
33.8. SPI Functional Description 579
33.9. TWI Functional Description 603
33.10. Register Summary 654

34. Quad Serial Peripheral Interface (QSPI) 798
34.1. Description 798
34.2. Embedded Characteristics 798
34.3. Block Diagram 799
34.4. Signal Description 799
34.5. Product Dependencies 799
34.6. Functional Description 800
34.7. Register Summary 826

35. Segment LCD Controller (SLCDC) 858
35.1. Description 858
35.2. Embedded Characteristics 858
35.3. Block Diagram 859
35.4. I/O Lines Description 860
35.5. Product Dependencies 860
35.6. Functional Description 860
35.7. Waveform Specifications 873
35.8. Register Summary 874

36. Timer Counter (TC) 890
36.1. Description 890
36.2. Embedded Characteristics 890
36.3. Block Diagram 891
36.4. Pin List 892
36.5. Product Dependencies 892
36.6. Functional Description 892
36.7. Register Summary 917

37. Analog-to-Digital Converter (ADC) Controller 963
37.1. Description 963
37.2. Embedded Characteristics 963
37.3. Block Diagram 964
37.4. Product Dependencies 964
37.5. Functional Description 966

© 2026 Microchip Technology Inc. and its subsidiaries

37.6. Register Summary 984

38. Analog Comparator Controller (ACC) 1026
38.1. Description 1026
38.2. Embedded Characteristics 1026
38.3. Block Diagram 1026
38.4. Signal Description 1026
38.5. Product Dependencies 1027
38.6. Functional Description 1027
38.7. Register Summary 1029

39. Advanced Encryption Standard (AES) 1040
39.1. Description 1040
39.2. Embedded Characteristics 1040
39.3. Product Dependencies 1041
39.4. Functional Description 1041
39.5. Register Summary 1065

40. Advanced Encryption Standard Bridge (AESB) 1095
40.1. Description 1095
40.2. Embedded Characteristics 1095
40.3. Product Dependencies 1095
40.4. Functional Description 1096
40.5. Register Summary 1102

41. Secure Hash Algorithm (SHA) 1121
41.1. Description 1121
41.2. Embedded Characteristics 1121
41.3. Product Dependencies 1121
41.4. Functional Description 1122
41.5. Register Summary 1132

42. True Random Number Generator (TRNG) 1152
42.1. Description 1152
42.2. Embedded Characteristics 1152
42.3. Block Diagram 1152
42.4. Product Dependencies 1152
42.5. Functional Description 1153
42.6. Register Summary 1156

43. Integrity Check Monitor (ICM) 1169
43.1. Description 1169
43.2. Embedded Characteristics 1170
43.3. Block Diagram 1171
43.4. Product Dependencies 1171
43.5. Functional Description 1172
43.6. Register Summary 1185

44. Classical Public Key Cryptography Controller (CPKCC) 1204
44.1. Description 1204
44.2. Product Dependencies 1204

© 2026 Microchip Technology Inc. and its subsidiaries

44.3. Functional Description....1204

45. Multi Channel Serial Peripheral Interface (MCSPI)....1205
45.1. Description....1205
45.2. Embedded Characteristics....1205
45.3. Block Diagram....1206
45.4. Application Block Diagram....1206
45.5. Signal Description....1207
45.6. Product Dependencies....1207
45.7. Functional Description....1207
45.8. Register Summary....1233

46. Pulse Width Modulation Controller (PWM)....1267
46.1. Description....1267
46.2. Embedded Characteristics....1267
46.3. Block Diagram....1269
46.4. I/O Lines Description....1269
46.5. Product Dependencies....1270
46.6. Functional Description....1271
46.7. Register Summary....1313

47. Universal Asynchronous Receiver Transmitter (UART)....1379
47.1. Description....1379
47.2. Embedded Characteristics....1379
47.3. Block Diagram....1380
47.4. Product Dependencies....1380
47.5. Functional Description....1380
47.6. Register Summary....1391

48. Electrical Characteristics....1413
48.1. Electrical Parameters Usage....1413
48.2. Absolute Maximum Ratings....1414
48.3. Recommended Operating Conditions....1415
48.4. Recommended Power Supply Sequencing for Externally Supplied VDDCORE/VDDPLL....1417
48.5. I/O Characteristics....1418
48.6. Digital Peripherals Characteristics....1420
48.7. Embedded Analog Peripherals Characteristics....1428
48.8. Embedded Flash Characteristics....1442
48.9. Power Supply Current Consumption....1443

49. Mechanical Characteristics....1450
49.1. 64-lead VQFN Mechanical Characteristics....1451
49.2. 128-pin EP-TQFP Mechanical Characteristics....1454

50. Marking....1457

51. Ordering Information....1458

52. Revision History....1459

53. Product Identification System....1460

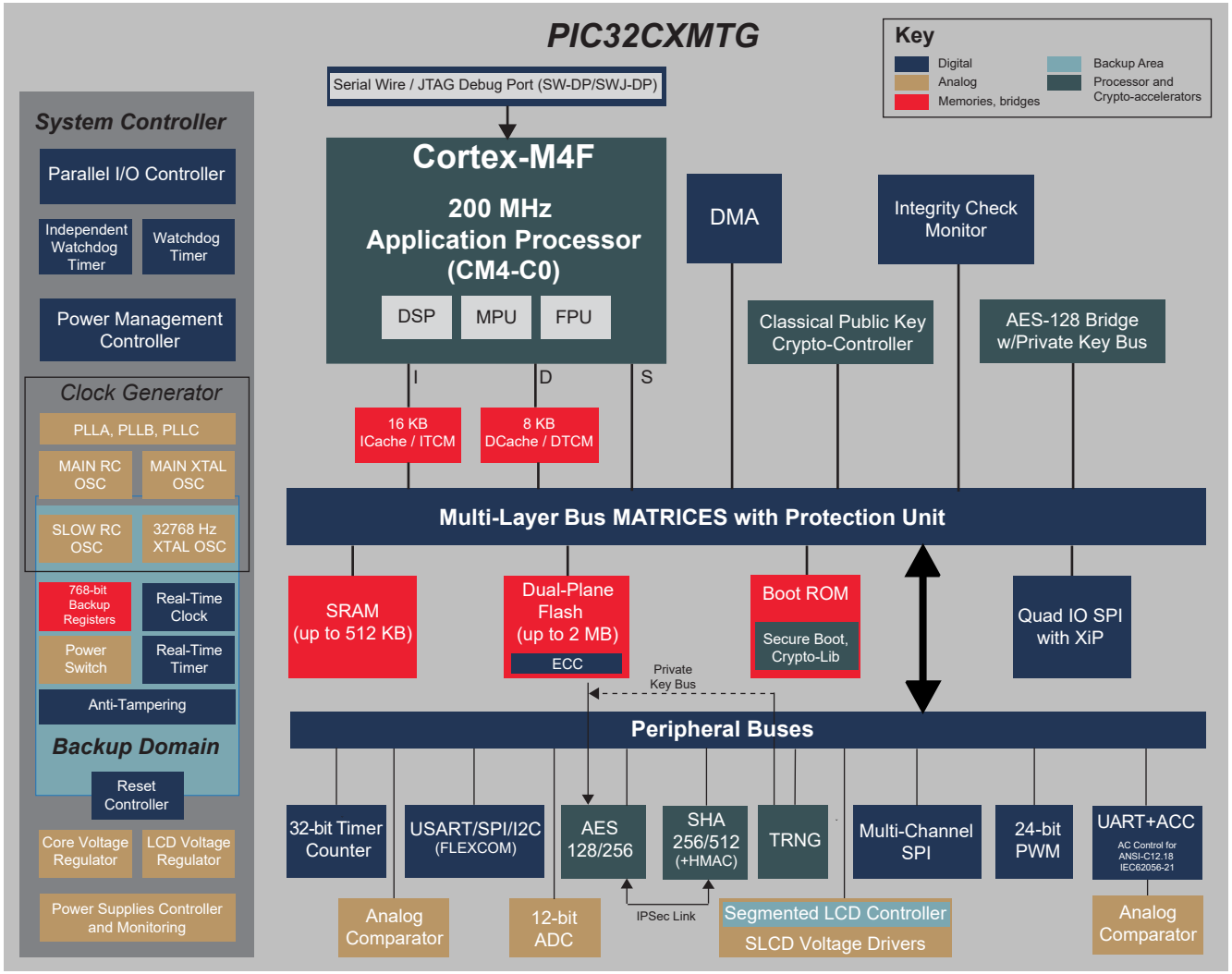
© 2026 Microchip Technology Inc. and its subsidiaries

Microchip Information....1462
Trademarks....1462
Legal Notice....1462
Microchip Devices Code Protection Feature....1462
Product Page Links....1463

© 2026 Microchip Technology Inc. and its subsidiaries

1. Block Diagram

Figure 1-1. PIC32CXMTG Block Diagram



© 2026 Microchip Technology Inc. and its subsidiaries

2. Configuration Summary

The devices differ in memory sizes, packages and features. The following table summarizes the different configurations.

Table 2-1. PIC32CXMTG Family Features

Device	Flash Memory Size (Kb)/ **Nb Subsystem x Size/2**	SRAM Memory (Kb)	TCM and Cache Memory (Kb)	FPU/MPU	Pins	Package	Digital Peripherals						Analog Peripherals		
							FLEXCOM (SPI, UART, TWI)	Quad I/O Serial Peripheral Interface	Timer Counter Channels	PWM Channels	DMA Channels	I/O Pins	Segment LCD Controller **(COMs+Seg)**	12-bit ADC (Channels) **External / Internal**	Analog Comparators
PIC32CX2051MTG64	2048 (2x1024)	512	16 + 8	Yes	64	VQFN	4x4-wire, 3x2-wire	Y	12	3	23	41	–	2 + 3	Yes
PIC32CX2051MTG128					128	EP-TQFP	8x4-wire					102	8x32	5 + 3	
PIC32CX1025MTG64	1024 (2x512)	256			64	VQFN	4x4-wire, 3x2-wire					41	–	2 + 3	
PIC32CX1025MTG128					128	EP-TQFP	8x4-wire					102	8x32	5 + 3	
PIC32CX5112MTG128	512 (2x256)	128			128	EP-TQFP	8x4-wire					102	8x32	5 + 3	

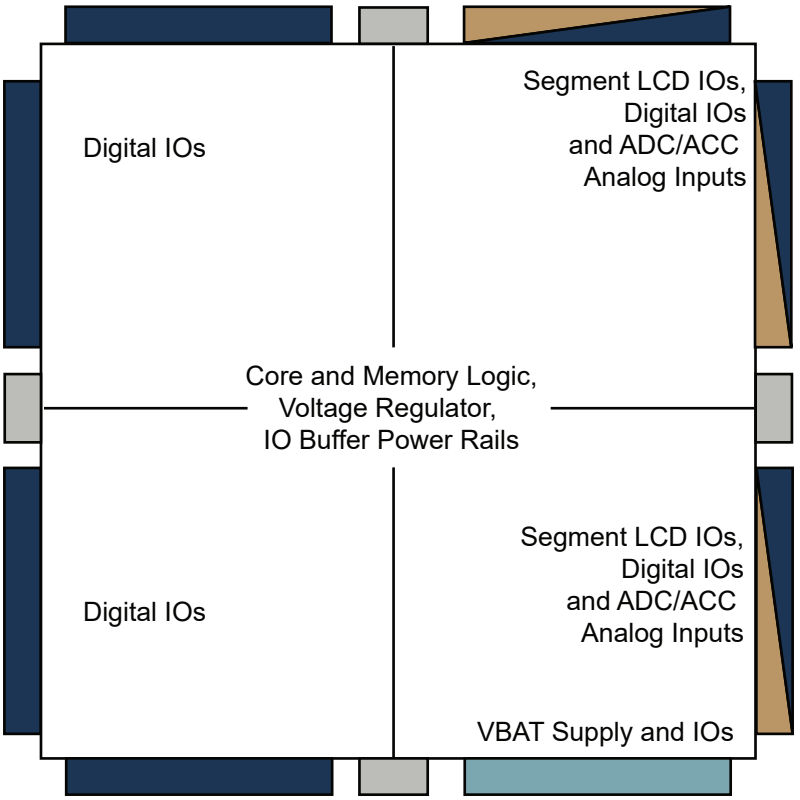
© 2026 Microchip Technology Inc. and its subsidiaries

3. Package and Pinout

In the following figures, the IO and power supply ring topology are shown for typical use cases of a Smart Energy meter and communication modules.

3.1. Pinout Topology

Figure 3-1. PIC32CXMTG Pinout Topology



3.2. Pinout and Multiplexing

3.2.1. Peripheral Signal Multiplexing on I/O Lines

The multiplexing table that follows defines how the I/O lines of the peripherals A, B, C and D are multiplexed on the PIO Controllers. Refer to the column *PIO Peripheral*, sub-column *Func*.

In the *Primary* and *Alternate* columns, the values in the *Dir* sub-column are applicable to the alternate function only if the primary signal is a PIO line (PAx, PBx, PCx, PDx). If only a primary function is available for a given pin, the value in the sub-column *Dir* applies.

The *IO Set* sub-column in the *PIO Peripheral* column gives IO set numbers per PIO line. For a given set of signals for a given peripheral, only IO peripheral lines which are in the same IO set can be used at the same time.

The column *Reset State* indicates the Reset state of the line.

- PIO, Alternate or Peripheral signal name — Indicates whether the PIO line resets in general purpose I/O mode, in Peripheral mode or in Alternate mode. If “PIO” is mentioned, the PIO line is in I/O mode. If a Peripheral or Alternate signal name is mentioned in the “Reset State” column, the PIO line is assigned to this function.

© 2026 Microchip Technology Inc. and its subsidiaries

- I or O — Indicates whether the signal is Input or Output state.
- PU or PD — Indicates whether Pull-up, Pull-down or nothing is enabled.
- ST— Indicates if Schmitt trigger is enabled.
- All PIO lines feature programmable slew rate control and Lock/Freeze configuration.

For details on FLEXCOM0/1/2/3/4, see FLEXCOM Features and the section Flexible Serial Communication Controller (FLEXCOM).

3.2.1.1. Pinout and Multiplexing

Table 3-1. 64-lead VQFN Multiplexing

VQFN64 Pin Number	Power Rail/ Voltage Reference	Primary	Alternate		PIO Peripheral				Reset State	Comments
		Signal	Signal	Dir	Func	Signal	Dir	IO Set	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER	
13	VDD3V3	PA4	–	–	A	FLEXCOM0_IO0	I/O	1,2	PIO, I, PU, ST	
14	VDD3V3	PA5	WKUP5	I	A	FLEXCOM0_IO1	I/O	1,2	PIO, I, PU, ST	
15	VDD3V3	PA6	–	–	A	FLEXCOM0_IO2	I/O	1	PIO, I, PU, ST	
					B	FLEXCOM0_IO4	O	2		
					C	TIOA6	I/O	1		
16	VDD3V3	PA7	–	–	A	FLEXCOM0_IO3	I/O	1,2	PIO, I, PU, ST	
					C	TIOB6	I/O	1		
7	VDD3V3	PA8	–	–	A	FLEXCOM1_IO0	I/O	1,2	PIO, I, PU, ST	
8	VDD3V3	PA9	WKUP6	I	A	FLEXCOM1_IO1	I/O	1,2	PIO, I, PU, ST	
17	VDD3V3	PA12	–	–	A	FLEXCOM2_IO0	I/O	1,2	PIO, I, PU, ST	
18	VDD3V3	PA13	WKUP7	I	A	FLEXCOM2_IO1	I/O	1,2	PIO, I, PU, ST	
34	VDD3V3	PA23	–	–	A	COM2	O	1	PIO, I, PD, ST	
					B	FLEXCOM5_IO1	I/O	5,6		
					D	TCLK0	I	1		
35	VDD3V3	PA24	–	–	A	COM3	O	1	PIO, I, PD, ST	
					B	FLEXCOM5_IO0	I/O	5,6		
					D	TIOB0	I/O	1		
36	VDD3V3	PA25	–	–	A	COM4	O	1	PIO, I, PD, ST	
					B	FLEXCOM6_IO0	I/O	4,5		
					D	TIOA0	I/O	1		
37	VDD3V3	PA29	AD0/ ACC_INP0/ XIN	I	A	SEG0	O	1	PIO, I, PD, ST	Clock input when oscillator is in Bypass mode
					B	FLEXCOM0_IO4	O	1		
					C	FLEXCOM1_IO4	O	1		
					D	FLEXCOM2_IO4	O	1		
39	VDD3V3	PB0	AD3/ ACC_INN0/ XOUT	I/O	A	SEG3	O	1	PIO, I, PD, ST	ADCx/ACCx signals:Pull-down must be disabled in PIO Controller when using these PIO pins as ADC/ACC inputs. XIN/OUT signals:Pull-down must be disabled in PIO Controller when using these PIO pins as Crystal Oscillator XIN/XOUT signals.
					B	FLEXCOM3_IO4	O	1		
					D	TIOB1	I/O	1		
40	VDD3V3	PB2	ERASE	I	A	SEG5	O	1	ERASE, I, PD, ST	
					B	FLEXCOM7_IO2	I/O	1		
					C	FLEXCOM7_IO4	O	2		
					D	TIOA2	I/O	1		
43	VDD3V3	PB3	–	–	A	SEG6	O	1	PIO, I, PD, ST	
					B	FLEXCOM7_IO1	I/O	1,2		
					D	TIOB2	I/O	1		
44	VDD3V3	PB4	–	–	A	SEG7	O	1	PIO, I, PD, ST	
					B	FLEXCOM7_IO0	I/O	1,2		
					D	TCLK2	I	1		
45	VDD3V3	PB9	WKUP9	I	A	SEG12	O	1	PIO, I, PD, ST	
					B	FLEXCOM3_IO0	I/O	3,4		

© 2026 Microchip Technology Inc. and its subsidiaries

Table 3-1. 64-lead VQFN Multiplexing (continued)

VQFN64 Pin Number	Power Rail/ Voltage Reference	Primary	Alternate		PIO Peripheral				Reset State	Comments
		Signal	Signal	Dir	Func	Signal	Dir	IO Set	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER	
46	VDD3V3	PB10	–	–	A	SEG13	O	1	PIO, I, PD, ST	
					B	FLEXCOM3_IO1	I/O	3,4		
					D	TRACESWO	O	1		
47	VDD3V3	PB11	–	–	A	SEG14	O	1	PIO, I, PD, ST	
					B	FLEXCOM3_IO2	I/O	3		
					C	FLEXCOM3_IO4	O	4		
48	VDD3V3	PB12	WKUP10	I	A	SEG15	O	1	PIO, I, PD, ST	
					B	FLEXCOM3_IO3	I/O	3,4		
49	VDD3V3	PB24	–	–	A	SEG27	O	1	PIO, I, PD, ST	
					B	FLEXCOM4_IO0	I/O	1,2		
					C	TIOA5	I/O	2		
50	VDD3V3	PB25	WKUP8	I	A	SEG28	O	1	PIO, I, PD, ST	
					B	FLEXCOM4_IO1	I/O	1,2		
					C	TIOB5	I/O	2		
51	VDD3V3	PB26	–	–	A	SEG29	O	1	PIO, I, PD, ST	
					B	FLEXCOM4_IO2	I/O	1		
					C	FLEXCOM4_IO4	O	2		
52	VDD3V3	PC0	–	–	A	SEG30	O	1	PIO, I, PD, ST	
					B	FLEXCOM4_IO3	I/O	1,2		
					C	TIOA3	I/O	2		
					D	TDI	I	1		
55	VDD3V3	PC2	–	–	A	SWCLK	I	1	SWCLK,ST	
					B	FLEXCOM1_IO0	I/O	3		
56	VDD3V3	PC3	–	–	A	SWDIO	I/O	1	SWDIO,ST	
					B	FLEXCOM1_IO1	I/O	3		
57	VDD3V3	PC5	–	–	A	PCK2	O	1	PIO, I, PU, ST	
					D	TCLK5	I	1,2		
59	VDD3V3	PC10	–	–	A	QIO3	I/O	1	PIO, I, PU, ST	
60	VDD3V3	PC11	–	–	A	QIO2	I/O	1	PIO, I, PU, ST	
					B	FLEXCOM7_IO4	O	3		
62	VDD3V3	PC12	–	–	A	QIO1	I/O	1	PIO, I, PU, ST	
					B	FLEXCOM7_IO1	I/O	3		
63	VDD3V3	PC13	–	–	A	QIO0	I/O	1	PIO, I, PU, ST	
					B	FLEXCOM7_IO0	I/O	3		
64	VDD3V3	PC14	WKUP13	I	A	QCS	O	1	PIO, I, PU, ST	
					B	FLEXCOM7_IO3	I/O	3		
					D	TIOA7	I/O	1		
1	VDD3V3	PC15	–	–	A	QSCK	O	1	PIO, I, PU, ST	
					B	FLEXCOM7_IO2	I/O	3		
					D	TIOB7	I/O	1		
2	VDD3V3	PC18	–	–	A	FLEXCOM6_IO2	I/O	1	PIO, I, PU, ST	
					B	FLEXCOM6_IO4	O	2		
					D	TIOB8	I/O	1		
3	VDD3V3	PC19	–	–	A	FLEXCOM6_IO3	I/O	1,2	PIO, I, PU, ST	
					D	TCLK8	I	1		
4	VDD3V3	PC20	–	–	A	FLEXCOM7_IO1	I/O	4	PIO, I, PU, ST	
5	VDD3V3	PC21	WKUP14	I	A	FLEXCOM7_IO0	I/O	4	PIO, I, PU, ST	
6	VDD3V3	PC22	RTCOUT1	O	A	PCK0	O	1	PIO, I, PD, ST	
10	VDD3V3	PD17	–	–	A	PWML0	O	1	PIO, I, PD, ST	
					C	TIOA11	I/O	1		

© 2026 Microchip Technology Inc. and its subsidiaries

Table 3-1. 64-lead VQFN Multiplexing (continued)

VQFN64 Pin Number	Power Rail/ Voltage Reference	Primary	Alternate		PIO Peripheral				Reset State	Comments
		Signal	Signal	Dir	Func	Signal	Dir	IO Set	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER	
11	VDD3V3	PD18	–	–	A	PWML1	O	1	PIO, I, PD, ST	
					B	–	I/O	1		
					C	TIOB11	I/O	1		
12	VDD3V3	PD19	–	–	A	PWML2	O	1	PIO, I, PD, ST	
					B	–	O	1		
					C	TCLK11	I	1		
22	VDD3V3	NRST	–	–	–	–	–	–	I, PU, ST	
28	VDDBU	FWUP	–	I	–	–	–	–	I,ST	Active low, external pull-up needed
29		JTAGSEL	–	I	–	–	–	–	I,PD	
33		SHDN	–	O	–	–	–	–	O	0: The device is in Backup mode 1: The device is not in Backup mode
32		TST	–	I	–	–	–	–	I,PD	
27		RTCOUT0	–	I	–	–	–	–	O	
26		WKUP2/ TMP2	–	I	–	–	–	–	I,ST	External pull-up needed
31		XOUT32	–	O	–	–	–	–	–	
30		XIN32	–	I	–	–	–	–	–	Clock input when oscillator is in Bypass mode
9	VDD3V3	POWER	–	–	–	–	–	–	–	
25	VBAT		–	–	–	–	–	–	–	
24	VDD3V3		–	–	–	–	–	–	–	
42	VDD3V3		–	–	–	–	–	–	–	
54	VDD3V3		–	–	–	–	–	–	–	
61	VDD3V3		–	–	–	–	–	–	–	
19	VDDCORE		–	–	–	–	–	–	–	
23	VDDCORE		–	–	–	–	–	–	–	
53	VDDCORE		–	–	–	–	–	–	–	
58	VDDCORE		–	–	–	–	–	–	–	
41	VDDPLL		–	–	–	–	–	–	–	
20	VDDOUT		–	–	–	–	–	–	–	
21	VDD3V3		–	–	–	–	–	–	–	
38	VREFP		–	–	–	–	–	–	–	
65	GND		–	–	–	–	–	–	–	Digital ground (exposed pad)

Table 3-2. 128-pin EP-TQFP Multiplexing

TQFP128 Pin Number	Power Rail/ Voltage Reference	Primary	Alternate		PIO Peripheral				Reset State	Comments
		Signal	Signal	Dir	Func	Signal	Dir	IO Set	Signal, Dir, PU, PD, HiZ, ST, FILTER	
16	VDD3V3	PA0	–	–	A	FLEXCOM6_IO0	–	–	–	
17	VDD3V3	PA1	–	–	A	FLEXCOM6_IO1	–	–	–	
27	VDD3V3	PA2	WKUP3/TMP3	I	A	PCK1	O	1	PIO, I, PU, ST	
					B	TIOA0	I/O	2		
28	VDD3V3	PA3	WKUP4/TMP4	I	A	TIOB0	I/O	2	PIO, I, PU, ST	
29	VDD3V3	PA4	–	–	A	FLEXCOM0_IO0	I/O	1,2	PIO, I, PU, ST	
30	VDD3V3	PA5	WKUP5	I	A	FLEXCOM0_IO1	I/O	1,2	PIO, I, PU, ST	
31	VDD3V3	PA6	–	–	A	FLEXCOM0_IO2	I/O	1	PIO, I, PU, ST	
					B	FLEXCOM0_IO4	O	2		
					C	TIOA6	I/O	1		

© 2026 Microchip Technology Inc. and its subsidiaries

Table 3-2. 128-pin EP-TQFP Multiplexing (continued)

TQFP128 Pin Number	Power Rail/ Voltage Reference	Primary	Alternate		PIO Peripheral				Reset State	Comments
		Signal	Signal	Dir	Func	Signal	Dir	IO Set	Signal, Dir, PU, PD, HiZ, ST, FILTER	
32	VDD3V3	PA7	–	–	A	FLEXCOM0_IO3	I/O	1,2	PIO, I, PU, ST	
					C	TIOB6	I/O	1		
5	VDD3V3	PA8	–	–	A	FLEXCOM1_IO0	I/O	1,2	PIO, I, PU, ST	
6	VDD3V3	PA9	WKUP6	I	A	FLEXCOM1_IO1	I/O	1,2	PIO, I, PU, ST	
25	VDD3V3	PA10	–	–	A	FLEXCOM1_IO2	I/O	1	PIO, I, PU, ST	
					B	FLEXCOM1_IO4	O	2		
					C	TCLK6	I	1		
26	VDD3V3	PA11	–	–	A	FLEXCOM1_IO3	I/O	1,2	PIO, I, PU, ST	
37	VDD3V3	PA12	–	–	A	FLEXCOM2_IO0	I/O	1,2	PIO, I, PU, ST	
38	VDD3V3	PA13	WKUP7	I	A	FLEXCOM2_IO1	I/O	1,2	PIO, I, PU, ST	
39	VDD3V3	PA14	–	–	A	FLEXCOM2_IO2	I/O	1	PIO, I, PU, ST	
					B	FLEXCOM2_IO4	O	2		
40	VDD3V3	PA15	–	–	A	FLEXCOM2_IO3	I/O	1,2	PIO, I, PU, ST	
33	VDD3V3	PA16	–	–	A	FLEXCOM3_IO0	I/O	1,2	PIO, I, PU, ST	
34	VDD3V3	PA17	–	–	A	FLEXCOM3_IO1	I/O	1,2	PIO, I, PU, ST	
35	VDD3V3	PA18	–	–	A	FLEXCOM3_IO2	I/O	1	PIO, I, PU, ST	
					B	FLEXCOM3_IO4	O	2		
36	VDD3V3	PA19	–	–	A	FLEXCOM3_IO3	I/O	1,2	PIO, I, PU, ST	
46	VDD3V3	PA20	–	–	A	PCK0	O	2	PIO, I, PU, ST	
63	VDD3V3	PA21	–	–	A	COM0	O	1	PIO, I, PD, ST	
					B	FLEXCOM5_IO3	I/O	5,6		
64	VDD3V3	PA22	–	–	A	COM1	O	1	PIO, I, PD, ST	
					B	FLEXCOM5_IO2	I/O	5		
					C	FLEXCOM5_IO4	O	6		
65	VDD3V3	PA23	–	–	A	COM2	O	1	PIO, I, PD, ST	
					B	FLEXCOM5_IO1	I/O	5,6		
					D	TCLK0	I	1		
66	VDD3V3	PA24	–	–	A	COM3	O	1	PIO, I, PD, ST	
					B	FLEXCOM5_IO0	I/O	5,6		
					D	TIOB0	I/O	1		
67	VDD3V3	PA25	–	–	A	COM4	O	1	PIO, I, PD, ST	
					B	FLEXCOM6_IO0	I/O	4,5		
					D	TIOA0	I/O	1		
68	VDD3V3	PA26	–	–	A	COM5	O	1	PIO, I, PD, ST	
					B	FLEXCOM6_IO1	I/O	4,5		
69	VDD3V3	PA27	–	–	A	COM6	O	1	PIO, I, PD, ST	
					B	FLEXCOM6_IO2	I/O	4		
					C	FLEXCOM6_IO4	O	5		
70	VDD3V3	PA28	–	–	A	COM7	O	1	PIO, I, PD, ST	
					B	FLEXCOM6_IO3	I/O	4,5		

© 2026 Microchip Technology Inc. and its subsidiaries

Table 3-2. 128-pin EP-TQFP Multiplexing (continued)

TQFP128 Pin Number	Power Rail/ Voltage Reference	Primary	Alternate		PIO Peripheral				Reset State	Comments
		Signal	Signal	Dir	Func	Signal	Dir	IO Set	Signal, Dir, PU, PD, HiZ, ST, FILTER	
71	VDD3V3	PA29	AD0/ ACC_INP0/ XIN	I	A	SEG0	O	1	PIO, I, PD, ST	Clock input when oscillator is in Bypass mode
					B	FLEXCOM0_IO4	O	1		ADCx/ACCx Signals: Pull-down must be disabled in PIO Controller when using these PIO pins as ADC/ACC inputs. XIN/OUT Signals: Pull-down must be disabled in PIO Controller when using these PIO pins as Crystal Oscillator XIN/XOUT signals.
					C	FLEXCOM1_IO4	O	1		
					D	FLEXCOM2_IO4	O	1		
72	VDD3V3	PA30	AD1/ACC_INP1	I	A	SEG1	O	1	PIO, I, PD, ST	
74	VDD3V3	PB1	AD4/ACC_INN1	I	A	SEG4	O	1	PIO, I, PD, ST	
					B	FLEXCOM7_IO3	I/O	1,2		
					C	PCK1	O	2		
					D	TCLK1	I	1		
75	VDD3V3	PB0	AD3/ ACC_INN0/ XOUT	I/O	A	SEG3	O	1	PIO, I, PD, ST	
					B	FLEXCOM3_IO4	O	1		
					D	TIOB1	I/O	1		
76	VDD3V3	PB2	ERASE	I	A	SEG5	O	1	ERASE, I, PD, ST	
					B	FLEXCOM7_IO2	I/O	1		
					C	FLEXCOM7_IO4	O	2		
					D	TIOA2	I/O	1		
77	VDD3V3	PA31	AD2/ACC_INP2	I	A	SEG2	O	1	PIO, I, PD, ST	
					D	TIOA1	I/O	1		
81	VDD3V3	PB3	–	–	A	SEG6	O	1	PIO, I, PD, ST	
					B	FLEXCOM7_IO1	I/O	1,2		
					D	TIOB2	I/O	1		
82	VDD3V3	PB4	–	–	A	SEG7	O	1	PIO, I, PD, ST	
					B	FLEXCOM7_IO0	I/O	1,2		
					D	TCLK2	I	1		
83	VDD3V3	PB5	–	–	A	SEG8	O	1	PIO, I, PD, ST	
					B	FLEXCOM4_IO0	I/O	3,4		
					C	TIOA5	I/O	1		
84	VDD3V3	PB6	–	–	A	SEG9	O	1	PIO, I, PD, ST	
					B	FLEXCOM4_IO1	I/O	3,4		
					C	TIOB5	I/O	1		
85	VDD3V3	PB7	–	–	A	SEG10	O	1	PIO, I, PD, ST	
					B	FLEXCOM4_IO2	I/O	3		
					C	FLEXCOM4_IO4	O	4		
86	VDD3V3	PB8	–	–	A	SEG11	O	1	PIO, I, PD, ST	
					B	FLEXCOM4_IO3	I/O	3,4		
					C	TIOA3	I/O	1		
87	VDD3V3	PB9	WKUP9	I	A	SEG12	O	1	PIO, I, PD, ST	
					B	FLEXCOM3_IO0	I/O	3,4		
88	VDD3V3	PB10	–	–	A	SEG13	O	1	PIO, I, PD, ST	
					B	FLEXCOM3_IO1	I/O	3,4		
					D	TRACESWO	O	1		
89	VDD3V3	PB11	–	–	A	SEG14	O	1	PIO, I, PD, ST	
					B	FLEXCOM3_IO2	I/O	3		
					C	FLEXCOM3_IO4	O	4		
90	VDD3V3	PB12	WKUP10	I	A	SEG15	O	1	PIO, I, PD, ST	
					B	FLEXCOM3_IO3	I/O	3,4		
91	VDD3V3	PB13	–	–	A	SEG16	O	1	PIO, I, PD, ST	
					C	TIOB3	I/O	1,2		
92	VDD3V3	PB14	–	–	A	SEG17	O	1	PIO, I, PD, ST	
					C	TCLK3	I	1,2		
93	VDD3V3	PB15	–	–	A	SEG18	O	1	PIO, I, PD, ST	
					B	FLEXCOM5_IO4	O	1		

© 2026 Microchip Technology Inc. and its subsidiaries

Table 3-2. 128-pin EP-TQFP Multiplexing (continued)

TQFP128 Pin Number	Power Rail/ Voltage Reference	Primary	Alternate		PIO Peripheral				Reset State	Comments
		Signal	Signal	Dir	Func	Signal	Dir	IO Set	Signal, Dir, PU, PD, HiZ, ST, FILTER	
94	VDD3V3	PB16	–	–	A	SEG19	O	1	PIO, I, PD, ST	
					B	FLEXCOM6_IO4	O	1		
95	VDD3V3	PB17	–	–	A	SEG20	O	1	PIO, I, PD, ST	
					C	TCLK4	I	1,2		
96	VDD3V3	PB18	–	–	A	SEG21	O	1	PIO, I, PD, ST	
					C	TIOA4	I/O	1,2		
97	VDD3V3	PB19	–	–	A	SEG22	O	1	PIO, I, PD, ST	
					C	TIOB4	I/O	1,2		
98	VDD3V3	PB20	–	–	A	SEG23	O	1	PIO, I, PD, ST	
					B	FLEXCOM5_IO0	I/O	1,2		
99	VDD3V3	PB21	–	–	A	SEG24	O	1	PIO, I, PD, ST	
					B	FLEXCOM5_IO1	I/O	1,2		
100	VDD3V3	PB22	–	–	A	SEG25	O	1	PIO, I, PD, ST	
					B	FLEXCOM5_IO2	I/O	1		
					C	FLEXCOM5_IO4	O	2		
101	VDD3V3	PB23	–	–	A	SEG26	O	1	PIO, I, PD, ST	
					B	FLEXCOM5_IO3	I/O	1,2		
102	VDD3V3	PB24	–	–	A	SEG27	O	1	PIO, I, PD, ST	
					B	FLEXCOM4_IO0	I/O	1,2		
					C	TIOA5	I/O	2		
103	VDD3V3	PB25	WKUP8	I	A	SEG28	O	1	PIO, I, PD, ST	
					B	FLEXCOM4_IO1	I/O	1,2		
					C	TIOB5	I/O	2		
104	VDD3V3	PB26	–	–	A	SEG29	O	1	PIO, I, PD, ST	
					B	FLEXCOM4_IO2	I/O	1		
					C	FLEXCOM4_IO4	O	2		
105	VDD3V3	PC0	–	–	A	SEG30	O	1	PIO, I, PD, ST	
					B	FLEXCOM4_IO3	I/O	1,2		
					C	TIOA3	I/O	2		
					D	TDI	I	1		
106	VDD3V3	PC1	–	–	A	SEG31	O	1	PIO, I, PD, ST	
					B	FLEXCOM4_IO4	O	1		
109	VDD3V3	PC2	–	–	A	SWCLK	I	1	SWCLK,ST	
					B	FLEXCOM1_IO0	I/O	3		
110	VDD3V3	PC3	–	–	A	SWDIO	I/O	1	SWDIO,ST	
					B	FLEXCOM1_IO1	I/O	3		
111	VDD3V3	PC4	–	–	–	–	I/O	–	PIO, I, PU, ST	
112	VDD3V3	PC5	–	–	A	PCK2	O	1	PIO, I, PU, ST	
					D	TCLK5	I	1,2		
113	VDD3V3	PC6	–	–	A	FLEXCOM5_IO3	I/O	3,4	PIO, I, PU, ST	
					B	FLEXCOM6_IO0	I/O	3		
114	VDD3V3	PC7	WKUP11	I	A	FLEXCOM5_IO2	I/O	3	PIO, I, PU, ST	
					B	FLEXCOM6_IO1	I/O	3		
					C	FLEXCOM5_IO4	O	4		
115	VDD3V3	PC8	–	–	A	FLEXCOM5_IO1	I/O	3,4	PIO, I, PU, ST	
116	VDD3V3	PC9	WKUP12	I	A	FLEXCOM5_IO0	I/O	3,4	PIO, I, PU, ST	
118	VDD3V3	PC10	–	–	A	QIO3	I/O	1	PIO, I, PU, ST	
119	VDD3V3	PC11	–	–	A	QIO2	I/O	1	PIO, I, PU, ST	
					B	FLEXCOM7_IO4	O	3		
121	VDD3V3	PC12	–	–	A	QIO1	I/O	1	PIO, I, PU, ST	
					B	FLEXCOM7_IO1	I/O	3		

© 2026 Microchip Technology Inc. and its subsidiaries

Table 3-2. 128-pin EP-TQFP Multiplexing (continued)

TQFP128 Pin Number	Power Rail/ Voltage Reference	Primary	Alternate		PIO Peripheral				Reset State	Comments
		Signal	Signal	Dir	Func	Signal	Dir	IO Set	Signal, Dir, PU, PD, HiZ, ST, FILTER	
122	VDD3V3	PC13	–	–	A	QIO0	I/O	1	PIO, I, PU, ST	
					B	FLEXCOM7_IO0	I/O	3		
123	VDD3V3	PC14	WKUP13	I	A	QCS	O	1	PIO, I, PU, ST	
					B	FLEXCOM7_IO3	I/O	3		
					D	TIOA7	I/O	1		
124	VDD3V3	PC15	–	–	A	QSCK	O	1	PIO, I, PU, ST	
					B	FLEXCOM7_IO2	I/O	3		
					D	TIOB7	I/O	1		
125	VDD3V3	PC16	–	–	A	FLEXCOM6_IO0	I/O	1,2	PIO, I, PU, ST	
					D	TCLK7	I	1		
126	VDD3V3	PC17	–	–	A	FLEXCOM6_IO1	I/O	1,2	PIO, I, PU, ST	
					D	TIOA8	I/O	1		
127	VDD3V3	PC18	–	–	A	FLEXCOM6_IO2	I/O	1	PIO, I, PU, ST	
					B	FLEXCOM6_IO4	O	2		
					D	TIOB8	I/O	1		
128	VDD3V3	PC19	–	–	A	FLEXCOM6_IO3	I/O	1,2	PIO, I, PU, ST	
					D	TCLK8	I	1		
1	VDD3V3	PC20	–	–	A	FLEXCOM7_IO1	I/O	4	PIO, I, PU, ST	
2	VDD3V3	PC21	WKUP14	I	A	FLEXCOM7_IO0	I/O	4	PIO, I, PU, ST	
3	VDD3V3	PC22	RTCOUT1	O	A	PCK0	O	1	PIO, I, PD, ST	
4	VDD3V3	PD0	–	–	A	TIOA9	I/O	1	PIO, I, PU, ST	
					B	–	I	1		
42	VDD3V3	PD1	–	–	A	URXD, PWMFI0, PWMEXTRG1	I	1	PIO, I, PU, ST	PWMFI0 is enabled in PWM_FMR; PWMEXTRG1 is enabled in PWM_ETRGx (refer to section Pulse Width Modulation Controller (PWM))
					B	–	O	1		
43	VDD3V3	PD2	–	–	A	UTXD	O	1	PIO, I, PU, ST	
					B	–	I/O	1		
7	VDD3V3	PD3	–	–	A	TCLK9	I	1	PIO, I, PU, ST	
					C	PWMH0	O	1		
8	VDD3V3	PD4	–	–	A	TIOB9	I/O	1	PIO, I, PU, ST	
			–	–	B	PCK2	O	2		
9		PD5	–	–	A	MCSPI_SPCK	O	2,3,4	PIO, I, PU, ST	
10		PD6	–	–	A	MCSPI_MISO	O	2	PIO, I, PU, ST	
11		PD7	–	–	A	MCSPI_MOSI0	I	2,3,4	PIO, I, PU, ST	
12		PD8	–	–	A	MCSPI_NPCS0	I/O	2,3	PIO, I, PU, ST	
			–	–	B	MCSPI_MOSI1	I	4		
13		PD9	–	–	A	MCSPI_NPCS1	O	2,3	PIO, I, PU, ST	
			–	–	B	MCSPI_MOSI2	I	4		
14		PD10	–	–	A	MCSPI_NPCS2	O	2,3	PIO, I, PU, ST	
			–	–	B	MCSPI_MOSI3	O	4		
15		PD11	–	–	A	MCSPI_NPCS3	O	2,3	PIO, I, PU, ST	
44	VDD3V3	PD12	–	–	A	URXD, PWMFI0, PWMEXTRG1	I	2	PIO, I, PU, ST	PWMFI0 is enabled in PWM_FMR; PWMEXTRG1 is enabled in PWM_ETRGx (refer to section Pulse Width Modulation Controller (PWM))
45	VDD3V3	PD13	–	–	A	UTXD	O	2	PIO, I, PU, ST	

© 2026 Microchip Technology Inc. and its subsidiaries

Table 3-2. 128-pin EP-TQFP Multiplexing (continued)

TQFP128 Pin Number	Power Rail/ Voltage Reference	Primary	Alternate		PIO Peripheral				Reset State	Comments
		Signal	Signal	Dir	Func	Signal	Dir	IO Set	Signal, Dir, PU, PD, HiZ, ST, FILTER	
18	VDD3V3	PD14	–	–	A	TIOA10	I/O	1	PIO, I, PU, ST	
19		PD15	–	–	A	TIOB10	I/O	1	PIO, I, PD, ST	
			–	–	B	PWMH1	O	1	PIO, I, PD, ST	
20	VDD3V3	PD16	–	–	A	TCLK10	I	1	PIO, I, PD, ST	
					B	PWMH2	O	1		
22	VDD3V3	PD17	–	–	A	PWML0	O	1	PIO, I, PD, ST	
					C	TIOA11	I/O	1		
23	VDD3V3	PD18	–	–	A	PWML1	O	1	PIO, I, PD, ST	
					B	–	I/O	1		
					C	TIOB11, PWMEXTRG2	I/O	1		PWMEXTRG2 is enabled in PWM_ETRGx (refer to section Pulse Width Modulation Controller (PWM))
24	VDD3V3	PD19	–	–	A	PWML2	O	1	PIO, I, PD, ST	
					B	–	O	1		
					C	TCLK11	I	1		
49	VDD3V3	NRST	–	I/O	–	–	–	–	I, PU, ST	
57	VDDBU	FWUP	–	I	–	–	–	–	I,ST	Active low, external pull-up needed
58		JTAGSEL	–	I	–	–	–	–	I,PD	
62		SHDN	–	O	–	–	–	–	O	0: The device is in Backup mode 1: The device is not in Backup mode
61		TST	–	I	–	–	–	–	I,PD	–
56		RTCOUT0	–	O	–	–	–	–	O	–
55		WKUP0/ TMP0	–	I	–	–	–	–	I,ST	External pull-up needed
54		WKUP1/ TMP1	–	I	–	–	–	–	I,ST	External pull-up needed
53		WKUP2/ TMP2	–	I	–	–	–	–	I,ST	External pull-up needed
60		XOUT32	–	O	–	–	–	–	–	–
59		XIN32	–	I	–	–	–	–	–	Clock input when oscillator is in Bypass mode
21	VDD3V3	POWER	–	–	–	–	–	–	–	–
52	VBAT		–	–	–	–	–	–	–	–
51	VDD3V3		–	–	–	–	–	–	–	–
80	VDD3V3		–	–	–	–	–	–	–	–
108	VDD3V3		–	–	–	–	–	–	–	–
120	VDD3V3		–	–	–	–	–	–	–	–
41	VDDCORE		–	–	–	–	–	–	–	–
50	VDDCORE		–	–	–	–	–	–	–	–
107	VDDCORE		–	–	–	–	–	–	–	–
117	VDDCORE		–	–	–	–	–	–	–	–
78	VDDPLL		–	–	–	–	–	–	–	–
47	VDDOUT		–	–	–	–	–	–	–	–
48	VDD3V3		–	–	–	–	–	–	–	–
73	VREFP		–	–	–	–	–	–	–	–
79	VDDLCD		–	–	–	–	–	–	–	–
129	GND		–	–	–	–	–	–	–	Digital ground (exposed pad)

© 2026 Microchip Technology Inc. and its subsidiaries

3.2.1.2. FLEXCOM Features

Table 3-3. FLEXCOM IOs

Signal Name	Function	Signal Name by Mode					
		SPI	TWI	2-Wire UART	4-Wire UART	USART	ISO7816
FLEXCOMx_IO0	Transmit Data	MOSI	TWD	TXD	TXD	TXD	TXD
FLEXCOMx_IO1	Receive Data	MISO	TWCK	RXD	RXD	RXD	–
FLEXCOMx_IO2	Serial Clock	SPCK	–	–	–	SCK	SCK
FLEXCOMx_IO3	Clear to send/Chip Select	NPCS0/NSS	–	–	CTS	–	–
FLEXCOMx_IO4	Request to send/Chip Select	NPCS1	–	–	RTS	–	–

© 2026 Microchip Technology Inc. and its subsidiaries

4. Power Supply and Power Control

4.1. Power Supplies

The device has several types of power supply pins. The figure and table below show power domains depending on the different power supply pins and supply description.

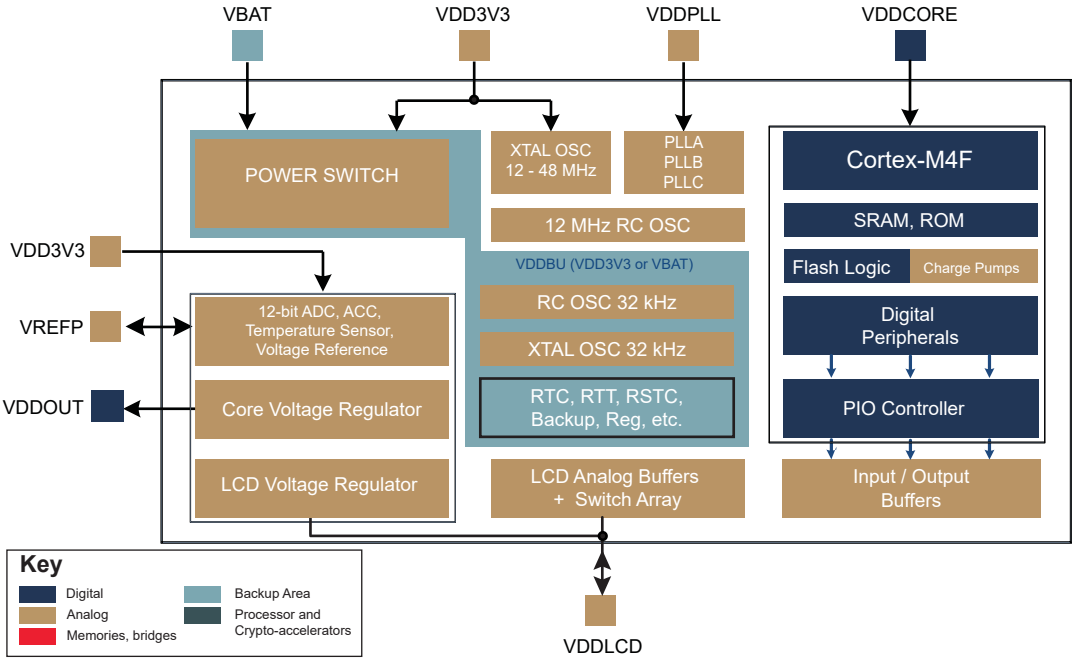
Table 4-1. Power Supplies

Supply Name	Description
VDD3V3[1]	• Input/Output buffers supply • Flash memory charge pumps supply for read, erase and program operations • Internal power switch input • Core voltage regulator input • Analog block supply (ADC, Temp Sensor, voltage reference, Analog Comparator)
VDDOUT	Output of the embedded core voltage regulator
VDDLCD[4]	LCD voltage regulator input/output for the LCD Driver and its logic
VDDCORE[2, 3]	Core, memories and peripheral logic supply
VDDPLL[2, 3]	PLLA, PLLB and PLLC supply
VBAT	Internal power switch input
VDDBU	Output of the internal power switch (VBAT or VDD3V3) powering the backup area

Notes:

1. VDD3V3 is powered from one single power source such that V(VDD3V3) ≤ ±50mV.
2. Applies only when VDDCORE and VDDPLL are powered from an external regulator.
3. VDDCORE and VDDPLL are powered from one single power source such that V(VDDCORE,VDDPLL) ≤ ±50mV.
4. At any time, VDDLCD ≤ VDD3V3.

Figure 4-1. PIC32CXMTG Power Domains



© 2026 Microchip Technology Inc. and its subsidiaries

4.1.1. Core Voltage Regulator

The core voltage regulator is managed by the Supply Controller (SUPC). It features three operating modes:

- In Normal mode, internal adaptive biasing adjusts the regulator quiescent current depending on the required load current.
- In Wait mode, the regulator is still ON but in low power mode with a very low quiescent current.
- In Backup mode, the voltage regulator is OFF.

For further information, refer to Core Voltage Regulator Characteristics.

4.1.2. LCD Voltage Regulator

The device embeds an adjustable LCD voltage regulator that is managed by the SUPC.

This internal regulator is designed to supply the Segment LCD outputs. The LCD regulator output voltage is software selectable with 16 levels to adjust the display contrast (only when the internal LCD Voltage regulator is used).

If not used, its output (VDDLCD) can be bypassed (Hi-Z mode) and an external power supply can be provided onto the VDDLCD pin. In this case, VDD3V3 still needs to be supplied.

The LCD voltage regulator can be used in all power modes (Backup, Wait, Sleep and Active).

For adequate input and output power supply decoupling/bypassing, refer to LCD Voltage Regulator and LCD Output Buffers.

4.1.3. Power Switch

The device features a power switch between VBAT and VDD3V3. Its output, VDDBU, powers the backup area.

The power switch features a Manual mode controlled by software to select VBAT or VDD3V3 (default at power-up). An Automatic mode is also available when used with the VDD3V3 Supply Monitor (SMVDD3V3) as a fold-back source to switch VDDBU to VBAT in case VDD3V3 is falling under the SMVDD3V3 threshold. A second Automatic mode (not user programmable and not correlated to the SMVDD3V3) is available, in case of the power switch is on VBAT and VBAT is falling, the switch is commuted automatically to VDD3V3. Refer to the sections Supply Controller (SUPC) and Reset Controller (RSTC) for details on the power switch.

4.1.4. Typical Powering Schematics

The device supports single-supply operation. Restrictions on this range may apply depending on enabled features. Refer to the section Electrical Characteristics.

Note that in the typical power supply schemes that follow, with or without a backup battery, the SLCD and the internal LCD voltage regulator can be used in all power modes. The LCD voltage regulator can be disabled to save its operating current. VDDLCD must then be provided externally and thus the contrast level must be managed by adjusting this supply. Internal and external wake-up sources (WKUPx, TMPx, FWUP) can be used in the power schemes illustrated below. Extra tamper or wake-up input pins referenced to VDD3V3 can be used in Backup mode as long as VDD3V3 is supplied in Backup mode.

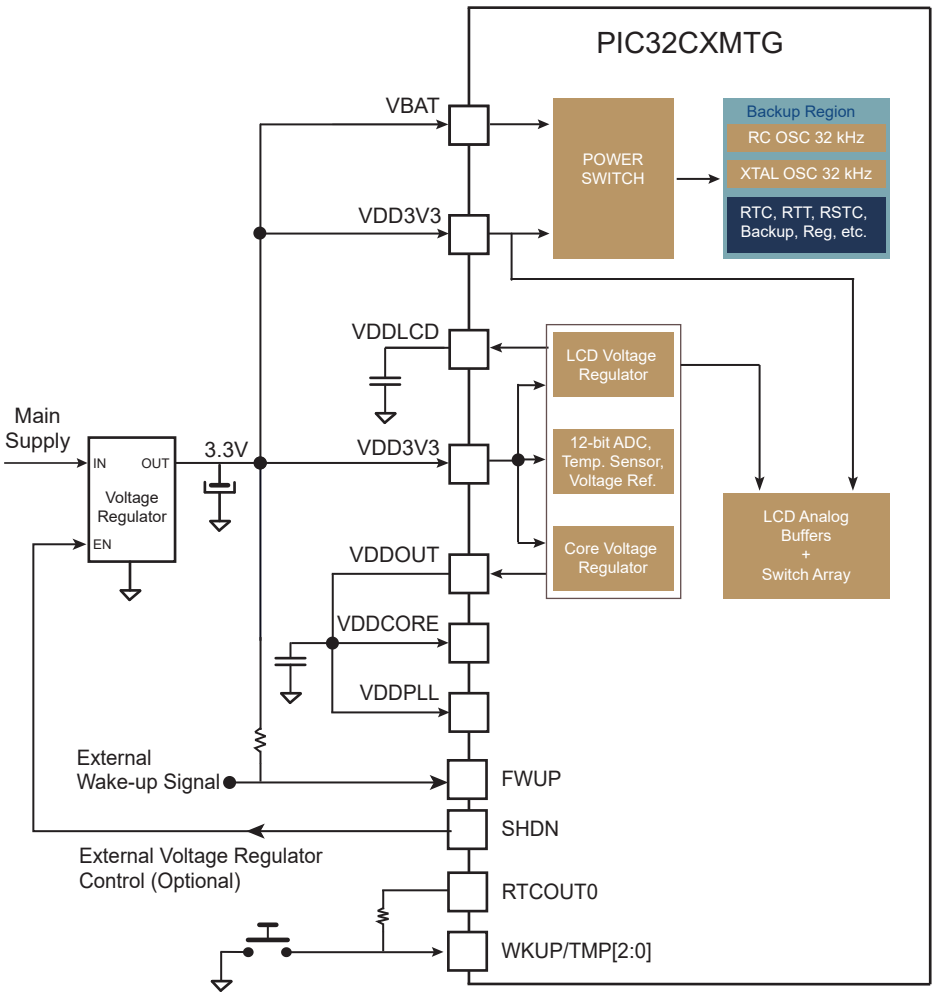
Important: The figures in the following sections show simplified schematics of the power section. Underwriters Laboratories (UL) standards, and specifically the UL standard 60950-1 battery protection circuit, are not represented.

© 2026 Microchip Technology Inc. and its subsidiaries

4.1.4.1. Single Supply Operation without Backup Battery

The following figure shows a typical power supply scheme with a single power source. VDD3V3 and VBAT are derived from the main power source (typically a 3.3V regulator output) while VDDCORE, VDDPLL, and VDDLCD are fed by the embedded regulator outputs.

Figure 4-2. Single Supply Operation without Backup Battery

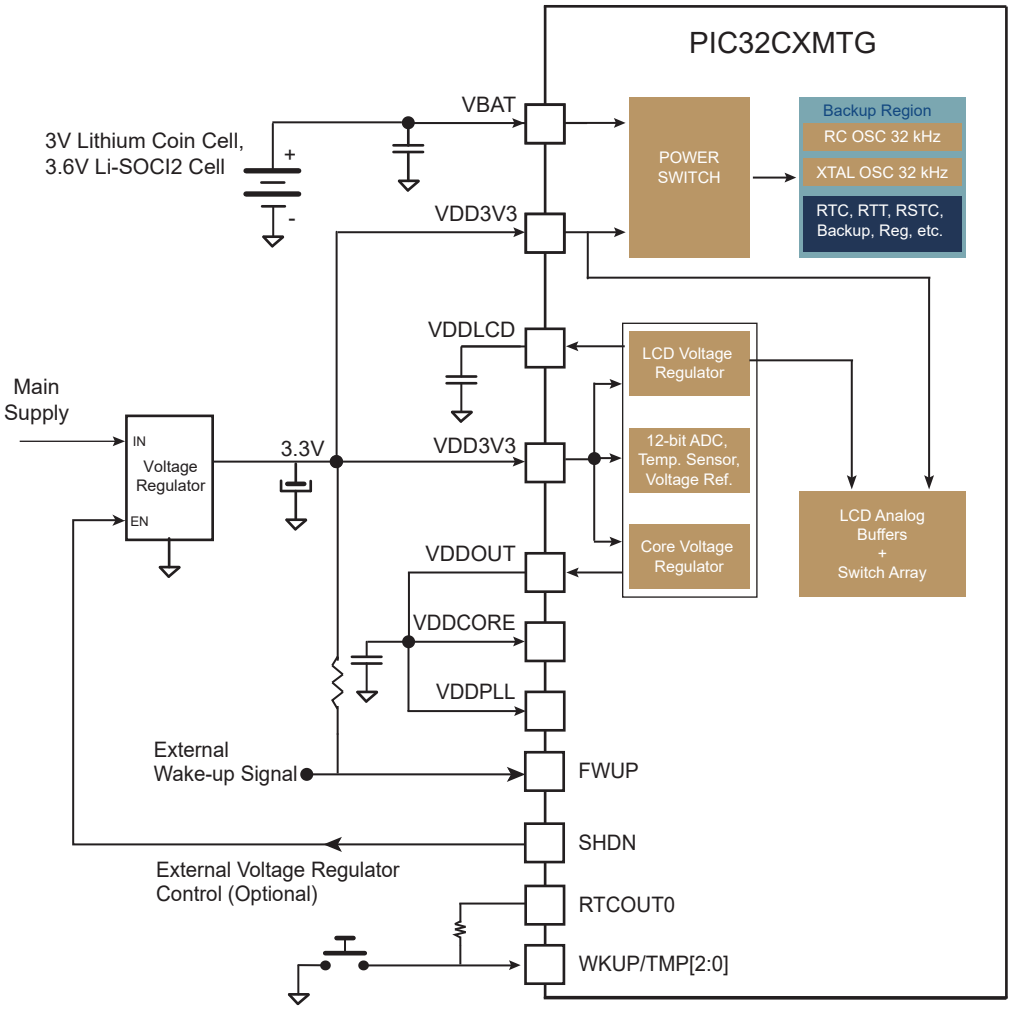


4.1.4.2. Single Supply Operation with Backup Battery

The following figure shows the single-supply operation schematic improved by adding a backup capability when main power source is removed. VBAT is supplied with a separate backup battery while VDD3V3 is still connected to the main power source. This power supply use case is for systems where saving of the backup area, mainly the RTC must remain active between full power cycles.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 4-3. Single Supply with Backup Battery



4.1.4.3. Backup Area/Supply Controller Pins

In all power supply figures shown above, FWUP, SHDN, WKUP/TMP[2:0] and RTCOUT0 are input/output pads referenced to the output of the power switch. As a result, the logic levels of the corresponding pins shift up or down depending on the voltage level of VBAT and VDD3V3 to which the power switch is connected. If more generic wake-up pins other than WKUP[2..0]/TMP[2..0] are needed as wake-up sources from Backup mode, or as anti-tamper inputs, VDD3V3 must be present in Backup mode.

4.1.4.4. Segment LCD IOs

Segment LCD IOs are referenced to VDDLCD when used as LCD IOs and referenced to VDD3V3 otherwise.

4.1.4.5. General Purpose IO (GPIO) State in Backup Mode

If the VDD3V3 power supply shown in the figures above is maintained before entering Backup mode, the state of any PIO line can be defined as input (pull-up or pull-down) or output (low or high level). All PIO lines can therefore be configured to a stable and known state in order to avoid any extra power consumption or any current path with the input/output lines of the external on-board devices. Configuration of the GPIO lines is maintained in the same state as before entering Backup mode.

By default, when waking up from Backup mode, all PIO lines are reset to their default state. In some cases, it may be necessary to postpone resetting some PIO lines until the main application program

© 2026 Microchip Technology Inc. and its subsidiaries

has reached a certain point. In these cases, IO retention can be configured in the IO Retention register. The granularity of the configuration is 8. Refer to the IO Retention register in the section Special Function Registers (SFR) for further details. Only removal of VBAT and VDD3V3 will reset the IO Retention configuration to the default mode.

4.1.4.6. Default General Purpose IO (GPIO) State after Reset

The Reset state of the GPIO lines after reset is given in the section Pinout and Multiplexing. For further details about the GPIO and system lines, wake-up sources and wake-up time, and typical power consumption in different Low-Power modes, see the table Low-Power Mode Summary Table.

4.2. System State

The device always boots from the ROM even if a valid application is present in the Flash memory. The device configuration is defined by the SAM-BA®/Secure SAM-BA boot program. Then, the device boot mode at next power-up or reset is defined by the boot mode defined in the GPNVM bit. Refer to the section ROM Code and Boot Strategies for details.

4.2.1. Device Configuration after a Power Cycle or General Reset

After a power cycle of all the power supply rails including VBAT, the system peripherals, such as the Flash controller, the Clock Generator, the Power Management Controller (PMC) and the SUPC, are in the following states prior to ROM code execution:

- Backup area is reset
- Slow Clock (SLCK) source is the internal 32 kHz RC Oscillator (32 kHz crystal oscillator is disabled)
- Main Clock (MAINCK) source is set to the 12 MHz internal RC Oscillator
- 12–48 MHz crystal oscillator and PLLs are disabled
- VDDCORE Supply Monitor and Core Reset are enabled
- VDD3V3 Supply Monitor is disabled
- Flash Wait State (FWS) bit in the Flash Mode register (EEFC_FMR) is set to 15 (0xF)

4.2.2. Device Configuration after a Wake-Up from Backup Mode

Before entering Backup mode, the backup area must be configured for the device to be able to wake up from an event (internal or external) as long as VBAT or VDD3V3 is supplied.

After wake-up, the device configuration is the following:

- Configuration of the peripherals the backup area remains the same as before entering Backup mode
- Main Clock (MAINCK) source is set to the 12 MHz internal RC Oscillator
- 12–48 MHz crystal oscillator and PLLs are disabled
- VDDCORE Supply Monitor and Core Reset are enabled
- Flash Wait State (FWS) bit in the Flash Mode register (EEFC_FMR) is set to 15 (0xF)

4.2.3. Boot ROM Clock Speed Configuration

For systems required to run on battery with low peak current capability during a power outage (read without Power mode), specific GPBR registers can be programmed by the user with predefined processor and bus clock speed and ratios. Refer to the section ROM Code and Boot Strategies.

4.3. Active Mode

Active mode is the normal running mode, with the single coreexecuting code. The system clock can be the fast RC oscillator, the main crystal oscillator or the PLLs. The PMC can be used to adapt the frequency and to disable the peripheral clocks when unused.

© 2026 Microchip Technology Inc. and its subsidiaries

4.4. Low-Power Modes

The various low-power modes (Backup, Wait and Sleep modes) are described below. Note that the SLCDC can be used in all low-power modes.

The Wait For Event instruction (WFE) of the Cortex-M4 core can be used to enter any of the low-power modes, however this may add complexity to the design of application state machines. This is due to the fact that the WFE instruction is associated with an event flag of the Cortex core that cannot be managed by the software application. The event flag can be set by interrupts, a debug event or an event signal from another processor. When an event occurs just before WFE execution, the processor takes it into account and does not enter Low-power mode. Microchip has made provision to avoid using the WFE instruction. The workarounds to ease application design, including the use of the WFE instruction, are given in the following descriptions of the Low-Power mode sequences.

4.4.1. Backup Mode

The purpose of Backup mode is to achieve the lowest possible power consumption in a system that executes periodic wake-ups to perform tasks but which does not require fast start-up time. In this mode, the backup area is running. The core voltage regulator and the core supplies are off. Wake-up from Backup mode can be done through several internal or external sources. The SLCDC can be used in Backup mode. The purpose is to maintain the displayed message on the LCD display after entering Backup mode. The current consumption on VDD3V3 to maintain the SLCDC is a few µA (typical). Refer to the section Electrical Characteristics.

In case the VDD3V3 power supply is maintained when entering Backup mode, it is up to the application to configure all PIO lines in a stable and known state to avoid extra power consumption or possible current path with the input/output lines of the external on-board devices.

4.4.1.1. Entering and Exiting Backup Mode

To enter Backup mode, follow the steps in the sequence below:

1. If VDD3V3 is maintained when in Backup mode, all IOs except SLCD IOs if used in Backup mode, have to be set back to PIO input pull-up or pull-down or output low or high level to avoid power consumption with external on-board devices.
2. Switch the complete clock tree (Processors and bus clocks) to the 12 MHz RC, and stop all PLLs and fast crystal oscillator if used.
3. Enable the RTT in 1-Hz mode.
4. Disable Normal mode of the RTT (RTT runs in 1-Hz mode).
5. Program one or several wake-up sources(1) and clear any pending wake-up sources.
6. Disable the VDDCORE Supply Monitor.
7. Adjust the power switch position in RSTC_CR according to the power supply used for VDDBU (internal switch output)
8. Select one of the following methods to enter Backup mode:
    a. If the internal core voltage regulator is used, write a 1 to SUPC_CR.VROFF. The pin SHDN remains high in this case.
    b. If an external voltage regulator is used for VDDCORE/VDDPLL, write a 1 to SUPC_CR.SHDW. The pin SHDN is asserted low, allowing an external voltage regulator to be turned OFF.
    c. If the SLCD is used in Backup mode to keep a static message, write a 1 to SUPC_CR.SHDWEOF. At the end of the SLCD frame, the SHDN pin is asserted low

Note:

1. If a WKUP wake-up pin with tamper functionality is used as wake-up source, the wake-up debouncer (WKUPDBC) value must be set so that its period is greater than or equal to 1/2 x RTCOUT0 period set in the RTC.

© 2026 Microchip Technology Inc. and its subsidiaries

After this step, the core voltage regulator is shut down. None of the digital internal logic is powered. Whether VROFF, SHDN or SHDWEOF was used to enter Backup mode, the system exits Backup mode if one of the enabled wake-up events listed in the table Low-Power Mode Summary occurs.

After exiting Backup mode, the device is in Reset state. Only the configuration of the backup area peripherals remains unchanged.

Note that the device does not automatically enter Backup mode if VDD3V3 is disconnected, or if it falls below minimum voltage.

For current consumption in Backup mode, refer to the section Electrical Characteristics.

4.4.2. Wait Mode

The purpose of Wait mode is to achieve low power consumption while maintaining the whole device in a powered state for a start-up time of a few µs. For current consumption in Wait mode, refer to the section Electrical Characteristics.

In this mode, all clocks are stopped when Wait mode is entered. However, the power supplies of digital logic and memories are maintained using Standby mode of the core voltage regulator to ensure memory and CPU context retention.

The device can handle external and internal events in order to perform a wake-up. This is done by configuring one or several internal or external wake-up sources.

4.4.2.1. Entering and Exiting Wait Mode

To enter Wait mode, follow the steps in the sequence below:

1. Apply the sequence for PMC configuration as defined in Ultra Low Power Modes and Fast Start-Up in the section Power Management Controller (PMC).
2. In order to avoid power consumption with an external on-board device, depending on peripheral use in the application, all IOs must be set to a known state either externally or internally via input pull-up or pull-down, or via output low or high level, or to an inactive state. Set peripheral IOs used previously into GPIO mode such as XIN/XOUT, ADx if used
3. Set the FWS bit in EEFC_FMR to 0.
4. Program one or several wake-up sources and clear any pending wake-up sources.
5. Select one of the following methods to enter Wait mode and complete the sequence:
   a. Set the WAITMODE bit to 1 in the PMC Main Oscillator register (CKGR_MOR). Ensure that no access to PMC registers is attempted immediately after accessing the WAITMODE bit.
   b. Use the WFE instruction:
      - Set the LPM bit in PMC_FSMR.
      - Write a 0 to the SLEEPDEEP bit of the Cortex-M4 processor.
      - Execute the Wait-For-Event (WFE) instruction of the processor.

Whether the WAITMODE bit or the WFE instruction was used to enter Wait mode, the system exits Wait mode if one of the enabled wake-up events listed in the table Low-Power Mode Summary occurs.

After exiting Wait mode, the PIO controllers have the same configuration state as before entering Wait mode. The device is clocked back to the RC oscillator frequency which was used before entering Wait mode. The core starts fetching from Flash or SRAM at this frequency.

4.4.3. Sleep Mode

The purpose of Sleep mode is to optimize power consumption of the device versus response time. In this mode, only the core clock is stopped. Some of the peripheral clocks can be enabled depending on the application needs. The current consumption in this mode is application-dependent.

© 2026 Microchip Technology Inc. and its subsidiaries

This mode is entered using Wait for Interrupt (WFI) or Wait for Event (WFE) instructions of the Cortex-M4. The processor can be awakened from an interrupt if the WFI instruction of the Cortex-M4 is used to enter Sleep mode, or from a wake-up event if the WFE instruction is used. The WFI instruction can also be used to enter Sleep mode with the SLEEPONEXIT bit set to 1 in the System Control register (SCB_SCR) of the Cortex-M4. If SCB_SCR.SLEEPONEXIT is set to 1, when the processor completes the execution of an exception handler it returns to Thread mode and immediately enters Sleep mode. This mechanism can be used in applications that require the processor to run only when an exception occurs. Setting the SLEEPONEXIT bit to 1 enables an interrupt-driven application in order to avoid returning to an empty main application.

4.4.4. Low-Power Mode Summary Table

The modes detailed above are the main Low-Power modes. The following table provides a configuration summary of the Low-Power modes. For more information on power consumption, refer to the section Electrical Characteristics.

Table 4-2. Power-Up and Start-Up Scenarios

Previous Power Mode	Wake-Up or Power-Up Mode to Active Mode						
	Power Signals				MCU Power Mode (Core & IPs)	Digital BU State	Wake-Up Source
	VDD3V3	VBAT	VDDBU	VDDCORE			
Not powered	Not Present (floating or 0V)	Floating	–	–	Not powered	Not powered	–
	Rising	Floating	VDD3V3	Started by SUPC	Not powered to Active	Active	–
	Rising	Under VBAT POR level, or 0V	VDD3V3	Started by SUPC	Not powered to Active	Active	–
	Not Present (floating or 0V)	Present (Established)	OFF	OFF	Not powered	OFF	–
	Rising	Present (Established)	VDD3V3	Started by SUPC	Not powered to Active	Active	–
	Rising	Present, slow rise > 1s	VDD3V3	Started by SUPC	Not powered to Active	Active	–
Powered, BU configured by user	Not Present (floating or 0V)	Present (Established)	VBAT	0V	Backup	Active	–
Backup Mode (Powered & BU configured by user)(1)	Rising	Present (Established)	VBAT to VDD3V3	Started by SUPC	From Backup to Active	Active	Any programmed wake-up sources
Backup Mode, with or without SLCD (Powered & BU configured by user)(1)	Present (Established)	Present (Established)	VBAT to VDD3V3	Started by SUPC	From Backup to Active	Active	Any programmed wake-up sources

Note:

1. VDDBU switched from VBAT to VDD3V3 by user.

Table 4-3. Low-Power Mode Summary Table

Core/SLCD Voltage Regulator	Core, Memory & Peripherals	Wake-Up Sources	Core State at Wake-Up	PIO State in Low-Power Mode	PIO State at Wake-Up
Backup Mode					

© 2026 Microchip Technology Inc. and its subsidiaries

Table 4-3. Low-Power Mode Summary Table (continued)

Core/SLCD Voltage Regulator	Core, Memory & Peripherals	Wake-Up Sources	Core State at Wake-Up	PIO State in Low-Power Mode	PIO State at Wake-Up
OFF/OFF	OFF (Not powered)	Any event among: - WKUP pins (WKUP) - Anti-tamper pins (TMP) (1) - Force Wake-up pin (FWUP) - VDD3V3 Supply Monitor (if VDD3V3 is present, and VDD3V3 supply falling) - 32 kHz crystal failure detection - VBAT Supply Monitor - VDDCORE POR Reset (if VDDCORE is powered by an external regulator) - RTC alarm - RTT alarm	Reset	Previous state saved	- Reset state. - Selected PIO Group state maintained if VDD3V3 present in Backup mode
Backup Mode with SLCD					
OFF/ON	OFF (Not powered)	Any event among: - WKUP pins (WKUP) - Anti-tamper pins (TMP)(1) - Force Wake-up pin (FWUP) - VDD3V3 Supply Monitor (at any time, VDDLCD ≤ VDD3V3) - 32 kHz crystal failure detection - VBAT Supply Monitor - VDDCORE POR Reset (if VDDCORE is powered by an external regulator) - RTC alarm - RTT alarm	Reset	Previous state saved	- SLCD IOs: Unchanged - Selected PIO Group state maintained - Other IOs: Reset state
Wait Mode					
ON/ON or OFF	Powered, not clocked, Selected SRAM0 blocks in Low-Power mode	Any event among: - WKUP pins (WKUP) configured as Fast Start-up in PMC(2) - VDD3V3 Supply Monitor (at any time, VDDLCD ≤ VDD3V3) - 32 kHz crystal failure detection - RTC alarm - RTT alarm - Wake-up on debug request (connected internally to WKUP15) - Asynchronous Partial Wake-up Event (FLEXCOM)	Clocked back	Previous state saved	Unchanged
Sleep Mode					

© 2026 Microchip Technology Inc. and its subsidiaries

Table 4-3. Low-Power Mode Summary Table (continued)

Core/SLCD Voltage Regulator	Core, Memory & Peripherals	Wake-Up Sources	Core State at Wake-Up	PIO State in Low-Power Mode	PIO State at Wake-Up
ON/ON or OFF	Powered (Not clocked)	Entry mode = WFI - Any enabled Interrupts Entry mode = WFE- Any enabled event: --- WKUP pins (WKUP) configured as Fast Start-up in PMC(2) --- VDD3V3 Supply Monitor (at any time, VDDLCD ≤ VDD3V3) --- 32 kHz crystal failure detection --- RTC alarm --- RTT alarm --- Wake-up on debug request (connected internally to WKUP15) --- Asynchronous Partial Wake-up Event (FLEXCOM)	Clocked back	Previous state saved	Unchanged

Notes:

1. For wake-up from WKUP pins in anti-tamper mode, the WKUPENx bit in SUPC_WUIMR corresponding to the WKUPx/TMPx pin must also be enabled.
2. While in Wait mode or Sleep mode, the device can be woken up by fast start-up pins (Wake-up pins WKUPx in fast start-up mode) which also generate a tamper event on related WKUP pins with tamper function. A low-power debouncer must be enabled on the corresponding WKUP pins to trigger a tamper event in parallel with a fast start-up event to the PMC.

© 2026 Microchip Technology Inc. and its subsidiaries

5. Input/Output Lines

The device has two types of input/output (I/O) lines—general purpose I/Os (GPIO) and system I/Os. GPIOs have alternate functionality due to multiplexing capabilities of the PIO controllers. The same PIO line can be used whether in I/O mode or by the multiplexed peripheral.

5.1. General Purpose I/O Lines

General purpose I/O (GPIO) lines are managed by PIO Controllers. All I/Os have several input or output modes, such as pull-up or pull-down, input Schmitt Triggers, multidrive (open-drain), glitch filters, debouncing or input change interrupt. Programming of these modes is performed independently for each I/O line through the PIO controller user interface. For more details, refer to the section Parallel Input/Output Controller (PIO).

5.2. TST Pin

The TST pin is used for JTAG Boundary Scan Manufacturing Test or Fast Flash programming mode. For details on entering Fast Flash programming mode, refer to the section FFPI Monitor. For more information on the manufacturing and test modes, refer to the section Debug and Test.

5.3. NRST Pin

The NRST pin is bidirectional. It is handled by the on-chip Reset Controller (RSTC) and can be driven low to provide a reset signal to the external components, or asserted low externally to reset the microcontroller. It resets the core and the peripherals, with the exception of the Backup region. There is no constraint on the length of the reset pulse, and the RSTC can guarantee a minimum pulse length. The NRST pin integrates a permanent pull-up resistor to VDD3V3 of about 100 kΩ. By default, the NRST pin is configured as an input.

5.4. Anti-Tamper (TMPx) Pins

Anti-tamper pins detect intrusion, for example, into a smart meter case. Upon detection through a tamper switch, automatic, asynchronous and immediate actions are performed. Refer to the section Supply Controller (SUPC) for more details.

Anti-tamper pins can be used in all modes. Date and number of tampering events are stored automatically. TMP0 to TMP2 buffers are referenced to the output of the power switch. All other tamper pins are referenced to VDD3V3.

5.5. RTCOUTx Pins

RTCOUT0 can be used to generate waveforms from the RTC in order to take advantage of its inherent prescalers while the RTC is the only powered circuitry (Low-Power mode, Backup mode) or in any active mode. Entering Backup or Low-Power operating modes does not affect the waveform generation outputs; the RTCOUT0 pad is referenced to the output of the power switch.

RTCOUT1 is shared in the PIO multiplexing and is referenced to VDD3V3. RTCOUT1 has the same function as RTCOUT0 but cannot be used as a sampling signal for tamper pins.

Anti-tampering pin detection can be synchronized with RTCOUT0 only.

5.6. Shutdown (SHDN) Pin

The purpose of the SHDN pin is to control the enable signal typically found on the voltage regulator. Depending on the application use case and needs in terms of Low-Power mode and power supply strategy, the SHDN pin can be asserted low (SUPC_CR.SHDWEOF=1 or SUPC_CR.SHDW=1) when the device enters Backup mode. At wake-up event detection, the SHDN pin is asserted high, and thus re-enables the external voltage regulator. See Typical Powering Schematics in the section Power Supply and Power Control for more details.

© 2026 Microchip Technology Inc. and its subsidiaries

5.7. Force Wake-up (FWUP) Pin

The FWUP pin can be used as a wake-up source in all low-power modes as it is referenced to the output of the power switch. An external pull-up is required on this pin.

5.8. Flash Memory Hardware ERASE Signal

A hardware erase is when the ERASE signal on PB2 pin is used to reinitialize the Flash content (memory array), the lock bits, GPNVM bits and security bit to an erased state. After a hardware erase, the memory array bits read as logic level 1, GPNVM bits read as logic level 0 and lock bits are in an unlocked state. The ERASE signal on PB2 pin integrates a pull-down resistor of about 100 kΩ to GND, so that it can be left unconnected for normal operations. The ERASE signal is the default mode on the PB2 pin.

If necessary, the hardware ERASE signal may be partially or permanently disabled by programming the Erase Function Lock (EFL) GPNVM bit [4:2]. Note that the security bit must be set in these cases. Otherwise, regardless of the value of the GPNVM bit [4:2], the ERASE signal will trigger an ERASE sequence if set to logic level high.

Note that it is strongly advised to disable the Erase function using the EFL once the production software has been fully validated. Refer to the GPNVM bit description in the section ROM Code and Boot Strategies for further details and correct use.

The ERASE signal is a system I/O signal that is available in the IO multiplexing. As a result, this pin can also be used as a standard general purpose I/O (GPIO). Configuration of this pin as a GPIO is done through the PIO Controller. Care must be taken when changing this pin between GPIO and peripheral function. Pull-up on this line must be disabled before assigning this pin to its peripheral function (i.e., the ERASE function). At start-up, the system I/O pin defaults to the ERASE function if it has not been disabled.

To avoid unexpected erase at power-up due to glitches, this pin is debounced by SLCK to improve glitch tolerance and minimum ERASE pin assertion time is required. Note that it is also mandatory to not power-down the device once the hardware erase sequence has been started until the hardware erase time has elapsed. Refer to timings in Electrical Characteristics.

If the ERASE signal is used as a standard I/O in Input or Output mode, note the following considerations and behavior:
Note: If the EFL mode has been set to partially or permanently disabled, the recommendations below do not apply.

- GPIO Input mode—At device start-up, the logic level of the pin must be low to prevent unwanted erasing until the user application has configured this system I/O pin to a standard I/O pin.
- GPIO Output mode—Asserting the pin to low does not erase the Flash.
- GPIO Input mode—Pull-up on this pin (PB2) must be disabled before setting this pin in ERASE signal mode.

During software development, faulty software may put the device into a deadlock and prevent the user from recovering the device.

This may be due to:

- programming an incorrect clock switching sequence
- using pin PB2 as a standard I/O pin
- entering Wait mode without any wake-up events programmed

The only way to recover normal behavior is to erase the Flash by following the steps below:

1. Apply a logic “1” level on the pin PB2 (ERASE signal).
2. Apply a logic “0” level on the pin NRST.
3. Power down, then power up the device.

© 2026 Microchip Technology Inc. and its subsidiaries

4. Maintain the pin PB2 (ERASE signal) at logic “0” level for at least the minimum assertion time after releasing the NRST pin to logic “1” level.

© 2026 Microchip Technology Inc. and its subsidiaries

6. Core and Interconnect

6.1. Cortex-M4 Processor

The Arm Cortex-M4 processor is a high-performance 32-bit processor, offering the following significant benefits to developers:

- Outstanding processing performance combined with fast interrupt handling
- Enhanced system debug with extensive breakpoint and trace capabilities
- Efficient processor core, system and memories
- Ultra-low power consumption with integrated Sleep modes
- Platform security robustness, with integrated memory protection unit (MPU)

The implemented Arm Cortex-M4 is revision r0p1.

For additional information, refer to http://www.arm.com.

The Cortex-M4 processor is built on a high-performance processor core with a 3-stage pipeline Harvard architecture, making it ideal for demanding embedded applications. The processor delivers exceptional power efficiency through an efficient instruction set and extensively optimized design, providing high-end processing hardware including IEEE 754-compliant single-precision floating-point computation, a range of single-cycle and SIMD multiplication and multiply-with-accumulate capabilities, saturating arithmetic and dedicated hardware division. To facilitate the design of cost-sensitive devices, the Cortex-M4 processor implements tightly-coupled system components that reduce processor area while significantly improving interrupt handling and system debug capabilities. The Cortex-M4 processor implements a version of the Thumb instruction set based on Thumb-2 technology, ensuring high code density and reduced program memory requirements.

The Cortex-M4 instruction set provides the exceptional performance expected of a modern 32-bit architecture, with the high code density of 8-bit and 16-bit microcontrollers.

The Cortex-M4 processor closely integrates a configurable NVIC, to deliver industry-leading interrupt performance. The NVIC includes a Non-Maskable interrupt (NMI), and provides up to 8 interrupt priority levels. The tight integration of the processor core and NVIC provides fast execution of Interrupt Service Routines (ISRs), dramatically reducing the interrupt latency. This is achieved through the hardware stacking of registers, and the ability to suspend load-multiple and store-multiple operations. Interrupt handlers do not require wrapping in assembler code, removing any code overhead from the ISRs. A tailchain optimization also significantly reduces the overhead when switching from one ISR to another. To optimize low-power designs, the NVIC integrates with the sleep modes, that include a deep sleep function that enables the entire device to be rapidly powered down while still retaining program state.

6.1.1. System Level Interface

The Cortex-M4 processor provides multiple interfaces to provide high-speed, low-latency memory accesses. It supports unaligned data accesses and implements atomic bit manipulation that enables faster peripheral controls, system spinlocks and thread-safe Boolean data handling.

The Cortex-M4 processor has a memory protection unit (MPU) that provides fine grain memory control, enabling applications to utilize multiple privilege levels, separating and protecting code, data and stack on a task-by-task basis.

6.1.2. Integrated Configurable Debug

The Cortex-M4 processor implements a complete hardware debug solution. This provides high system visibility of the processor and memory through a 2-pin Serial Wire Debug (SWD) port that is ideal for microcontrollers and other small package devices. For system trace the processor integrates an Instrumentation Trace Macrocell (ITM) alongside data watchpoints and a profiling unit.

© 2026 Microchip Technology Inc. and its subsidiaries

The Embedded Trace Macrocell (ETM) delivers unrivaled instruction trace capture in an area far smaller than traditional trace units, enabling many low cost MCUs to implement full instruction trace for the first time. To enable simple and cost-effective profiling of the system events these generate, a stream of software-generated messages, data trace, and profiling information is exported over three different ways:

- Output off-chip using the TPIU, through a single pin, called Serial Wire Viewer (SWV). Limited to ITM system trace.
- Output off-chip using the TPIU, through a 4-bit pin interface. Bandwidth is limited.
- Internally stored in RAM, using the CoreSight ETB. Bandwidth is then optimal but capacity is limited.

The Flash Patch and Breakpoint Unit (FPB) provides up to 8 hardware breakpoint comparators that debuggers can use. The comparators in the FPB also provide remap functions of up to 8 words in the program code in the code memory region. This enables applications stored on a non-erasable, ROM-based microcontroller to be patched if a small programmable memory, for example Flash memory, is available in the device. During initialization, the application in ROM detects, from the programmable memory, whether a patch is required. If a patch is required, the application programs the FPB to remap a number of addresses. When those addresses are accessed, the accesses are redirected to a remap table specified in the FPB configuration, which means the program in the non-modifiable ROM can be patched.

6.1.3. Cortex-M4 Processor Features and Configuration

- Thumb instruction set combines high code density with 32-bit performance
- IEEE 754-compliant single-precision Floating Point Unit (FPU)
- Integrated Sleep modes for low power consumption
- Fast code execution permits slower processor clock or increases Sleep mode time
- Hardware division and fast digital-signal-processing oriented multiply accumulate
- Saturating arithmetic for signal processing
- Deterministic, high-performance interrupt handling for time-critical applications
- Memory Protection Unit (MPU) for safety-critical applications
- Extensive debug and trace capabilities: Serial Wire Debug and Serial Wire Trace reduce the number of pins required for debugging, tracing, and code profiling.

Features	Cortex-M4 Options	Device Configuration
Interrupts	1 to 240	97
Number of priority bits	3 to 8	3 = eight levels of priority
Data endianness	Little-endian or big-endian	Little-endian
SysTick Timer calibration value	SysTick calibration value	25000
MPU	Present or Not present	Present
Debug support level	0 = No debug. No DAP, breakpoints, watchpoints, Flash patch, or halting debug. 1 = Minimum debug. Two breakpoints, one watchpoint, no Flash patch. 2 = Full debug minus DWT data matching. 3 = Full debug plus DWT data matching.	3
Trace support level	0 = No trace. No ETM, ITM or DWT triggers and counters. 1 = Standard trace. ITM and DWT triggers and counters, but no ETM. 2 = Full trace. Standard trace plus ETM. 3 = Full trace plus HTM port.	1
JTAG	Present or Not present	Present

© 2026 Microchip Technology Inc. and its subsidiaries

Cortex-M4 Processor Features and Configuration (continued)

Features	Cortex-M4 Options	Device Configuration
Bit banding	Present or Not present	Present
FPU	Present or Not present	Present

© 2026 Microchip Technology Inc. and its subsidiaries

7. Product Mapping and Peripheral Access

The following figure shows the default memory mapping of the Arm Cortex-M core.

Figure 7-1. Cortex-M Memory Mapping

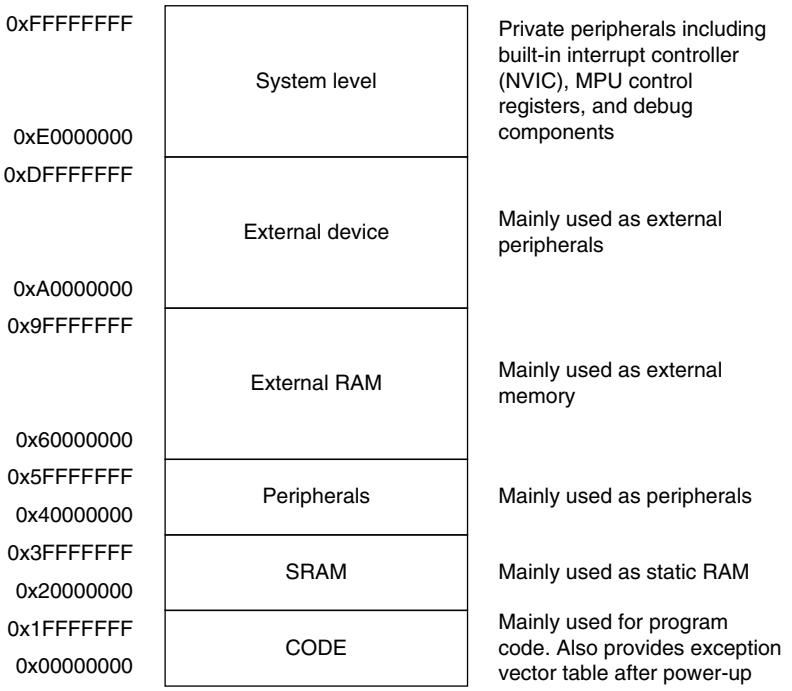


Table 7-1. Code Area

Base Address	End Address	Memory	Size	Comment
0x00000000	–	Boot Memory	16 Mbytes	ROM Code
0x01000000	–	Internal Flash	16 Mbytes	Code - Non Cached
0x02000000	0x0201FFFF	Reserved	–	–
0x02020000	–	CPKCC ROM	64 Kbytes	–
0x02030000	0x0203FFFF	Reserved	–	–
0x02031000	0x02031FFF	CPKCC RAM	–	–
0x02032000	0x03FFFFFF	Undefined (Abort)	–	–
0x04000000	–	QSPI MEM	32 Mbytes	Code - Non Cached
0x06000000	–	QSPI MEM AESB	32 Mbytes	Code through AESB - Non Cached
0x08000000	0x0FFFFFFF	Undefined (Abort)	–	–
0x10000000	–	Undefined (Abort)	16 Mbytes	–
0x11000000	–	Internal Flash	16 Mbytes	Code - Cached
0x12000000	–	Undefined (Abort)	16 Mbytes	–
0x13000000	–	Undefined (Abort)	16 Mbytes	–
0x14000000	–	QSPI MEM	32 Mbytes	Code - Cached
0x16000000	0x1FFF9FFF	QSPI MEM AESB	32 Mbytes	Code through AESB - Cached
0x1FFFA000	0x1FFFBFFF	DTCM	8 Kbytes	–
0x1FFFC000	0x1FFFFFFF	ITCM	16 Kbytes	–

© 2026 Microchip Technology Inc. and its subsidiaries

Table 7-2. SRAM Area

Base Address	End Address	Memory
0x20000000	0x2007FFFF	SRAM0[1]

Note:

1. In a small product configuration, depending on the real size of the embedded SRAM0, the expected responses when accessing SRAM0 area are:
   - accessing 256 Kb/half upper area -> ABORT (SRAM0 size < 512 Kb)
   - accessing 2nd quarter (128 to 256 Kb) -> ABORT (SRAM0 size < 256 Kb

Table 7-3. AHB-to-APB Bridges

Base Address	Instance
0x40000000	BRIDGE 0
0x44000000	BRIDGE 2
0x46000000	BRIDGE 3
0x48000000	BRIDGE 1
0x4A000000	BRIDGE 4

Table 7-4. BRIDGE 0 Peripheral Mapping

Base Address	Peripheral	Comments
0x40000000	FLEXCOM0	
0x40004000	FLEXCOM1	
0x40008000	FLEXCOM2	
0x4000C000	FLEXCOM3	
0x40010000	FLEXCOM4	
0x40014000	FLEXCOM5	
0x40018000	FLEXCOM6	
0x4001C000	FLEXCOM7	
0x40020000	QSPI	
0x40024000	ADC	
0x40028000	ACC	
0x40034000	MEM2MEM0	
0x40038000	TC0	
0x4003C000	TC1	
0x40040000	TC2	
0x40044000	MATRIX1	
0x40048000	PIOA, PIOB, PIOC	
0x4004C000	Reserved	
0x40050000	System Controller	
0x40054000	Reserved	

Table 7-5. BRIDGE 1 Peripheral Mapping

Base Address	Peripherals	Comments
0x48004000	MEM2MEM1	
0x48008000	TC3	
0x4800C000	PIOD	
0x48010000	UART	
0x48018000	MCSPI	

© 2026 Microchip Technology Inc. and its subsidiaries

Table 7-5. BRIDGE 1 Peripheral Mapping (continued)

Base Address	Peripherals	Comments
0x4801C000	PWM	
0x48020000	MATRIX3	
0x48028000	Reserved	

Table 7-6. BRIDGE 2 Peripheral Mapping

Base Address	Peripherals	Comments
0x44000000	AES	
0x44004000	AESB	
0x44008000	SHA	
0x4400C000	TRNG	
0x44010000	ICM	
0x44014000	Reserved	

Table 7-7. BRIDGE 3 Peripheral Mapping

Base Address	Peripherals	Comments
0x46000000	CPKCC	
0x46004000	MATRIX0	
0x46008000	CMCC0	
0x4600C000	CMCC1	
0x46010000	Reserved	
0x460E0000	SEFC0	
0x460E0200	SEFC1	
0x46000000	Reserved	
0x460E0400	Reserved	
0x46800000	PMC	
0x46800200	Reserved	
0x46800400	Reserved	

Table 7-8. BRIDGE 4 Peripheral Mapping

Base Address	Peripherals	Comments
0x4A000000	MATRIX2	
0x4A004000	Reserved	

Table 7-9. System Controller

Base Address	Peripherals	Comments
0x40050000	Reserved	
0x40050200	CHIPID	
0x40050400	SFR	
0x40050600	SFRBU	
0x40050800	Reserved	
0x40052000	SYSC	

Table 7-10. External SRAM

Base Address	Peripherals	Comments
0x60000000 to 0x9FFFFFFF	Undefined (Abort)	

© 2026 Microchip Technology Inc. and its subsidiaries

Table 7-11. External Peripherals

Base Address	Peripherals	Comments
0xA0000000	Undefined	
0xA1000000	Internal Flash Write	Strongly ordered Alias address for Flash Controller Page Buffer
0xA2000000	Undefined	
0xA3000000	Undefined	
0xA4000000	Undefined (Abort)	

© 2026 Microchip Technology Inc. and its subsidiaries

8. Memories

8.1. Embedded Memories

8.1.1. Internal SRAM

The device embeds up to 512 Kbytes of high-speed SRAM with zero wait state access time.

The multibank and multiport RAM architecture provides an efficient way to optimize the access latency to the memory, whatever types of access are performed by the hosts.

A maximum of one system bus clock cycle is required to complete an access to the RAM. The predictability is therefore optimal.

8.1.2. Cache Controller and Tightly Coupled Memory (TCM) Interface

The device features two cache controllers with TCM memories. The total cache memory can be used as cache only, or as a mix of cache and TCM RAM.

Important: When used as TCM, SRAM memories embedded in both cache controllers (CMCC0 and CMCC1) are 32-bit accessible memories only. 8-bit or 16-bit accesses are not allowed. The code to load the ITCM during application initialization must use a **"32-bit wide"** memory copy (memcpy) function.

The CMCC0 is dedicated to Program Code (Instruction Fetch) and CMCC1 is dedicated to Program Data (RO Data, Literal pool). Both cache controllers can be used either for the internal Flash memory or for an external serial memory connected to the QSPI or both.

Cache and tightly-coupled memories allow different run-time memory mapping scenarios using TCM to adapt for "real-time response" and/or increase execution speed from Flash using cache. Both cache and TCM can be used at the same time, and memory size for each can be allocated at build time.

The DTCM accessible only by the core may be used as storage memory for the main stack typically used by the kernel of OS/RTOS. The process stack used by OS/RTOS tasks can be located in SRAM0. This creates a physical separation between the two stacks and, as a result, a safety level is added.

8.1.3. System ROM

The device embeds an Internal ROM for the boot program, which contains SAM-BA Monitor, Secure SAM-BA Monitor, and the Fast Flash Programming Interface (FFPI). The device always boots from this memory regardless of the settings of the GPNVM bits.

8.1.4. CPKCC ROM

The ROM contains a cryptographic library using the CPKCC cryptographic accelerator peripheral (CPKCC) to provide support for Rivest Shamir Adleman (RSA), Elliptic Curve Cryptography (ECC), Digital Signature Algorithm (DSA) and Elliptic Curve Digital Signature Algorithm (ECDSA).

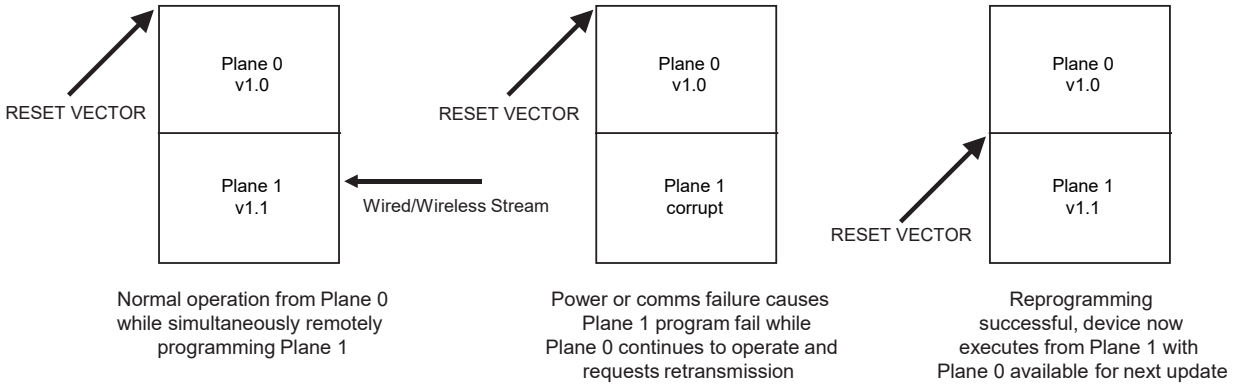
8.1.5. Embedded Flash

8.1.5.1. Flash Overview

The device features a dual-plane Flash to program or erase a memory plane while reading from the other plane. The dual-plane capability also provides the dual boot scheme. The benefit of the dual plane and the dual boot is that the firmware can be upgraded while the main application is running and that it is possible to switch to the new firmware in the other plane. The figure below shows the operating principle of firmware upgrade by using the dual bank/dual boot. Suspend and resume of write and erase operations are also available on each plane.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 8-1. Dual Plane and Dual Boot Firmware Upgrade



8.1.5.2. Memory Plane and Sector Organization

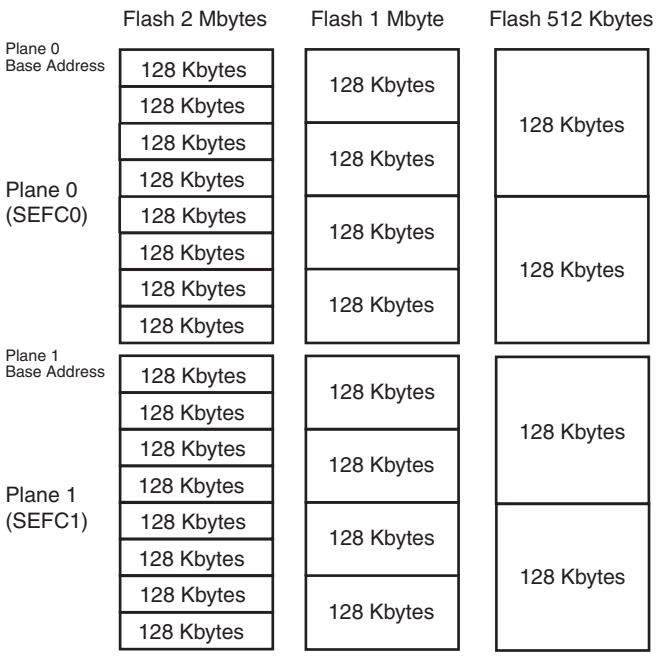
Each memory plane is organized in sectors. Each sector has a physical size of 128 Kbytes.

Each sector is organized in 32 blocks of 4 Kbytes. One block is made of 8 pages of 512 bytes.

- Write Operation
  - By page
- Erase Operation
  - By 4-Kbyte blocks
  - By 2 x 4-Kbyte blocks
  - By 4 x 4-Kbyte blocks
  - By 8 x 4-Kbyte blocks
  - By sectors of 128 Kbytes
  - Full Chip Erase by HW erase signal

The figure below illustrates the organization of the Flash depending on its size.

Figure 8-2. Flash Size



© 2026 Microchip Technology Inc. and its subsidiaries

8.1.5.2.1. Secure Embedded Flash Controller (SEFC)

The Secure Embedded Flash Controller (SEFC) manages accesses performed by hosts of the system. It enables reading the Flash and writing the write buffer. It also contains a user interface, mapped on the APB.

The SEFC ensures the interface of the Flash block. It manages the programming, erasing, locking and unlocking sequences of the Flash using the full set of commands.

For dual-plane devices, each plane is managed by an SEFC. Flash Wait State and Flash Optimization mode must be set up in both SEFCs.

8.1.5.2.2. Flash Speed

The user must set the number of Wait states depending on the frequency used.

For more details, refer to Embedded Flash Characteristics of the section Electrical Characteristics.

8.1.5.2.3. Lock Regions

Several lock bits are used to protect write and erase operations on lock regions. A lock region is composed of several consecutive pages, and each lock region has its associated lock bit.

Table 8-1. Lock Bit Number

Flash Size	Number of Lock Bits	Lock Region Size
2 Mbytes	256 (128 + 128)	8 Kbytes
1 Mbyte	128 (64 + 64)	8 Kbytes
512 Kbytes	64 (32 + 32)	8 Kbytes

The lock bits are software programmable through the SEFC user interface. The command “Set Lock Bit” enables the protection. The command “Clear Lock Bit” unlocks the lock region.

Asserting the ERASE signal clears the lock bits.

8.1.5.2.4. Unique Identifier

Each device integrates its own 128-bit unique identifier. These bits are factory-configured and cannot be changed by the user. The ERASE signal has no effect on the unique identifier.

8.1.5.2.5. User Signature

The memory has 8 x 4 Kbytes of additional user reprogrammable pages. The ROM code uses the User Signature Area [0]. The User Signature Area [0] is locked for read/erase/write before starting the customer application in Flash. The User Signature Areas [1...7] can be used by the customer. User Signature Areas 0 to 3 erase upon hardware erase signal assertion. User signature areas 4 to 7 are not erased.

8.1.5.2.6. Boot Process

The device always boots at address 0x0 from the ROM. Depending on the GPNVM bits [8:5], the ROM code instructs the Core 0 to execute code in Flash either by a non-secure boot process (SAM-BA Monitor or Standard Boot) or by a secure boot process (Secure SAM-BA Monitor or Secure Boot).

After power-up or full system reset, the second peripheral set is maintained in reset and with no clock.

The second peripheral set consists of the following peripherals: MEM2MEM1 (ID78), TC3(ID79 to 84) , PIOD (ID85/86), UART (ID87), MCSPI (ID89), PWM (ID90) and are clocked by the Main System Bus Clocks MCK1, MCK1DIV and the reset line managed by CPEREN bit (second peripheral set enable in RSTC.MR).

© 2026 Microchip Technology Inc. and its subsidiaries

8.2. External Memories

The QSPI provides the interface to a wide range of external serial memories for Execute-In-Place for code execution or raw data storage. Code execution may benefit from the use of the cache controller.

© 2026 Microchip Technology Inc. and its subsidiaries

9. Safety and Security Features

9.1. Design for Safety

Specific design techniques have been used in the device to resolve general purpose safety concerns. This allows reduced software development and code size, as well as savings on external hardware circuitry, since built-in self-tests are already embedded in the device. The table below gives the list of peripherals which incorporate these techniques for general purpose safety considerations.

Table 9-1. Safety Features

Peripheral	Component	Fault/Error/Feature
PMC, SUPC	Clock	MCK frequency monitor - MCK out-of-range operation
		32.768 kHz crystal oscillator frequency monitor - Abnormal frequency deviation - Failure
		Main crystal oscillator failure detector - Crystal failure detection
ICM	Memories	Any error detection
SEFC	Embedded nonvolatile memory	Single bit error correction
		Double bit error detection
System Controller	All	Safety critical peripheral logic or circuitry is fed by the always-on slow RC oscillator - WDT, RSTC, start-up counters, timeout counters, etc.
PIO	I/O lines	Digital I/O - Plausibility check
ADC		Analog I/O and ADC converter - Plausibility check
WDT	Watchdog	Watchdog is driven by an internal always ON clock - Program counter stuck at faults
		Watchdog configuration can be locked until the next reset - Errant writes (programming errors, errors introduced by system or hardware failures)
		Watchdog overflow generates a reset or interrupt
Cortex-M4 MPU	Memory Protection Unit	Cortex-M4 Memory Protection Unit
MATRIX, RTC, RTT, RSTC, PMC, PIOC, FLEXCOM, QSPI, TC, ADC	Peripherals	Configuration, Interrupt Enable/Disable and Control registers can be independently write-protected - Errant writes (programming errors, errors introduced by system or hardware failures)
RTC	Peripheral	Robust against crystal oscillator glitches. The design of the RTC 32.768 kHz divider does not propagate glitches downstream to time/date counter.

9.2. Design for Security

The device embeds peripherals with security features to prevent counterfeiting, to secure external communication, and to authenticate the system.

The table below provides the list of peripherals and an overview of their security function. For more information, refer to the section on each peripheral.

Peripheral	Function	Description	Comments
Secure Boot	Trusted Code Authentication	Based on hardware accelerated cryptographic modules	Software + AES, SHA

© 2026 Microchip Technology Inc. and its subsidiaries

Design for Security (continued)			
Peripheral	**Function**	**Description**	**Comments**
Cortex-M4 MPU	Memory Protection Unit	AES/SHA	–
PIO	I/O Control/ Peripheral Access	When a peripheral is not selected (PIO-controlled), I/O lines have no access to the peripheral.	–
Classical Public Key Crypto Library (CPKCL)	Cryptography standards	Software ECC (Asymmetric key algorithm, elliptic curves)	Software library
		Software RSA (Asymmetric key algorithm)	–
AES		Hardware-accelerated AES up to 256 bits	FIPS-compliant
SHA		SHA up to 512 bits and HMAC-SHA	FIPS-compliant
TRNG	Cryptography	True Random Number Generator	–
AES	Anti-tampering	Immediate clear of keys in case of external tamper event detection (if enabled)	–
AES, SHA	Reinforced security	Improved robustness against attack	–
	Integrity check	User-configurable immediate stop on integrity error detection	When the security error status flag is set in the user interface, an integrity error has been detected. These errors can occur only under abnormal operating conditions (electromagnetic attacks, VDD glitches, etc.)
AES, SHA, TC, ICM, TRNG	Software access monitoring and integrity check	Dedicated interrupt line for security event	Non-maskable in the peripheral. When a security error status flag is set in the user interface, an interrupt is triggered. These errors can occur only under abnormal operating conditions (electromagnetic attacks, VDD glitches, etc.)
Flash	Security bit	Reinforced Security bit to disable JTAG access Secure area key storage	–
Flash, GPBR	HW Flash Erase signal	Flash memory array is erased prior to Security bit erase	–
		SRAM content can be configured to be erased on ERASE pin activation	
		GPBR can be configured to be erased on ERASE pin activation	
		AES keys and QSPI scrambling keys are erased on ERASE pin activation	
QSPI	Scrambling	On-the-fly zero-wait state or zero latency scrambling/unscrambling	–

© 2026 Microchip Technology Inc. and its subsidiaries

10. Real-Time Event Management

The events generated by peripherals are designed to be directly routed to peripherals managing/using these events without processor intervention. Peripherals receiving events contain logic to select the required event.

10.1. Real-Time Event Mapping List

Table 10-1. Real-Time Event Mapping List

Function	Description	Source	Destination
Security	Immediate clear (asynchronous) on anti-tamper detection through pins	Anti-tamper inputs (TMPx)	GPBR, AES/AESB/QSPI Keys SHA Lock
Safety	Automatic switch to reliable RC oscillators in case of crystal failure	PMC	SUPC
Safety	Invalid 12-bit ADC supply voltage	ADC	ADC IRQ
Safety	Invalid VBAT or VDD3V3 voltage	Supply monitors	Power switch, SUPC
Safety	Puts the PWM outputs in a pre-defined Safe state in case of main crystal clock failure	PMC	PWM
	Puts the PWM outputs in a pre-defined Safe state in case of ADC Comparison event	ADC	
	Puts the PWM outputs in a pre-defined Safe state in case of Timer Counter overflow on speed and/or position event	TC	
	Puts the PWM outputs in a pre-defined Safe state in case of IO lines level change	PD1, PD12	
Safety	32 kHz crystal oscillator failure in Backup mode	SUPC	SUPC event
Measurement Trigger	Trigger source selection in ADC	Timer Counter Block 0 - TIOA0	ADC trigger
		Timer Counter Block 0 - TIOA1	
		Timer Counter Block 0 - TIOA2	
		Timer Counter Block 1 - TIOA0	
		Timer Counter Block 1 - TIOA1	
		PWM Event Line 0	
	Trigger source selection in TC	PWM triggers	TC trigger
	Channel measurement	RTCOUT0	ADC trigger
	Low sampling rate temperature sensor measurement	RTCOUT1	

© 2026 Microchip Technology Inc. and its subsidiaries

11. Peripherals

11.1. Peripheral Identifiers

The table below defines the peripheral identifiers. A peripheral identifier is required for the control of the peripheral interrupt with the NVIC of the core, and for the control of the peripheral clock with the PMC.

The two Arm Cortex-M4 processors share the same interrupt mapping, and thus, they share the interrupts of the peripherals common to both cores.

To prevent any single software error from corrupting a peripheral's behavior, certain registers in the peripheral's address space can be write-protected. For further details, refer to the information on register write protection in the section of the corresponding peripheral, as well as to the section Register Write Protection in System Controller Write Protection (SYSCWP).

Each peripheral that has both a peripheral clock and a GCLK needs its peripheral clock enabled to work correctly. The GCLK alone is not sufficient.

Moreover, a minimum ratio between the peripheral clock (PCK) and the GCLK may be required (configuration of GCLK source, GCLKCSS, and GCLK ratio, GCLKDIV). For more information, refer to the corresponding peripheral section.

Table 11-1. Peripheral Identifiers

Instance ID	Instance Name	NVIC Interrupt	PMC Clock Control	Generic Clock	Main System Bus Clock	Register Write Protection	Instance Description
0	SUPC	X	–	–	–	X	Supply Controller
1	RSTC	X	–	–	–	X	Reset Controller
2	RTC	X	–	–	–	X	Real Time Clock
3	RTT	X	–	–	–	X	Real Time Timer
4	WDT0	X	–	–	–	X	Dual Watchdog Timer 0
5	WDT1	X	–	–	–	–	Dual Watchdog Timer 1
6	PMC	X	–	–	–	X	Power Management Controller
7	SEFC0	X	–	–	MCK0[1]	X	Embedded Flash Controller 0
8	SEFC1	X	–	–	MCK0[1]	X	Embedded Flash Controller 1
9	FLEXCOM0	X	X	X	MCK0DIV	X	FLEXCOM 0 (USART0/SPI0/TWI0)
10	FLEXCOM1	X	X	X	MCK0DIV	X	FLEXCOM 1 (USART1/SPI1/TWI1)
11	FLEXCOM2	X	X	X	MCK0DIV	X	FLEXCOM 2 (USART2/SPI2/TWI2)
12	FLEXCOM3	X	X	X	MCK0DIV	X	FLEXCOM 3 (USART3/SPI3/TWI3)
13	FLEXCOM4	X	X	X	MCK0DIV	X	FLEXCOM 4 (USART4/SPI4/TWI4)
14	FLEXCOM5	X	X	X	MCK0DIV	X	FLEXCOM 5 (USART5/SPI5/TWI5)
15	FLEXCOM6	X	X	X	MCK0DIV	X	FLEXCOM 6 (USART6/SPI6/TWI6)
16	FLEXCOM7	X	X	X	MCK0DIV	X	FLEXCOM 7 (USART7/SPI7/TWI7)

© 2026 Microchip Technology Inc. and its subsidiaries

Table 11-1. Peripheral Identifiers (continued)

Instance ID	Instance Name	NVIC Interrupt	PMC Clock Control	Generic Clock	Main System Bus Clock	Register Write Protection	Instance Description
17	PIOA	X	X	–	MCK0DIV	X	Parallel I/O Controller A (PIOA)
18	PIOA	SEC	–	–	MCK0DIV	X	Parallel I/O Controller A (PIOA) Secure Event Interrupt
19	PIOB	X	–	–	MCK0DIV	X	Parallel I/O Controller (B PIOB)
20	PIOB	SEC	–	–	MCK0DIV	X	Parallel I/O Controller (B PIOB) Secure Event Interrupt
21	PIOC	X	–	–	MCK0DIV	X	Parallel I/O Controller C (PIOC)
22	PIOC	SEC	–	–	MCK0DIV	X	Parallel I/O Controller C (PIOC) Secure Event Interrupt
23	QSPI	X	X	X	MCK0DIV	X	Quad IO Serial Peripheral Interface
24	ADC	X	X	X	MCK0DIV	X	Analog to Digital Converter
25	ACC	X	X	–	MCK0DIV	X	Analog Comparator
26	ARM	FPU	–	–	–	–	Floating Point Unit except IXC
27	ARM	IXC	–	–	–	–	FPU Interrupt IXC associated with FPU cumulative exception bit
30	MEM2MEM0	X	X	–	MCK0DIV	X	Memory to Memory Transfer Controller (MEM2MEM0)
31	TC0	CHANNEL0	X	X	MCK0DIV	X	Timer Counter 0, Channel 0
32	TC0	CHANNEL1	X	–	MCK0DIV	–	Timer Counter 0, Channel 1
33	TC0	CHANNEL2	X	–	MCK0DIV	–	Timer Counter 0, Channel 2
34	TC1	CHANNEL0	X	X	MCK0DIV	X	Timer Counter 1, Channel 0
35	TC1	CHANNEL1	X	–	MCK0DIV	–	Timer Counter 1, Channel 1
36	TC1	CHANNEL2	X	–	MCK0DIV	–	Timer Counter 1, Channel 2
37	TC2	CHANNEL0	X	X	MCK0DIV	X	Timer Counter 2, Channel 0
38	TC2	CHANNEL1	X	–	MCK0DIV	–	Timer Counter 2, Channel 1
39	TC2	CHANNEL2	X	–	MCK0DIV	–	Timer Counter 2, Channel 2
40	TC0	C0SEC	–	–	MCK0DIV	X	Timer Counter 0, Channel 0, Secure IRQ
41	TC0	C1SEC	–	–	MCK0DIV	–	Timer Counter 0, Channel 1, Secure IRQ
42	TC0	C2SEC	–	–	MCK0DIV	–	Timer Counter 0, Channel 2, Secure IRQ

© 2026 Microchip Technology Inc. and its subsidiaries

Table 11-1. Peripheral Identifiers (continued)

Instance ID	Instance Name	NVIC Interrupt	PMC Clock Control	Generic Clock	Main System Bus Clock	Register Write Protection	Instance Description
43	TC1	C0SEC	–	–	MCK0DIV	X	Timer Counter 1, Channel 0, Secure IRQ
44	TC1	C1SEC	–	–	MCK0DIV	–	Timer Counter 1, Channel 1, Secure IRQ
45	TC1	C2SEC	–	–	MCK0DIV	–	Timer Counter 1, Channel 2, Secure IRQ
46	TC2	C0SEC	–	–	MCK0DIV	X	Timer Counter 2, Channel 0, Secure IRQ
47	TC2	C1SEC	–	–	MCK0DIV	–	Timer Counter 2, Channel 1, Secure IRQ
48	TC2	C2SEC	–	–	MCK0DIV	–	Timer Counter 2, Channel 2, Secure IRQ
49	AES	X	X	–	MCK0DIV	X	Advanced Encryption Standard
50	AES	AESSEC	–	–	MCK0DIV	–	AES Secure Event Interrupt
51	AESB	X	X	–	MCK0DIV	X	AES Bridge
52	AESB	AESBSEC	–	–	MCK0DIV	X	AES Bridge Secure Interrupt
53	SHA	X	X	–	MCK0DIV	X	Secure Hash Algorithm
54	SHA	SHASEC	–	–	MCK0DIV	–	SHA Secure Event Interrupt
55	TRNG	X	X	–	MCK0DIV	X	True Random Number Generator
56	TRNG	TRNGSEC	–	–	MCK0DIV	–	TRNG Secure Event Interrupt
57	ICM	X	X	–	MCK0DIV	X	Integrity Check Module
58	ICM	ICMSEC	–	–	MCK0DIV	X	Integrity Check Module
59	CPKCC	X	X	–	MCK0	–	Public Key Cryptography Controller
60	MATRIX0	X	–	–	MCK0	X	High-Speed Matrix (MATRIX0)
61	MATRIX1	X	–	–	MCK0DIV	X	Low-Speed Matrix (MATRIX1)
62	SUPC	WKUP3	–	–	–	–	External interrupt 3
63	SUPC	WKUP4	–	–	–	–	External interrupt 4
64	SUPC	WKUP5	–	–	–	–	External interrupt 5
65	SUPC	WKUP6	–	–	–	–	External interrupt 6
66	SUPC	WKUP7	–	–	–	–	External interrupt 7
67	SUPC	WKUP8	–	–	–	–	External interrupt 8
68	SUPC	WKUP9	–	–	–	–	External interrupt 9
69	SUPC	WKUP10	–	–	–	–	External interrupt 10
70	SUPC	WKUP11	–	–	–	–	External interrupt 11
71	SUPC	WKUP12	–	–	–	–	External interrupt 12
72	SUPC	WKUP13	–	–	–	–	External interrupt 13
73	SUPC	WKUP14	–	–	–	–	External interrupt 14
74	–	–	–	–	–	–	–

© 2026 Microchip Technology Inc. and its subsidiaries

Table 11-1. Peripheral Identifiers (continued)

Instance ID	Instance Name	NVIC Interrupt	PMC Clock Control	Generic Clock	Main System Bus Clock	Register Write Protection	Instance Description
78	MEM2MEM1	X	X	–	MCK1DIV	X	Memory to Memory Transfer Controller 1 (MEM2MEM1)
79	TC3	CHANNEL0	X	X	MCK0DIV	X	Timer Counter 3, Channel 0
80	TC3	CHANNEL1	X	–	MCK0DIV	X	Timer Counter 3, Channel 1
81	TC3	CHANNEL2	X	–	MCK0DIV	X	Timer Counter 3, Channel 2
82	TC3	C0SEC	–	–	MCK0DIV	X	Timer Counter 3, Channel 0, Secure IRQ
83	TC3	C1SEC	–	–	MCK0DIV	–	Timer Counter 3, Channel 1, Secure IRQ
84	TC3	C2SEC	–	–	MCK0DIV	–	Timer Counter 3, Channel 2, Secure IRQ
85	PIOD	X	X	–	MCK1DIV	–	Parallel I/O Controller D (PIOD)
86	PIOD	SEC	–	–	MCK1DIV	X	Parallel I/O Controller D Secure Interrupt (named PIOD)
87	UART	X	X	X	MCK1DIV	X	Optical UART
89	MCSPI	X	X	X	MCK1DIV	X	Multi-Channel SPI
90	PWM	X	X	–	MCK1DIV	–	Pulse Width Modulation
94	MATRIX2	X	–	–	MCK1	X	High-SpeedMatrix (MATRIX2)
95	MATRIX3	X	–	–	MCK1DIV	X	Low-Speed Matrix (MATRIX3)

Note:

1. SEFC0/SEFC1 clocks are MCK0DIV2 clock after power-up and reset and during the whole execution of the ROM Code. The end user application must set the SEFC clock to CPU clock (MCK0) for maximum performance. Refer to the Flash configuration register SFR_FLASH in the section Special Function Registers (SFR).

11.2. Peripheral DMA Controller (PDC)

Features of the PDC include:

- Data Transfer Handling between Peripherals and Memories
- Low Bus Arbitration Overhead
  - One Main System Bus clock cycle needed for a transfer from memory to peripheral
  - Two Main System Bus clock cycles needed for a transfer from peripheral to memory
- Next Pointer Management Reduces Interrupt Latency Requirement

The PDC handles transfer requests from the channel according to the priorities given in the following tables (low-to-high priorities).

Table 11-2. Peripheral DMA Controller (PDC0)

Instance Name	Channel T/R	Channel Number
ADC	Receive	10
FLEXCOM0	Transmit	14

© 2026 Microchip Technology Inc. and its subsidiaries

Table 11-2. Peripheral DMA Controller (PDC0) (continued)

Instance Name	Channel T/R	Channel Number
FLEXCOM0	Receive	0
FLEXCOM1	Transmit	15
FLEXCOM1	Receive	1
FLEXCOM2	Transmit	16
FLEXCOM2	Receive	2
FLEXCOM3	Transmit	17
FLEXCOM3	Receive	3
FLEXCOM4	Transmit	20
FLEXCOM4	Receive	6
FLEXCOM5	Transmit	21
FLEXCOM5	Receive	7
FLEXCOM6	Transmit	22
FLEXCOM6	Receive	8
FLEXCOM7	Transmit	23
FLEXCOM7	Receive	9
MEM2MEM0	Transmit	19
MEM2MEM0	Receive	5
QSPI	Transmit	18
QSPI	Receive	4
TC0	Receive	13
TC1	Receive	12
TC2	Receive	11

Table 11-3. Peripheral DMA Controller (PDC1)

Instance Name	Channel T/R	Channel Number
MCSPI	Transmit	6
MCSPI	Receive	1
MEM2MEM1	Transmit	9
MEM2MEM1	Receive	3
PWM	Transmit	7
TC3	Receive	5
UART	Transmit	8
UART	Receive	2

Table 11-4. Peripheral DMA Controller (PDC2)

Instance Name	Channel T/R	Channel Number
SHA	Transmit	2
AES	Transmit	1
AES	Receive	0

© 2026 Microchip Technology Inc. and its subsidiaries

12. Cortex-M4 Processor (Arm)

12.1. Description

The Cortex-M4 processor is a high performance 32-bit processor designed for the microcontroller market. It offers significant benefits to developers, including outstanding processing performance combined with fast interrupt handling, enhanced system debug with extensive breakpoint and trace capabilities, efficient processor core, system and memories, ultra-low power consumption with integrated sleep modes, and platform security robustness, with integrated memory protection unit (MPU).

The Cortex-M4 processor is built on a high-performance processor core, with a 3-stage pipeline Harvard architecture, making it ideal for demanding embedded applications. The processor delivers exceptional power efficiency through an efficient instruction set and extensively optimized design, providing high-end processing hardware including IEEE754-compliant single-precision floating-point computation, a range of single-cycle and SIMD multiplication and multiply-with-accumulate capabilities, saturating arithmetic and dedicated hardware division.

To facilitate the design of cost-sensitive devices, the Cortex-M4 processor implements tightly-coupled system components that reduce processor area while significantly improving interrupt handling and system debug capabilities. The Cortex-M4 processor implements a version of the Thumb® instruction set based on Thumb-2 technology, ensuring high code density and reduced program memory requirements. The Cortex-M4 instruction set provides the exceptional performance expected of a modern 32-bit architecture, with the high code density of 8-bit and 16-bit microcontrollers.

The Cortex-M4 processor closely integrates a configurable NVIC, to deliver industry-leading interrupt performance. The NVIC includes a non-maskable interrupt (NMI), and provides up to 256 interrupt priority levels. The tight integration of the processor core and NVIC provides fast execution of interrupt service routines (ISRs), dramatically reducing the interrupt latency. This is achieved through the hardware stacking of registers, and the ability to suspend load-multiple and store-multiple operations. Interrupt handlers do not require wrapping in assembler code, removing any code overhead from the ISRs. A tail-chain optimization also significantly reduces the overhead when switching from one ISR to another.

To optimize low-power designs, the NVIC integrates with the sleep modes, that include a deep sleep function that enables the entire device to be rapidly powered down while still retaining program state.

12.2. Reference Documents

Document Title	Available
Arm Cortex-M4 Technical Reference Manual	developer.arm.com/
ARMv7-M Architecture Reference Manual	

12.3. Embedded Characteristics

- Tight integration of system peripherals reduces area and development costs
- Thumb instruction set combines high code density with 32-bit performance
- IEEE754-compliant single-precision FPU
- Code-patch ability for ROM system updates
- Power control optimization of system components
- Integrated sleep modes for low power consumption
- Fast code execution permits slower processor clock or increases sleep mode time

© 2026 Microchip Technology Inc. and its subsidiaries

- Hardware division and fast digital-signal-processing oriented multiply accumulate
- Saturating arithmetic for signal processing
- Deterministic, high-performance interrupt handling for time-critical applications
- Memory Protection Unit (MPU) for safety-critical applications
- Extensive debug and trace capabilities:
  - Serial Wire Debug and Serial Wire Trace reduce the number of pins required for debugging, tracing, and code profiling

© 2026 Microchip Technology Inc. and its subsidiaries

13. Flash Programming, Debug and Test Features

13.1. Flash Programming Sequence

The device can be programmed in several ways during development and during production.

During development, debugger and/or Flash downloader tools program the application in Flash through the Serial Wire/JTAG Debug Port and set the corresponding GPNVM bit according to the desired Boot mode on the next start-up. Lock bits and the Security bit are generally set in the application code.

During end-product manufacturing, the main memory array, GPNVM bits, Lock bits and the Security bit may be set.

The following sequence must be respected when considering setting the security bit either during the development or the production phase:

1. Program the main memory array (one or several binary images).
2. Program the lock bits (optional).
3. Program the GPNVM bits according to the desired Boot mode.
4. Program the Security bit.

13.2. Description

A number of complementary debug and test capabilities are available to users. The Serial Wire/JTAG Debug Port (SWJ-DP) combines a Serial Wire Debug port (SW-DP) and a JTAG Debug port (JTAG-DP). The SWJ-DP is used for standard debugging functions, such as downloading code and single-stepping through programs. It also embeds a serial wire trace.

13.2.1. Associated Documentation

The device implements the standard Arm CoreSight™ Macrocell. For information on CoreSight, the following reference documents are available from the Arm website:

- CoreSight Technology System Design Guide (ARM DGI 0012D)
- CoreSight Components Technical Reference Manual (ARM DDI 0314H)
- Arm Debug Interface v5 Architecture Specification (Doc. ARM IHI 0031A)

13.2.2. Embedded Characteristics

- Debugging with Serial Wire Debug Port (SW-DP) or JTAG Debug Port (JTAG-DP) Debug Access Port
- Intrumentation Trace Macrocell (ITM) on Both Cores for Support of 'printf' Style Debugging
- Debug Access to All Memory and Registers in the System, including Cortex-M4 Register Bank when the Core is Running, Halted, or Held in Reset
- Flash Patch and Breakpoint (FPB) Unit for Implementing Breakpoints and Code Patches
- Data Watchpoint and Trace (DWT) Unit for Implementing Watchpoints, Data Tracing, and System Profiling
- IEEE 1149.1 JTAG Boundary Scan on All Digital Pins

13.2.2.1. JTAG Debug Port (JTAG-DP) and Serial Wire Debug Port (SW-DP) Pins

The SWJ-DP is a combined JTAG-DP and SW-DP. SWJ-DP pins are TCK/SWCLK, TMS/SWDIO, TDO/TRACESWO, TDI and commonly provided on a standard 20-pin JTAG connector defined by Arm.

At start-up, SWJ-DP pins are configured in SW-DP mode. SWJ-DP pins can be used as standard I/Os to provide users with more general input/output pins when the debug port is not needed in the end application. Mode selection between SWJ-DP modes is performed through the PIO Controller

© 2026 Microchip Technology Inc. and its subsidiaries

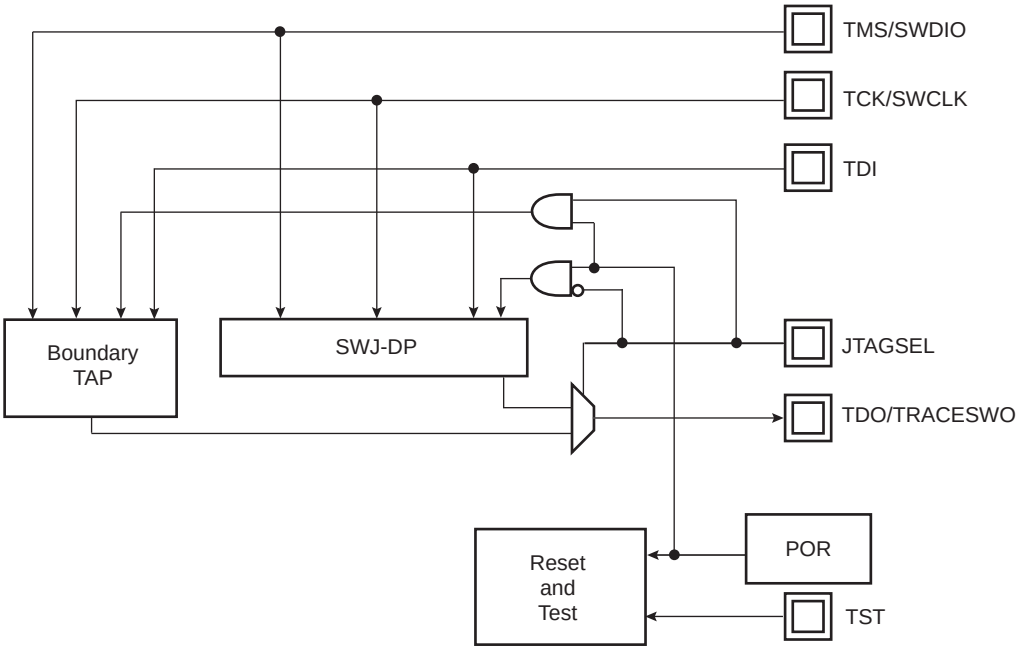
Registers. Configuration of the pad for pull-up, triggers, debouncing and glitch filters is possible regardless of the mode.

The JTAGSEL pin is used to select the JTAG boundary scan when asserted at a high level. It integrates a permanent pull-down resistor of about 15 k to GND, so that it can be left unconnected for normal operations.

By default, the Serial Wire Debug Port (SW-DP) is active. If the debugger host wants to switch to the JTAG-DP, it must provide a dedicated JTAG sequence on TMS/SWDIO and TCK/SWCLK which disables the SWD-DP and enables the JTAG-DP. When the Serial Wire Debug Port is active, TDO/TRACESWO can be used for instrumentation trace.

The asynchronous TRACE output (TRACESWO) is multiplexed with TDO. The asynchronous trace can only be used with SW-DP, not JTAG-DP. The SWJ-DP pins are used for debug access to both cores.

Figure 13-1. Debug and Test Block Diagram



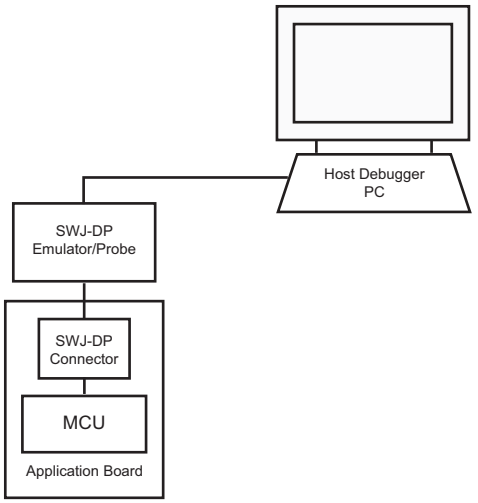
13.2.3. Application Examples

13.2.3.1. Debug Environment

The figure below shows a complete debug environment example. The ICE/JTAG interface is used for standard debugging functions, such as downloading code and single-stepping through the program. A software debugger running on a personal computer provides the user interface for configuring a Trace Port interface utilizing the ICE/JTAG interface.

© 2026 Microchip Technology Inc. and its subsidiaries

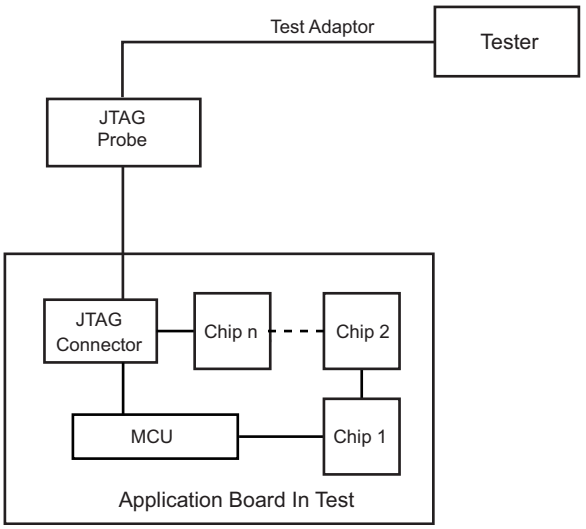
Figure 13-2. Application Debug and Trace Environment Example



13.2.3.2. Test Environment

The figure below shows a test environment example. Test vectors are sent and interpreted by the tester. In this example, the “board in test” is designed using a number of JTAG-compliant devices. These devices can be connected to form a single scan chain.

Figure 13-3. Application Test Environment Example



13.2.4. Debug and Test Signal Description

Table 13-1. Debug and Test Signal List

Pin Name	Function	Type	Active Level
Reset/Test			
NRST	Microcontroller Reset	I/O	Low
TST	Test Select	Input	–
SWD/JTAG			
TCK/SWCLK	Test Clock/Serial Wire Clock	Input	–

© 2026 Microchip Technology Inc. and its subsidiaries

Table 13-1. Debug and Test Signal List (continued)

Pin Name	Function	Type	Active Level
TDI	Test Data In	Input	–
TDO/TRACESWO	Test Data Out/Trace Asynchronous Data Out	Output	–
TMS/SWDIO	Test Mode Select/Serial Wire Input/Output	Input	–
JTAGSEL	JTAG Selection	Input	High

13.2.5. Functional Description

13.2.5.1. Test Pin

One dedicated pin, TST, is used to define the device operating mode. When this pin is at low level during power-up, the device is in normal operating mode. When at high level, the device is in Test mode or FFPI mode. The TST pin integrates a permanent pull-down resistor of about 15 kΩ, so that it can be left unconnected for normal operation. Note that when setting the TST pin to low or high level at power-up, the pin must remain in the same state for the duration of the operation.

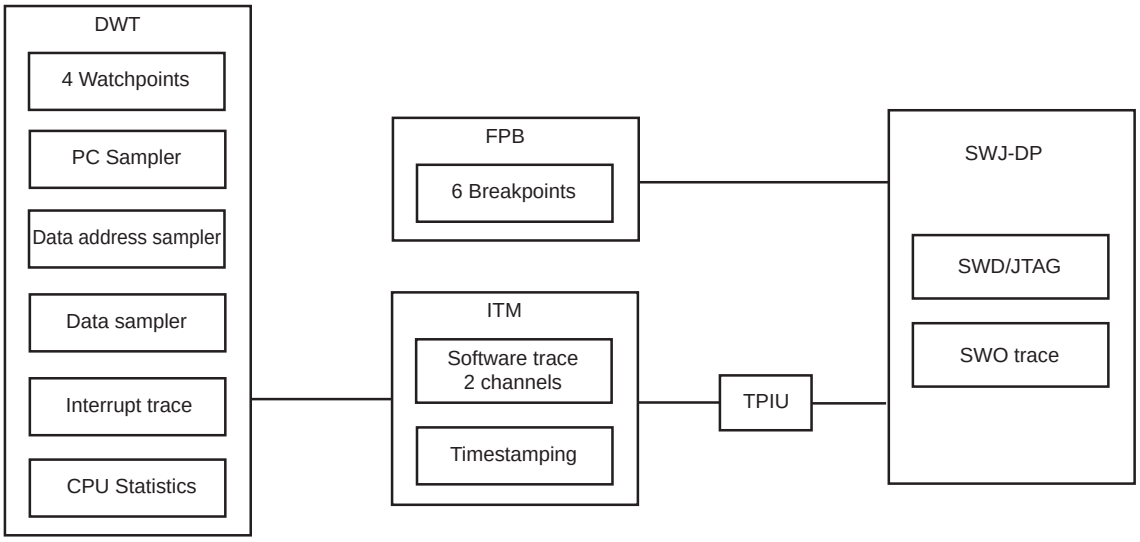
13.2.5.2. Debug Architecture

The figure below illustrates the debug architecture. The Cortex-M4F embeds four functional units for debug:

- SWJ-DP (Serial Wire/JTAG Debug Port)
- FPB (Flash Patch Breakpoint)
- DWT (Data Watchpoint and Trace)
- ITM (Instrumentation Trace Macrocell)
- TPIU (Trace Port Interface Unit)

The information that follows is mainly dedicated to developers of SWJ-DP Emulators/Probes and debugging tool vendors for Cortex-M4 based microcontrollers. For further details on SWJ-DP, see the Cortex-M4 technical reference manual.

Figure 13-4. Debug Architecture



13.2.5.3. Serial Wire/JTAG Debug Port (SWJ-DP)

The Cortex-M4 embeds a SWJ-DP Debug port which is the standard CoreSight debug port. It combines Serial Wire Debug Port (SW-DP), from 2 to 3 pins and JTAG Debug Port (JTAG-DP), 5 pins.

© 2026 Microchip Technology Inc. and its subsidiaries

By default, the SW-DP is active. If the host debugger wants to switch to the JTAG-DP, it must provide a dedicated JTAG sequence on TMS/SWDIO and TCK/SWCLK. This disables SW-DP and enables JTAG-DP.

When SW-DP is active, TDO/TRACESWO can be used for trace. The asynchronous TRACE output (TRACESWO) is multiplexed with TDO and thus the asynchronous trace can only be used with SW-DP.

SW-DP or JTAG-DP mode is selected when JTAGSEL is low. It is not possible to switch directly between SWJ-DP and JTAG boundary scan operations. A chip reset must be performed after JTAGSEL is changed.

Table 13-2. SWJ-DP Pin List

Pin Name	JTAG Port	Serial Wire Debug Port
TMS/SWDIO	TMS	SWDIO
TCK/SWCLK	TCK	SWCLK
TDI	TDI	–
TDO/TRACESWO	TDO	TRACESWO (optional: trace)

13.2.5.3.1. SW-DP and JTAG-DP Selection

Debug port selection is done by sending a specific SWDIOTMS sequence.

- Switch from JTAG-DP to SW-DP. The sequence is:
  - Send more than 50 SWCLKTCK cycles with SWDIOTMS = 1
  - Send the 16-bit sequence on SWDIOTMS = 0111100111100111 (0x79E7 MSB first)
  - Send more than 50 SWCLKTCK cycles with SWDIOTMS = 1
- Switch from SWD to JTAG. The sequence is:
  - Send more than 50 SWCLKTCK cycles with SWDIOTMS = 1
  - Send the 16-bit sequence on SWDIOTMS = 0011110011100111 (0x3CE7 MSB first)
  - Send more than 50 SWCLKTCK cycles with SWDIOTMS = 1

13.2.5.4. Flash Patch Breakpoint (FPB)

The Flash Patch Breakpoint (FPB):

- implements hardware breakpoints
- patches code and data from code space to system space

The FPB unit contains:

- two literal comparators for matching against literal loads from Code space, and remapping to a corresponding area in System space
- six instruction comparators for matching against instruction fetches from Code space and remapping to a corresponding area in System space

Alternatively, comparators can also be configured to generate a breakpoint instruction to the processor core on a match

13.2.5.5. Data Watchpoint and Trace (DWT)

The Data Watchpoint and Trace (DWT) contains four comparators which can be configured to generate the following:

- PC sampling packets at set intervals
- PC or Data watchpoint packets
- Watchpoint event to halt core

The DWT contains counters for the items that follow:

© 2026 Microchip Technology Inc. and its subsidiaries

- Clock cycle (CYCCNT)
- Folded instructions
- Load Store Unit (LSU) operations
- Sleep Cycles
- CPI (all instruction cycles except for the first cycle)
- Interrupt overhead

13.2.5.6. Instrumentation Trace Macrocell (ITM)

The Instrumentation Trace Macrocell (ITM) is an application-driven trace source that supports 'printf 'style debugging to trace operating system (OS) and application events, and emits diagnostic system information. The ITM emits trace information as packets which can be generated by three different sources with several priority levels:

- Software trace—Software can write directly to ITM stimulus registers. This can be done using the 'printf' function. For more information, see How to Configure the ITM.
- Hardware trace—The ITM emits packets generated by the DWT.
- Timestamping—Timestamps are emitted relative to packets. The ITM contains a 21-bit counter to generate the timestamp.

13.2.5.6.1. How to Configure the ITM

The following example describes how to output trace data in asynchronous trace mode.

1. Configure the TPIU for asynchronous trace mode (see How to Configure the TPIU)
2. Enable the write accesses into the ITM registers by writing 0xC5ACCE55 into the Lock Access Register (Address: 0xE0000FB0)
3. Write 0x00010015 into the Trace Control Register:
   - Enable ITM.
   - Enable Synchronization packets.
   - Enable SWO behavior.
   - Fix the ATB ID to 1.
4. Write 0x1 into the Trace Enable Register:
   - Enable the Stimulus port 0.
5. Write 0x1 into the Trace Privilege Register:
   - Stimulus port 0 only accessed in privileged mode (Clearing a bit in this register will result in the corresponding stimulus port being accessible in user mode.)
6. Write into the Stimulus port 0 register: TPIU (Trace Port Interface Unit). The TPIU acts as a bridge between the on-chip trace data and the ITM. It formats and transmits trace data off-chip at frequencies asynchronous to the core.

13.2.5.6.2. Asynchronous Mode

The TPIU is configured in asynchronous mode, trace data are output using the single TRACESWO pin. The TRACESWO signal is multiplexed with the TDO signal of the JTAG Debug Port. As a consequence, asynchronous trace mode is only available when the Serial Wire Debug mode is selected since TDO signal is used in JTAG debug mode.

Two encoding formats are available for the single pin output:

- Manchester encoded stream. This is the reset value.
- NRZ-based UART byte structure

13.2.5.6.3. How to Configure the TPIU

This example only concerns the asynchronous trace mode.

© 2026 Microchip Technology Inc. and its subsidiaries

1. Set the TRCENA bit to 1 into the Debug Exception and Monitor register (0xE000EDFC) to enable the use of trace and debug blocks.
2. Write 0x2 into the Selected Pin Protocol register:
   - Select the Serial Wire Output – NRZ
3. Write 0x100 into the Formatter and Flush Control register.
4. Set the suitable clock prescaler value into the Async Clock Prescaler register to scale the baud rate of the asynchronous output (this can be done automatically by the debugging tool).

13.2.5.7. IEEE 1149.1 JTAG Boundary Scan

IEEE 1149.1 JTAG Boundary Scan allows pin-level access independent of the device packaging technology.

IEEE 1149.1 JTAG Boundary Scan is enabled when TST is tied low, while JTAGSEL is high during the power-up, and must be kept in this state during the whole boundary scan operation. The SAMPLE, EXTEST and BYPASS functions are implemented. In SWD/JTAG Debug mode, the Arm processor responds with a non-JTAG chip ID that identifies the processor. This is not IEEE 1149.1 JTAG-compliant.

It is not possible to switch directly between JTAG Boundary Scan and SWJ Debug Port operations. A chip reset must be performed after JTAGSEL is changed.

A Boundary-scan Descriptor Language (BSDL) file is provided on www.microchip.com for test setup.

13.2.5.7.1. JTAG Boundary-Scan Register (BSR)

The Boundary-scan Register (BSR) contains a number of bits which correspond to active pins and associated control signals.

Each input/output pin corresponds to a 3-bit register in the BSR. The OUTPUT bit contains data that can be forced on the pad. The INPUT bit facilitates the observability of data applied to the pad. The CONTROL bit selects the direction of the pad.

For more information, refer to the BDSL files available for the device on www.microchip.com.

13.2.5.8. Reset Control

When in Debug mode, debug probes can reset the device (core and peripherals) via the NRST pin or the Reset Controller (RSTC) registers.

When in JTAG Boundary Scan mode, the device can be reset via the NRST pin or the IEEE 1149.1 reset method.

Important: In Debug mode, resetting the core and peripherals by setting the SYSRESETREQ bit in the AIRCR via the SW-DP/JTAG-DP interface is not possible.

© 2026 Microchip Technology Inc. and its subsidiaries

13.2.5.9. ID Code Register

Name: ID Code Register
Property: Read-only

JTAG ID Code value is 0x05B4503F.

Bit	31	30	29	28	27	26	25	24
	VERSION[3:0]				PART NUMBER[15:12]			
Access	R	R	R	R	R	R	R	R
Reset								

Bit	23	22	21	20	19	18	17	16
	PART NUMBER[11:4]							
Access	R	R	R	R	R	R	R	R
Reset								

Bit	15	14	13	12	11	10	9	8
	PART NUMBER[3:0]				MANUFACTURER IDENTITY[10:7]			
Access	R	R	R	R	R	R	R	R
Reset								

Bit	7	6	5	4	3	2	1	0
	MANUFACTURER IDENTITY[6:0]							1
Access	R	R	R	R	R	R	R	R
Reset								

Bits 31:28 – VERSION[3:0] Product Version Number
Set to 0x0.

Bits 27:12 – PART NUMBER[15:0] Product Part Number
Product part number is 0x05B45.

Bits 11:1 – MANUFACTURER IDENTITY[10:0]
Set to 0x01F.

Bit 0 – 1
Required by IEEE Std. 1149.1. Set to 1.

© 2026 Microchip Technology Inc. and its subsidiaries

14. ROM Code and Boot Strategies

14.1. Description

The system always boots from the ROM code, the first-stage bootloader stored in ROM. The main function of the ROM code is to execute the user application, stored in the internal Flash at address 0x01000000. Boot may be done in Standard mode, or in Secure mode if the user application requires authentication before execution.

SAM-BA Monitor modes are available to configure the system, to program the user application into regular pages of the internal Flash, or to save boot-related data such as customer secret keys into dedicated pages of the internal Flash.

The boot mode selection is done through GPNVM[8:5] of the GPNVM word. Other boot and security features, such as Flash memory plane exchange or JTAG access, are also managed by the GPNVM word.

14.1.1. Boot Modes and System Clock Tree

The ROM code configures the system clock tree based on the boot mode selected by GPNVM[8:5] and on the value of the GPBR[15] register.

14.1.1.1. Standard Boot Mode or FFPI Monitor

When Standard Boot mode is selected, the ROM code jumps directly into the internal Flash memory to execute the user application. When entering FFPI Monitor, the ROM code keeps the system clock configuration that was set at power-on or reset: no PLL is enabled, Core and MCK0 are sourced by the MAIN clock running from the 12 MHz internal RC oscillator. The value of the GPBR[15] register is ignored.

14.1.1.2. Secure Boot Modes or SAM-BA Monitors

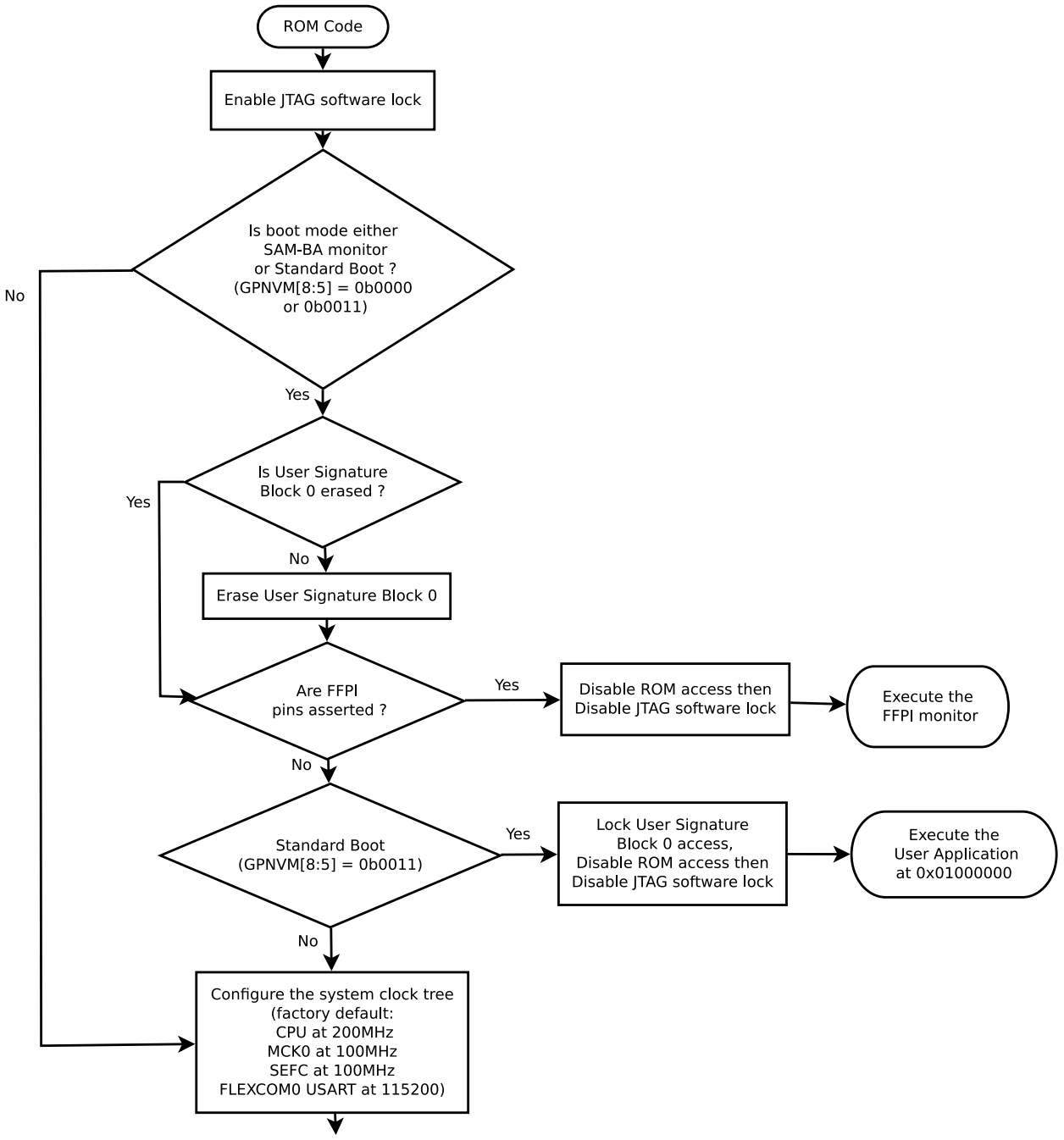
For all other boot modes other than Standard Boot and FFPI Monitor, the ROM code modifies the system clock tree that was configured at power-on or reset and sets a new configuration based on the value of the GPBR[15] register. For more details, see the section System Clock Tree.

14.1.2. Boot Sequence

The figure below illustrates the ROM code boot sequence depending on the Boot mode.

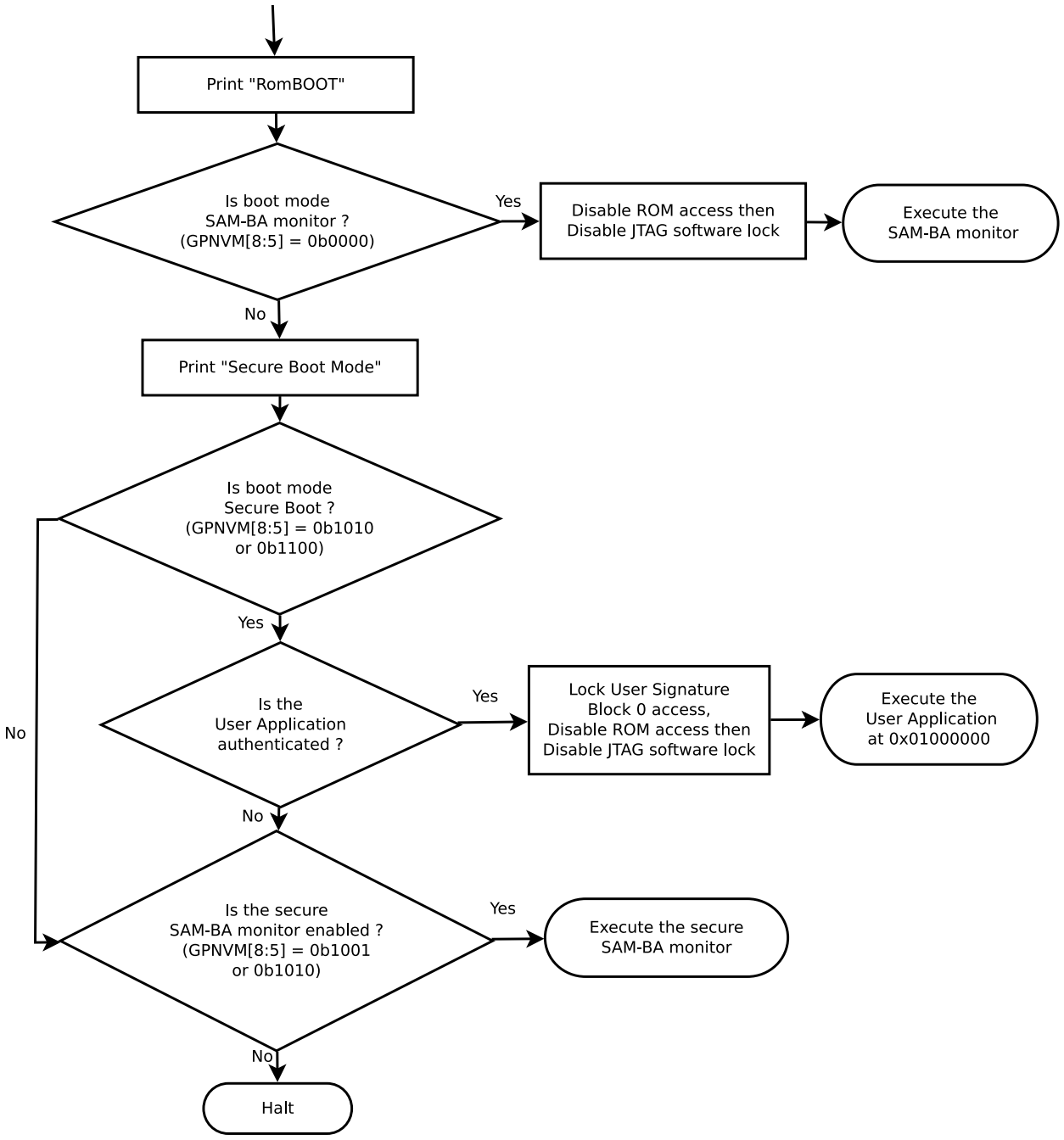
© 2026 Microchip Technology Inc. and its subsidiaries

Figure 14-1. ROM Code Flowchart (Part 1)



© 2026 Microchip Technology Inc. and its subsidiaries

Figure 14-2. ROM Code Flowchart (Part 2)



© 2026 Microchip Technology Inc. and its subsidiaries

Important:
Boot memory is always the ROM regardless of the value of GPNVM bits [8:5]. The ROM Code sets the Core clock to 200 MHz on PLLB with a 12 MHz RC oscillator as input clock, only if the ROM code, via the configuration of the GPNVM bits, is instructed to start the SAM-BA monitor, the secure SAM-BA monitor, Secure boot with or without fall back to secure SAM-BA monitor.

Because the PMC is set up by the ROM code, care must be taken when the main application running from Flash sets a new clock configuration scheme for PLLB. Any switch of the input clock of the PLLB must avoid leading to an output frequency higher than 200 MHz. To avoid any potential over-clocking and provide more flexibility, Microchip software examples make use of the down-clocking feature for the ROM code available when programming GPBR15. For more information, see System Clock Tree.

14.2. Functional Description

14.2.1. ROM Code Console

The console of the ROM code is a serial link to interact with the ROM code and its SAM-BA Monitors.

By default, the system communicates through FLEXCOM0 (FLEXCOM0_IO0 on PA4, and FLEXCOM0_IO1 on PA5) configured in USART mode with the following attributes:

- Baud rate: 115200 Baud
- Data bits: 8
- Parity: none
- Stop bit: 1

With the GPBR[15] word, the baud rate can be reduced to 57600 if required.

14.2.2. GPNVM Bits

The device features up to eight GPNVM bits and one security bit in the SEFC0/Plane 0. The GPNVM bits configure the software or physical connection to hardware in the device for Boot mode selection and other functions.

Note that all GPNVM bits are erased upon assertion of the ERASE signal.

14.2.2.1. GPNVM Bits Overview

For more information on using GPNVM bits, refer to GPNVM Bits.

GPNVM Bit Number	Value	Function
Security bit		
0	–	When set to 1, the security bit can only be erased by the ERASE signal
Memory Plane Selection (Plane 0 or Plane 1)		
1	0	The Flash addresses are the default one: order Plane 0, Plane 1
	1	The Flash addresses are exchanged: order Plane 1, Plane 0
Hardware Erase Function Lock (EFL)		
[4:2]	000	Not locked, default
	110	HW Erase function disabled, EFL bits are read-only (active only if security bit is set)
	Others	HW Erase function disabled, EFL bits can still be written by software (active only if security bit is set)
Boot Mode		

© 2026 Microchip Technology Inc. and its subsidiaries

GPNVM Bits Overview (continued)		
GPNVM Bit Number	**Value**	**Function**
[8:5]	0000	Standard SAM-BA Monitor (default). Not Secured.
	0011	Standard Boot: Run the user application from the internal Flash at offset 0. Not Secured.
	1001	Secure SAM-BA Monitor. Secured.
	1010	Secure Boot (with fallback to Secure SAM-BA Monitor). Secured.
	1100	Secure Boot (Secure SAM-BA Monitor disabled). Secured.
	Others	Halt (the CPU runs an endless loop)

14.2.2.2. GPNVM[0]: Security Bit

The security bit is based on a specific general-purpose NVM bit (GPNVM bit 0).

When security is enabled, any access to the Flash, SRAM, core registers and internal peripherals, either through the SW-DP/JTAG-DP interface or through the FFPI, is forbidden. This ensures the confidentiality of the code programmed in the Flash.

This bit is enabled through the command “Set General Purpose NVM Bit” in EEFC_FCR.FCMD. Disabling the security bit can only be achieved by asserting the ERASE signal on pin PB2 at 1, and after a full Flash erase is performed.

14.2.2.3. GPNVM[1]: Memory Plane Addresses Exchange (MPAE) (Plane 0 or Plane 1)

When the device contains two Flash controllers, SEFC0 and SEFC1, the Flash memory is constituted of two planes, Plane 0 and Plane 1.

The Flash controller register interfaces have fixed addresses in the system memory map. However, their associated memory planes are contiguous in the system memory map but can be dynamically swapped by changing the value of the bit MPAE.

When planes are swapped for dual-boot purposes, the SEFC follows the Flash plane it is connected to. MPAE has no effect on a device with a single Flash controller ((SEFC0)/Plane 0).

14.2.2.4. GPNVM[4:2]: Erase Function Lock (EFL)

To avoid erase of the device, the ERASE signal can be locked by programming the GPNVM[4:2].

The EFL function is managed at application level. By default, the ERASE signal is active. The software can lock the ERASE signal by setting the Erase Function Lock (EFL). These bits can be set to 0 or 1 by software and they apply only to the ERASE signal. These bits do not prevent erase commands requested by the software.

When the EFL locks the ERASE signal and the security bit (GPNVM[0]) is enabled, the device cannot be recovered. This function should be used only at the end of the production process. EFL has three bits to allow the majority filter mechanism.

14.2.2.5. GPNVM[8:5]: Boot Mode

GPNVM[8:5] are used to select the boot mode. If the Flash is erased via the ERASE signal, the GPNVM are erased and the next boot mode is the Standard SAM-BA Monitor. To allow at least two bits of difference between two values, four GPNVM bits are implemented.

14.2.3. SW-DP/JTAG-DP Access

SW-DP/JTAG-DP access is during execution of the ROM code. It is re-enabled when:

- jumping into the user application in Flash after the execution of the Standard boot or Secure boot mode
- executing the FFPI Monitor
- executing the SAM-BA Monitor

Note that the JTAG access remains locked during the execution of the Secure SAM-BA Monitor and its secured applets.

© 2026 Microchip Technology Inc. and its subsidiaries

14.2.4. User Signature Block 0 of SEFC0

The User Signature Block 0 of SEFC0 is reserved for ROM code usage. This is a 4096-byte special area, split into eight 512-byte pages.

In “non-secure” boot modes (SAM-BA Monitor, Standard boot or FFPI Monitor), the ROM code scans the User Signature Block 0, erases it if not already erased, and keeps the block unlocked to allow read, write and erase permissions from the EEFC_USR register of SEFC0. This is a safety mechanism, erasing the customer keys when the selected boot mode in GPNVM[8:5] is altered from a “secure” boot mode to a “non-secure” boot mode.

If the selected boot mode in GPNVM[8:5] is SAM-BA Monitor, the user is allowed to program the secure boot data expected in Secure boot mode, without executing the Secure SAM-BA Monitor.

In “secure” boot modes (Secure boot, Secure SAM-BA Monitor), the ROM code stores its secure boot configuration (authentication algorithm and customer keys) in the User Signature Block 0. Before executing an authenticated user application, the ROM code locks the access to the User Signature Block 0 and removes read, write and erase permissions through the EEFC_USR register of SEFC0. This way the customer keys are protected against any further access from the software.

Finally, the secret customer key, used during AES-256-CMAC computation for user application authentication, is sent from the Flash controller (SEFC0) to the AES hardware through the key bus. This key never goes through the system bus, the CPU or the internal SRAM.

© 2026 Microchip Technology Inc. and its subsidiaries

14.2.5. System Clock Tree

Name: System Clock Tree
Reset: –

The ROM code configures the system clock tree according to the value of the GPBR[15] word:

Bit	31	30	29	28	27	26	25	24
	KEY[23:16]							
Access								
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	KEY[15:8]							
Access								
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	KEY[7:0]							
Access								
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	CPU_CK[3:0]				PATCH_BYPASS	BR_57600	MCK0DIV2	MCK0DIV
Access								
Reset	–	–	–	–	–	–	–	–

Bits 31:8 – KEY[23:0]
The ROM code ignores the value of GPBR[15] word unless the value of KEY is 0x524F4D. For any other value, the ROM code configures the system clock tree from its default settings:

- CPU_CK: 0
- MCK0DIV: 1
- MCK0DIV2: 1
- BR_57600: 0
- PATCH_BYPASS: 1

Bits 7:4 – CPU_CK[3:0] CPU Clock Frequency

Value	Description
0	CPU clock frequency is 200 MHz
1	CPU clock frequency is 100 MHz
2	CPU clock frequency is 50 MHz
Others	CPU clock frequency is 200 MHz

Bit 3 – PATCH_BYPASS Flash Patch Bypass
When the ROM code configures the system clock tree, it sets the value of the PATCH_BYPASS bit from the GPBR[15] word into the SFR_FLASH register.

Bit 2 – BR_57600 ROM Code Console Baud Rate is 57600
Depending on the system clock tree, the FLEXCOM0 USART baud rate cannot reach the targeted 115200 baud rate. In such cases, the baud rate should be reduced to 57600.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	ROM code console baud rate is 115200
1	ROM code console baud rate is 57600

Bit 1 – MCK0DIV2

When the ROM code configures the system clock tree, it sets the value of the MCK0DIV2 bit from the GPBR[15] word into the PMC_CPU_CKR register.

Bit 0 – MCK0DIV

When the ROM code configures the system clock tree, it sets the value of the MCK0DIV bit from the GPBR[15] word into the PMC_CPU_CKR register.

14.3. Secure Boot Mode

In Secure boot mode, the ROM code authenticates the user application before execution. If the ROM code fails to authenticate the user application , then this user application is not executed and either:

- the Secure SAM-BA Monitor is run (boot mode is 0b1010)
- the ROM code halts, running an endless loop (boot mode is 0b1100)

A cryptographic signature is appended to the user application in the internal Flash. The ROM code verifies this signature based on cryptographic data and algorithm choice stored in the first two 512-byte pages of the User Signature Block 0 (USB0) in SEFC0.

Table 14-1. User Signature Block 0 Page 0

Byte Offset	Bits[127:96]	Bits[95:64]	Bits[63:32]	Bits[31:0]
0x000	Reserved			
0x010				
0x020	AES_256_CMAC_KEY			
0x030				
0x040	Reserved			
...				
0x1F0				

AES_256_CMAC_KEY

256-bit secret key that the ROM code uses to compute the AES-CMAC signature of the user application before comparing it to the signature appended to the user application in the internal Flash. This private key is only used if AUTH_TYPE is 0 in USB0 page1.

Table 14-2. User Signature Block 0 Page 1

Byte Offset	Bits[127:96]	Bits[95:64]	Bits[63:32]	Bits[31:0]
0x000	PUBLIC_KEY_DIGEST			
0x010				
0x020				
0x030				
0x040	Reserved			
0x050				
0x060	Reserved			AUTH_TYPE
...				
0x1F0	Reserved			

PUBLIC_KEY_DIGEST:

© 2026 Microchip Technology Inc. and its subsidiaries

SHA-512 digest of the X.509 root certificate in DER format. This public key digest is used by the ROM code only when AUTH_TYPE is 1. A chain of X.509 certificates in DER format is appended to the user application signature. The root certificate is the first certificate in this chain.

AUTH_TYPE: Authentication Type

Selects the cryptographic algorithm to be used for the user application authentication.

Value	Description
0	AES-256-CMAC algorithm
1	Public Key algorithm (RSA or ECDSA)
Others	The ROM code fails to authenticate the user application

14.3.1. User Application Padding and Signature Alignment

When the Secure boot mode is enabled, padding is always inserted between the user application and its signature so the offset of this signature in internal Flash is always aligned to 128 bits. The padding is included in the computation of the user application signature.

In the sections that follow, the 128-bit aligned user signature refers to the user signature plus its padding.

14.3.2. AES-256-CMAC Authentication (AUTH_TYPE = 0)

In AES-256-CMAC Authentication mode, the 128-bit signature of the user application is computed using the AES-256-CMAC algorithm, hence using the 256-bit secret key, AES_256_CMAC_KEY, stored in USB0 page 0.

The computed signature is compared to the signature appended to the 128-bit aligned user application in Flash. The ROM code locates the position of the signature in Flash reading the 8th vector in the ARM Cortex-M4 exception table read from 0x01000000 address. Indeed, this 32-bit word, in little endian format, is written with the size in bytes of the 128-bit aligned user application size plus the 16 bytes of the 128-bit AES-CMAC signature.

Table 14-3. User Application in Internal Flash at 0x01000000

Byte Offset	Bits[127:96]	Bits[95:64]	Bits[63:32]	Bits[31:0]
0x000	Hard Fault	NMI	**Reset**	**Stack Pointer**
0x010	**(User App + Signature) Size**	Usage Fault	Bus Fault	Mem Manage
0x020				
0x040				
...				
User App Size	128-bit AES-256-CMAC signature			
(User App + Signature) Size				

As an example, for a 1000-byte unsigned user application, the size of the 128-bit aligned user application is ((1000 + 16 -1) / 16) * 16 = 1008. Then the User App Size written into the 8th vector is 1008 + 16 = 1024.

14.3.3. Public Key Signature for Authentication (AUTH_TYPE = 1)

The signature of 128-bit aligned user application is appended to the 128-bit aligned user application itself and stored in the internal Flash. The size of this signature depends on the signature algorithm (RSA or ECDSA) and its parameters.

A chain of X.509 version 3 certificates in DER format is stored in the internal Flash, immediately after the user application signature. The 1st certificate of this chain (lowest address in internal Flash) is called the root certificate. The root certificate is self-signed. A SHA-512 digest of the whole root certificate is stored in PUBLIC_KEY_DIGEST inside page 1 of the User Block Signature 0 in SEFC0.

© 2026 Microchip Technology Inc. and its subsidiaries

The ROM code computes the SHA-512 digest of the root-certificate stored in internal Flash then compares the computed digest with the digest stored in PUBLIC_KEY_DIGEST in page 1 of USB0.

If the two digests do not match, then the ROM code rejects the root certificate and fails to authenticate the user application. If the two digests match, then the ROM code continues it authentication process.

Once the root certificate passes the SHA-512 digest test, the ROM code iterates on each X.509 certificate next in the chain and verifies its signature with the public key of the previous X.509 certificate in the chain. If the ROM code fails to verify the signature of any certificate in the chain, it fails to authenticate the certificate chain, therefore fails to authenticate the user application.

When the whole certificate chain is authenticated, the public key of the last X.509 certificate in the chain (highest address in internal Flash) is used to verify the signature of the user application. If this final signature verification is passed, then the user application is authenticated and is executed. Otherwise, the ROM code fails to authenticate the user application.

In this security model, the root of trust is established by the root certificate and its SHA-512 digest stored in page 1 of USB0.

The 8th and 9th vectors in the exception table of ARM Cortex-M4 are used to store

1. the 32-bit size in bytes of the 128-bit aligned user application plus the size of its signature
2. the 31-bit size in bytes of the chain of X.509 certificates

X.509 certificates are appended in the chain without padding or any alignment consideration.

Bit 31 in the 9th vector tells whether the ROM code can skip the signature verification of the self-signed root certificate:

0: The signature of the self-signed root certificate if verified before the SHA-512 digest verification

1: The ROM code skips the verification of signature of the self-signed root certificate; only the SHA-512 digest verification is done to authenticate the root certificate.

Restriction: The ROM code has a known limitation when parsing X.509 certificates. Therefore, it is not fully compliant with https://datatracker.ietf.org/doc/html/rfc5280#section-4.1.2.2, as serial numbers, which MUST be positive numbers, are limited to 18 bytes instead of 20 bytes.

14.3.3.1. ECDSA Signature

For ECDSA algorithm only, the user signature is organized based on the following layout:

ECDSA user application signature:

OpenSSL ECDSA signature	Padding for 32-bit alignment	User App Size (32-bit word)

The OpenSSL ECDSA signature is located at the beginning (lowest address in internal Flash) of the ECDSA user application signature. The 32-bit trailing word value is the size in bytes of the 128-bit aligned user application in little-endian format.

As an example, for a 1000-byte user application, the size of the 128-bit aligned user application is ((1000 + 16 - 1) / 16) * 16 = 1008 bytes. Hence the 32-bit word is 0x000003F0 and its byte sequence from lowest addresses to highest addresses is 0xF0, 0x03, 0x00, 0x00.

Thus the total size of the ECDSA user application signature, as verified by the ROM code, is always aligned to 32 bits. Therefore, the offset of the X.509 certificate chain in the internal Flash, given by the value of the 8th vector, is also always aligned to 32 bits. However, the size of this X.509 certificate chain, given by the value of the 9th vector, has no alignment constraint.

© 2026 Microchip Technology Inc. and its subsidiaries

The layout of the user signature, its signature and the X.509 certificate chain is given in the table below.

Byte Offset	Data
0x000	**Stack Pointer**
0x004	**Reset**
0x008	NMI
0x00C	Hard Fault
0x010	Mem Manage
0x014	Bus Fault
0x018	Usage Fault
0x01C	**(User App + Signature) Size**
0x020	**X.509 Certificate Chain Size**
...	
User App Size	OpenSSL ECDSA Signature + Padding for 32-bit alignment
...	
(User App + Signature) Size - 0x008	
(User App + Signature) Size - 0x004	**User App Size**
(User App + Signature) Size	X.509 Certificate Chain
...	
(User App + Signature) Size + X.509 Certificate Chain Size - 0x001	
(User App + Signature) Size + X.509 Certificate Chain Size	–

14.3.3.2. RSA Signature

For the RSA algorithm only, the user signature is directly the RSA signature alone. Unlike the ECDSA case, the size of the 128-bit aligned user application (without its signature) is not stored because the size of the RSA signature is always equal to the size of the modulo of the public key from the last X.509 certificate in the chain.

This modulo/signature size is extracted during the parsing and the authentication of the X.509 certificate chain. This signature size is subtracted to 32-bit size stored in the 8th vector to compute the offset of the RSA signature. The ROM code supports 2048 and 4096-bit RSA signature size.

The layout of the user signature, its signature and the X.509 certificate chain is given in the table below.

Byte Offset	Data
0x000	**Stack Pointer**
0x004	**Reset**
0x008	NMI
0x00C	Hard Fault
0x010	Mem Manage
0x014	Bus Fault
0x018	Usage Fault
0x01C	**(User App + Signature) Size**
0x020	**X.509 Certificate Chain Size**
...	

© 2026 Microchip Technology Inc. and its subsidiaries

RSA Signature (continued)	
Byte Offset	**Data**
User App Size	2048 or 4096-bit RSA signature
...	
(User App + Signature) Size - 0x001	
(User App + Signature) Size	X.509 Certificate Chain
...	
(User App + Signature) Size + X.509 Certificate Chain Size - 0x001	
(User App + Signature) Size + X.509 Certificate Chain Size	

14.4. SAM-BA Monitors

The ROM code embeds two distinct SAM-BA Monitors. Both transfer data through FLEXCOM0 USART. Most transfers, including all commands, exchange raw text, or small sequences of ASCII characters. However, bigger data transfers use the Xmodem protocol to improve the data integrity.

The communication is always initiated by the monitor client, likely the SAM-BA tool, which acts as the Host in the communication protocol between the target (PIC32CXMTx) and the Host computed.

14.4.1. Standard SAM-BA Monitor

Standard SAM-BA Monitor is the factory default for the boot mode. It supports a small set of basic commands. Most of these commands allow the user to perform 8-, 16- and 32-bit read or write access to the entire system memory map, including peripheral registers. Two additional commands download or upload files into the system memory so that a third command may be run to execute any previously loaded code. Such small binary programs are called applets. They help to extend the features of the SAM-BA Monitor. For example, one applet could be dedicated to the internal Flash management so that the user application may be programmed into the internal Flash prior to switching into the Standard boot mode.

The general syntax of any non-secure SAM-BA Monitor command is:

command ::=[*op_code* [“,” *address* [“,” *value*]]] “#”

op_code ::= “S” | “R” | “O” | “H” | “W” | “o” | “h” | “w” | “G” | “T” | “N” | “K” | “V”

address ::= *hex_value*

value ::= *hex_value*

hex_value ::= [*hex_digit*] | *hex_digit hex_value*

hex_digit ::= “0” | “1” | “2” | “3” | “4” | “5” | “6” | “7” | “8” | “9” | “A” | “B” | “C” | “D” | “E” | “F” | “a” | “b” | “c” | “d” | “e” | “f”

The following array lists all supported commands and their expected parameters:

op_code	*address*	*value*	Description	Examples
S	X	X	Send a ***value***-byte file at ***address*** into the system memory	S,300000,1000#
R	X	X	Receive a ***value***-byte file from ***address*** in the system memory	R,20000000,40000#
O	X	X	Write the 8-bit ***value*** at ***address*** into the system memory	O,F8034060,7F#
H	X	X	Write the 16-bit ***value*** at ***address*** into the system memory	H,00300000,1337#
W	X	X	Write the 32-bit ***value*** at ***address*** into the system memory	W,300010,deadbeef#
o	X	–	Read the 8-bit data from ***address*** in the system memory	o,1234,#
h	X	–	Read the 16-bit data from ***address*** in the system memory	h,00001234,#

© 2026 Microchip Technology Inc. and its subsidiaries

Standard SAM-BA Monitor (continued)				
op_code	***address***	***value***	**Description**	**Examples**
w	X	–	Read the 32-bit data from ***address*** in the system memory	w,f802c000,#
G	X	–	Execute code at ***address*** in the system memory	G,300000#
T	–	–	Enter terminal mode	T#
N	–	–	Exit terminal mode	N#
K	X	X	Sets the boot mode to Secure SAM-BA Monitor for further power-on/resets (only if ***address*** is **0xcafe4fab** and ***value*** is **0xcafedeca**)	K,cafe4fab,cafedeca#
V	–	–	Read the ROM code version string	V#

When the ROM code starts executing its SAM-BA Monitor, Terminal mode is enabled. If Terminal mode is enabled, each command output is enclosed by an opening “\n\r” sequence and a closing “>” prompt. Neither the “\n\r” sequence nor the “>” prompt is sent if Terminal mode is disabled. However, a “\n\r” sequence alone is sent in reply to any “N#” received command even if Terminal mode is disabled.

In order to help SAM-BA clients to parse and process data, it is recommended to exit Terminal mode before sending or receiving binary files through the SAM-BA Monitor.

The *address* and *value* parameters are case-insensitive and are limited to the last 8 *hex_digits* received. All previously received hexadecimal digits are dropped, limiting both *address* and *value* to 32-bit words.

Data transfers following “R” and “S” commands use the Xmodem protocol. All other transfers are raw text only.

14.4.2. Secure SAM-BA Monitor

Secure SAM-BA Monitor restricts the number of supported commands compared to the non-secure SAM-BA Monitor. Direct read and write access to the system memory is now discarded and new commands are introduced to help the user configure the Secure boot mode.

Note that the syntax of commands differs from that of the non-secure SAM-BA Monitor.

Commands are the only transfers using the raw text format. All other transfers, including replies and payloads, use the Xmodem protocol.

The syntax for Secure SAM-BA Monitor commands is:

command ::= *op_code* “,” *address* “,” *length* “,” *id* “,” *rw* “#”

op_code ::= “RVER” | “WCKY” | “SAPT” | “SMBX” | “RMBX” | “EAPP” | “SFIL” | “RFIL” | “SJTD” | “CRST” | “SSEC” | “SSNM”

address ::= *hex_value*

length ::= *hex_value*

id ::= *hex_value*

rw ::= *hex_value*

hex_value ::= [*hex_digit*] | *hex_digit hex_value*

hex_digit ::= “0” | “1” | “2” | “3” | “4” | “5” | “6” | “7” | “8” | “9” | “A” | “B” | “C” | “D” | “E” | “F” | “a” | “b” | “c” | “d” | “e” | “f”

The *op_code* parameter is always a string of four upper-case letters. The *address* and *length* parameters must not exceed eight hexadecimal digits.

The *id* and *rw* parameters are not used but kept for backward compatibility purpose. They should be left empty.

© 2026 Microchip Technology Inc. and its subsidiaries

Depending on the command, a data payload can be added after the command. In this case, the payload must be sent after the Secure SAM-BA Monitor has acknowledged the command and communicated how the data payload must be split (size of the payload).

The following array lists all supported commands and their expected parameters:

op_code	*address*	*length*	Description	Examples
RVER	_	_	Read the ROM code version string	RVER,,,,#
WCKY	_	X	Write the payload of the ***length***-byte Customer Key message into page 0 and page 1 of the User Signature Block 0.	WCKY,,C0,,#
SAPT	_	X	Send a ***length***-byte Secure SAM-BA applet into the internal SRAM0.	SAPT,,13D0,,#
SMBX	_	X	Send the 128-byte applet mailbox into the internal SRAM0.	SMBX,,80,,#
RMBX	_	_	Received the 128-byte applet mailbox from internal SRAM0	RMBX,,,,#
EAPP	_	_	Execute the Secure SAM-BA applet previously loaded with the SAPT command	EAPP,,,,#
SFIL	X	X	Send a ***length***-byte file to later program it at the ***address*** offset in some applet-dependent memory	SFIL,2000,400,,#
RFIL	X	X	Receive a ***length***-byte file from the ***address*** offset in some applet-dependent memory	RFIL,2000,400,,#
SJTD	_	_	Set the 'SECURITY' bit in GPNVM to persistently disable JTAG and Debug ports	SJTD,,,,#
CRST	_	_	Chip Reset	CRST,,,,#
SSEC	_	_	Set the boot mode to Secure Boot (with fallback to Secure SAM-BA Monitor) for further power-on/resets	SSEC,,,,#
SSNM	_	_	Set the boot mode to Secure Boot (Secure SAM-BA Monitor disabled) for further power-on/resets	SSNM,,,,#

There are three types of Secure SAM-BA command replies. The first type is used as a Command ACKnowledge for all but EAPP and SVER commands. In this case, the syntax of the reply is:

reply ::= "CACK," *errcode* "," *length* "#" [*payload*]

errcode ::= *hex_value*

length ::= *hex_value*

hex_value ::= [*hex_digit*] | *hex_digit hex_value*

hex_digit ::= "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9" | "A" | "B" | "C" | "D" | "E" | "F" | "a" | "b" | "c" | "d" | "e" | "f"

payload ::= [BYTE]| BYTE *payload*

The second type replies to EAPP (Execute APPlet) commands. The syntax of this reply is the same as the first type except that "CACK" is replaced by "ASTA", which stands for Applet STAtus.

Finally, the third type replies to RVER (Read VERsion) commands. . The syntax of this reply is the same as the first type except that "CACK" is replaced by "SVER", which stands for Send VERsion.

Table 14-4. List of *errcode*

Hexadecimal Value	Description
00000000	No error
FFFFFFFD	Bad value for *address* argument
FFFFFFFC	Bad value for *length* argument
FFFFFFF9	Bad *op_code*
FFFFFFF8	Bad customer key length
FFFFFFF6	The customer key has already been written
FFFFFFF5	AES-256-CMAC error
FFFFFFF4	AES-256-CBC error

© 2026 Microchip Technology Inc. and its subsidiaries

Table 14-4. List of *errcode* (continued)

Hexadecimal Value	Description
FFFFFFF3	Key expansion error
FFFFFFF0	SEFC UID read error
FFFFFFED	SEFC write error
FFFFFFE7	Data transfer error
FFFFFFE6	Invalid argument (*payload*)

14.4.2.1. Secure SAM-BA Monitor Protocol

14.4.2.1.1. Read ROM Code Version (RVER)

For this specific command, the Secure SAM-BA Monitor replies with the SVER opcode and indicates that the ROM Code version string length is 48 (0x30) bytes, so the external tool knows how many characters it has to receive.

(PC to Device) >> RVER,,,,#

(Device to PC) << SVER,00000000,00000030#v1.0 Jul 4 2019 11:58:23

14.4.2.1.2. Write Customer Key (WCKY)

Sending the customer key to the chip involves sending a binary payload (the ciphered and signed message containing the customer key and all secure boot settings). This payload is to be sent by the tool after receiving the acknowledge from the Secure SAM-BA Monitor, telling how many bytes it expects.

(PC to Device) >> WCKY,,c0,,#

(Device to PC) << CACK,00000000,000000C0#

(PC to Device) >> <customer_key_file.cip>

(Device to PC) << CACK,00000000,00000000#

14.4.2.1.3. Executing Secured Applets

Secured applets are small programs running in the target internal SRAM and extending the Secure SAM-BA Monitor features. Secured applet binaries are ciphered, signed and bundled with the SAM-BA tool. They cannot be modified by the user and only Microchip can provide them.

First, send the ciphered and signed applet to the target. This is done by the Send Applet command:

(PC to Device) >> SAPT,,9870,,#

(Device to PC) << CACK,00000000,00009870#

(PC to Device) >> <applet_binary.cip>

(Device to PC) << CACK,00000000,00000000#

In the example above, the SAM-BA tool requests sending an applet of size 0x9870 and the Secure SAM-BA Monitor acknowledges this. The SAM-BA tool sends the ciphered and signed applet binary (the file applet_binary.cip), and after checking the signature and deciphering the applet in SRAM, the Secure SAM-BA Monitor sends the status (0x0: successful).

Once the applet has been loaded into the internal SRAM, there is no need to load it again between applet command executions.

Once the applet is in SRAM, before executing its code, its mailbox must be filled. The mailbox is the 32-word buffer at the beginning of the applet area, which allows input parameters and output results to be exchanged with the applet. To do so, the Send Mailbox command must be issued.

The mailbox is not ciphered, and is automatically written at the correct address in SRAM0 by the Secure SAM-BA Monitor.

(PC to Device) >> SMBX,,80,,#

© 2026 Microchip Technology Inc. and its subsidiaries

(Device to PC) << CACK,00000000,00000080#

(PC to Device) >> <applet_init_mailbox.bin>

(Device to PC) << CACK,00000000,00000000#

Now we can run the applet program with Execute Applet command.

(PC to Device) >> EAPP,,,,#

(Device to PC) << 0x06

(Device to PC) << ASTA,00000000,00000000#

First, the applet sends a 0x06 byte to notify completion, then the Secure SAM-BA Monitor replies with the status of the applet execution (0x0: successful). The status of applets are specific to each applet and are not related to the Secure SAM-BA Monitor error codes.

Also, depending on the applet and the executed command, the mailbox may have been updated with output results. Then the mailbox should be read back to get those outputs running the Read Mailbox command.

(PC to Device) >> RMBX,,,,#

(Device to PC) << CACK,00000000,00000080#<applet_output_mailbox.bin>

Finally, some applet commands need to exchange a large amount of data that does not fit into the 128-byte mailbox. In this case, applet commands can expect to exchange data through the applet buffer. The Send File (SFIL) and Receive File (RFIL) Secure SAM-BA Monitor commands are responsible for large data exchange through the applet buffer.

The Send File (SFIL) command should be executed before the Execute Applet (EAPP), whereas the Receive File (RFIL) should be executed after it.

(PC to Device) >> SFIL,,10000,,#

(Device to PC) << CACK,00000000,00010000#

(PC to Device) >> <data.bin>

(Device to PC) << CACK,00000000,00000000#

Or

(PC to Device) >> RFIL,00000000,00000400#

(Device to PC) << CACK,00000000,00000400#<data.bin>

The “initialize” command should be the first applet command to be executed for any applet. Then, the size of the applet buffer can be retrieved by reading the output mailbox.

14.5. FFPI Monitor

14.5.1. Description

FFPI Monitor provides parallel high-volume programming using a standard gang programmer. The parallel interface is fully handshaked and offers an optimized access to all the embedded Flash functionalities

Although the Fast Flash Programming mode is a dedicated mode for high-volume programming, this mode is not designed for in-situ programming.

14.5.2. Embedded Characteristics

- Programming Mode for High-volume Flash Programming Using Gang Programmer
  - Offers read and write access to the Flash memory plane

© 2026 Microchip Technology Inc. and its subsidiaries

  - Enables control of lock bits and general-purpose NVM bits
  - Enables security bit activation
  - Disabled once security bit is set
- Parallel Fast Flash Programming Interface
  - Provides an 16-bit parallel interface to program the embedded Flash
  - Full handshake protocol

14.5.3. Parallel Fast Flash Programming

14.5.3.1. Device Configuration

In Fast Flash Programming mode, the device is in a specific test mode. Only a certain set of pins is significant. The rest of the PIOs are used as inputs with a pull-up. The crystal oscillator is in bypass mode. Other pins must be left unconnected.

The figure and table below provide details on the signals and power supplies used when FFPI is implemented.

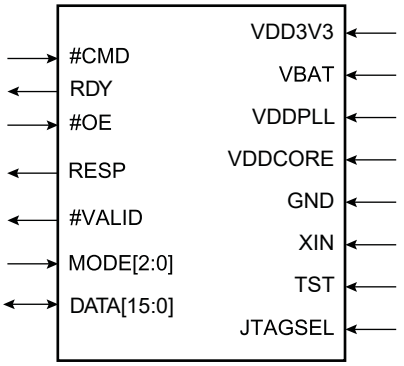


Table 14-5. Signal Description List

Signal Name	PIO Lines	Function	Type	Active Level	Comments
Power					
VDD3V3	–	I/O Lines and Main Power Supplies Input	Power	–	Connect to 3.3V
VBAT	–	Backup Area and Backup I/O Lines Power Supply Input	Power	–	Connect to 3.3V
VDDCORE	–	Core Power Supply Input	Power	–	Connect to 1.1V
VDDPLL	–	PLL Power Supply Input	Power	–	Connect to 1.1V
GND	–	Ground	Ground	–	Connect to GND
Clocks					
XIN	PA29	Main Clock Input	Input	–	3.3V Square wave signal from 10 to 50 MHz
Test					
TST	–	Test Mode Select	Input	High	Must be at high (3.3V) during power-up, then low (0V) after reset period.
JTAGSEL	–	JTAG Test Mode Select	Input	High	Must be at high (3.3V) during power-up, then low (0V) after reset period.
PIO					
#CMD	PA8	Valid command available	Input	Low	Pulled-up input at reset
RESP	PB26	Error Response	Output	–	Pulled-up input at reset

© 2026 Microchip Technology Inc. and its subsidiaries

Table 14-5. Signal Description List (continued)

Signal Name	PIO Lines	Function	Type	Active Level	Comments
RDY	PB24	0: Device is busy 1: Device is ready for a new command	Output	High	Pulled-up input at reset
#OE	PA9	Output Enable (active low) 0: the host releases the DATA[15:0] lines and asks the target device to drive them 1: the host drives the DATA[15:0] lines	Input	Low	Pulled-up input at reset
#VALID	PB25	When #OE is 0, #VALID is 0 if a valid data can by read by the host from DATA[15:0] and #VALID is 1 if the data on DATA[15:0] is not valid yet. When #OE is 1, the target device rises #VALID to 1	Output	Low	Pulled-up input at reset
MODE0	PA23	Specifies DATA type	Input	–	Pulled-up input at reset
MODE1	PA24				
MODE2	PA25				
DATA0	PA4	Bi-directional data bus	Input/Output	–	Pulled-up input at reset
DATA1	PA5				
DATA2	PA6				
DATA3	PA7				
DATA4	PB3				
DATA5	PB4				
DATA6	PB9				
DATA7	PB10				
DATA8	PB11				
DATA9	PB12				
DATA10	PC10				
DATA11	PC11				
DATA12	PC12				
DATA13	PC13				
DATA14	PC14				
DATA15	PC15				

14.5.3.2. Entering FFPI Mode

To enter FFPI mode, follow the steps below:

1. Apply the supplies as described in figure and table above.
2. Apply XIN within the VDDCORE POR reset time-out period, as defined in the section “Electrical Characteristics”.
3. Wait for the end of this reset period. Set JTAGSEL and TST to 0V during the entire FFPI operation.

The FFPI Monitor also requires that:

- Boot mode (BM) in GPNVM is either 0b0000 (SAM-BA Monitor) or 0b0011 (Standard boot)
- SECURITY bit is 0 in GPNVM

© 2026 Microchip Technology Inc. and its subsidiaries

14.5.4. FFPI Protocol

Despite the XIN clock, the FFPI protocol is asynchronous, as there is no clock signal to synchronize other FFPI signals. Instead, the target device and the host are kept synchronized by the RDY and CMD signals.

14.5.4.1. RDY , #CMD and MODE[2:0] Lines

The RDY signal is driven by the target device so its ready/busy state can be monitored by the host. Hence, the target device raises the RDY line to level 1 (ready) when it is ready to process a new command and it keeps the line to level 0 (busy) while processing a command. The host must wait for the RDY line level to be 1 before sending any command.

Once the command processing is finished for the target device, it must wait for the host to release the #CMD line before raising the RDY line to level 1 again.

When the target device is ready to process a command, the host first drives the MODE[2:0] lines to encode the type of the next command to be processed by the target device. Then it asserts the #CMD to query the target to process the requested command. The table below lists four different types of commands:

MODE[2:0]	Command	Command Description
0	CMDE	Read the value of DATA[7:0] lines and set it as the new value of the OP[7:0] register
1	ADDR0	Read the value of DATA[15:0] lines and set it as the new value of ADDR[15:0] register
2	ADDR1	Read the value of DATA[15:0] lines and set it as the new value of ADDR[31:16] register
5	DATA	Exchange data and/or start/resume the selected operation

For details on OP[7:0] and ADDR[31:0], see the section Internal Registers.

The host must keep the levels of the MODE[2:0] and #CMD lines until a falling edge is detected on the RDY line. This falling edge of the RDY line indicates that the target device has acknowledged the new command and has started to process it. Now, the host can release the MODE[2:0] and #CMD lines.

Next, the host must release the #CMD line because the target device waits for the #CMD line to be released before raising the RDY line again.

14.5.4.2. DATA[15:0], #OE and VALID Lines

16-bit data are exchanged between the target device and the host through the DATA[15:0] lines. This data transfer can be bidirectional. The level of the #OE line, driven by the host, indicates the direction of the transfer:

- 0: target device to host (output data)
- 1: host to target device (input data)

Input data are always sampled by the target device at the beginning of any command but output data can only be sent by the target device then sampled by the host at the end of DATA commands.

14.5.4.2.1. Input Data Transfer (all commands)

At the beginning of any command, the host should drive the DATA[15:0] lines and release the #OE line.

If the host has to send an input data, then it must set the proper value on the DATA[15:0] lines (and on the MODE[2:0] lines) before it starts a new command by lowering the #CMD line. Also, similar to the MODE[2:0] lines, the host must maintain the value on the DATA[15:0] lines until it detects a falling edge on the RDY line. The target device always samples the DATA[0:15] lines to read the input data at the same time as it samples the #CMD and #MODE[2:0] lines, just before lowering the RDY line.

14.5.4.2.2. Output Data Transfer (DATA Command Only)

The host can assert #OE only during DATA commands to query output data. The host must wait for the RDY line to be lowered before asserting the #OE line. Then, the host must maintain #OE

© 2026 Microchip Technology Inc. and its subsidiaries

asserted in order to allow the target device to drive the DATA[15:0] lines and write the output data. When it has completed processing the DATA command, the target device always waits for the host to assert the #OE line before driving the DATA[15:0] lines then asserting the #VALID line.

The #VALID line notifies the host that the output data is available for reading. Consequently, the target device must keep the #VALID signal asserted and the output value on the DATA[15:0] lines until it detects the next rising edge of the #OE signal.

Once it has detected the assertion of the #VALID line, the host reads the output value on the DATA[15:0] lines. Then, the host must deassert the #OE line to notify the target device that the output data has been read. Finally, the target device can release the DATA[15:0] lines before raising again the RDY line.

14.5.4.2.3. RESP Line

The target device raises the RESP line at the end of a command if an error has occurred during processing. The target device can raise the RESP line only during CMDE or DATA commands. The host keeps the RESP line to 1 until the host acknowledges it. The procedure to acknowledge the error depends only on the type of command during which the target device has raised the RESP line.

- If the RESP line has been raised to 1 at the end of a CMDE command, then the host only needs to send a single CMDE or DATA command to acknowledge the error
- If the RESP line has been raised to 1 at the end of a DATA command, then the host must send two commands: a CMDE command first, then either a CMDE or a DATA command for the second command.

The input data carried by the DATA[15:0] lines during the commands sent to acknowledge the error are not relevant. However, CMDE or DATA commands are processed normally by the target device, as if the RESP line were 0.

On error acknowledge, the target device lowers the RESP line just after it has lowered the RDY line, hence at the beginning of the CMDE or DATA command processing.

14.5.4.3. Internal Registers

The FFPI Monitor uses two internal registers: OP[7:0] and ADDR[31:0].

14.5.4.3.1. OP[7:0] Register

OP[7:0] is an 8-bit register that stores an operation code to select which operation the target device performs when it processes DATA commands. The host updates the value of the OP[7:0] register with CMDE commands. Supported operations are listed in the table below.
Note: Operation codes listed in the table below are not the low-level Flash Commands at SEFC level. These are high-level commands within the FFPI Monitor.

OP[7:0]	Operation	Description
0x00	SYNC	No operation
0x11	READ	Read Pages
0x12	WP	Write Pages
0x21	USRP	User Signature Read Pages
0x22	WPL	Write Pages then Lock Pages
0x32	EWP	Erase Block then Write Pages
0x42	EWPL	Erase Block, Write Pages then Lock Pages
0x14	SLB	Set Lock Bit
0x24	CLB	Clear Lock Bit
0x34	SGB	Set GPNVM Bit
0x44	CGB	Clear GPNVM Bit
0x52	USWP	User Signature Write Pages
0x54	SSB	Set SECURITY Bit

© 2026 Microchip Technology Inc. and its subsidiaries

OP[7:0] Register (continued)		
OP[7:0]	**Operation**	**Description**
0x62	USEWP	User Signature Erase Block then Write Pages
0x15	GLB	Get Lock Bit
0x25	GGB	Get GPNVM Bit
0x35	GSB	Get SECURITY Bit
0x06	SFB	Select Flash Bank
0x1E	VERSION	Read FFPI Monitor Version
0x1F	WRAM	Write into RAM
0x2F	RRAM	Read from RAM
0x3F	JRAM	Jump in RAM

14.5.4.3.2. ADDR[31:0] Register

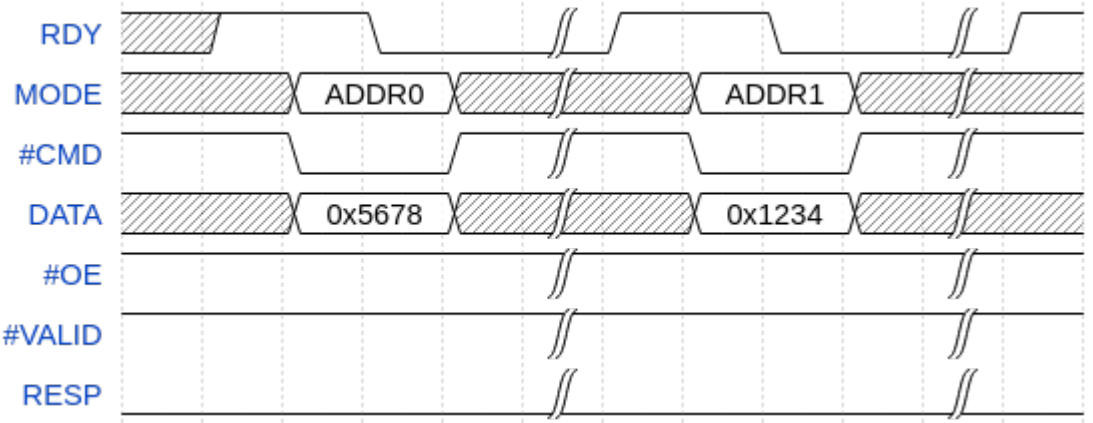
ADDR[31:0] is a 32-bit register that stores the address/offset value used as an argument by some operations. Operations making use of the ADDR[31:0] register are:

- READ
- WP
- USRP
- WPL
- EWP
- EWPL
- USWP
- USEWP
- RRAM
- WRAM
- JRAM

Before selecting any of those operations, the user should execute ADDR0 and ADDR1 commands to set the value of ADDR[31:0].

The example below sets the ADDR[31:0] register to 0x12345678.

The ADDR[31:0] register is automatically incremented by 2 after each DATA command in the operations listed above except JRAM.

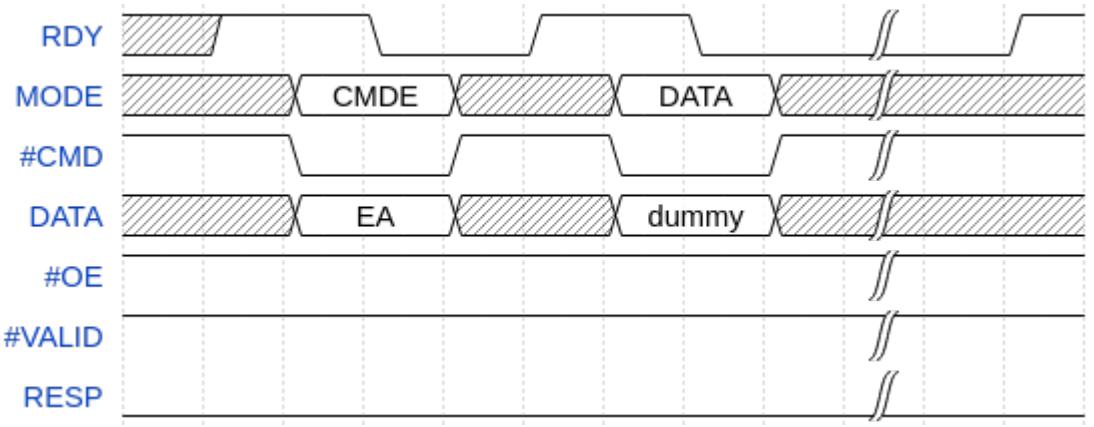


© 2026 Microchip Technology Inc. and its subsidiaries

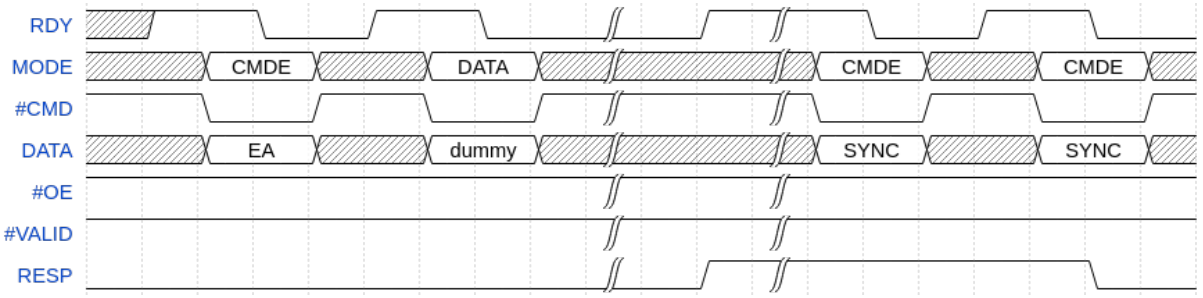
14.5.5. Erasing Flash Sectors

Prior to programming a Flash page, the 4K block containing this Flash page must be erased by using the EWP command.

Note: If a large amount of data must be programmed, for example, data spanning several 4K or 32K blocks or 128K sectors, it is advisable to use the HW Erase signal to erase the entire Flash prior to writing data into the Flash.

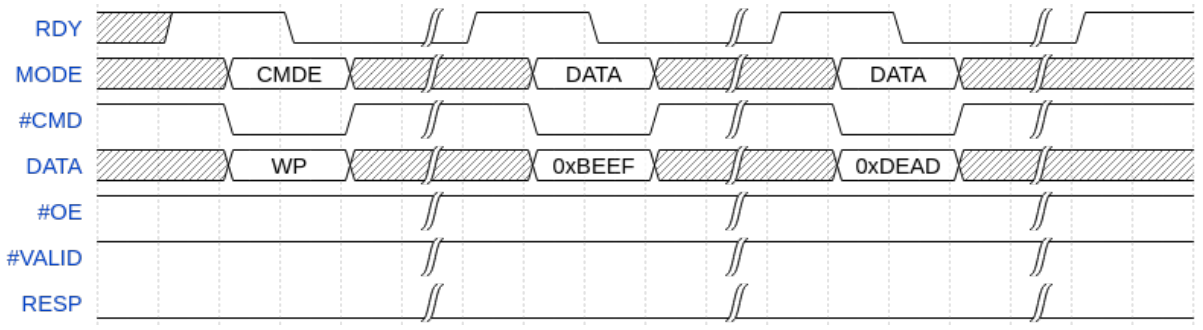


The above example shows a successful EWP command. However, if an error occurs during the erase operation the FPPI Monitor raises the RESP line as shown below:



14.5.6. Programming Flash Pages

Page programming into the internal Flash uses a page cache in internal SRAM0 to speed up to the process, writing all page data at once in the internal Flash.



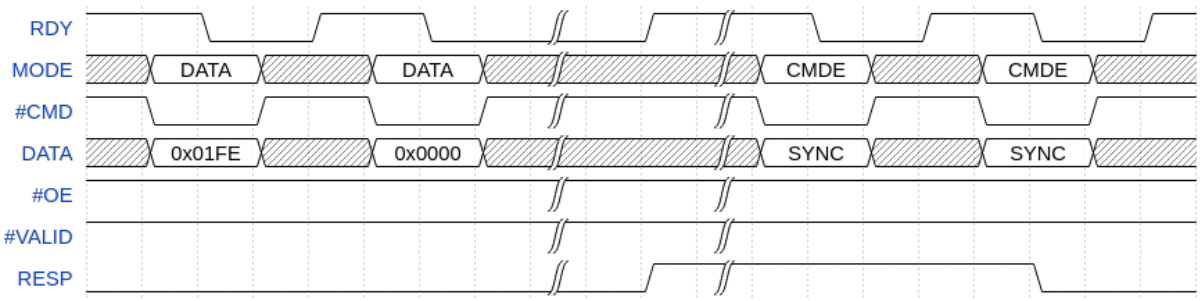
The above example shows how to write the [0xEF, 0xBE, 0xAD, 0xDE] byte sequence into the page cache. Since Flash pages are 512 bytes, the offset in the page cache where any 16-bit data is written is given by current value ADDR[8:0] before the auto increment. In turn, the page index where the page cache will be flushed into is given in ADDR[31:9]. A value of zero for page index makes the address parameter point to the very first page of the internal Flash.

© 2026 Microchip Technology Inc. and its subsidiaries

At the end of a DATA command, if the auto increment of ADDR[31:0] crosses the page boundary, then a page cache flush into the internal Flash is automatically triggered at the beginning of the next DATA command, before writing the new data into the page cache. If no more data are to be written, a page cache flush can be manually triggered by sending a CMDE command, similar to a simple SYNC operation.

Before executing a DATA command during a Write Page operation (WP, EWP, WPL, EWPL)s, the ADDR[31:0] register should always be aligned to 2 bytes, meaning ADDR[0] = 0.

If an error occurs while flushing the page cache, the FPPI Monitor raises the RESP line:

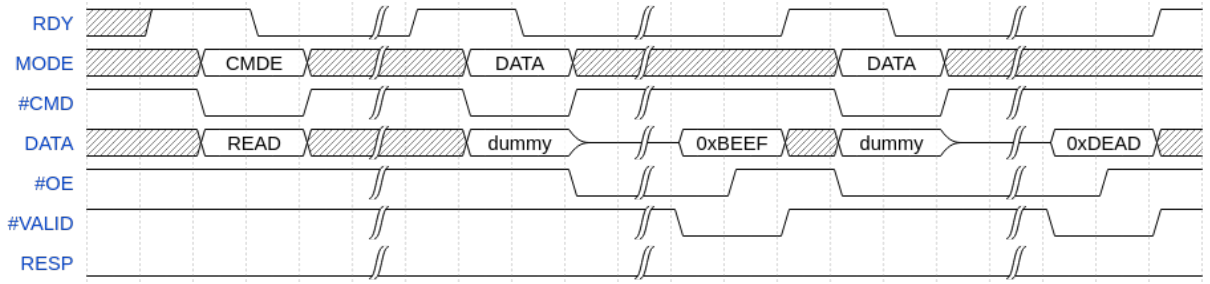


In the above example, the sequence of bytes [0xFE, 0x01, 0x00, 0x00] is written into the page cache. The page boundary is crossed after byte 0x01 so the 2nd DATA command triggers a page cache flush. This flush operation fails, so the FFPI Monitor raises the RESP line. Later the host sends two CMDE commands to acknowledge the error so the FFPI Monitor lowers the RESP line.

14.5.7. Reading Flash Pages

Flash pages are read executing READ operations. The ADDR[31:0] register must be positioned first. A zero value for ADDR[31:0] sets the current address parameter to the beginning of the internal Flash.

In the example below, the sequence of bytes [0xEF, 0xBE, 0xAD, 0xDE] is read from the internal Flash.

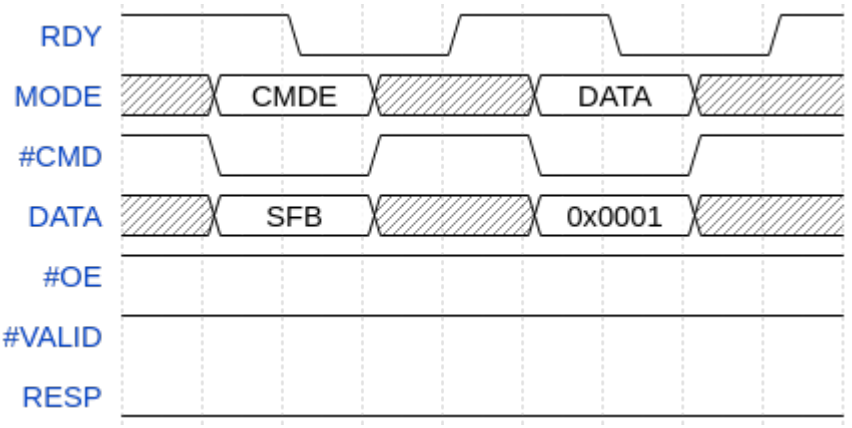


14.5.8. Selecting the Flash Bank (SEFC0 or SEFC1)

Some product parts embed two Flash banks, each controlled by the corresponding Flash controllers: either SEFC0 or SEFC1. All Flash operations, such as EA or SLB, access only the selected Flash bank. By default, the bank controlled by SEFC0 is selected. The SFB operation changes the selected Flash bank according to the input value of the DATA command following the CMDE command:

- 0: Flash Bank 0 (SEFC0) is selected
- 1: Flash Bank 1 (SEFC1) is selected

© 2026 Microchip Technology Inc. and its subsidiaries

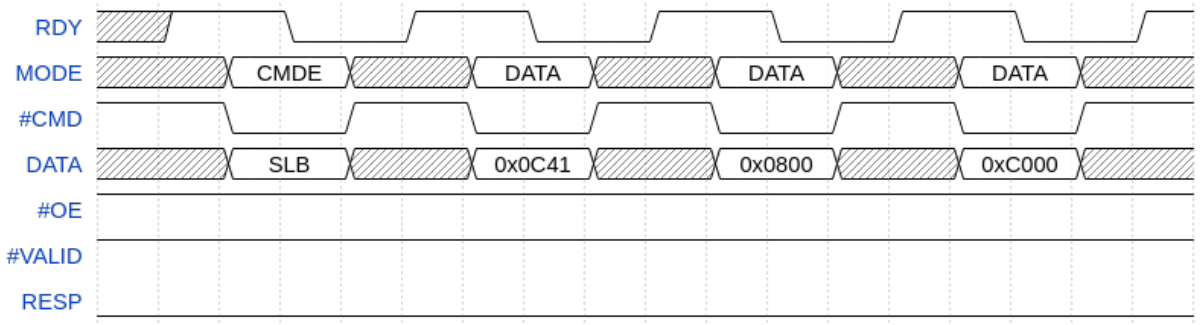


14.5.9. Managing Flash Bits

The FFPI Monitor supports a set of command to set, clear or get Flash bits. Three kinds of Flash bits can be managed:

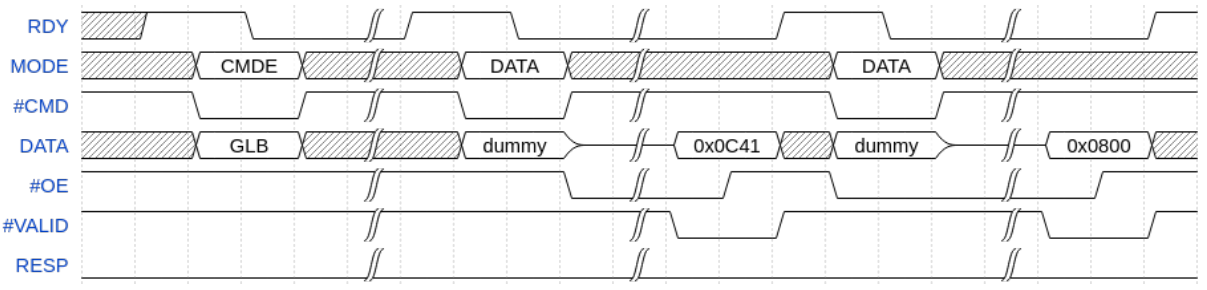
- GPNVM bits
- Lock bits
- SECURITY bit (GPNVM bit 0)

All set/clear/get operations start by a CMDE command with the relevant op code, followed by one or more DATA commands carrying data. The first data carries bits [15:0], the second data carries bits [31:16], the third data carries bits [47:32], etc.



As an example, the above command sets Lock bits 0, 6, 10, 11, 27, 46 and 47 in the selected Flash bank.

Then the Get Lock Bit operation can be executed to read back:



Set Bit operations only update the value of bits set in the bitmask carried by DATA commands but leave the values of other bits unchanged. Also, Clear Bit operations only update the value of bits set in the bitmask, leaving values of bits not set in the bitmask unchanged.

© 2026 Microchip Technology Inc. and its subsidiaries

14.5.10. Read, Write or Jump into RAM

RRAM and WRAM operations perform 16-bit read/write accesses into any RAM area of the system memory. The ADDR[31:0] register must be updated first with the address of in the system memory of the 16-bit word in RAM to be read or written. Then, the ADDR[31:0] register is automatically incremented by 2 after each DATA command within the RRAM or WRAM operation. There is no alignment constraint when accessing the system RAM.

Finally, the JRAM operation makes the FFPI Monitor jump into an address in the system RAM to execute the binary code previously loaded into this address. The binary code is executed after the target device has processed a DATA command within a JRAM operation.

© 2026 Microchip Technology Inc. and its subsidiaries

15. Supply Controller (SUPC)

15.1. Description

The Supply Controller (SUPC) is a VDDBU-powered (resulting voltage from VBAT/VDD3V3 selection) peripheral in charge of:

- Sequencing the internal circuits (VDDCORE regulator, oscillators, etc.) at device power-up or wake-up, as well as when entering a low-power mode (Backup mode, Wait mode),
- Slow clock (MD_SLCK and TD_SLCK) generation
- Internal supply regulator control
- Control of external power supply circuit (via SHDN pin)
- Supply Monitoring functions (Power-on Reset, Supply Monitors) management
- Wake-up event detection
- Tamper event detection

The SUPC configuration is saved as long as VDDBU supply is maintained.

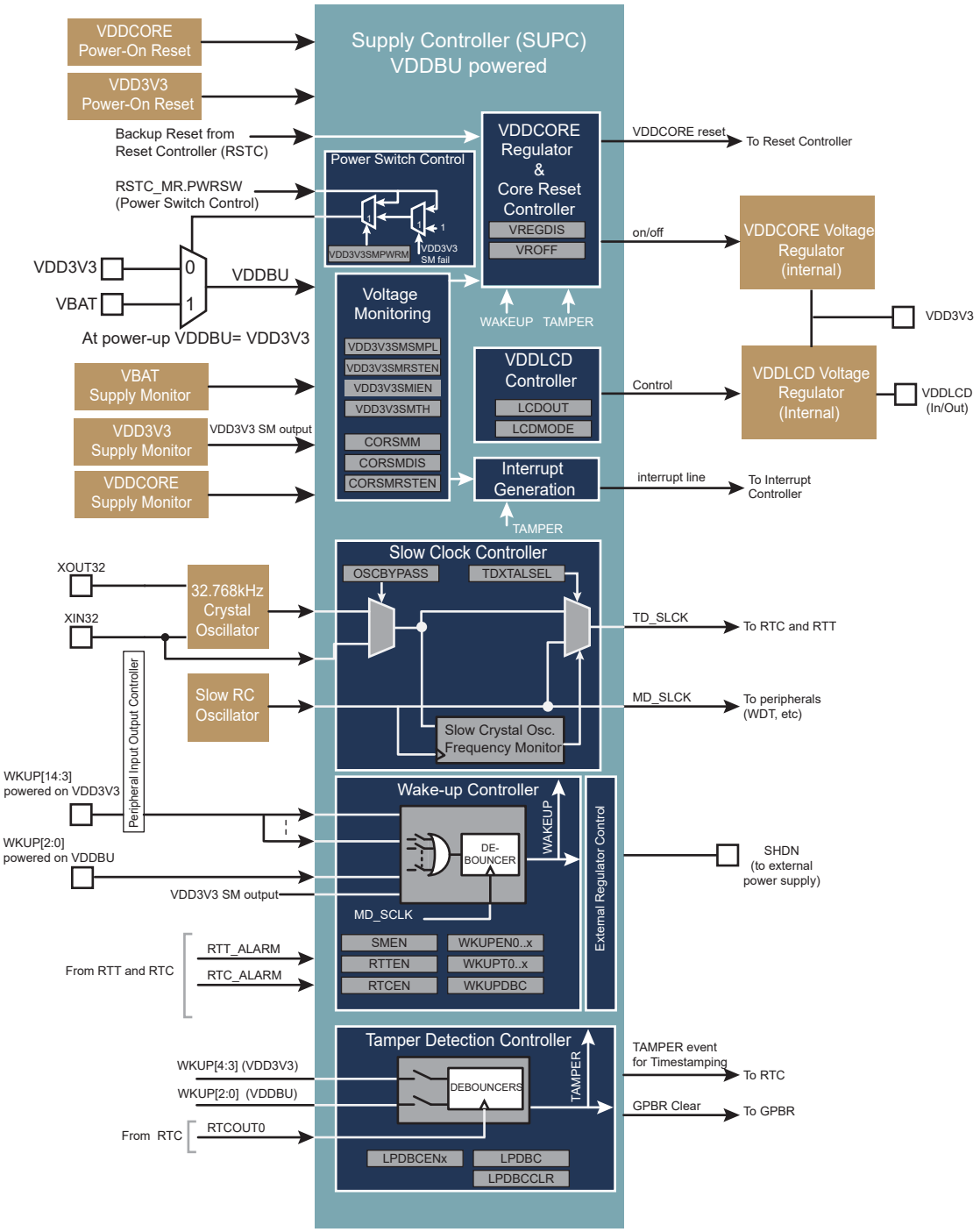
15.2. Embedded Characteristics

- Management of the Embedded Voltage Regulator to Enter In and Exit from Backup Mode
- Management of Supply Monitors on VDD3V3 and VDDCORE to Trigger a Reset
- Shutdown Logic for External VDDCORE Generation
  - Software assertion of the Shutdown Output pin (SHDN)
  - Automatic de-assertion from the configurable wake-up events
- Slow Clock Generation
  - MD_SLCK—Monitoring domain slow clock. This clock, sourced from the Slow RC oscillator, is the only permanent clock of the system and feeds the safety-critical functions of the device (WDT, RSTC, SUPC, frequency monitors and detectors, PMC start-up time counters). Its source cannot be modified.
  - TD_CLK—Timing domain slow clock. This clock, generally sourced from the 32.768 kHz crystal oscillator, is routed to the RTC and RTT peripherals.
- Independent Tamper Detection on 5 Inputs with Configurable Backup Mode Exit
- Programmable Immediate Clear of the General-Purpose Backup Registers (GPBR) on Tamper Events
- Support of Multiple Wake-Up Sources to Exit from Backup Mode
  - 15 wake-up inputs with programmable debouncers
  - Real-Time Clock alarm
  - Real-Time Timer alarm
  - Supply monitor detection on VDD3V3, with programmable sampling period and voltage threshold
  - Detection of VDDCORE rising edge when externally supplied

© 2026 Microchip Technology Inc. and its subsidiaries

15.3. Block Diagram

Figure 15-1. SUPC Block Diagram



15.4. Functional Description

15.4.1. Overview

The Supply Controller (SUPC) is powered by VDDBU and clocked by MD_SLCK (slow clock). MD_SLCK is always driven by the slow RC oscillator for safety/security reasons.

© 2026 Microchip Technology Inc. and its subsidiaries

At power-up, as soon as the VDDBU voltage passes over the VDD3V3 POR threshold voltage, the SUPC sequentially starts the VDDCORE voltage regulator and deactivates the reset of the VDDCORE domain.

The system can be put in Backup mode (Low-power mode) by configuring the SUPC; exit from Backup mode can be triggered by different programmable sources (RTC, RTT, WKUPx pins, Supply Monitor).

When entering or exiting Backup mode, the SUPC controls the VDDCORE voltage regulator and the reset of the VDDCORE domain.

At power-up, TD_SLCK is driven by the slow RC oscillator. Once configured on the 32.768 kHz crystal oscillator, TD_SLCK is driven by this 32.768 kHz crystal oscillator as long as VDDBU voltage is active.

A slow crystal oscillator monitoring circuitry is active in all system modes including Backup mode, and can be configured to automatically switch the TD_SLCK clock source from the slow crystal oscillator to the slow RC oscillator.

The Supply Monitoring section of the SUPC includes a programmable supply monitor on VDD3V3. The output can either generate a software interrupt or a reset of the VDDCORE domain.

The SUPC features a tamper detection circuit on the WKUP[14:0] signals. This detection is active in all power modes. When only VBAT is supplied, 3 tamper detection pins are available (WKUP[2:0]. Upon a tamper detection event, the SUPC can be configured to automatically clear the General-Purpose Backup Registers (GPBR) and the tamper event is routed to the RTC for timestamping purposes.

The SUPC can be configured to control an external power supply, for example an external voltage regulator, by driving the SHDN pin.

To perform a shutdown of the external voltage regulator, the software asserts the SHDN pin by writing SHDW=1 in the Control register (SUPC_CR). The shutdown is active only two MD_SLCK cycles after the write of SUPC_CR. This register is password-protected and so the value written must contain the correct key for the command to be taken into account. As a result, the SHDN pin is driven low and the system is powered down (VDDCORE voltage equals 0).

15.4.2. Slow Clock Generator

As soon as VDD3V3 is supplied, both the 32.768 kHz crystal oscillator and the slow RC oscillator are powered, but only the slow RC oscillator is enabled. The slow RC oscillator provides a faster start-up time than the 32.768 kHz crystal oscillator.

The user can select the 32.768 kHz crystal oscillator to be the source of TD_SLCK, as it provides higher accuracy than the slow RC oscillator especially when an RTC is required by the application. The slow crystal oscillator is selected when SUPC_CR.TDXTALSEL=1.

For MD_SLCK clock, the slow RC oscillator is always the source. There is no way to modify the source of this clock. This ensures a reliable monitoring clock for the system due to the monolithic nature of the slow RC oscillator.

The following sequences must be followed to switch from the slow RC oscillator to the 32.768 kHz crystal oscillator:

Case 1: The 32.768 kHz crystal oscillator was not previously enabled.

1. Start and select the 32.768 kHz crystal oscillator as the source of TD_SLCK by setting SUPC_CR.TDXTALSEL.
2. The switching of TD_SLCK is effective when TDOSCSEL is set in the Status register (SUPC_SR).
   Note: This flag is set at the end of the 32.768 kHz crystal oscillator start-up period, which can last from a few hundreds of millisecond to a few seconds.

Case 2: The 32.768 kHz crystal oscillator is already enabled.

© 2026 Microchip Technology Inc. and its subsidiaries

1. Select the 32.768 kHz crystal oscillator as the source of TD_SLCK by setting SUPC_CR.TDXTALSEL.
2. The switching of TD_SLCK is effective when SUPC_SR.TDOSCSEL is set.

The switching time may vary depending on the slow RC oscillator output frequency. The switch of the slow clock source is glitch-free.

Reverting to the slow RC oscillator as the clock source of TD_SLCK is only possible by powering down the selected voltage source of VDDBU (VDD3V3 or VBAT).

The user can also set the 32.768 kHz crystal oscillator in Bypass mode instead of connecting a crystal. In this case, the user has to provide the external clock signal on XIN32. The input characteristics of the XIN32 pin are given in the section “Electrical Characteristics”. To enter Bypass mode, SUPC_MR.OSCBYPASS must be set to 1 before setting TDXTALSEL.

15.4.3. Core Voltage Regulator and Low-Power Modes

The SUPC controls the embedded VDDCORE voltage regulator. It controls its power-up sequence to ensure an optimal load.

The user can switch off the voltage regulator, thus putting the device into Backup mode, by writing SUPC_CR.VROFF to 1.

If an external power supply is used, the device is put into Backup mode by writing SUPC_CR.SHDW to 1.

When the internal voltage regulator is not used and VDDCORE is externally supplied, the voltage regulator can be disabled by writing SUPC_MR.VREGDIS to 1. In this case, SUPC_EMR.COREBGEN must be set to 1 once the external VDDCORE is supplied.

This asserts the VDDCORE domain reset signal after the write resynchronization time, which lasts two slow clock cycles (worst case). Once the VDDCORE domain reset signal is asserted, the processor and the peripherals are stopped one slow clock cycle before the core power supply shuts off.

15.4.4. VDD3V3 Supply Monitor

The SUPC embeds a supply monitor which monitors the VDD3V3 rail.

The VDD3V3 supply monitor can be used to prevent the processor from falling into an unpredictable state if the main power supply drops below a certain level.

The threshold of the VDD3V3 supply monitor is programmable in the IOSMTH field of the Supply Monitor Mode register (SUPC_SMMR). Refer to the section “Electrical Characteristics”.

The VDD3V3 supply monitor can also be enabled during one slow clock period on every one of either 32, 256 or 2048 slow clock periods, depending on the user selection. This is configured in the IOSMSMPL field in SUPC_SMMR.

Enabling the VDD3V3 supply monitor for such reduced times divides the typical supply monitor power consumption by factors of 2, 16 and 128, respectively, if continuous monitoring of the VDD3V3 power supply is not required.

A VDD3V3 supply monitor detection generates either a reset of the VDDCORE domain or a wake-up (exit from Backup mode). Generating a VDDCORE domain reset when a VDD3V3 supply monitor detection occurs is enabled by writing SUPC_SMMR.IOSMRSTEN=1.

Waking up the device when a VDD3V3 supply monitor event occurs can be enabled by writing IOSMWKEN=1 in the Backup Mode register (SUPC_BMR).

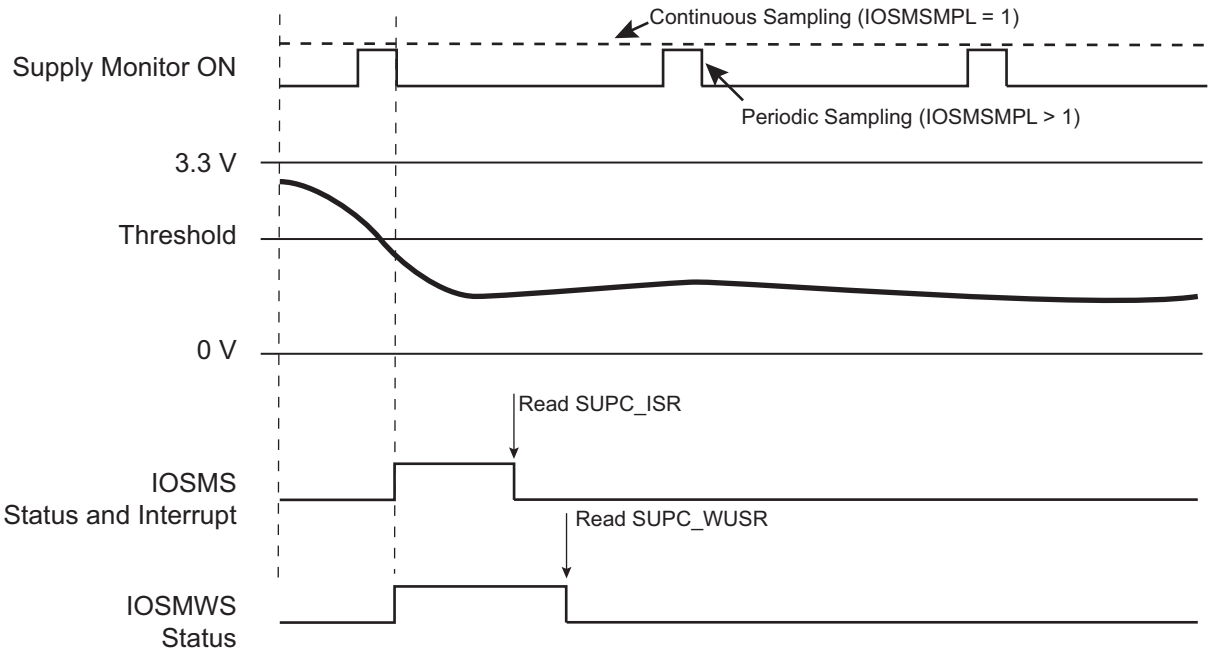
A VDD3V3 supply monitor under voltage event can trigger an interrupt by writing IOSMEV=1 in the Interrupt Enable register (SUPC_IER).

The SUPC provides the following status bit:

© 2026 Microchip Technology Inc. and its subsidiaries

- SUPC_SR.IOSMWS: The flag rises when a VDD3V3 supply monitor event triggers a wake-up of the system. This flag is a copy of IOSMWS in the Wakeup Status register (SUPC_WUSR) and thus is cleared when SUPC_WUSR is read.
- SUPC_SR.IOSMRSTS: The flag rises when VDD3V3 supply monitor under voltage event triggers a VDDCORE reset. This flag is a copy of SUPC_WUSR.IOSMRSTS and thus is cleared when SUPC_WUSR is read.
- SUPC_SR.IOSMS: VDD3V3 supply monitor output.
- SUPC_SR.IOSMIS: The flag rises when VDD3V3 supply monitor under voltage event occurs. This flag is a copy of SUPC_ISR.IOSMEV and thus is cleared when the SUPC_ISR is read.

Figure 15-2. VDD3V3 Supply Monitor Status Bit and Associated Interrupt



15.4.5. Power-up, Power-down and SUPC Reset

The SUPC is reset (Backup area reset) by powering down the selected voltage source of VDDBU (VDD3V3 or VBAT).

As long as a Backup area reset is asserted, the system is in the following state:

- The SUPC is in reset state, and consequently,
- The VDDCORE core voltage regulator is off and the VDDCORE reset signal is asserted. The I/O pins default to their reset state as described in the section “Package and Pinout”.

The slow RC oscillator, powered by VDDBU, is off as long as VDDBU voltage remains low.

As soon as VDDBU voltage is valid, the SUPC exits reset state. Five slow RC oscillator cycles later, the SUPC enables the VDDCORE voltage regulator and the VDDCORE POR. When the output signal of this VDDCORE POR indicates a valid VDDCORE voltage for at least one slow RC oscillator cycle, the VDDCORE core logic reset signal is de-asserted and the Reset Controller (RSTC) de-asserts the downstream reset signals in the VDDCORE domain.

At power-down, the SUPC is reset and its configuration is immediately lost. Without any sequencing, the core voltage regulator is turned off, the core reset signal (vddcore_nreset) is asserted, the I/O configuration is lost (I/Os default to their reset state), the oscillators and PLLs are switched off. Considering the uncontrolled nature of this power-down, it is strongly recommended to have the device in an idle state before reaching VDDBU low voltage threshold.

© 2026 Microchip Technology Inc. and its subsidiaries

15.4.6. VDDCORE Domain Reset

The SUPC manages the VDDCORE reset signal sent to the Reset Controller, as described in Power-up, Power-down and SUPC Reset. The VDDCORE domain reset signal is asserted before shutting down the VDDCORE power supply and released as soon as the VDDCORE power supply is valid.

There are two additional sources (different from watchdog, user reset) which can be programmed to trigger a VDDCORE domain reset:

- a VDD3V3 supply monitor under voltage detection
- a VDDCORE supply monitor under voltage detection

15.4.6.1. VDD3V3 Supply Monitor Reset

The VDD3V3 supply monitor can be configured to generate a reset of the VDDCORE domain This is enabled by setting the SUPC_SMMR.VDD3V3SMRSTEN=1.

If SUPC_SMMR.VDD3V3SMRSTEN=1 and a VDD3V3 supply monitor event occurs, the VDDCORE source reset signal is immediately activated for a minimum of one slow clock cycle.

15.4.6.2. VDDCORE Supply Monitor Reset

The VDDCORE supply monitor can be configured to generate a reset of the VDDCORE domain. If the VDDCORE supply monitor output is low for more than one slow clock period, the SUPC asserts a VDDCORE reset if SUPC_MR.CORSMRSTEN=1.

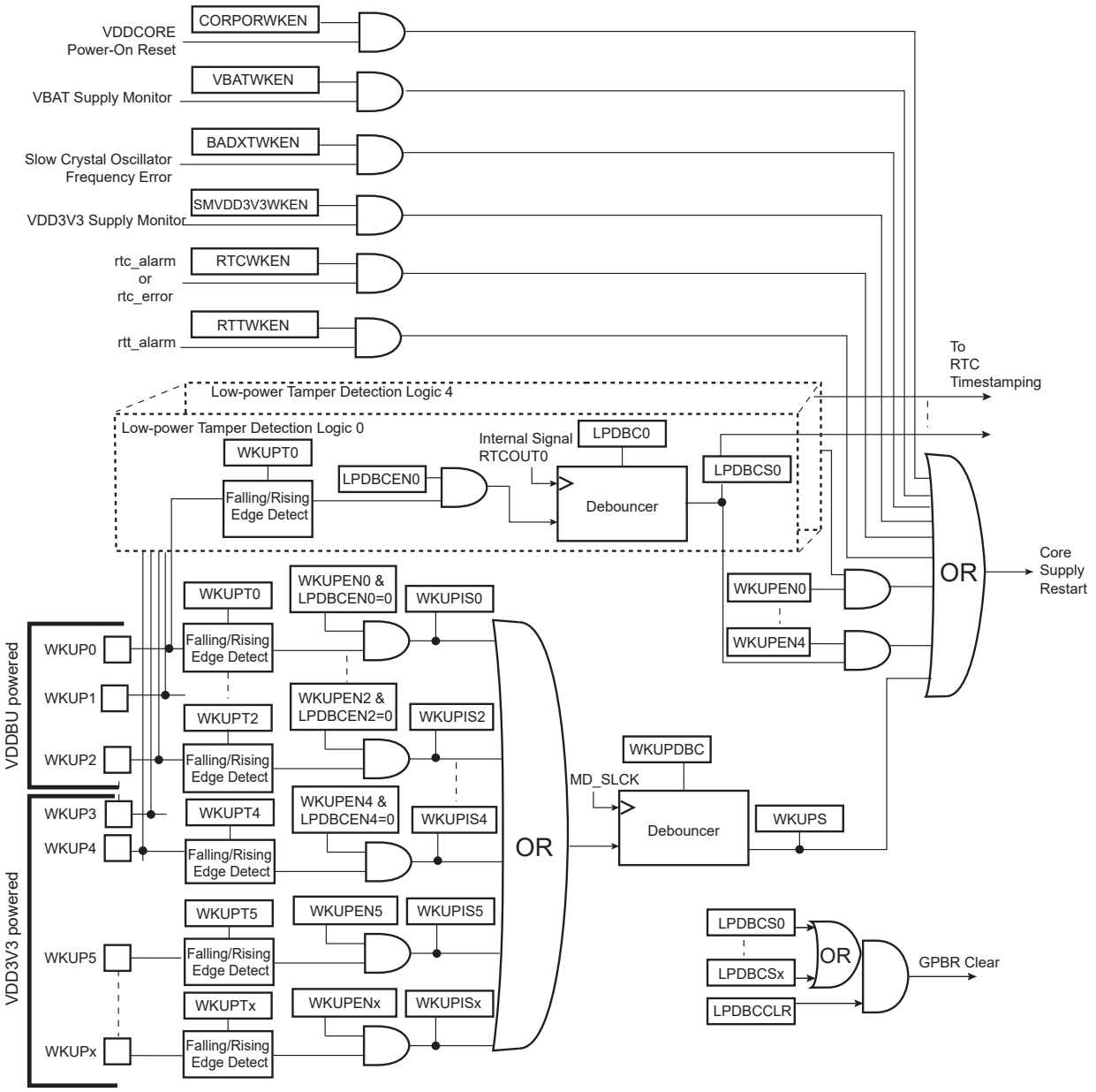
If SUPC_MR.CORSMRSTEN=1 and the VDDCORE voltage regulation is below the threshold configured in the supply monitor, the VDDCORE reset signal is asserted for a minimum of one slow clock cycle and de-asserted as soon as supply monitor indicates a valid voltage.

15.4.7. Wake-up Sources

The wake-up events allow the device to exit from Backup mode. When a wake-up event is detected, the SUPC performs a sequence that automatically enables the core voltage regulator.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 15-3. Wake-up Sources



15.4.7.1. Wake-up Inputs

The wake-up inputs, WKUPx, can be programmed to perform a wake-up of the device from Backup mode. Each input can be enabled by writing a 1 to the corresponding bit, WKUPENx, in the Wakeup Inputs register (SUPC_WUIR). The wake-up level can be selected with the corresponding polarity bit, WKUPTx, also located in SUPC_WUIR.

WKUP0 to WKUP2 pins are located on the VDDBU power rail. All other wake-up inputs are located on the VDD3V3 power rail, and so they cannot be used if VDD3V3 is not supplied.

The resulting signals are wired-ORed to trigger a debounce counter, which is programmed with the WKUPDBC field in the Wakeup Mode register (SUPC_WUMR). SUPC_WUMR.WKUPDBC selects a debouncing period of 3, 32, 512, 4096 or 32768 slow clock cycles. The duration of these periods corresponds, respectively, to about 100 µs, about 1 ms, about 16 ms, about 128 ms and about 1 second (for a typical slow clock frequency of 32 kHz). Programming WKUPDBC to 0x0 selects an

© 2026 Microchip Technology Inc. and its subsidiaries

immediate wake-up, i.e., an enabled WKUP pin must be active according to its polarity during a minimum of one slow clock period to wake up the core power supply. Once a wake-up event has been detected, the SHDN pin is released.

If an enabled WKUP pin is asserted for a duration longer than the debouncing period, a wake-up of the core voltage regulator is started and the wake-up signals are latched in SUPC_SR. Refer to the figure Wake-up Sources. This allows the user to identify the source of the wake-up. However, if a new wake-up condition occurs, the primary information is lost. No new wake-up can be detected since the primary wake-up condition has disappeared.

15.4.7.2. Wake-Up by RTC and RTT Alarm

The RTC and the RTT can be programmed to perform a wake-up of the device from Backup mode. This can be enabled by writing SUPC_BMR.RTCWKEN=1 and/or SUPC_BMR.RTTWKEN=1.

When the RTC is enabled to perform a system wake-up, the triggering event can be either the alarm event configured by writing RTC alarm registers (RTC_TIMALR, RTC_CALALR) or an abnormal behavior of the RTC reported by the on-the-fly RTC integrity circuitry. To determine which source triggered the RTC wake-up, refer to the RTC_SR in the section “Real-Time Controller (RTC)”.

Refer to the sections “Real-Time Controller (RTC)” or “Real-Time Timer (RTT)” to configure the alarms (condition of wake-up).

The SUPC does not provide any status as the information is available in the user interface of either the RTT or the RTC.

15.4.7.3. Wake-up by VDD3V3 Supply Monitor Detection

The VDD3V3 supply monitor can generate a wake-up of the VDDCORE domain if SUPC_BMR.VDD3V3SMWKEN=1. See VDD3V3 Supply Monitor.

15.4.7.4. Wake-up by VBAT Supply Monitor Detection

The VBAT supply monitor can generate a wake-up of the VDDCORE domain if the VBAT voltage drops down a predefined threshold (Power-on Reset threshold) if SUPC_BMR.VBATWKEN=1.

15.4.7.5. Wake-up by Slow Crystal Oscillator Frequency Monitor

The slow crystal oscillator frequency monitor can generate a wake-up of the VDDCORE domain if SUPC_BMR.BADXTWKEN=1.

15.4.7.6. Tamper Detection

Tamper detection is done through wake-up pins (WKUP) with a specific configuration. Up to 5 WKUP inputs can be enabled to act as tamper detection. Some wake-up/tamper pins are referenced in the VDDBU domain (output of the power switch) and some others in the VDD3V3 domain.

Refer to the table “Pinout and Multiplexing”.

A tamper detection can perform a partial or full immediate clear of the General-Purpose Backup, an immediate clear of the keys of the AES and the AESB crypto engine as well as the scrambling keys of the Quad SPI (QSPI). In addition, a tamper detection can lock the SHA.

Each tamper input has a dedicated Low Power Debouncer (LPDBC). For wake-up pins to be enabled as tamper to perform the actions defined above, follow the steps:

1. Configure the corresponding LPDBCENx bit in SUPC_WUMR to 1.
2. Configure the corresponding LPDBCx bit field in SUPC_WUMR to a value other than 0.

For the different possible actions on tamper events, configure the corresponding peripheral.

Some power reduction can be performed in the tamper circuitry. For example, if the tamper sensor is biased through a resistor and constantly driven by the power supply when the tamper is active, this leads to power consumption as long as the tamper detection switch is in its active state. To reduce the energy when the switch is in active state, the tamper sensor circuitry can be intermittently powered, and thus a specific waveform must be applied to the sensor circuitry.

© 2026 Microchip Technology Inc. and its subsidiaries

The waveform is generated using RTCOUT0 in all modes including Backup mode. Refer to the section “Real-Time Clock (RTC)” for waveform generation.

Figure 15-4 and Figure 15-5 below show examples of optimized power consumption circuitry where two tamper switches are used. RTCOUT0 powers the external pullup used by the tamper sensor circuitry.

The SUPC provides two modes for driving the pull-up/down resistor with RTCOUT0. The waveform provided by the RTCOUT0 pin differs slightly depending on the mode configured in SUPC_BMR.MRTCOUT.

When SUPC_BMR.MRTCOUT=1, the RTCOUT0 pin is stuck at 1 while there is no tamper detected on any inputs, thus no dynamic power consumption due to RTCOUT0 swicthing. As soon as a tamper detection occur (SUPC_SR.LPDBCSx differs from 0), RTCOUT0 starts powering the tamper detection circuitry in an intermittent way to reduce the energy when the switch is in active state.

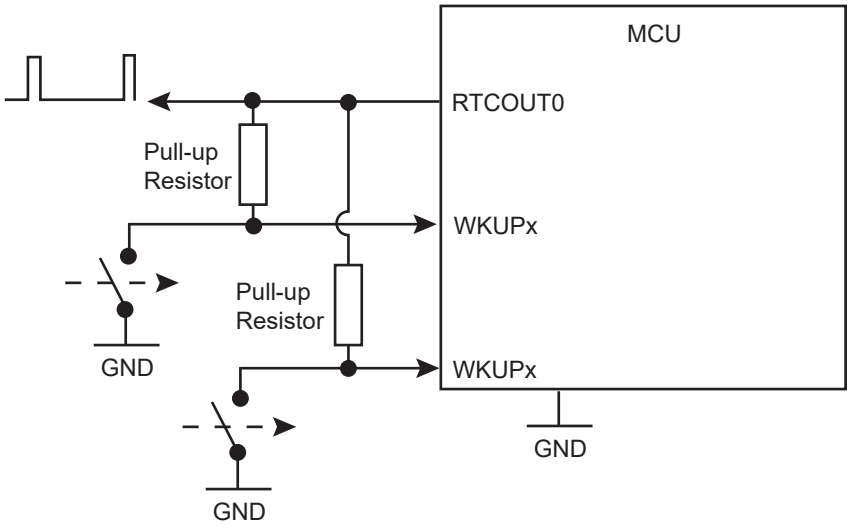
Figure 15-6 shows the waveforms provided by RTCOUT0 according to the configuration of SUPC_BMR.MRTCOUT.

The WKUP inputs enabled for tamper detection can be configured to perform a system wake-up upon tamper detection. This is enabled by writing SUPC_WUMR.LPDBCENx=1 and SUPC_WUIR.WKUPENx=1.

The WKUP inputs enabled for tamper detection can also be used when VDDCORE is powered.

Low-power tamper detection requires the RTC to be configured to generate a duty cycle programmable pulse (i.e., OUT0 = 0x7 in RTC_MR) in order to create the sampling points of both debouncers. For the debouncer circuitry, the sampling point is the falling edge of the RTCOUT0 waveform (carried on internal wire of the system).

Figure 15-4. Low-power Debouncer (Push-to-Make Switch, Pull-up Resistors)



© 2026 Microchip Technology Inc. and its subsidiaries

Figure 15-5. Low-power Debouncer (Push-to-Break Switch, Pull-down Resistors)

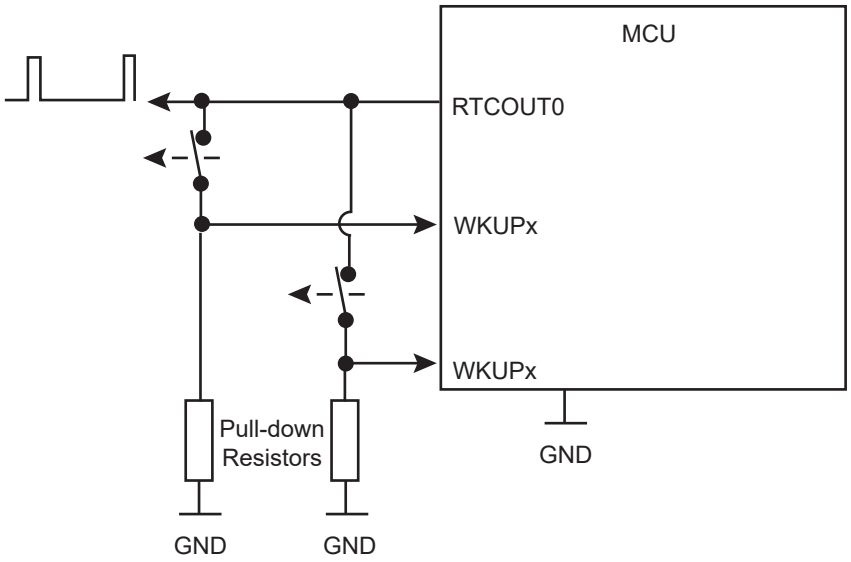
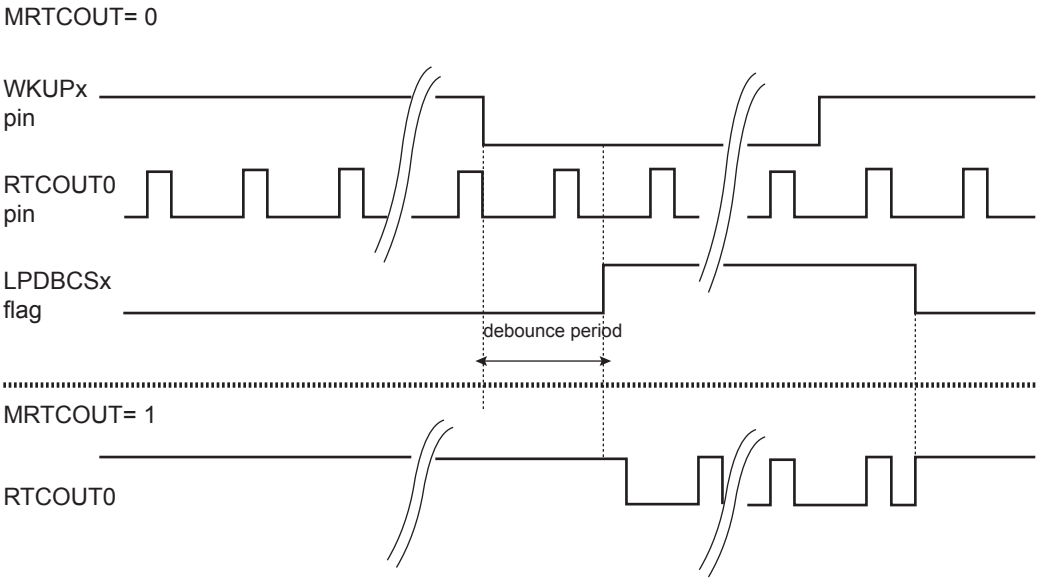


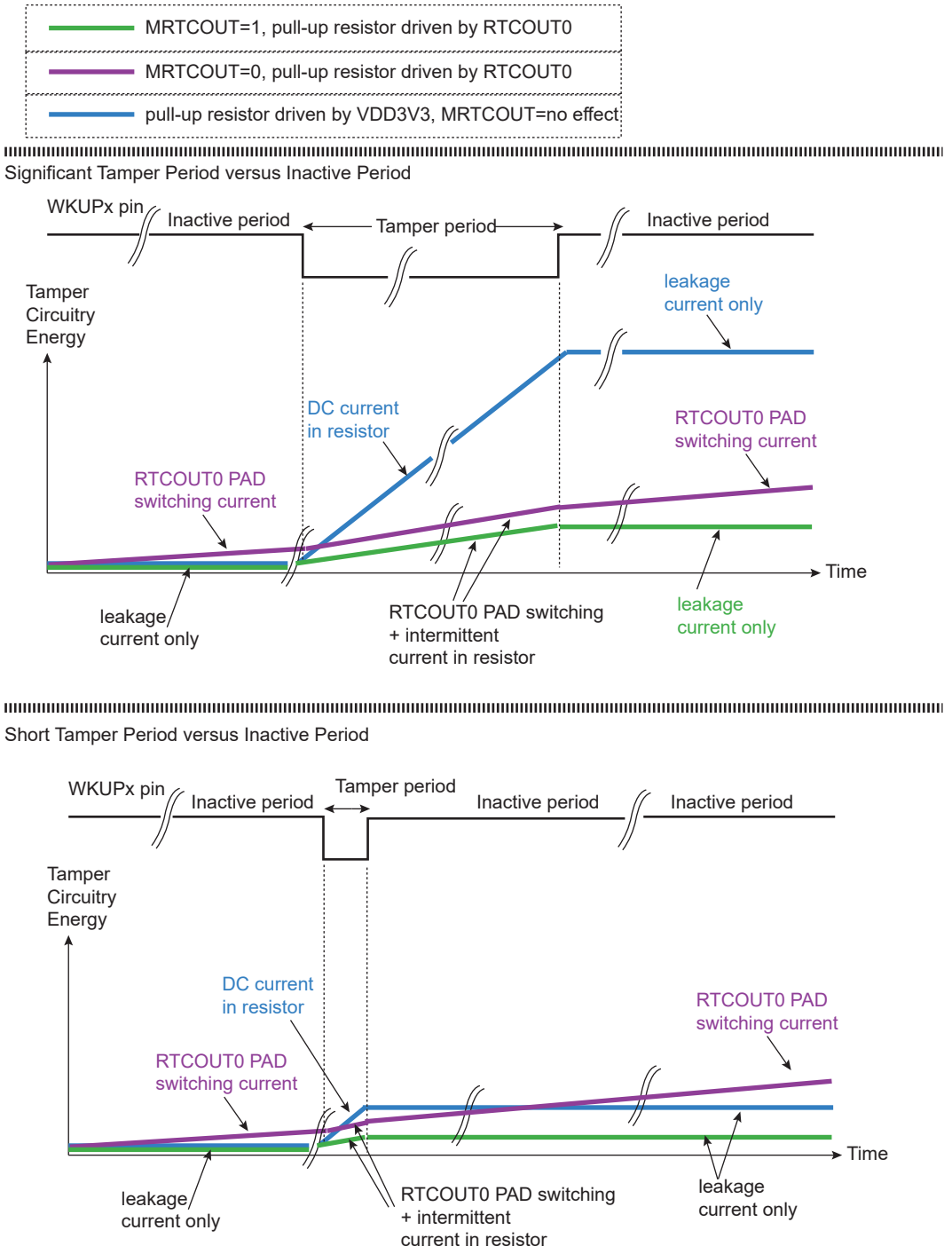
Figure 15-6. Low-power Debouncer Waveforms



The figure below shows the energy consumed by the tamper detection circuitry when configured in different operating modes. When the system is in Backup mode, the energy required by the tamper circuitry depends on the ratio tamper period versus inactive period.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 15-7. Energy Required by Tamper Detection Circuitry



The debouncing period duration is configurable. The period is set for all debouncers; the duration cannot be adjusted for each debouncer. The number of successive identical samples to wake up the system can be configured from 2 up to 8 in SUPC_WUMR.LPDBC. The period of time between two samples can be configured by programming RTC_MR.TPERIOD. Power parameters can be adjusted by modifying the period of time in RTC_MR.THIGH.

The wake-up polarity of the inputs can be independently configured by writing SUPC_WUIR.WKUPTx.

© 2026 Microchip Technology Inc. and its subsidiaries

In order to determine which wake-up/tamper pin triggers the system wake-up, a status flag is associated for each low-power debouncer. These flags can be read in SUPC_SR (no clear on read) or in SUPC_ISR (cleared on read).

A tamper detection can perform a partial or full immediate clear of the General-purpose Backup registers by writing RTC_TAMPER.TAMPCLR=1 (refer to the section “Real-Time Controller (RTC)”).

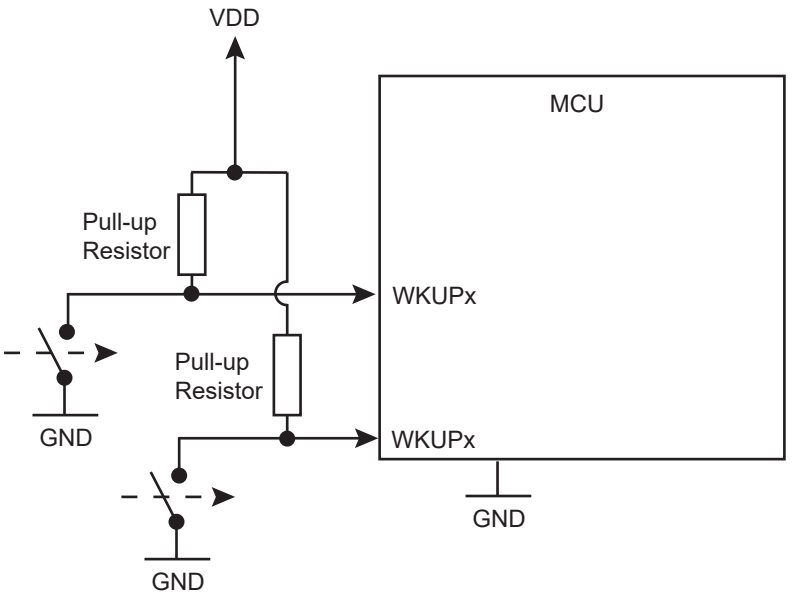
A tamper detection can perform an immediate clear of the keys of the AES crypto engine. Refer to the section “Advanced Encryption Standard (AES)” to enable the immediate clear.

A tamper detection can perform an immediate clear of the keys of the AES Bridge (AESB) and the scrambling keys of the Quad SPI (QSPI). Refer to the sections “AES Bridge (AESB)” and “Quad SPI (QSPI)” to enable the immediate clear.

Note that it is not mandatory to use the RTCOUT0 pin when using the WKUPx pins as tampering inputs in any mode. Using the RTCOUT0 pin provides a “sampling mode” to further reduce the power consumption of the tamper detection circuitry. If RTCOUT0 is not used to bias external pull-ups or pull-down, the RTC.OUT0 field must be configured to create an internal sampling point for the debouncer logic. The period of time between two samples can be configured by programming RTC_MR.TPERIOD.

The following figure illustrates the use of WKUPx without the RTCOUT0 pin. Note that VDD in the figure below can be VBAT or VDD3V3 depending on the wake-up/tamper pins used.

Figure 15-8. Using WKUP Pins Without RTCOUT0 Pins



15.4.8. IO Settings Retention Mode

Before entering Backup mode with VDD3V3 voltage always supplied, the IO lines can be configured to keep their current states (input/output/pull-up/pull-down) when VDDCORE voltage is no longer supplied. This retention mode prevents any extra power consumption, current path or electrical compatibility with external devices connected to the IO lines.

When waking up from Backup mode (with no VDD3V3 voltage supplied), all IO lines are reset to their default state. Refer to “Reset State” in the table “Pin Description”.

IO retention can be configured in the IO Retention register. The granularity of the configuration is 8. Refer to the IO Retention register in the section “Special Function Registers (SFR)” for further details.

© 2026 Microchip Technology Inc. and its subsidiaries

15.4.9. Backup Domain IO Isolation

If VDD3V3 is removed when the device is in Backup mode, isolation must be enabled (SUPC_MR.IO_BACKUP_ISO=0) before entering Backup mode and must disabled (SUPC_MR.IO_BACKUP_ISO=1) after a wake-up from Backup mode. If VDD3V3 is not removed when the device is in Backup mode, SUPC_MR.IO_BACKUP_ISO must be kept at '1'.

15.4.10. Register Write Protection

To prevent any single software error from corrupting SYSC behavior, certain registers in the address space can be write-protected by writing SYSC_WPMR.WPEN=1.

The following registers can be write-protected:

- SUPC Control Register
- SUPC Supply Monitor Mode Register
- SUPC Mode Register
- SUPC Wakeup Mode Register
- SUPC Wakeup Inputs Register
- SUPC Extended Mode Register
- SUPC Backup Mode Register

15.4.11. Register Bits in Backup Domain (VDDBU)

The following configuration registers, or certain bits of these registers, are physically located in the product backup domain:

- SUPC Control Register (see register description for details)
- SUPC Supply Monitor Mode Register (all bits)
- SUPC Mode Register (see register description for details)
- SUPC Wakeup Mode Register (all bits)
- SUPC Wakeup Inputs Register (all bits)
- SUPC Status Register (all bits)
- SUPC Extended Mode Register (all bits)
- SUPC Backup Mode Register (all bits)

© 2026 Microchip Technology Inc. and its subsidiaries

15.5. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	SUPC_CR	31:24	KEY[7:0]							
		23:16								
		15:8								
		7:0					TDXTALSEL	VROFF	SHDWEOF	SHDW
0x04	SUPC_SMMR	31:24								
		23:16								
		15:8			VDD3V3SMPWRM	VDD3V3SMRSTEN		VDD3V3SMSMPL[2:0]		
		7:0					VDD3V3SMTH[3:0]			
0x08	SUPC_MR	31:24	KEY[7:0]							
		23:16				OSCBYPASS				
		15:8		IO_BACKUP_ISO	CORSMDIS	CORSMRSTEN				
		7:0	VREGDIS	CORSMM	LCDMODE[1:0]		LCDOUT[3:0]			
0x0C	SUPC_WUMR	31:24		LPDBC4[2:0]			LPDBC3[2:0]			LPDBC2[2]
		23:16	LPDBC2[1:0]		LPDBC1[2:0]			LPDBC0[2:0]		
		15:8		WKUPDBC[2:0]				FWUPDBC[2:0]		
		7:0				LPDBCEN4	LPDBCEN3	LPDBCEN2	LPDBCEN1	LPDBCEN0
0x10	SUPC_WUIR	31:24		WKUPT14	WKUPT13	WKUPT12	WKUPT11	WKUPT10	WKUPT9	WKUPT8
		23:16	WKUPT7	WKUPT6	WKUPT5	WKUPT4	WKUPT3	WKUPT2	WKUPT1	WKUPT0
		15:8		WKUPEN14	WKUPEN13	WKUPEN12	WKUPEN11	WKUPEN10	WKUPEN9	WKUPEN8
		7:0	WKUPEN7	WKUPEN6	WKUPEN5	WKUPEN4	WKUPEN3	WKUPEN2	WKUPEN1	WKUPEN0
0x14	SUPC_SR	31:24	RTCS	RTTS	FWKUPS	BADXTS	WKUPS	SXFME	SXFMS[1:0]	
		23:16				LPDBCS4	LPDBCS3	LPDBCS2	LPDBCS1	LPDBCS0
		15:8								LCDS
		7:0		VDD3V3SMS	VDD3V3SMIS	VDD3V3SMRSTS	CORSMRSTS	VDD3V3SMWS		TDOSCSEL
0x18 ... 0x1B	Reserved									
0x1C	SUPC_EMR	31:24								
		23:16						COREBGEN	FULLGPBRC	FLRSGPBR
		15:8								
		7:0								
0x20	SUPC_BMR	31:24	KEY[7:0]							
		23:16								
		15:8						BADXTWKEN	MRTCOUT	VBATREN
		7:0		VDD3V3SMWKEN	CORPORWKEN	FWUPEN		VBATWKEN	RTCWKEN	RTTWKEN
0x24	SUPC_WUSR	31:24	WKUPIS15	WKUPIS14	WKUPIS13	WKUPIS12	WKUPIS11	WKUPIS10	WKUPIS9	WKUPIS8
		23:16	WKUPIS7	WKUPIS6	WKUPIS5	WKUPIS4	WKUPIS3	WKUPIS2	WKUPIS1	WKUPIS0
		15:8								VDD3V3SMRSTS
		7:0			RTCS	RTTS	VDD3V3SMWS	FWUPS	BADXTWKS	WKUPS
0x28	SUPC_IER	31:24								
		23:16							VBATSMEV	VDD3V3SMEV
		15:8								
		7:0				LPDBC4	LPDBC3	LPDBC2	LPDBC1	LPDBC0
0x2C	SUPC_IDR	31:24								
		23:16							VBATSMEV	VDD3V3SMEV
		15:8								
		7:0				LPDBC4	LPDBC3	LPDBC2	LPDBC1	LPDBC0
0x30	SUPC_IMR	31:24								
		23:16							VBATSMEV	VDD3V3SMEV
		15:8								
		7:0				LPDBC4	LPDBC3	LPDBC2	LPDBC1	LPDBC0

© 2026 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x34	SUPC_ISR	31:24								
		23:16							VBATSMEV	VDD3V3SMEV
		15:8								
		7:0				LPDBC4	LPDBC3	LPDBC2	LPDBC1	LPDBC0

© 2026 Microchip Technology Inc. and its subsidiaries

15.5.1. SUPC Control Register

Name: SUPC_CR
Offset: 0x00
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode Register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
	KEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					TDXTALSEL	VROFF	SHDWEOF	SHDW
Access					W	W	W	W
Reset					–	–	–	–

Bits 31:24 – KEY[7:0] Password

Value	Name	Description
0xA5	PASSWD	Writing any other value in this field aborts the write operation.

Bit 3 – TDXTALSEL Timing Domain Clock on Slow Crystal Oscillator
This bit is located in the VDDBU domain.

Value	Name	Description
0	NO_EFFECT	No effect.
1	XTAL32K	If KEY=0xA5, TDXTALSEL switches the slow clock of the timing domain (TD_SLCK) on the 32.768 kHz crystal oscillator output.

Bit 2 – VROFF Voltage Regulator Off
This bit is located in the VDDBU domain.

Value	Name	Description
0	NO_EFFECT	No effect.
1	STOP_VREG	If KEY=0xA5, VROFF asserts the VDDCORE domain reset and stops the voltage regulator.

Bit 1 – SHDWEOF Shutdown End Of Frame

Value	Description
0	No effect.
1	If KEY=0xA5, activates the Shutdown pin once the end of frame of the LCD driver occurs.

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 0 – SHDW Shutdown

Value	Description
0	No effect.
1	If KEY=0xA5, activates the Shutdown pin.

© 2026 Microchip Technology Inc. and its subsidiaries

15.5.2. SUPC Supply Monitor Mode Register

Name: SUPC_SMMR
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode register (SYSC_WPMR).

This register is located in the VDDBU domain.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			VDD3V3SMPWRM	VDD3V3SMRSTEN		VDD3V3SMSMPL[2:0]		
Access			R/W	R/W		R/W	R/W	R/W
Reset			0	0		0	0	0

Bit	7	6	5	4	3	2	1	0
					VDD3V3SMTH[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 13 – VDD3V3SMPWRM VDD3V3 Supply Monitor Power Supply Mode

Value	Name	Description
0	MANUAL	The VDDBU power source selection is controlled by configuring the bit RSTC_MR.PWRSW.
1	AUTO_IOSM	The VDDBU power source is VBAT when a VDD3V3 under voltage is detected by VDD3V3 Supply Monitor.

Bit 12 – VDD3V3SMRSTEN VDD3V3 Supply Monitor Reset Enable

Value	Description
0	The VDDCORE reset is not asserted when a VDD3V3 supply monitor event occurs.
1	The VDDCORE reset is asserted when a VDD3V3 supply monitor event occurs.

Bits 10:8 – VDD3V3SMSMPL[2:0] VDD3V3 Supply Monitor Sampling Period

Value	Name	Description
0x0	DISABLED	VDD3V3 supply monitor is disabled
0x1	ALWAYS_ON	Continuous VDD3V3 supply monitoring
0x2	32SLCK	VDD3V3 supply monitor is enabled for 1 period every 32 MD_SLCK periods Energy optimization in Backup mode with VDD3V3 supplied.
0x3	256SLCK	VDD3V3 supply monitor is enabled for 1 period every 256 MD_SLCK periods Energy optimization in Backup mode with VDD3V3 supplied.
0x4	2048SLCK	VDD3V3 supply monitor is enabled for 1 period every 2048 MD_SLCK periods Energy optimization in Backup mode with VDD3V3 supplied.

© 2026 Microchip Technology Inc. and its subsidiaries

Bits 3:0 – VDD3V3SMTH[3:0] VDD3V3 Supply Monitor Threshold
Selects the threshold voltage of the VDD3V3 supply monitor. Refer to the section “Electrical Characteristics” for voltage values.

© 2026 Microchip Technology Inc. and its subsidiaries

15.5.3. SUPC Mode Register

Name: SUPC_MR
Offset: 0x08
Reset: 0x00005000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
	KEY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
				OSCBYPASS				
Access				R/W				
Reset				0				

Bit	15	14	13	12	11	10	9	8
		IO_BACKUP_ISO	CORSMDIS	CORSMRSTEN				
Access		R/W	R/W	R/W				
Reset		1	0	1				

Bit	7	6	5	4	3	2	1	0
	VREGDIS	CORSMM	LCDMODE[1:0]		LCDOUT[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – KEY[7:0] Password Key

Value	Name	Description
0xA5	PASSWD	Writing any other value in this field aborts the write operation.
0x59	PASSWD1	Modifies the VREGDIS bit. Other bits are not affected.

Bit 20 – OSCBYPASS Slow Crystal Oscillator Bypass
This bit is located in the VDDBU domain.

Value	Name	Description
0	NO_EFFECT	No effect. Clock selection depends on the value of the bit SUPC_CR.TDXTALSEL.
1	BYPASS	The slow crystal oscillator is bypassed if SUPC_CR.TDXTALSEL=1. The bit OSCBYPASS must be set prior to setting the bit TDXTALSEL.

Bit 14 – IO_BACKUP_ISO Backup Domain IO Isolation Control

Value	Description
0	Isolation mode is enabled.
1	Isolation mode is disabled.

Bit 13 – CORSMDIS VDDCORE Supply Monitor Disable
This bit is located in the VDDBU domain.

Value	Description
0	The VDDCORE supply monitor is enabled.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
1	The VDDCORE supply monitor is disabled.

Bit 12 – CORSMRSTEN VDDCORE Supply Monitor Reset Enable
This bit is located in the VDDBU domain.

Value	Description
0	The VDDCORE domain reset signal is not asserted when a VDDCORE supply monitor under-voltage detection occurs.
1	The VDDCORE domain reset signal is asserted when a VDDCORE supply monitor under-voltage detection occurs.

Bit 7 – VREGDIS Internal VDDCORE Voltage Regulator Disable
To write this bit, SUPC_MR.KEY must be set to 0x59.
This bit is located in the VDDBU domain.

Value	Name	Description
0	INT_VREG	Internal VDDCORE voltage regulator is enabled.
1	EXT_VREG	Internal VDDCORE voltage regulator is disabled (external power supply is used).

Bit 6 – CORSMM VDDCORE Supply Monitor Output Mode

Value	Name	Description
0	NO_EFFECT	VDDCORE supply monitor output value has no effect. Power-on is performed whatever the value of the supply monitor output.
1	VALID_VDD	VDDCORE supply monitor output value is checked to validate the VDDCORE domain power-on.

Bits 5:4 – LCDMODE[1:0] LCD Controller Mode of Operation

Value	Name	Description
0x0	LCDOFF	The internal supply source and the external supply source are both deselected (OFF Mode).
0x2	LCDON_EXTVR	The external supply source for LCD (VDDLCD) is selected (the embedded LCD voltage regulator is in High-impedance mode).
0x3	LCDON_INVR	The internal voltage regulator for VDDLCD is selected (Active mode).

Bits 3:0 – LCDOUT[3:0] LCD Voltage Regulator Output
Adjusts the output voltage of the embedded LCD Voltage Regulator. Refer to the section “Electrical Characteristics” for voltage values.

© 2026 Microchip Technology Inc. and its subsidiaries

15.5.4. SUPC Wakeup Mode Register

Name: SUPC_WUMR
Offset: 0x0C
Reset: 0x0000_000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode register (SYSC_WPMR).

This register is located in the VDDBU domain.

Bit	31	30	29	28	27	26	25	24
		LPDBC4[2:0]			LPDBC3[2:0]			LPDBC2[2]
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	LPDBC2[1:0]		LPDBC1[2:0]			LPDBC0[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
		WKUPDBC[2:0]				FWUPDBC[2:0]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bit	7	6	5	4	3	2	1	0
				LPDBCEN4	LPDBCEN3	LPDBCEN2	LPDBCEN1	LPDBCEN0
Access				R/W	R/W	R/W	R/W	R/W
Reset				_	0	0	0	0

Bits 16:18, 19:21, 22:24, 25:27, 28:30 – LPDBCx Low-power Debouncer Period of WKUPx
Refer to the section “Real-Time Controller (RTC)” to configure the RTCOUT0 clock period.

Value	Name	Description
0	DISABLE	Disables the low-power debouncers.
1	2_RTCOUT	WKUPx in active state for at least 2 RTCOUTx clock periods
2	3_RTCOUT	WKUPx in active state for at least 3 RTCOUTx clock periods
3	4_RTCOUT	WKUPx in active state for at least 4 RTCOUTx clock periods
4	5_RTCOUT	WKUPx in active state for at least 5 RTCOUTx clock periods
5	6_RTCOUT	WKUPx in active state for at least 6 RTCOUTx clock periods
6	7_RTCOUT	WKUPx in active state for at least 7 RTCOUTx clock periods
7	8_RTCOUT	WKUPx in active state for at least 8 RTCOUTx clock periods

Bits 14:12 – WKUPDBC[2:0] Wake-up Inputs Debouncer Period

Value	Name	Description
0	IMMEDIATE	Immediate, no debouncing, detected active at least on one Slow Clock edge.
1	3_SK	WKUPx shall be in its active state for at least 3 MD_SLCK periods
2	32_SK	WKUPx shall be in its active state for at least 32 MD_SLCK periods
3	512_SK	WKUPx shall be in its active state for at least 512 MD_SLCK periods
4	4096_SK	WKUPx shall be in its active state for at least 4,096 MD_SLCK periods
5	32768_SK	WKUPx shall be in its active state for at least 32,768 MD_SLCK periods

Bits 10:8 – FWUPDBC[2:0] Force Wake-up Inputs Debouncer Period

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Name	Description
0	IMMEDIATE	Immediate, no debouncing, detected active at least on one Slow Clock edge.
1	3_SK	WKUPx shall be in its active state for at least 3 MD_SLCK periods
2	32_SK	WKUPx shall be in its active state for at least 32 MD_SLCK periods
3	512_SK	WKUPx shall be in its active state for at least 512 MD_SLCK periods
4	4096_SK	WKUPx shall be in its active state for at least 4,096 MD_SLCK periods
5	32768_SK	WKUPx shall be in its active state for at least 32,768 MD_SLCK periods

Bits 0, 1, 2, 3, 4 – LPDBCENx Tamper Enable for WKUPx Pin

Value	Description
0	The WKUPx input pin is not used as a tamper pin. It can be used as a wake-up pin by writing SUPC_WUIR.WKUPENx=1.
1	The WKUPx input pin acts as a tamper pin and a tamper event forces a system wake-up.

© 2026 Microchip Technology Inc. and its subsidiaries

15.5.5. SUPC Wakeup Inputs Register

Name: SUPC_WUIR
Offset: 0x10
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode register (SYSC_WPMR).

This register is located in the VDDBU domain.

When the system is in Backup mode and VDD3V3 voltage is not supplied, only WKUP[2:0] pins can perform a wake-up.

Bit	31	30	29	28	27	26	25	24
		WKUPT14	WKUPT13	WKUPT12	WKUPT11	WKUPT10	WKUPT9	WKUPT8
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WKUPT7	WKUPT6	WKUPT5	WKUPT4	WKUPT3	WKUPT2	WKUPT1	WKUPT0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
		WKUPEN14	WKUPEN13	WKUPEN12	WKUPEN11	WKUPEN10	WKUPEN9	WKUPEN8
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	WKUPEN7	WKUPEN6	WKUPEN5	WKUPEN4	WKUPEN3	WKUPEN2	WKUPEN1	WKUPEN0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30 – WKUPTx Wake-up Input Type x

Value	Name	Description
0	LOW	A falling edge followed by a low level for a period defined by WKUPDBC on the corresponding wake-up input forces the wake-up of the core power supply.
1	HIGH	A rising edge followed by a high level for a period defined by WKUPDBC on the corresponding wake-up input forces the wake-up of the core power supply.

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14 – WKUPENx Wake-up Input Enable x

Value	Description
0	The corresponding wake-up input has no wake-up effect.
1	The corresponding wake-up input is enabled for a wake-up of the core power supply.

© 2026 Microchip Technology Inc. and its subsidiaries

15.5.6. SUPC Status Register

Name: SUPC_SR
Offset: 0x14
Reset: 0x00000000
Property: Read-only

This register is located in the VDDBU domain.

Bit	31	30	29	28	27	26	25	24
	RTCS	RTTS	FWKUPS	BADXTS	WKUPS	SXFME	SXFMS[1:0]	
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
				LPDBCS4	LPDBCS3	LPDBCS2	LPDBCS1	LPDBCS0
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
								LCDS
Access								R
Reset								0

Bit	7	6	5	4	3	2	1	0
		VDD3V3SMS	VDD3V3SMIS	VDD3V3SMRSTS	CORSMRSTS	VDD3V3SMWS		TDOSCSEL
Access		R	R	R	R	R		R
Reset		0	0	0	0	0		0

Bit 31 – RTCS RTC Wake-up Status (cleared by reading SUPC_WUSR)

Value	Description
0	No wake-up due to the assertion of the RTC alarm has occurred since the last read of SUPC_WUSR.
1	At least one wake-up due to the assertion of the RTC alarm has occurred since the last read of SUPC_WUSR.

Bit 30 – RTTS RTT Wake-up Status (cleared by reading SUPC_WUSR)

Value	Description
0	No wake-up due to the assertion of the RTT alarm has occurred since the last read of SUPC_WUSR.
1	At least one wake-up due to the assertion of the RTT alarm has occurred since the last read of SUPC_WUSR.

Bit 29 – FWKUPS FWUP Wake-up Status (cleared by reading SUPC_WUSR)

Value	Description
0	No wake-up due to the assertion of the FWUP pin has occurred since the last read of SUPC_WUSR.
1	At least one wake-up due to the assertion of the FWUP pin has occurred since the last read of SUPC_WUSR.

Bit 28 – BADXTS Slow Crystal Oscillator Wake-up Status (cleared by reading SUPC_WUSR)

Value	Description
0	No wake-up due to slow crystal oscillator failure has occurred since the last read of SUPC_WUSR.
1	At least one wake-up due to slow crystal oscillator failure has occurred since the last read of SUPC_WUSR.

Bit 27 – WKUPS WKUP Wake-up Status (cleared by reading SUPC_WUSR)

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	No wake-up due to the assertion of the WKUP pins has occurred since the last read of SUPC_WUSR.
1	At least one wake-up due to the assertion of the WKUP pins has occurred since the last read of SUPC_WUSR.

Bit 26 – SXFME Slow Crystal Oscillator Frequency Monitor Error (cleared on read)

Value	Description
0	No error detected on slow crystal oscillator frequency since the last read of SUPC_SR.
1	At least one error (SXFMS=1 or 2) detected on slow crystal oscillator frequency since the last read of SUPC_SR.

Bits 25:24 – SXFMS[1:0] Slow Crystal Oscillator Frequency Monitor Status (cleared on read)

Value	Name	Description
0	GOOD	No frequency error detected.
1	FREQ_ERROR	The frequency has not been correct over 4 consecutive monitoring periods but there are still edges detected on the slow crystal oscillator output.
2	FAIL	No edge detected in the slow crystal oscillator output for at least 2 consecutive monitoring periods.

Bits 16, 17, 18, 19, 20 – LPDBCSx Tamper Detection Wake-up Status (cleared by reading SUPC_ISR)

Value	Description
0	No wake-up due to a tamper detection on WKUPx pin has occurred since the last read of SUPC_ISR.
1	At least one system wake-up due to a tamper detection on WKUPx pin has occurred since the last read of SUPC_ISR.

Bit 8 – LCDS LCD Power Domain Status

Value	Description
0	VDDLCD voltage is not supplied.
1	VDDLCD voltage is supplied.

Bit 6 – VDD3V3SMS VDD3V3 Supply Monitor Output Status

Value	Description
0	VDD3V3 supply monitor output reports a valid voltage.
1	VDD3V3 supply monitor output reports an invalid voltage.

Bit 5 – VDD3V3SMIS VDD3V3 Supply Monitor Interrupt Status (cleared by reading SUPC_ISR)

Value	Description
0	No VDD3V3 supply monitor under voltage detection since the last read of the SUPC_ISR.
1	At least one VDD3V3 supply monitor under voltage detection since the last read of the SUPC_ISR.

Bit 4 – VDD3V3SMRSTS VDD3V3 Supply Monitor Reset Status (cleared by reading SUPC_WUSR)

Value	Description
0	No VDD3V3 supply monitor detection has generated a VDDCORE reset since the last read of the SUPC_WUSR.
1	At least one VDD3V3 supply monitor detection has generated a VDDCORE reset since the last read of the SUPC_WUSR.

Bit 3 – CORSMRSTS VDDCORE Supply Monitor Reset Status (cleared by reading SUPC_WUSR)

Value	Description
0	No VDDCORE reset generated by VDDCORE POR event has been detected since the last read of the SUPC_WUSR.
1	A VDDCORE reset has been generated by VDDCORE POR event since the last read of the SUPC_WUSR.

Bit 2 – VDD3V3SMWS VDD3V3 Supply Monitor Wake-up Status (cleared by reading SUPC_WUSR)

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	No wake-up due to a VDD3V3 supply monitor event has occurred since the last read of SUPC_WUSR.
1	At least one wake-up due to a VDD3V3 supply monitor event has occurred since the last read of SUPC_WUSR

Bit 0 – TDOSCSEL Timing Domain 32 kHz Oscillator Selection Status

Value	Name	Description
0	RC	The timing domain slow clock (TD_SLCK) source is the slow RC oscillator output.
1	XTAL	The timing domain slow clock source is the 32.768 kHz crystal oscillator output.

© 2026 Microchip Technology Inc. and its subsidiaries

15.5.7. SUPC Extended Mode Register

Name: SUPC_EMR
Offset: 0x1C
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode register (SYSC_WPMR).

This register is located in the VDDBU domain.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						COREBGEN	FULLGPBRC	FLRSGPBR
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 18 – COREBGEN VDDCORE Voltage Regulator Bandgap Enable

Value	Description
0	The bandgap of the VDDCORE internal voltage regulator is disabled.
1	The bandgap of the VDDCORE internal voltage regulator is enabled (mandatory when VDDCORE external voltage regulator is used).

Bit 17 – FULLGPBRC Full GPBR Clean

Value	Description
0	When a GPBR clear is asserted, half GPBR registers are cleared.
1	When a GPBR clear occurs, GPBR registers are all cleared.

Bit 16 – FLRSGPBR Flash Erase GPBR

Value	Description
0	When a Flash Erase occurs, there is no action on GPBR registers.
1	When a Flash Erase occurs, GPBR registers are cleared.

© 2026 Microchip Technology Inc. and its subsidiaries

15.5.8. SUPC Backup Mode Register

Name: SUPC_BMR
Offset: 0x20
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode register (SYSC_WPMR).

This register is located in the VDDBU domain.

Bit	31	30	29	28	27	26	25	24
	KEY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						BADXTWKEN	MRTCOUT	VBATREN
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
		VDD3V3SMWKEN	CORPORWKEN	FWUPEN		VBATWKEN	RTCWKEN	RTTWKEN
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bits 31:24 – KEY[7:0] Password Key

Value	Name	Description
0xA5	PASSWD	Writing any other value in this field aborts the write operation.

Bit 10 – BADXTWKEN Slow Crystal Oscillator Frequency Error Wake-up Enable

Value	Description
0	A slow crystal oscillator frequency error does not wake up the system.
1	A slow crystal oscillator frequency error wakes up the system.

Bit 9 – MRTCOUT RTCOUT0 Outputs Drive Mode

Value	Name	Description
0	USERDEF	RTCOUT0 output is driven according to the configuration of the field RTC_MR.OUT0.
1	TAMP_OPT	In Backup mode, RTCOUT0 output is stuck at 1 while there is no tamper detection and driven according to the configuration of the field RTC_MR.OUT0 when a tamper is detected on a tamper input.

Bit 8 – VBATREN Battery Voltage Event Report Enable
0 (DISABLE): Disables the report of event on VBAT voltage.
1 (ENABLE): Enables the report of event on VBAT voltage.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Name	Description
0	DISABLE	Disables the report of event on VBAT voltage.
1	ENABLE	Enables the report of event on VBAT voltage.

Bit 6 – VDD3V3SMWKEN VDD3V3 Supply Monitor Wake-up Enable

Value	Name	Description
0	DISABLE	Wake-up on VDD3V3 supply monitor under voltage detection is disabled.
1	ENABLE	Wake-up on VDD3V3 supply monitor under voltage detection is enabled.

Bit 5 – CORPORWKEN VDDCORE POR Wake-up Enable

Value	Name	Description
0	DISABLE	Wake-up on VDDCORE Power-On Reset Event is disabled.
1	ENABLE	Wake-up on VDDCORE Power-On Reset Event is enabled.

Bit 4 – FWUPEN Force Wake-up Pin Wake-up Enable

Value	Name	Description
0	DISABLE	The FWUP pin has no wake-up effect.
1	ENABLE	The FWUP pin forces the wake-up of the core power supply.

Bit 2 – VBATWKEN VBAT Supply Monitor Wake-up Enable

Value	Name	Description
0	DISABLE	Wake-up on VBAT supply monitor under voltage detection is disabled.
1	ENABLE	The RTC alarm signal forces the wake-up of the core power supply.

Bit 1 – RTCWKEN Real-time Clock Wake-up Enable

Value	Name	Description
0	DISABLE	The RTC alarm signal has no wake-up effect.
1	ENABLE	The RTC alarm signal forces the wake-up of the core power supply.

Bit 0 – RTTWKEN Real-time Timer Wake-up Enable

Value	Name	Description
0	DISABLE	The RTT alarm signal has no wake-up effect.
1	ENABLE	The RTT alarm signal forces the wake-up of the core power supply.

© 2026 Microchip Technology Inc. and its subsidiaries

15.5.9. SUPC Wakeup Status Register

Name: SUPC_WUSR
Offset: 0x24
Reset: 0x00000000
Property: Read-only

This register is located in the VDDBU domain.

Bit	31	30	29	28	27	26	25	24
	WKUPIS15	WKUPIS14	WKUPIS13	WKUPIS12	WKUPIS11	WKUPIS10	WKUPIS9	WKUPIS8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WKUPIS7	WKUPIS6	WKUPIS5	WKUPIS4	WKUPIS3	WKUPIS2	WKUPIS1	WKUPIS0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
								VDD3V3SMRSTS
Access								R
Reset								0

Bit	7	6	5	4	3	2	1	0
			RTCS	RTTS	VDD3V3SMWS	FWUPS	BADXTWKS	WKUPS
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bits 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – WKUPISx WKUPx Input Wake-up Status (cleared on read)

Value	Description
0	No wake-up due to the assertion of the WKUPx pin has occurred since the last read of SUPC_WUSR.
1	At least one wake-up due to the assertion of the WKUPx pin has occurred since the last read of SUPC_WUSR.

Bit 8 – VDD3V3SMRSTS VDD3V3 Supply Monitor Reset Status (cleared on read)

Value	Description
0	No VDDCORE reset due to VDD3V3 supply monitor event since the last read of the SUPC_WUSR.
1	VDDCORE reset due to VDD3V3 supply monitor event since the last read of the SUPC_WUSR.

Bit 5 – RTCS RTC Wake-up Status (cleared on read)

Value	Description
0	No wake-up due to the assertion of the RTC alarm has occurred since the last read of SUPC_WUSR.
1	At least one wake-up due to the assertion of the RTC alarm has occurred since the last read of SUPC_WUSR.

Bit 4 – RTTS RTT Wake-up Status (cleared on read)

Value	Description
0	No wake-up due to the assertion of the RTT alarm has occurred since the last read of SUPC_WUSR.
1	At least one wake-up due to the assertion of the RTT alarm has occurred since the last read of SUPC_WUSR.

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 3 – VDD3V3SMWS VDD3V3 Supply Monitor Wake-up Status (cleared on read)
The VDD3V3 supply monitor event occurs if SUPC_SMMR.VDD3V3SMSMPL > 0.

Value	Description
0	No wake-up due to VDD3V3 supply monitor event has occurred since the last read of SUPC_WUSR.
1	At least one wake-up due to VDD3V3 supply monitor event has occurred since the last read of SUPC_WUSR.

Bit 2 – FWUPS FWUP Wake-up Status (cleared on read)

Value	Description
0	No wake-up due to the assertion of the FWUP pin has occurred since the last read of SUPC_WUSR.
1	At least one wake-up due to the assertion of the FWUP pin has occurred since the last read of SUPC_WUSR.

Bit 1 – BADXTWKS Slow Crystal Oscillator Frequency Error Wake-up Status (cleared on read)

Value	Description
0	No wake-up due to slow crystal oscillator frequency error has occurred since the last read of SUPC_WUSR.
1	At least one wake-up due to slow crystal oscillator frequency error has occurred since the last read of SUPC_WUSR.

Bit 0 – WKUPS WKUP Wake-up Status (cleared on read)
The flags WKUPISx report the input(s) that triggered the wake-up.
In case the wake-up is issued from tamper detection circuitry (SUPC_WUMR.LPDBCENx=1), the SUPC_SR.LPDBCSx flags report the input(s) that triggered the wake-up.

Value	Description
0	No wake-up due to the assertion of the WKUP pins has occurred since the last read of SUPC_WUSR.
1	At least one wake-up due to the assertion of the WKUP pins has occurred since the last read of SUPC_WUSR.

© 2026 Microchip Technology Inc. and its subsidiaries

15.5.10. SUPC Interrupt Enable Register

Name: SUPC_IER
Offset: 0x28
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the System Controller Write Protection Mode register (SYSC_WPMR).

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							VBATSMEV	VDD3V3SMEV
Access							W	W
Reset							–	–

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				LPDBC4	LPDBC3	LPDBC2	LPDBC1	LPDBC0
Access				W	W	W	W	W
Reset				–	–	–	–	–

Bit 17 – VBATSMEV VBAT Supply Monitor Event Interrupt Enable
The interrupt on VBAT voltage event is enabled if SUPC_BMR.VBATREN=1.

Bit 16 – VDD3V3SMEV VDD3V3 Supply Monitor Event Interrupt Enable
The VDD3V3 supply monitor event occurs if SUPC_SMMR.VDD3V3SMSMPL > 0.

Bits 0, 1, 2, 3, 4 – LPDBCx WKUPx Pin Tamper Detection Interrupt Enable

© 2026 Microchip Technology Inc. and its subsidiaries

15.5.11. SUPC Interrupt Disable Register

Name: SUPC_IDR
Offset: 0x2C
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the System Controller Write Protection Mode Register (SYSC_WPMR).

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							VBATSMEV	VDD3V3SMEV
Access							W	W
Reset							–	–

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				LPDBC4	LPDBC3	LPDBC2	LPDBC1	LPDBC0
Access				W	W	W	W	W
Reset				–	–	–	–	–

Bit 17 – VBATSMEV VBAT Supply Monitor Event Interrupt Disable

Bit 16 – VDD3V3SMEV VDD3V3 Supply Monitor Event Interrupt Disable

Bits 0, 1, 2, 3, 4 – LPDBCx WKUPx Pin Tamper Detection Interrupt Disable

© 2026 Microchip Technology Inc. and its subsidiaries

15.5.12. SUPC Interrupt Mask Register

Name: SUPC_IMR
Offset: 0x30
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							VBATSMEV	VDD3V3SMEV
Access							R	R
Reset							0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				LPDBC4	LPDBC3	LPDBC2	LPDBC1	LPDBC0
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit 17 – VBATSMEV VBAT Supply Monitor Event Interrupt Mask

Bit 16 – VDD3V3SMEV VDD3V3 Supply Monitor Event Interrupt Mask

Bits 0, 1, 2, 3, 4 – LPDBCx WKUPx Pin Tamper Detection Interrupt Mask

© 2026 Microchip Technology Inc. and its subsidiaries

15.5.13. SUPC Interrupt Status Register

Name: SUPC_ISR
Offset: 0x34
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							VBATSMEV	VDD3V3SMEV
Access							R	R
Reset							0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				LPDBC4	LPDBC3	LPDBC2	LPDBC1	LPDBC0
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit 17 – VBATSMEV VBAT Supply Monitor Event Interrupt Status (cleared on read)
The VBAT voltage event is reported if SUPC_BMR.VBATREN=1.

Value	Description
0	No VBAT voltage drop down event has occurred since the last read of SUPC_ISR.
1	At least one VBAT voltage drop down event has occurred since the last read of SUPC_ISR.

Bit 16 – VDD3V3SMEV VDD3V3 Supply Monitor Event Interrupt Status (cleared on read)
The VDD3V3 supply monitor event occurs if SUPC_SMMR.IOSMSMPL > 0.

Value	Description
0	No VDD3V3 supply monitor event has occurred since the last read of SUPC_ISR.
1	At least one VDD3V3 supply monitor event has occurred since the last read of SUPC_ISR.

Bits 0, 1, 2, 3, 4 – LPDBCx WKUPx Pin Tamper Detection Interrupt Status (cleared on read)

Value	Description
0	No tamper detection event has occurred on WKUPx since the last read of SUPC_ISR.
1	At least one tamper detection event has occurred since the last read of SUPC_ISR.

© 2026 Microchip Technology Inc. and its subsidiaries

16. Reset Controller (RSTC)

16.1. Description

The Reset Controller (RSTC) handles all the resets of the system and is driven by power-on reset cells (POR), supply monitor cells (SM), software, the external reset pin, and peripheral events. The RTSC reports the type of reset.

The RSTC drives independently or simultaneously the external reset and the peripheral and processor resets.

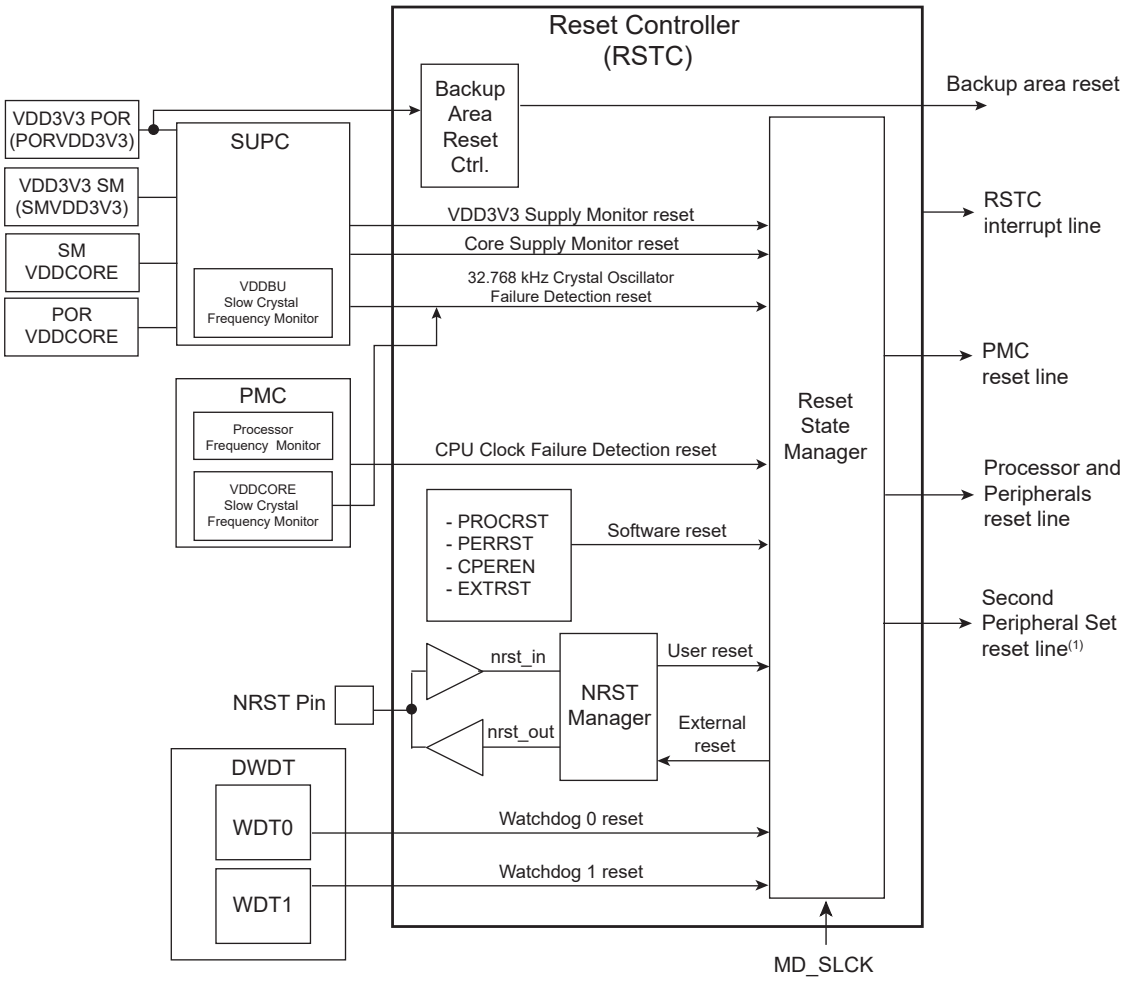
16.2. Embedded Characteristics

- Driven by Embedded Power-On Reset, Supply Monitor, Software, External Reset Pin and Peripheral Events
- Management of All Internal Resets Lines in the System
- Reset Source Report:
- External Reset Line Control
- Backup Area Voltage Selection

© 2026 Microchip Technology Inc. and its subsidiaries

16.3. Block Diagram

Figure 16-1. RSTC Block Diagram



Note (1): Peripherals driven by clocks MCK1 and MCK1DIV.
(2): See the table “Peripheral Identifiers” for the main system bus clock driving each peripheral.

16.4. Functional Description

16.4.1. Overview

The RSTC comprises an NRST pin manager and a Reset State manager. The RSTC clock is MD_SLCK (monitoring slow clock) generated by the always-on slow RC oscillator. The RSTC generates the following reset signals:

- Processor and peripherals reset line (also resets the Watchdog Timers)
- Power Management Controller (PMC) reset line
- NRST pin

These reset signals are controlled by the RSTC, either on events generated by peripherals, events on NRST pin, watchdog fault or overflow, or on software action. The Reset State manager controls the generation of reset signals and provides a signal to the NRST manager when an assertion of the NRST pin is required.

© 2026 Microchip Technology Inc. and its subsidiaries

The NRST manager asserts the NRST pin during a programmable time, thus controlling external device resets.

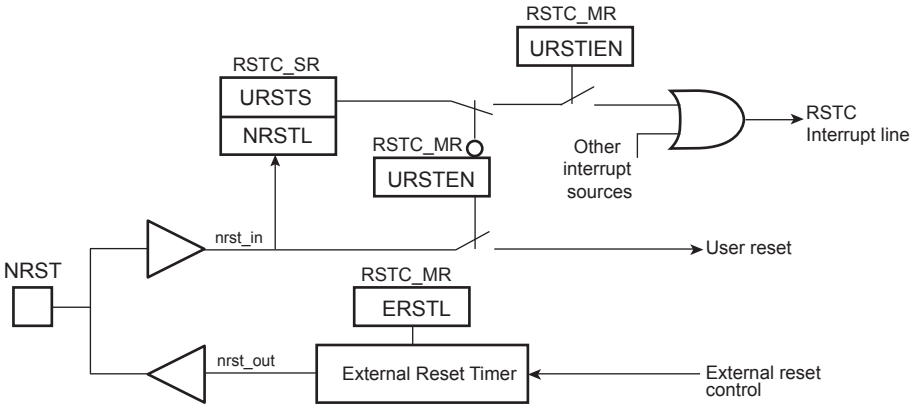
The Mode register (RSTC_MR), used to configure the RSTC, is powered by VDDBU.

The RSTC can reset separately the processor and its peripherals.

16.4.2. NRST Manager

The NRST manager samples the NRST input pin and drives this pin low when required by the Reset State manager. The figure below shows the block diagram of the NRST manager.

Figure 16-2. NRST Manager



16.4.2.1. NRST Signal or Interrupt

The NRST manager handles the NRST input line asynchronously if RSTC_MR.URSTASYNC =1. When the NRST input is low, a User reset is immediately reported to the Reset State manager and the internal reset signals are asserted even if there is a clock failure on MD_SLCK (safe reset).

The NRST manager handles the NRST input line synchronously if RSTC_MR.URSTASYNC=0. When the line is low, it is first resynchronized on slow clock before it is reported to the Reset State manager. In both cases, when the NRST goes from low to high, the internal reset is synchronized with the monitoring slow clock to provide a safe internal de-assertion of reset (if enabled).

If RSTC_MR.URSTEN=0, the assertion of the NRST input pin does not trigger a VDDCORE domain reset.

The level of the pin NRST is reported in NRSTL of the Status register (RSTC_SR).

As soon as the pin NRST is asserted (low level), RSTC_SR.URSTS=1. This bit is cleared on read.

If RSTC_MR.URSTIEN=1, the assertion of the NRST pin triggers an interrupt and not a VDDCORE reset.

16.4.2.2. NRST External Reset Control

The RSTC can be configured to assert the external reset line (NRST). The NRST pin is driven low for a time programmed by RSTC_MR.ERSTL. This assertion duration lasts $2^{(ERSTL+1)}$ MD_SLCK cycles. This assertion duration time in range 60 µs to 2 seconds. If ERSTL=0 , a two slow clock period duration is generated on the NRST pin.

This feature allows the NRST line to be compliant with any external devices connected on the system reset (i.e., when external devices require a longer start-up time than the processor system).

If WDTx_MR.WDNRST_DIS = 0, the NRST pin is asserted in case of a Watchdog reset.

If WDTx_MR.WDNRST_DIS = 1, the NRST pin is not asserted in case of a Watchdog reset.

© 2026 Microchip Technology Inc. and its subsidiaries

16.4.2.3. VDDBU Power Switch Control

The Backup section power supply VDDBU is powered by either VBAT or VDD3V3. The voltage can be selected by configuring RSTC_MR.PWRSW. At first power-up (General reset), the backup section is powered by VDD3V3. If RSTC_MR.PWRSW is written to 1, the backup section is powered by VBAT if the voltage on VBAT is above the VBAT Power-On Reset threshold; otherwise, a General reset is performed.

The VDDBU voltage can be automatically switched from VDD3V3 to VBAT in case a VDD3V3 failure is detected by the VDD3V3 supply monitor by setting SUPC_SMMR.IOSMPWRM to 1. Refer to the section “Supply Controller (SUPC)” for details.

16.4.2.4. Second Peripheral Set Reset Control

The reset of the second peripheral set is managed by RSTC_MR.CPEREN. When this bit is set to ‘1’, the reset is de-asserted. When RSTC_MR.CPEREN=0, the reset is asserted.

Note: The second peripheral set comprises peripherals driven by MCK1 and MCK1DIV. See the table “Peripheral Identifiers” for the main system bus clock driving each peripheral.

16.4.3. Reset States

The Reset State manager handles the different reset sources and generates the internal reset signals. It reports the reset status in the field RSTTYP of the Status register (RSTC_SR). RSTC_SR.RSTTYP is updated when the processor reset is released.

If more than one reset event occurred since the last read of RST_SR, the field RSTTYP reports the first reset that occurred.

16.4.3.1. General Reset

A General reset occurs when the VDD3V3 voltage is below the threshold of the VDD3V3 POR and the VBAT voltage is below the VBAT POR threshold (first power-up).

In Active mode, a General reset occurs if VBAT is selected as the voltage source of VDDBU and is below the VBAT POR threshold (erroneous voltage selection resulting in a loss of data in the backup domain).

The NRST line rises two cycles after the VDDCORE reset line, as ERSTL defaults at value 0x0.

The PMC reset is de-asserted once the NRST pin has been sampled as inactive.

The processor and peripheral reset lines are de-asserted once the Flash power-up sequence has ended.

Once all the reset signals are released, RSTC_SR.RSTTYP reports a General reset.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 16-3. General Reset Timing Diagram

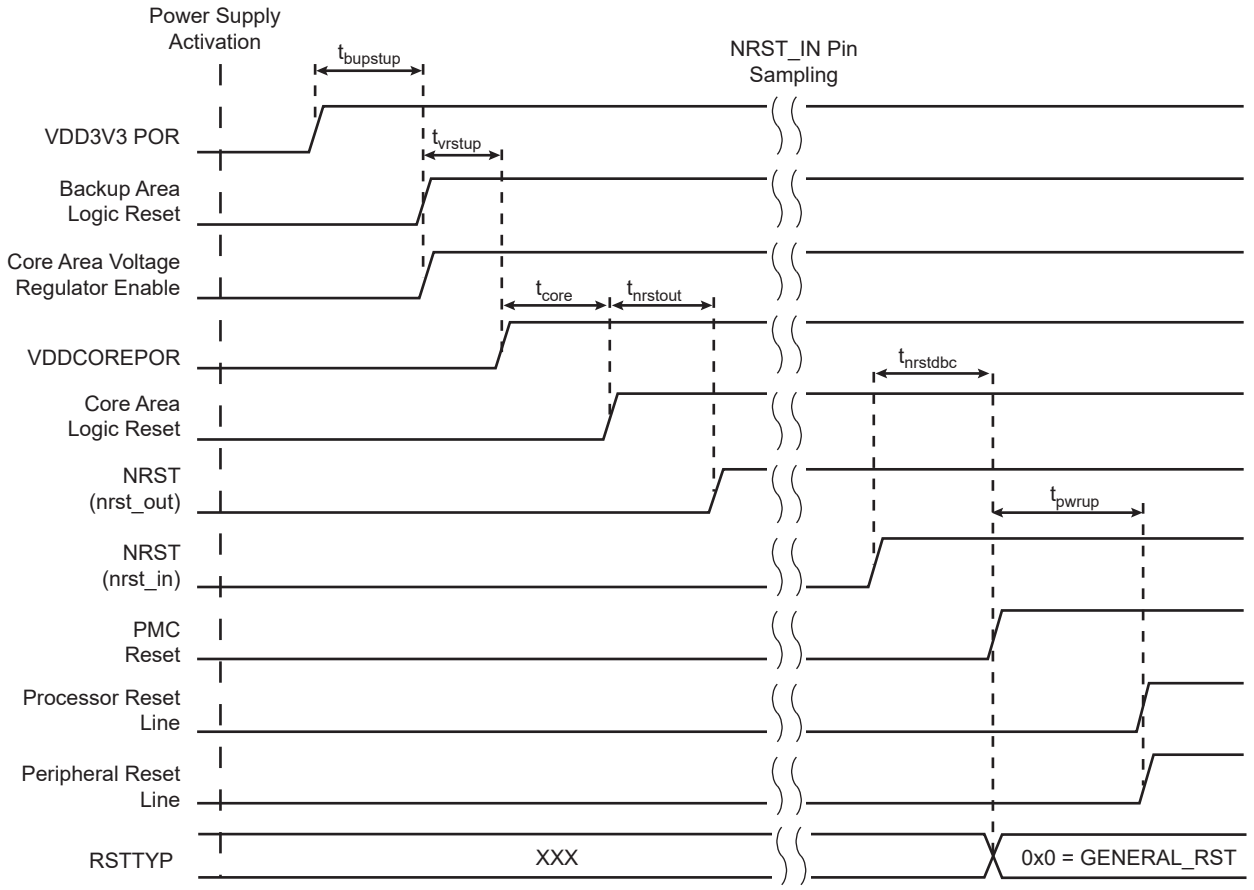


Table 16-1. General Reset Timings

Name	Value	Unit	Description
$t_{bupstup}$	16	MD_SLCK	Backup start-up time. VDD3V3 POR rising to backup area logic reset release
t_{vrstup}	8	MD_SLCK	Voltage regulator start-up time. Voltage regulator power on to VDDCORE POR assertion
t_{core}	7	MD_SLCK	VDDCORE POR to core area logic reset
$t_{nrstout}$	5	MD_SLCK	Core area logic reset to NRST pin release
$t_{nrstdbc}$	2	MD_SLCK	NRST pin debouncing time
t_{pwrup}	5	ms	Flash power-up time

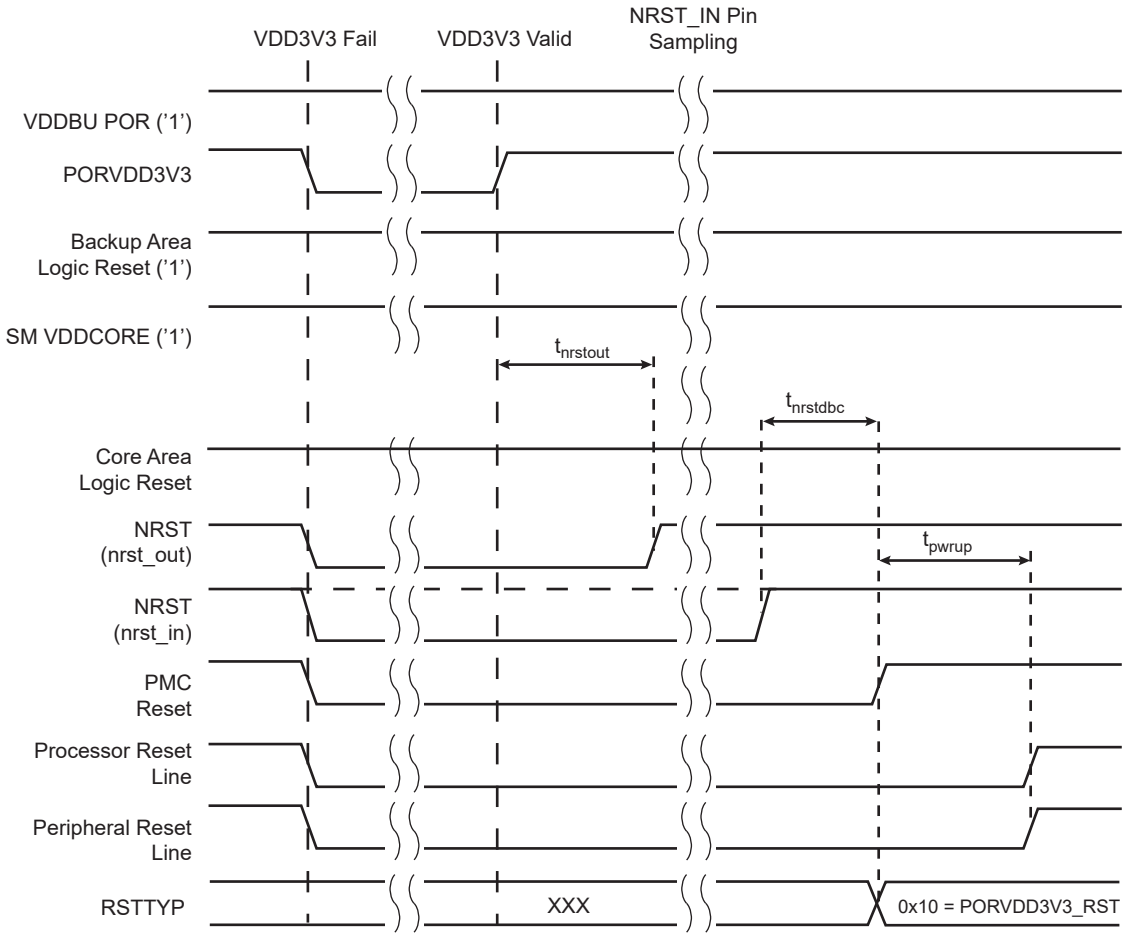
16.4.3.2. VDD3V3 POR Reset

The system embeds a VDD3V3 Power-On Reset (POR) which generates a reset of the VDDCORE domain.

When this reset occurs, the whole VDDCORE domain is reset except for the RSTC, which reports the reset type as a PORVDD3V3 reset.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 16-4. VDD3V3 POR Reset Timing Diagram



Note: Timings are defined in General Reset Timings.

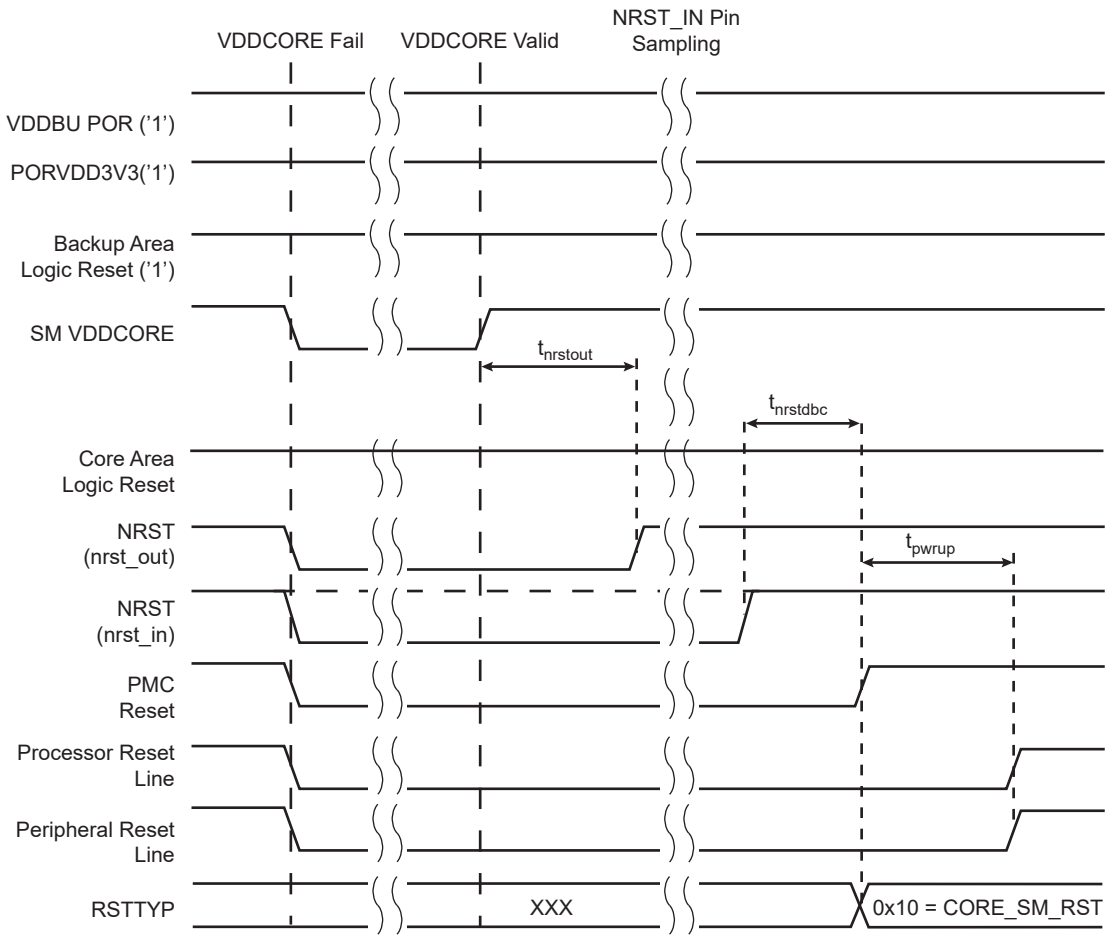
16.4.3.3. VDDCORE Supply Monitor Reset

The system embeds a VDDCORE Supply Monitor (SM VDDCORE) which generates a reset of the VDDCORE domain.

When this reset occurs, the VDDCORE domain is reset except for the RSTC, which reports the reset type as a CORE_SM_RST reset.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 16-5. VDDCORE Supply Monitor Reset Timing Diagram



Note: Timings are defined in General Reset Timings.

16.4.3.4. VDDCORE Reset

A VDDCORE reset occurs when VDDCORE is switched off.

The following actions result in a VDDCORE reset:

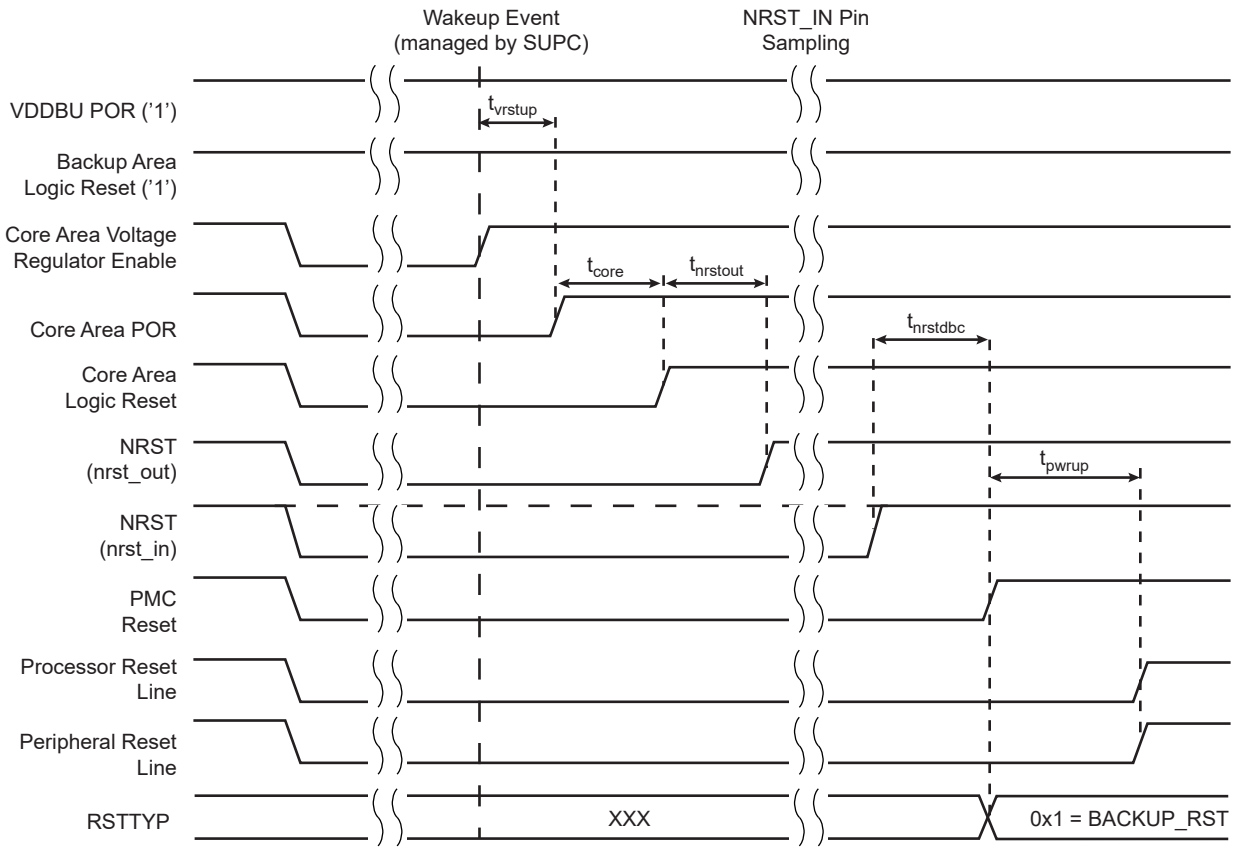
- switching off the internal VDDCORE voltage regulator by writing SUPC_CR.VROFF to '1'
- asserting the SHDN pin by writing SUPC_CR.SHDW to '1' to switch off the external VDDCORE voltage regulator.

While exiting Backup mode, the VDDCORE reset signal is de-asserted by the Supply Controller.

RSTC_SR.RSTTYP is updated to report a backup reset.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 16-6. VDDCORE Reset Timing Diagram



Note: Timings are defined in General Reset Timings.

16.4.3.5. 32.768 kHz Crystal Oscillator Failure Detection Reset

The system embeds two slow crystal frequency monitors: one is implemented in the Power Management Controller (PMC) and located in the VDDCORE power domain. The second frequency monitor is located in the SUPC powered by VDDBU. The monitor embedded in the PMC can be enabled by setting CKGR_MOR.XT32KFME to '1'. The monitor located in the backup area is always enabled and monitors the slow crystal frequency even if the system is in Backup mode.

The slow crystal oscillator failure detection reset is done when one of the two frequency monitors detects a failure and RSTC_MR.BADXTRST=1. This reset signal lasts three slow clock cycles.

When RSTC_MR.BADXTRST=0, the 32.768 kHz crystal oscillator fault has no impact on the RSTC.

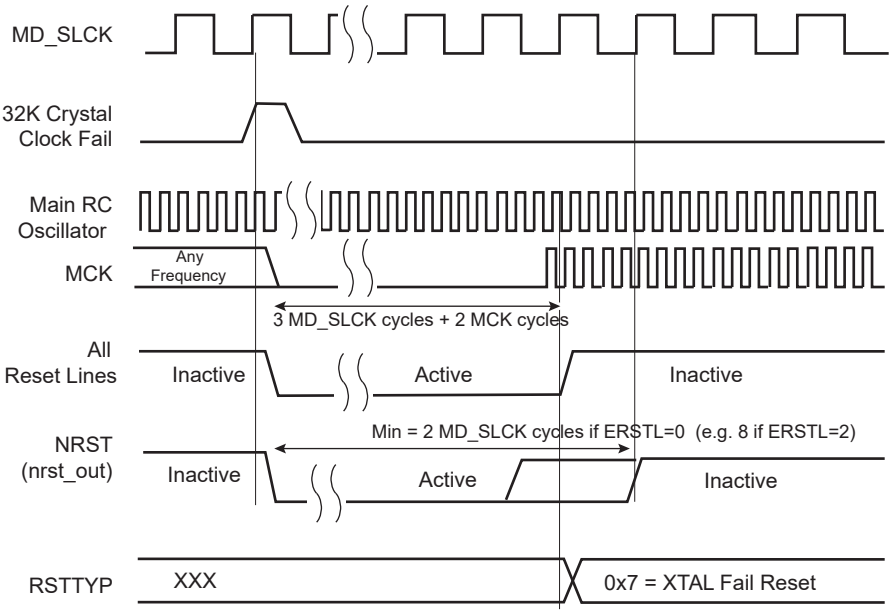
When RSTC_MR.SCKSW=1 and RSTC_MR.BADXTRST=0, the 32.768 kHz crystal oscillator fault leads to an automatic TD_SLCK source switching from slow crystal oscillator to slow RC oscillator.

If the slow crystal oscillator failure detection is enabled to perform a system reset, the processor reset and the peripheral reset are asserted. The NRST line is also asserted, depending on the value of RSTC_MR.ERSTL.

When the slow crystal oscillator failure generates a VDDCORE reset, PMC_SR.XT32KERR is automatically cleared by the peripheral and core reset.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 16-7. 32.768 kHz Crystal Oscillator Failure Detection Reset Timing Diagram



16.4.3.6. CPU Clock Failure Detection Reset

The system embeds a CPU clock frequency monitor that is located in the PMC. It can be enabled by setting CKGR_MOR.BMCKRST.

The CPU Clock Failure Detection reset is done when the CPU frequency monitor detects a failure and RSTC_MR.CPUFEN=1. This reset lasts three MD_SLCK cycles.

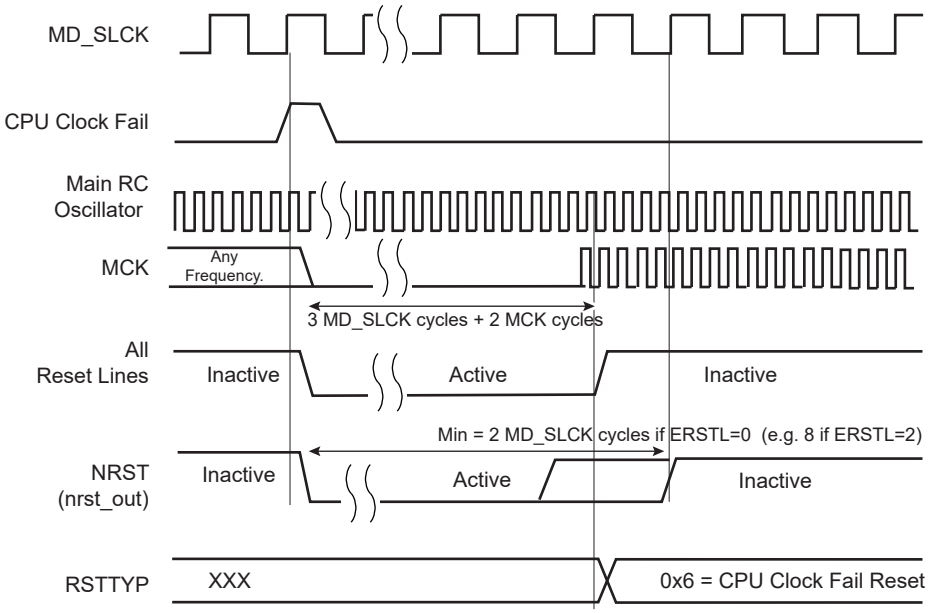
When RSTC_MR.CPUFEN=0, the slow crystal oscillator fault has no impact on the RSTC.

During a CPU Clock Failure Detection reset, the processor reset and the peripheral reset are asserted. The NRST line is also asserted, depending on the value of RSTC_MR.ERSTL.

When the CPU clock failure generates a VDDCORE reset, PMC_SR.MCKMON is automatically cleared by the peripheral and core reset.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 16-8. CPU Clock Failure Detection Reset Timing Diagram



16.4.3.7. Watchdog Reset

A Watchdog reset is entered when a Watchdog 0 or Watchdog 1 fault occurs and if the corresponding watchdog is configured to generate a reset (WDTx_MR.RPTHRST or WDTx_MR.PERIODRST). This reset lasts three MD_SLCK cycles.

A Watchdog reset can reset the circuit fully or partially:

- The PMC controller is reset if RSTC_MR.WDTPMCx=1. If the PMC is not reset, the clock configuration is maintained after the Watchdog reset.
- The NRST pin is asserted if WDTx_MR.WDNRSTDIS=0. The duration of the NRST pin assertion depends on the RSTC_MR.ERSTL value. However, the resulting low level on NRST does not result in a User reset.

Note: The second peripheral set is not reset by a Watchdog reset.

The watchdog timer is always reset after a Watchdog reset, and the watchdog is enabled by default and with a period set to a maximum.

When WDTx_MR.WDRSTEN=0, the Watchdog fault has no impact on the RSTC.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 16-9. Watchdog Reset Timing Diagram RSTC_MR.WDTPMCx=1

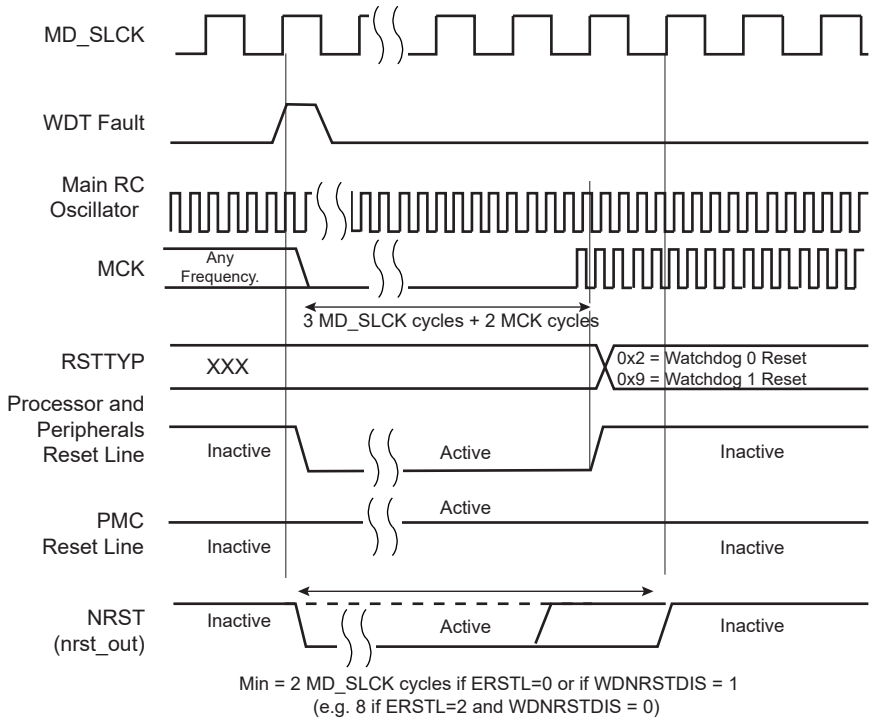
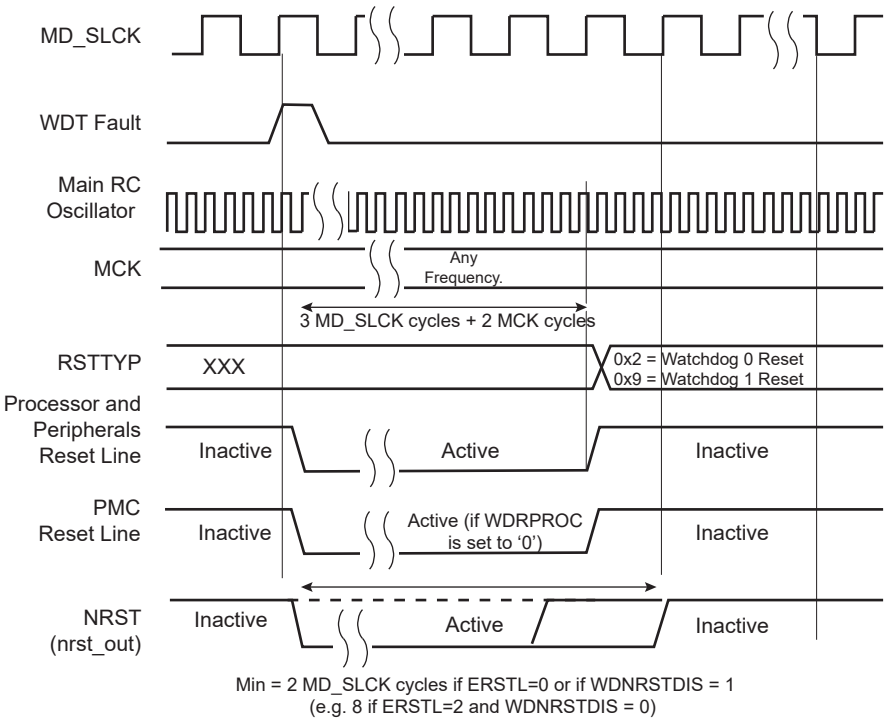


Figure 16-10. Watchdog Reset Timing Diagram RSTC_MR.WDTPMCx=0



16.4.3.8. Software Reset

The RSTC offers commands to assert the different reset signals. These commands are performed by writing the Control register (RSTC_CR) with the following bits at '1':

© 2026 Microchip Technology Inc. and its subsidiaries

- RSTC_CR.PROCRST and RSTC_CR.PERRST—Writing a '1' to PROCRST and PERRST resets the processor, its peripherals and the Watchdog Timer, whereas the second peripheral set is not reset, including the memory system. If RSTC_MR.SFTPMCRS=1, the PMC is reset. If RSTC_MR.SFTPMCRS=0, the PMC is not reset. PERRST must always be used in conjunction with PROCRST (PERRST and PROCRST set both at 1 simultaneously).
- RSTC_CR.EXTRST: Writing a '1' to EXTRST asserts low the NRST pin during a time defined by the field RSTC_MR.ERSTL.

The reset of the second peripheral set is managed by RSTC_MR.CPEREN.

The Software reset is entered if at least one of these bits is written to '1' by the software. All these commands can be performed independently or simultaneously. The Software reset lasts three MD_SLCK cycles.

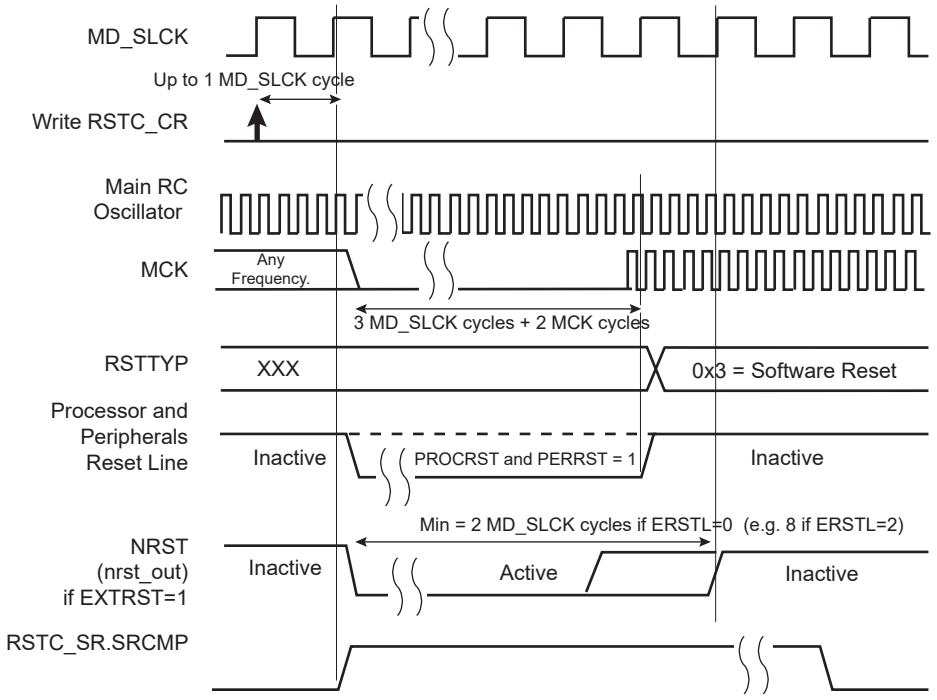
The internal reset signals are asserted as soon as the register write is performed. This is detected on the Main System Bus Clock (MCK). They are released when the Software reset has ended, i.e., synchronously to MD_SLCK.

If EXTRST=1, the NRST is driven low depending on the configuration of RSTC_MR.ERSTL. However, the assertion of NRST pin does not lead to a User reset.

If RSTC_CR.PROCRST and RSTC_CR.PERRST=1, the RSTC reports the software status in RSTC_SR.RSTTYP. Other Software resets are not reported in RSTTYP.

As soon as a software operation is detected, RSTC_SR.SRCMP=1. SRCMP is cleared at the end of the Software reset. No other Software reset can be performed while SRCMP=1, and writing any value in RSTC_CR has no effect.

Figure 16-11. Software Reset Timing Diagram



16.4.3.9. User Reset

The User reset is entered when a low level is detected on the NRST pin and RSTC_MR.URSTEN=1. If URSTASYNC=1, a falling edge of the NRST input signal immediately asserts internal reset lines. If

© 2026 Microchip Technology Inc. and its subsidiaries

URSTASYNC=0, the NRST input signal is resynchronized and internal reset lines are asserted once a falling edge has been detected on the resynchronized NRST input signal.

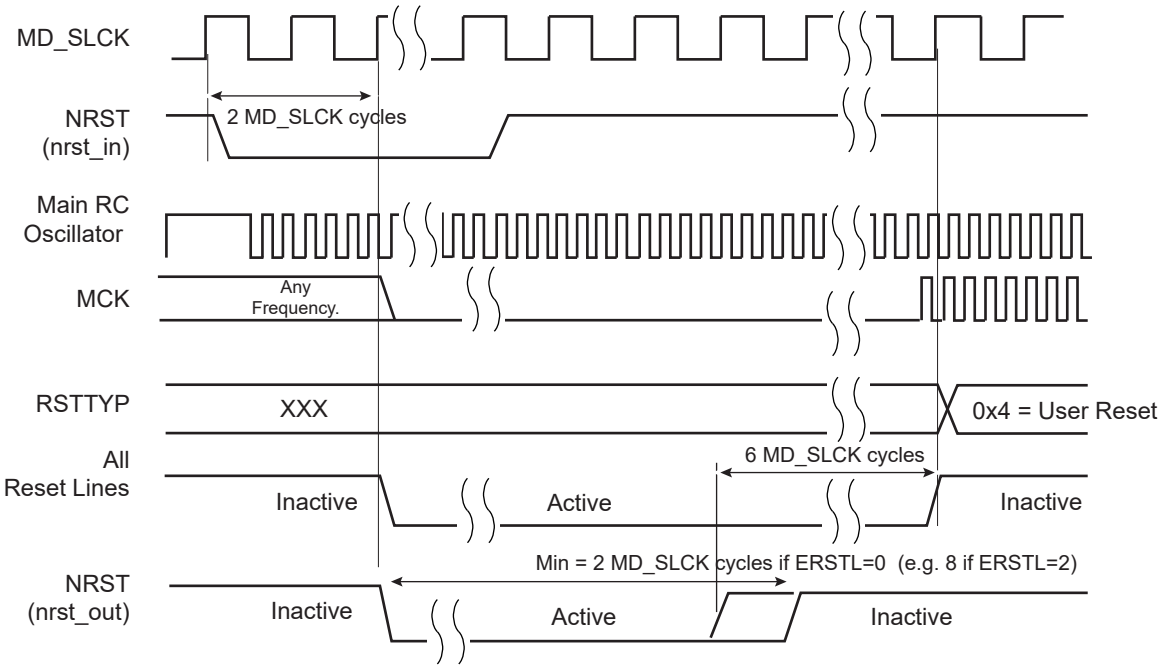
In case of a User reset, the processor and all peripheral resets are asserted.

The User reset is released when NRST rises, after a two-cycle resynchronization time and a two-cycle processor start-up. The processor clock is re-enabled as soon as NRST is confirmed high.

When the processor reset signal is released, RSTC_SR.RSTTYP is loaded with the value 0x4, indicating a User reset.

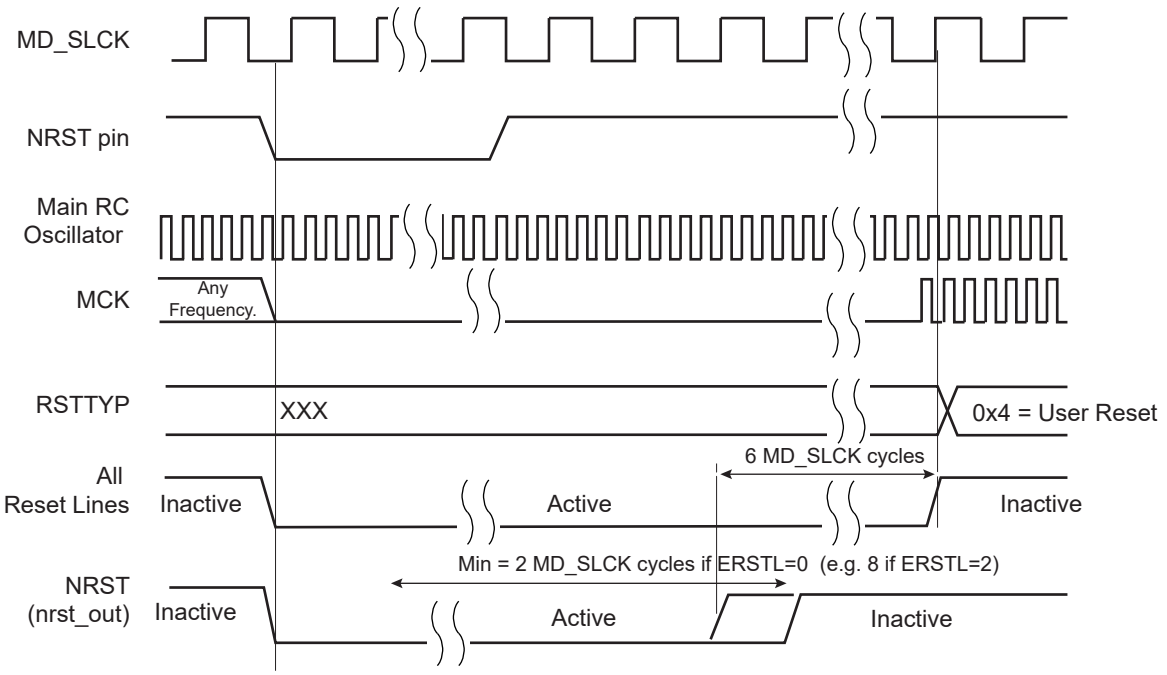
The NRST manager ensures that the NRST line is asserted for a number of slow clock cycles configured in RSTC_MR.ERSTL. However, if NRST pin does not rise during the configured period, because it is driven low externally, the internal reset lines remain asserted until NRST rises.

Figure 16-12. User Reset State (URSTASYNC = '0')



© 2026 Microchip Technology Inc. and its subsidiaries

Figure 16-13. User Reset State (URSTASYNC = '1')



16.4.4. Reset State Priorities

The Reset State manager manages the priorities among the different reset sources. The resets are listed in order of priority as follows:

1. General reset
2. VDD3V3 Supply Monitor (SMVDD3V3) or VDD3V3 POR (PORVDD3V3) reset
3. VDDCORE Supply Monitor Reset
4. VDDCORE POR reset (backup reset)
5. User reset if RSTC_MR.URSTASYNC is set to 1
6. 32.768 kHz Crystal Oscillator Failure Detection reset
7. CPU Clock Failure Detection reset
8. Watchdog 0 or 1 reset
9. Software reset
10. User reset if RSTC_MR.URSTASYNC is set to 0

Specific cases are listed below:

- When in Watchdog reset:
  - The processor reset is active and so a Software reset cannot be programmed.
  - If RSTC_MR.URSTASYNC is set to 1 and a User reset occurs, it has a higher priority and a User reset is performed.
- When in Software reset:
  - A watchdog event has priority over the current state.
  - The NRST pin has no effect.
- When in User reset:
  - A watchdog event is impossible because the watchdog timer is being reset.
  - A Software reset is impossible, since the processor reset is being asserted.

© 2026 Microchip Technology Inc. and its subsidiaries

16.4.5. Managing Resets at Application Level

If a reset source is enabled, it acts on the systems as described in the table below. See the corresponding peripheral sections for further details on resets.

- VDD3V3 Supply Monitor reset – Full system reset.
- VDDCORE Supply Monitor reset – Full system reset.
- 32.768 kHz Crystal Oscillator Failure Detection reset – Full system reset except the second peripheral set.
- CPU Clock Failure Detection reset – Full system reset.
- Watchdog 0 reset – Full system reset except the second peripheral set. The reset of the PMC can be configured with RSTC_MR.WDTPMC0. The second peripheral set must be managed by RSTC_MR.CPEREN.
- Watchdog 1 reset – Full system reset except the second peripheral set. The reset of the PMC can be configured with RSTC_MR.WDTPMC1. The second peripheral set must be managed by RSTC_MR.CPEREN.
- Software reset – Full system reset except the second peripheral set. The reset of the PMC can be configured with RSTC_MR.SFTPMCRS. The second peripheral set must be managed by RSTC_MR.CPEREN
- User reset (NRST pin) – Full system reset. To avoid this, the User reset must be configured to generate an interrupt and not a reset (RSTC_MR.URSTEN = 0 and RSTC_MR.URSTIEN = 1).

© 2026 Microchip Technology Inc. and its subsidiaries

16.5. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	RSTC_CR	31:24	KEY[7:0]							
		23:16								
		15:8								
		7:0					EXTRST	PERRST		PROCRST
0x04	RSTC_SR	31:24								
		23:16							SRCMP	NRSTL
		15:8					RSTTYP[3:0]			
		7:0							CORESMS	URSTS
0x08	RSTC_MR	31:24	KEY[7:0]							
		23:16		BADXTRST	PWRSW				CPEREN	CORSMIEN
		15:8					ERSTL[3:0]			
		7:0	WDTPMC1	WDTPMC0	SFTPMCRS	URSTIEN	CPUFEN	URSTASYNC	SCKSW	URSTEN

© 2026 Microchip Technology Inc. and its subsidiaries

16.5.1. RSTC Control Register

Name: RSTC_CR
Offset: 0x00
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	KEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					EXTRST	PERRST		PROCRST
Access					W	W		W
Reset					–	–		–

Bits 31:24 – KEY[7:0] System Reset Key

Value	Name	Description
0xA5	PASSWD	Writing any other value in this field aborts the write operation.

Bit 3 – EXTRST External Reset

Value	Description
0	No effect.
1	If KEY = 0xA5, asserts the reset on NRST pin.

Bit 2 – PERRST Peripheral Reset

Value	Description
0	No effect.
1	If KEY = 0xA5, resets the peripherals. Must be used in conjunction with PROCRST.

Bit 0 – PROCRST Processor Reset

Value	Description
0	No effect.
1	If KEY = 0xA5, resets the processor.

© 2026 Microchip Technology Inc. and its subsidiaries

16.5.2. RSTC Status Register

Name: RSTC_SR
Offset: 0x04
Reset: 0x00000000
Property: Read-only

The reset value given here assumes that a General reset has been performed, subject to change if other types of reset are generated.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access							R	R
Reset							0	0

Bit	15	14	13	12	11	10	9	8
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
Access							R	R
Reset							0	0

Bit 17 – SRCMP Software Reset Command in Progress
When set, this bit indicates that a Software reset command is in progress and that no further Software reset should be performed until the end of the current one. This bit is automatically cleared at the end of the current Software reset.

Value	Description
0	No Software reset command is being performed by the RSTC. The RSTC is ready for a Software reset command.
1	A Software reset command is being performed by the RSTC.

Bit 16 – NRSTL NRST Pin Level
Reports NRST pin level after sampling on MCK clock.

Bits 11:8 – RSTTYP[3:0] Reset Type
Reports the cause of the last processor reset. Reading RSTC_SR does not reset this field.
Values not listed below must be considered 'reserved'.

Value	Name	Description
0	GENERAL_RST	First power-up reset, Core and VDD3V3 Supply Monitor if not a PORVDD3V3 reset
1	BACKUP_RST	VDDCORE reset. Wake-up from Backup mode.
2	WDT0_RST	Watchdog 0 fault occurred
3	SOFT_RST	Processor reset required by the software
4	USER_RST	NRST pin detected low
5	CORE_SM_RST	Core Supply Monitor reset
6	CPU_FAIL_RST	CPU clock failure detection occurred

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Name	Description
7	SLCK_XTAL_RST	32.768 kHz crystal failure detection fault occurred
9	WDT1_RST	Watchdog 1 fault occurred
10	PORVDD3V3_RST	VDD3V3 (PORVDD3V3) reset occurred

Bit 1 – CORESMS VDDCORE Supply Monitor Reset Flag Status (cleared on read)

Value	Description
0	No VDDCORE reset occurred since the last read of RSTC_SR.
1	VDDCORE reset occurred since the last read of RSTC_SR.

Bit 0 – URSTS User Reset Status (cleared on read)

Set when a high-to-low transition of the NRST pin (reset assertion) occurs. This transition is also detected on the MCK rising edge. If the User reset is disabled (URSTEN = 0 in RSTC_MR) and if the interrupt is enabled by RSTC_MR.URSTIEN, URSTS triggers an interrupt. Reading the RSTC_SR resets URSTS and clears the interrupt.

Value	Description
0	No high-to-low edge on NRST pin happened since the last read of RSTC_SR.
1	At least one high-to-low transition of NRST pin has been detected since the last read of RSTC_SR.

© 2026 Microchip Technology Inc. and its subsidiaries

16.5.3. RSTC Mode Register

Name: RSTC_MR
Offset: 0x08
Reset: 0x000000F5
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode Register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
	KEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
		BADXTRST	PWRSW				CPEREN	CORSMIEN
Access		R/W	R/W				R/W	R/W
Reset		0	0				0	0

Bit	15	14	13	12	11	10	9	8
					ERSTL[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	WDTPMC1	WDTPMC0	SFTPMCRS	URSTIEN	CPUFEN	URSTASYNC	SCKSW	URSTEN
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	0	1	0	1

Bits 31:24 – KEY[7:0] Write Access Password

Value	Name	Description
0xA5	PASSWD	Writing any other value in this field aborts the write operation. Always reads as 0.

Bit 22 – BADXTRST Bad XTAL Fail Reset

Value	Description
0	The detection of a 32.768 kHz crystal failure has no effect.
1	The detection of a 32.768 kHz crystal failure resets the logic supplied by VDDCORE.

Bit 21 – PWRSW Backup Area Power Switch Control

Value	Description
0	VDDBU is supplied by VDD3V3.
1	VDDBU is supplied by VBAT.

Bit 17 – CPEREN Second Peripheral Set Enable

Value	Description
0	If KEY = 0xA5, resets the second peripheral set.
1	If KEY = 0xA5, deasserts the reset of the second peripheral set.

Bit 16 – CORSMIEN VDDCORE Supply Monitor Interrupt Enable

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	Disables VDDCORE supply monitor event interrupt.
1	Enables VDDCORE supply monitor event interrupt.

Bits 11:8 – ERSTL[3:0] External Reset Length

This field defines the external reset length. The external reset is asserted during a time of $2^{(ERSTL+1)}$ MD_SLCK cycles. This allows assertion duration to be programmed between 60 µs and 2 seconds. Note that synchronization cycles must also be considered when calculating the actual reset length as previously described.

Bit 7 – WDTPMC1 WDT1 PMC Reset

Value	Description
0	In case of a WDT1 reset, the PMC is not reset.
1	In case of a WDT1 reset, the PMC is reset.

Bit 6 – WDTPMC0 WDT0 PMC Reset

Value	Description
0	In case of a WDT0 reset, the PMC is not reset.
1	In case of a WDT0 reset, the PMC is reset.

Bit 5 – SFTPMCRS Software PMC Reset

Value	Description
0	In case of a Software reset, the PMC is not reset.
1	In case of a Software reset, the PMC is reset.

Bit 4 – URSTIEN User Reset Interrupt Enable

Value	Description
0	If RSTC_SR.USRTS =1 , no effect on the RSTC interrupt line.
1	If RSTC_SR.USRTS =1, asserts the RSTC interrupt line if URSTEN = 0.

Bit 3 – CPUFEN CPU Fail Enable

Value	Description
0	The detection of a CPU clock failure has no effect.
1	The detection of a CPU clock failure resets the logic supplied by VDDCORE.

Bit 2 – URSTASYNC User Reset Asynchronous Control

Value	Description
0	NRST input signal is managed synchronously.
1	NRST input signal is managed asynchronously.

Bit 1 – SCKSW Slow Clock Switching

Value	Description
0	The detection of a 32.768 kHz crystal failure has no effect.
1	The detection of a 32.768 kHz crystal failure automatically switches the TD_SLCK clock source to the slow RC oscillator.

Bit 0 – URSTEN User Reset Enable

Value	Description
0	The detection of a low level on the NRST pin does not trigger a User reset.
1	The detection of a low level on the NRST pin triggers a User reset.

© 2026 Microchip Technology Inc. and its subsidiaries

17. System Controller Write Protection (SYSCWP)

17.1. Functional Description

17.1.1. System Controller Peripheral Mapping

Table 17-1. System Controller Peripheral Mapping

Offset	System Controller Peripheral	Name
0x0000-0x002C	Watchdog 1 of Dual Watchdog Controller (DWDT)	DWDT.WDT1
0x1000-0x100C	Reset Controller	RSTC
0x1020-0x103C	Real-time Timer	RTT
0x1050-0x105C	Reserved	–
0x1060-0x10FC	General Purpose Backup Registers	GPBR
0x1100-0x119C	Real Time Clock	RTC
0x11A0	Write Protection Mode Register	SYSC_WPMR
0x11A4	Write Protection Status Register	SYSC_WPSR
0x11D0-0x120C	Supply Controller	SUPC
0x1210-0x123C	Watchdog 0 of Dual Watchdog Controller (DWDT)	DWDT.WDT0

17.1.2. Register Write Protection

To prevent any single software error from modifying the configuration of Reset Controller (RSTC), Supply Controller (SUPC), Real-time Timer (RTT), General Purpose Backup Register (GPBR), Real-time Clock (RTC) and Dual Watchdog Timer (DWDT), some registers of these peripherals can be write-protected by setting the WPEN and/or WPITEN bits in the System Controller Write Protection Mode register (SYSC_WPMR).

Note: The DWDT embeds additional write protection mechanisms.

When write protection is enabled, any attempt to write these registers is reported in the System Controller Write Protection Status register (SYSC_WPSR).

The following registers can be write-protected when SYSC_WPMR.WPEN=1:

- WDT 0 Control Register
- WDT 0 Mode Register
- WDT 0 Window Level Register
- WDT 0 Window Interrupt Register
- RSTC Mode Register
- RTC Control Register
- RTC Mode Register
- RTC Time Alarm Register
- RTC Calendar Alarm Register
- RTC Tamper Control Register
- RTT Mode Register
- RTT Alarm Register
- RTT Modulo Selection Register
- GPBR Full Clear Register (Write protection only. No violation report.)
- GPBR Registers

© 2026 Microchip Technology Inc. and its subsidiaries

- SUPC Control Register
- SUPC Control Register
- SUPC Supply Monitor Mode Register
- SUPC Mode Register
- SUPC Wakeup Mode Register
- SUPC Wakeup Input Register
- SUPC Extended mode Register
- SUPC Backup Mode Register
- WDT 1 Control Register
- WDT 1 Mode Register
- WDT 1 Window Level Register
- WDT 1 Window Interrupt Register

The following registers can be write-protected when SYSC_WPMR.WPITEN=1:

- WDT 0 Interrupt Enable Register
- WDT 0 Interrupt Disable Register
- RTC Interrupt Enable Register
- RTC Interrupt Disable Register
- SUPC Interrupt Enable Register
- SUPC Interrupt Disable Register
- WDT 1 Interrupt Enable Register
- WDT 1 Interrupt Disable Register

© 2026 Microchip Technology Inc. and its subsidiaries

17.2. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	SYSC_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0							WPITEN	WPEN
0x04	SYSC_WPSR	31:24								
		23:16								
		15:8	WVSRC[7:0]							
		7:0								WPVS

© 2026 Microchip Technology Inc. and its subsidiaries

17.2.1. SYSC Write Protection Mode Register

Name: SYSC_WPMR
Offset: 0x00
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
							WPITEN	WPEN
Access							R/W	R/W
Reset							0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x535943	PASSWD	Writing any other value in this field aborts the write operation of the WPEN and WPITEN bits. Always reads as 0.

Bit 1 – WPITEN Write Protection RTC Interrupt Enable

Value	Description
0	Disables the write protection of the interrupt enable/disable registers if WPKEY corresponds to 0x535943 (“SYC” in ASCII).
1	Enables the write protection of the interrupt enable/disable registers if WPKEY corresponds to 0x535943 (“SYC” in ASCII).

Bit 0 – WPEN Write Protection Enable

Value	Description
0	Disables the write protection of the configuration registers if WPKEY corresponds to 0x535943 (“SYC” in ASCII).
1	Enables the write protection of the configuration registers if WPKEY corresponds to 0x535943 (“SYC” in ASCII).

© 2026 Microchip Technology Inc. and its subsidiaries

17.2.2. SYSC Write Protection Status Register

Name: SYSC_WPSR
Offset: 0x04
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	WVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPVS
Access								R
Reset								0

Bits 15:8 – WVSRC[7:0] Write Violation Source
When bit WPVS is equal to 1, the field WVSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protection Register Violation Status

Value	Description
0	No write protection violation has occurred since the last read of SYSC_WPSR.
1	A write protection violation has occurred since the last read of SYSC_WPSR. The associated violation is reported into field WVSRC.

© 2026 Microchip Technology Inc. and its subsidiaries

18. Dual Watchdog Timer (DWDT)

18.1. Description

The Dual Watchdog Timer (DWDT) is used to prevent system lock-up if the software becomes trapped in a deadlock. The DWDT embeds two independent watchdogs (WDT0 and WDT1) and each watchdog provides a monitoring period of up to 16 seconds.

One watchdog may be used to monitor critical software task execution time in an OS/RTOS by generating an interrupt only. The second watchdog may be used to generate a system reset if the software is trapped in a deadlock.

WDT0 and WDT1 clocks are driven by monitoring the slow clock (MD_SLCK).

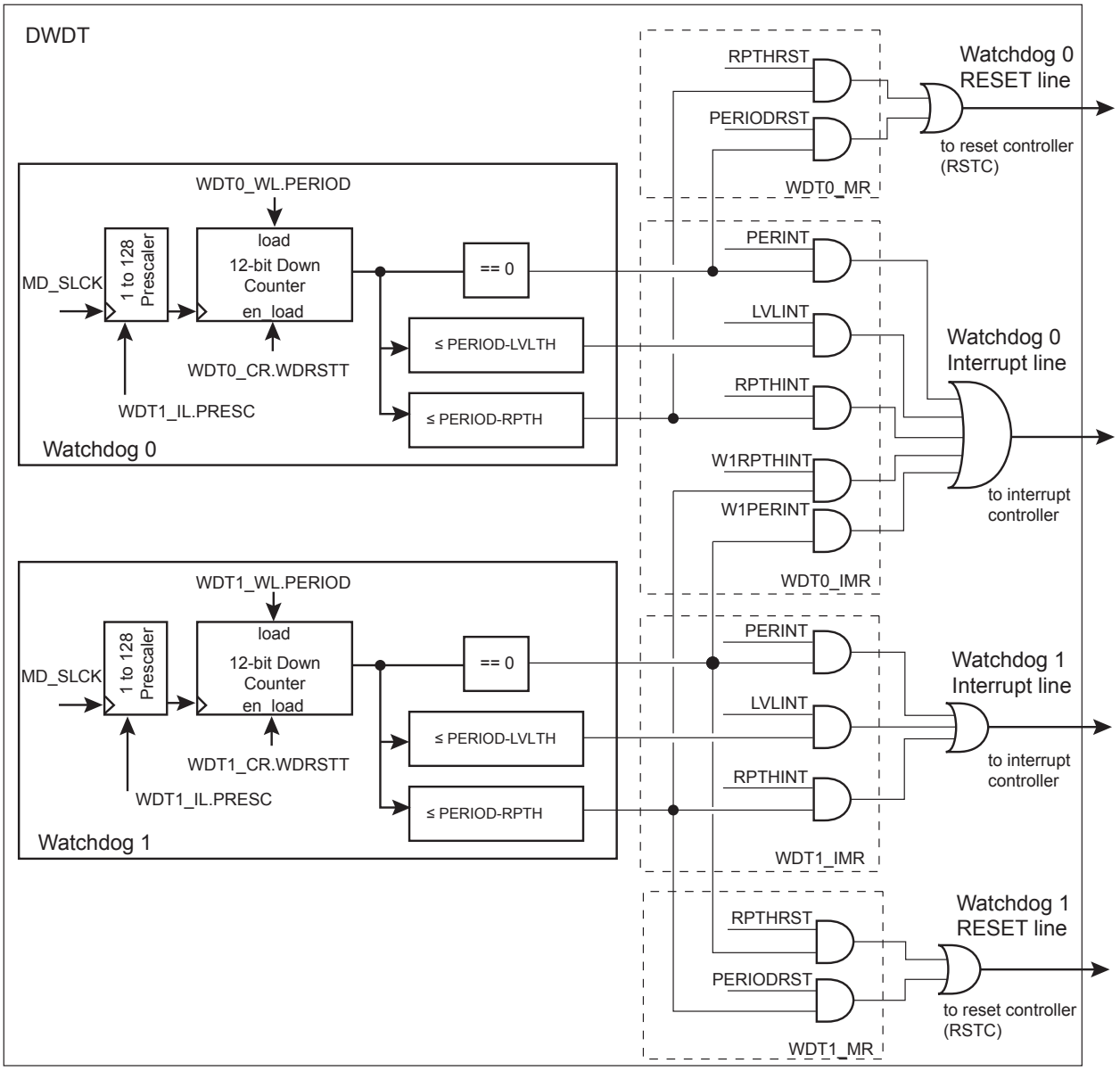
18.2. Embedded Characteristics

- 12-bit Key-Protected Programmable Counter
- Watchdog Clocks are Independent from Processor Clock
- Provides Reset or Interrupt Signals to the System
- Counters May be Stopped while the Core is in Debug State or in Sleep Mode
- Generates a General Reset

© 2026 Microchip Technology Inc. and its subsidiaries

18.3. Block Diagram

Figure 18-1. DWDT Block Diagram



18.4. Functional Description

18.4.1. Configuration

The DWDT is supplied with VDDCORE.

Each watchdog monitoring period (period corresponding to an overflow of the watchdog counter) is independent and can be configured by writing the field PERIOD in the Window Level register (WDT1_WL or WDT0_WL).

The WDT0 clock period is defined by MD_SLCK divided by $2^{(7\text{-WDT0\_IL.PRESC})}$.

The WDT1 clock period is defined by MD_SLCK divided by $2^{(7\text{-WDT1\_IL.PRESC})}$.

For each watchdog, the following parameters can be defined:

© 2026 Microchip Technology Inc. and its subsidiaries

- PRESC (Prescaler value)–Defines the clock of the 12-bit down counter. The watchdog counter is decreased by 1 each time the prescaler reaches the value defined by $2^{(7-WDTx\_IL.PRESC)}$.
- PERIOD (Watchdog Monitoring Period)–Value loaded each time a watchdog reset command is asserted. Once the down counter reaches 0, a watchdog event is generated. This event leads to either a reset (if WDT0_MR.PERIODRST=1 or WDT1_MR.PERIODRST=1) or an interrupt (if WDT0_IMR.PERINT=1 or WDT1_IMR.PERINT=1).
- RPTH (Repeat Threshold)–A watchdog restart done before the repeat threshold is elapsed leads to a repeat violation. A repeat violation leads to either a reset (if WDT0_MR.RPTHRST=1 or WDT1_MR.RPTHRST=1) or an interrupt (if WDT0_IMR.RPTHINT=1 or WDT1_IMR.RPTHINT=1).
- LVLTH (Interrupt Threshold)–Threshold after which an interrupt is generated (if WDT0_IMR.LVLINT=1 or WDT1_IMR.LVLINT=1).

After a processor reset, the value of PERIOD is 0xFFF and the value of PRESC is 0x000, corresponding to the maximum value of the counter with the external reset generation enabled (PERIODRST at 1 after a backup reset). This means that watchdogs are running at reset, that is, at power-up. The user can either disable the WDT by setting WDT0_MR.WDDIS=1 and/or WDT1_MR.WDDIS=1 or reprogram the WDTs to meet the maximum watchdog period the application requires.

The WDT1 and the WDT0 embed securities to avoid programming out of range values. The following inequality must be verified, otherwise the configuration is canceled:

$$\mathrm{RPTH} \leq \mathrm{LVLTH} < \mathrm{PERIOD}$$

In addition, the WDT0 has the possibility to control the range of operation of the WDT1. It can limit the period, the repeat threshold and the interrupt level of the WDT1 by programming WDT1_LVLLIM, WDT1_RLIM and WDT1_PLIM.

18.4.2. Watchdog Reload

In normal operation, the user reloads the watchdog at regular intervals before the down counter reaches 0, by configuring WDT0_CR.WDRSTT=1 or WDT1_CR.WDRSTT=1 with the correct KEY field. The watchdog counter is then immediately reloaded with the PERIOD value and restarted, and the MD_SLCK PRESC divider is reset and restarted. Writing WDT0_CR or WDT1_CR with an incorrect key raises an error flag.

Writing WDT0_MR, WDT0_WLR or WDT0_ILR immediately reloads the counter from PERIOD and restarts the MD_SLCK PRESC divider of WDT0.

Writing WDT1_MR, WDT1_WLR or WDT1_ILR immediately reloads the counter from PERIOD and restarts the MD_SLCK PRESC divider of WDT1.

18.4.3. Watchdog Lock

WDT0_MR and WDT1_MR can be written until a LOCKMR command is issued in the corresponding WDT0_CR or WDT1_CR. Once locked, only a processor reset resets WDT0_MR and WDT1_MR. As long as a WDT is not locked, writing WDT1_MR or WDT0_MR automatically reloads the corresponding Watchdog timer with the newly programmed mode parameters.

If the watchdog is restarted by writing into the corresponding Control register (WDT0_CR or WDT1_CR), the corresponding WDT0_MR or WDT1_MR must not be programmed during a period of time of three MD_SLCK periods following the WDT0_CR or WDT1_CR write access.

18.4.4. Repeat Threshold

A repeat threshold can be defined for each watchdog in order to protect against deadlocks that would repeatedly restart the watchdog. WDT0_WL.RPTH and WDT1_WL.RPTH define the minimum number of cycles to wait after a watchdog restart before the watchdog can be started again.
If a watchdog restart occurs before this limit is reached, a repeat threshold failure is asserted and WDT0_ISR.RPTHINT and/or WDT1_ISR.RPTHINT are/is set to 1. This feature can be disabled by programming a null RPTH value. In such a configuration, restarting the Watchdog Timer is permitted

© 2026 Microchip Technology Inc. and its subsidiaries

in the whole range [0; PERIOD] and does not generate an error. This is the default configuration on reset (RPTH is null).

18.4.5. Watchdog Reset Order

If the down counter of any watchdog reaches 0 or if one of the watchdogs is restarted before reaching the RPTH threshold limit, a Watchdog Reset Order is sent to the Reset Controller (RSTC) if WDT0_MR.PERIODRST=1 and/or WDT0_MR.RPTHRST=1 or if WDT1_MR.PERIODRST=1 and/or WDT1_MR.RPTHRST=1.

18.4.6. Watchdog Interrupt

Each watchdog drives a separate interrupt line.

Watchdog 0 interrupt line can be asserted on Watchdog 1 events (see WDT0_IER).

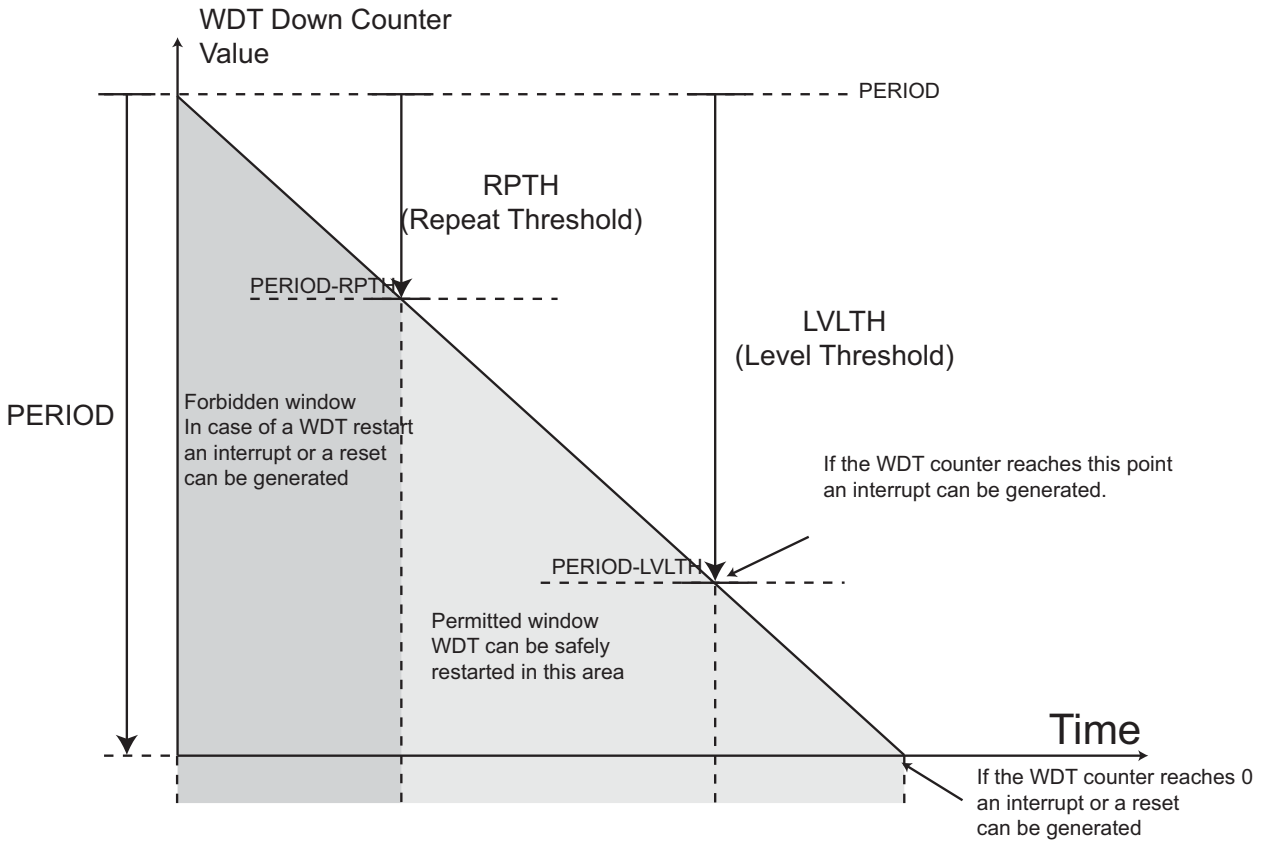
18.4.7. Watchdog Halt

The counters may be stopped depending on the value programmed for the bits:

- WDTx_MR.WDIDLEHLT: If high, the corresponding watchdog counter is stopped while the Core is in Sleep mode.
- WDT0_MR.WDDBGxHLT, WDT1_MR.WDDBGxHLT: If high, the corresponding watchdog counter is stopped while the Core is in Debug state.

18.4.8. Timing Diagrams

Figure 18-2. Timing Diagram



© 2026 Microchip Technology Inc. and its subsidiaries

18.5. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	WDT1_CR	31:24	KEY[7:0]							
		23:16								
		15:8								
		7:0				LOCKMR				WDRSTT
0x04	WDT1_MR	31:24			WDDBG0HLT	WDIDLEHLT				
		23:16								
		15:8				WDDIS			WDNRSTDIS	
		7:0			RPTHRST	PERIODRST				
0x08	WDT1_VR	31:24								
		23:16								
		15:8					COUNTER[11:8]			
		7:0	COUNTER[7:0]							
0x0C	WDT1_WL	31:24					RPTH[11:8]			
		23:16	RPTH[7:0]							
		15:8					PERIOD[11:8]			
		7:0	PERIOD[7:0]							
0x10	WDT1_IL	31:24								
		23:16						PRESC[2:0]		
		15:8					LVLTH[11:8]			
		7:0	LVLTH[7:0]							
0x14	WDT1_IER	31:24								
		23:16								
		15:8								
		7:0			RLDERR			LVLINT	RPTHINT	PERINT
0x18	WDT1_IDR	31:24								
		23:16								
		15:8								
		7:0			RLDERR			LVLINT	RPTHINT	PERINT
0x1C	WDT1_ISR	31:24								
		23:16								
		15:8								
		7:0			RLDERR			LVLINT	RPTHINT	PERINT
0x20	WDT1_IMR	31:24								
		23:16								
		15:8								
		7:0			RLDERR			LVLINT	RPTHINT	PERINT
0x24 ... 0x120F	Reserved									
0x1210	WDT0_CR	31:24	KEY[7:0]							
		23:16								
		15:8								
		7:0				LOCKMR				WDRSTT
0x1214	WDT0_MR	31:24			WDDBG0HLT	WDIDLEHLT				
		23:16								
		15:8				WDDIS			WDNRSTDIS	
		7:0			RPTHRST	PERIODRST				
0x1218	WDT0_VR	31:24								
		23:16								
		15:8					COUNTER[11:8]			
		7:0	COUNTER[7:0]							
0x121C	WDT0_WL	31:24					RPTH[11:8]			
		23:16	RPTH[7:0]							
		15:8					PERIOD[11:8]			
		7:0	PERIOD[7:0]							
0x1220	WDT0_IL	31:24								
		23:16						PRESC[2:0]		
		15:8					LVLTH[11:8]			
		7:0	LVLTH[7:0]							

© 2026 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x1224	WDT0_IER	31:24								
		23:16								
		15:8								
		7:0			RLDERR	W1RPTHINT	W1PERINT	LVLINT	RPTHINT	PERINT
0x1228	WDT0_IDR	31:24								
		23:16								
		15:8								
		7:0			RLDERR	W1RPTHINT	W1PERINT	LVLINT	RPTHINT	PERINT
0x122C	WDT0_ISR	31:24								
		23:16								
		15:8								
		7:0			RLDERR	W1RPTHINT	W1PERINT	LVLINT	RPTHINT	PERINT
0x1230	WDT0_IMR	31:24								
		23:16								
		15:8								
		7:0			RLDERR	W1RPTHINT	W1PERINT	LVLINT	RPTHINT	PERINT
0x1234	WDT1_LVLLIM	31:24					LVLMAX[11:8]			
		23:16	LVLMAX[7:0]							
		15:8					LVLMIN[11:8]			
		7:0	LVLMIN[7:0]							
0x1238	WDT1_RLIM	31:24					RPTHMAX[11:8]			
		23:16	RPTHMAX[7:0]							
		15:8					RPTHMIN[11:8]			
		7:0	RPTHMIN[7:0]							
0x123C	WDT1_PLIM	31:24					PERMAX[11:8]			
		23:16	PERMAX[7:0]							
		15:8					PERMIN[11:8]			
		7:0	PERMIN[7:0]							

© 2026 Microchip Technology Inc. and its subsidiaries

18.5.1. DWDT Watchdog 1 Control Register

Name: WDT1_CR
Offset: 0x0000
Reset: –
Property: Write-only

The WDT1_CR register values must not be modified within three MD_SLCK periods following a restart of the watchdog performed by a write access in WDT1_CR. Any modification causes the watchdog to trigger an end of period earlier than expected.

Bit	31	30	29	28	27	26	25	24
	KEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				LOCKMR				WDRSTT
Access				W				W
Reset				–				–

Bits 31:24 – KEY[7:0] Password

Value	Name	Description
0xA5	PASSWD	Writing any other value in this field aborts the write operation.

Bit 4 – LOCKMR Lock Mode Register Write Access

Value	Description
0	No effect.
1	Locks the configuration registers if KEY is written to 0xA5. Write accesses to WDT1_MR, WDT1_WL and WDT1_IL have no effect.

Bit 0 – WDRSTT Watchdog Restart

Value	Description
0	No effect.
1	Restarts the watchdog if KEY is written to 0xA5.

© 2026 Microchip Technology Inc. and its subsidiaries

18.5.2. DWDT Watchdog 1 Mode Register

Name: WDT1_MR
Offset: 0x0004
Reset: 0x00000030
Property: Read/Write

Write access to this register has no effect if the LOCKMR command is issued in WDT1_CR (unlocked on hardware reset).

The WDT1_MR register values must not be modified within three MD_SLCK periods following a restart of the watchdog performed by a write access in WDT1_CR. Any modification will cause the watchdog to trigger an end of period earlier than expected.

Bit	31	30	29	28	27	26	25	24
			WDDBG0HLT	WDIDLEHLT				
Access			R/W	R/W				
Reset			0	0				

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				WDDIS			WDNRSTDIS	
Access				R/W			R/W	
Reset				0			0	

Bit	7	6	5	4	3	2	1	0
			RPTHRST	PERIODRST				
Access			R/W	R/W				
Reset			1	1				

Bit 29 – WDDBG0HLT Watchdog Core Debug Halt

Value	Description
0	The Watchdog 1 runs when the processor is in Debug state.
1	The Watchdog 1 stops when the processor is in Debug state.

Bit 28 – WDIDLEHLT Watchdog Idle Halt

Value	Description
0	The watchdog runs when the system is in Idle state.
1	The watchdog stops when the system is in Idle state.

Bit 12 – WDDIS Watchdog Disable

Value	Description
0	Enables the Watchdog Timer.
1	Disables the Watchdog Timer.

Bit 9 – WDNRSTDIS Watchdog Reset NRST Pin Disable

Value	Description
0	A watchdog reset asserts the NRST pin.
1	A watchdog reset does not assert the NRST pin.

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 5 – RPTHRST Repeat Threshold Reset Enable

Value	Description
0	No reset is generated if the watchdog is restarted before the RPTH threshold (early restart).
1	A reset is generated if the watchdog is restarted before the RPTH threshold.

Bit 4 – PERIODRST Watchdog Overflow Period Reset Enable

Value	Description
0	No reset is generated if the watchdog reaches 0.
1	A reset is generated once the watchdog reaches 0.

© 2026 Microchip Technology Inc. and its subsidiaries

18.5.3. DWDT Watchdog 1 Value Register

Name: WDT1_VR
Offset: 0x0008
Reset: 0x00000FFF
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					COUNTER[11:8]			
Access					R	R	R	R
Reset					1	1	1	1

Bit	7	6	5	4	3	2	1	0
	COUNTER[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	1	1	1	1	1	1	1	1

Bits 11:0 – COUNTER[11:0] Watchdog Down Counter Value
Current value of the watchdog down counter.
Due to the asynchronous operation of the watchdog, it is necessary to read this register twice at the same value to get a valid read. Therefore, a minimum of two and a maximum of three accesses are required.

© 2026 Microchip Technology Inc. and its subsidiaries

18.5.4. DWDT Watchdog 1 Window Level Register

Name: WDT1_WL
Offset: 0x000C
Reset: 0x00000FFF
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
					RPTH[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RPTH[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
					PERIOD[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					1	1	1	1

Bit	7	6	5	4	3	2	1	0
	PERIOD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 27:16 – RPTH[11:0] Repeat Threshold
Defines the period before which a watchdog restart generates a Watchdog 1 and/or a Watchdog 0 interrupt.

Bits 11:0 – PERIOD[11:0] Watchdog Period
Defines the period after which the watchdog generates a Watchdog 1 and/or a Watchdog 0 interrupt.

© 2026 Microchip Technology Inc. and its subsidiaries

18.5.5. DWDT Watchdog 1 Interrupt Level Register

Name: WDT1_IL
Offset: 0x0010
Reset: 0x00000FFF
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						PRESC[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
					LVLTH[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					1	1	1	1

Bit	7	6	5	4	3	2	1	0
	LVLTH[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 18:16 – PRESC[2:0] Prescaler Ratio
Defines the Watchdog 1 Prescaler ratio.

Value	Name	Description
0x0	RATIO128	The watchdog counter decreased when the prescaler reaches 128.
0x1	RATIO64	The watchdog counter decreased when the prescaler reaches 64.
0x2	RATIO32	The watchdog counter decreased when the prescaler reaches 32.
0x3	RATIO16	The watchdog counter decreased when the prescaler reaches 16.
0x4	RATIO8	The watchdog counter decreased when the prescaler reaches 8.
0x5	RATIO4	The watchdog counter decreased when the prescaler reaches 4.
0x6	RATIO2	The watchdog counter decreased when the prescaler reaches 2.
0x7	RATIO1	The watchdog counter decreased when the prescaler reaches 1.

Bits 11:0 – LVLTH[11:0] Level Threshold
Defines the period after which the watchdog generates an interrupt.

© 2026 Microchip Technology Inc. and its subsidiaries

18.5.6. DWDT Watchdog 1 Interrupt Enable Register

Name: WDT1_IER
Offset: 0x0014
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			RLDERR			LVLINT	RPTHINT	PERINT
Access			W			W	W	W
Reset			–			–	–	–

Bit 5 – RLDERR Reload Command Error Interrupt Enable

Value	Description
0	No effect.
1	Enables the interrupt when there is an attempt to do a Watchdog 1 reload with an incorrect key (WDT1_CR.KEY).

Bit 2 – LVLINT Interrupt Level Threshold Interrupt Enable

Value	Description
0	No effect.
1	Enables the interrupt when the Watchdog 1 counter reaches the interrupt period defined in WDT1_IL.LVLTH.

Bit 1 – RPTHINT Reload Repeat Period Interrupt Enable

Value	Description
0	No effect.
1	Enables the interrupt when the Watchdog 1 is reloaded before the period configured in WDT1_WL.RPTH.

Bit 0 – PERINT Overflow Period Interrupt Enable

Value	Description
0	No effect.
1	Enables the interrupt when the Watchdog 1 overflow occurs (period configured in WDT1_WL.PERIOD).

© 2026 Microchip Technology Inc. and its subsidiaries

18.5.7. DWDT Watchdog 1 Interrupt Disable Register

Name: WDT1_IDR
Offset: 0x0018
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			RLDERR			LVLINT	RPTHINT	PERINT
Access			W			W	W	W
Reset			–			–	–	–

Bit 5 – RLDERR Reload Command Error Interrupt Disable

Value	Description
0	No effect.
1	Disables the interrupt when there is an attempt to do a Watchdog 1 reload with an incorrect key (WDT1_CR.KEY).

Bit 2 – LVLINT Interrupt Level Threshold Interrupt Disable

Value	Description
0	No effect.
1	Disables the interrupt when the Watchdog 1 counter reaches the interrupt period defined in WDT1_IL.LVLTH.

Bit 1 – RPTHINT Reload Repeat Period Interrupt Disable

Value	Description
0	No effect.
1	Disables the interrupt when the Watchdog 1 is reloaded before the period configured in WDT1_WL.RPTH.

Bit 0 – PERINT Overflow Period Interrupt Disable

Value	Description
0	No effect.
1	Disables the interrupt when the Watchdog 1 overflow occurs (period configured in WDT1_WL.PERIOD).

© 2026 Microchip Technology Inc. and its subsidiaries

18.5.8. DWDT Watchdog 1 Interrupt Status Register

Name: WDT1_ISR
Offset: 0x001C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			RLDERR			LVLINT	RPTHINT	PERINT
Access			R			R	R	R
Reset			0			0	0	0

Bit 5 – RLDERR Reload Command Error Status (cleared on read)

Value	Description
0	No attempt to do a Watchdog 1 reload with an incorrect key (WDT1_CR.KEY) since the last read of WDT1_ISR.
1	At least one attempt to do a Watchdog 1 reload with an incorrect key (WDT1_CR.KEY) since the last read of WDT1_ISR.

Bit 2 – LVLINT Interrupt Level Threshold Status (cleared on read)

Value	Description
0	The Watchdog 1 counter did not reach the period defined in WDT1_IL.LVLTH since the last read of WDT1_ISR.
1	The Watchdog 1 counter reached the period defined in WDT1_IL.LVLTH since the last read of WDT1_ISR.

Bit 1 – RPTHINT Reload Repeat Period Status (cleared on read)

Value	Description
0	No reload of the Watchdog 1 before the period configured in WDT1_WL.RPTH since the last read of WDT1_ISR.
1	At least one reload of the Watchdog 1 before the period configured in WDT1_WL.RPTH since the last read of WDT1_ISR.

Bit 0 – PERINT Overflow Period Status (cleared on read)

Value	Description
0	No Watchdog 1 overflow has occurred since the last read of WDT1_ISR.
1	At least one Watchdog 1 overflow has occurred since the last read of WDT1_ISR.

© 2026 Microchip Technology Inc. and its subsidiaries

18.5.9. DWDT Watchdog 1 Interrupt Mask Register

Name: WDT1_IMR
Offset: 0x0020
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			RLDERR			LVLINT	RPTHINT	PERINT
Access			R			R	R	R
Reset			0			0	0	0

Bit 5 – RLDERR Reload Command Error Interrupt Mask

Value	Description
0	The interrupt on Watchdog 1 attempt to reload with an incorrect key (WDT1_CR.KEY) is disabled.
1	The interrupt on Watchdog 1 attempt to reload with an incorrect key (WDT1_CR.KEY) is enabled.

Bit 2 – LVLINT Interrupt Level Threshold Interrupt Mask

Value	Description
0	The interrupt is disabled when the Watchdog 1 counter reaches the period defined in WDT1_IL.LVLTH.
1	The interrupt is enabled when the Watchdog 1 counter reaches the period defined in WDT1_IL.LVLTH.

Bit 1 – RPTHINT Reload Repeat Period Interrupt Mask

Value	Description
0	The interrupt is disabled when the Watchdog 1 is reloaded before the period configured in WDT1_WL.RPTH.
1	The interrupt is enabled when the Watchdog 1 is reloaded before the period configured in WDT1_WL.RPTH.

Bit 0 – PERINT Overflow Period Interrupt Mask

Value	Description
0	The interrupt is disabled when a Watchdog 1 overflow occurs.
1	The interrupt is enabled when a Watchdog 1 overflow occurs.

© 2026 Microchip Technology Inc. and its subsidiaries

18.5.10. DWDT Watchdog 0 Control Register

Name: WDT0_CR
Offset: 0x1210
Reset: –
Property: Write-only

The WDT0_CR register values must not be modified within three MD_SLCK periods following a restart of the watchdog performed by a write access in WDT0_CR. Any modification will cause the watchdog to trigger an end of period earlier than expected.

Bit	31	30	29	28	27	26	25	24
	KEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				LOCKMR				WDRSTT
Access				W				W
Reset				–				–

Bits 31:24 – KEY[7:0] Password

Value	Name	Description
0xA5	PASSWD	Writing any other value in this field aborts the write operation.

Bit 4 – LOCKMR Lock Mode Register Write Access

Value	Description
0	No effect.
1	Locks the configuration registers if KEY is written to 0xA5. Write accesses to WDT0_MR, WDT0_VR, WDT0_WL and WDT0_IL have no effect.

Bit 0 – WDRSTT Watchdog Restart

Value	Description
0	No effect.
1	Restarts the watchdog if KEY is written to 0xA5.

© 2026 Microchip Technology Inc. and its subsidiaries

18.5.11. DWDT Watchdog 0 Mode Register

Name: WDT0_MR
Offset: 0x1214
Reset: 0x00000030
Property: Read/Write

Write access to this register has no effect if the LOCKMR command is issued in WDT0_CR (unlocked on hardware reset).

The WDT0_MR register values must not be modified within three MD_SLCK periods following a restart of the watchdog performed by a write access in WDT0_CR. Any modification will cause the watchdog to trigger an end of period earlier than expected.

Bit	31	30	29	28	27	26	25	24
			WDDBG0HLT	WDIDLEHLT				
Access			R/W	R/W				
Reset			0	0				

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				WDDIS			WDNRSTDIS	
Access				R/W			R/W	
Reset				0			0	

Bit	7	6	5	4	3	2	1	0
			RPTHRST	PERIODRST				
Access			R/W	R/W				
Reset			1	1				

Bit 29 – WDDBG0HLT Watchdog Core Debug Halt

Value	Description
0	The Watchdog 0 runs when the processor is in Debug state.
1	The Watchdog 0 stops when the processor is in Debug state.

Bit 28 – WDIDLEHLT Watchdog Idle Halt

Value	Description
0	The watchdog runs when the system is in Idle state.
1	The watchdog stops when the system is in Idle state.

Bit 12 – WDDIS Watchdog Disable

Value	Description
0	Enables the Watchdog Timer.
1	Disables the Watchdog Timer.

Bit 9 – WDNRSTDIS Watchdog NRST Disable

Value	Description
0	A watchdog reset asserts the NRST pin.
1	A watchdog reset does not assert the NRST pin.

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 5 – RPTHRST Repeat Threshold Reset

Value	Description
0	No reset is generated if the watchdog is restarted before the RPTH threshold.
1	A reset is generated if the watchdog is restarted before the RPTH threshold.

Bit 4 – PERIODRST Period Reset

Value	Description
0	No reset is generated if the watchdog down counter reaches 0.
1	A reset is generated once the watchdog down counter reaches 0.

© 2026 Microchip Technology Inc. and its subsidiaries

18.5.12. DWDT Watchdog 0 Value Register

Name: WDT0_VR
Offset: 0x1218
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					COUNTER[11:8]			
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COUNTER[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 11:0 – COUNTER[11:0] Watchdog Down Counter Value
Current value of the watchdog down counter.
Due to the asynchronous operation of the watchdog, it is necessary to read this register twice at the same value to get a valid read. Therefore, a minimum of two and a maximum of three accesses are required.

© 2026 Microchip Technology Inc. and its subsidiaries

18.5.13. DWDT Watchdog 0 Window Level Register

Name: WDT0_WL
Offset: 0x121C
Reset: 0x00000FFF
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
					RPTH[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RPTH[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
					PERIOD[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					1	1	1	1

Bit	7	6	5	4	3	2	1	0
	PERIOD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 27:16 – RPTH[11:0] Repeat Threshold
Defines the period before which the Watchdog 0 restart generates an interrupt.

Bits 11:0 – PERIOD[11:0] Watchdog Period
Defines the period after which the Watchdog 0 generates an interrupt.

© 2026 Microchip Technology Inc. and its subsidiaries

18.5.14. DWDT Watchdog 0 Interrupt Level Register

Name: WDT0_IL
Offset: 0x1220
Reset: 0x00000FFF
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						PRESC[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
					LVLTH[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					1	1	1	1

Bit	7	6	5	4	3	2	1	0
	LVLTH[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 18:16 – PRESC[2:0] Prescaler Ratio
Defines the Watchdog 0 Prescaler ratio.

Value	Name	Description
0x0	RATIO128	The watchdog counter decreased when the prescaler reaches 128.
0x1	RATIO64	The watchdog counter decreased when the prescaler reaches 64.
0x2	RATIO32	The watchdog counter decreased when the prescaler reaches 32.
0x3	RATIO16	The watchdog counter decreased when the prescaler reaches 16.
0x4	RATIO8	The watchdog counter decreased when the prescaler reaches 8.
0x5	RATIO4	The watchdog counter decreased when the prescaler reaches 4.
0x6	RATIO2	The watchdog counter decreased when the prescaler reaches 2.
0x7	RATIO1	The watchdog counter decreased when the prescaler reaches 1.

Bits 11:0 – LVLTH[11:0] Level Threshold
Defines the period after which the Watchdog generates an interrupt.

© 2026 Microchip Technology Inc. and its subsidiaries

18.5.15. DWDT Watchdog 0 Interrupt Enable Register

Name: WDT0_IER
Offset: 0x1224
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			RLDERR	W1RPTHINT	W1PERINT	LVLINT	RPTHINT	PERINT
Access			W	W	W	W	W	W
Reset			–	–	–	–	–	–

Bit 5 – RLDERR Reload Command Error Interrupt Enable

Value	Description
0	No effect.
1	Enables the interrupt when there is an attempt to do a Watchdog 0 reload with an incorrect key (WDT0_CR.KEY).

Bit 4 – W1RPTHINT Watchdog 1 Repeat Threshold Interrupt Enable

Value	Description
0	No effect.
1	Enables the interrupt when Watchdog 1 is reloaded before the period configured in WDT1_WL.RPTH.

Bit 3 – W1PERINT Watchdog 1 Overflow Interrupt Enable

Value	Description
0	No effect.
1	Enables the interrupt when Watchdog 1 overflows.

Bit 2 – LVLINT Interrupt Level Threshold Interrupt Enable

Value	Description
0	No effect.
1	Enables the interrupt when the Watchdog 0 counter reaches the interrupt period defined in WDT0_IL.LVLTH.

Bit 1 – RPTHINT Reload Repeat Period Interrupt Enable

Value	Description
0	No effect.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
1	Enables the interrupt when the Watchdog 0 is reloaded before the period configured in WDT0_WL.RPTH.

Bit 0 – PERINT Overflow Period Interrupt Enable

Value	Description
0	No effect.
1	Enables the interrupt when the Watchdog 0 overflow occurs (period configured in WDT0_WL.PERIOD).

© 2026 Microchip Technology Inc. and its subsidiaries

18.5.16. DWDT Watchdog 0 Interrupt Disable Register

Name: WDT0_IDR
Offset: 0x1228
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			RLDERR	W1RPTHINT	W1PERINT	LVLINT	RPTHINT	PERINT
Access			W	W	W	W	W	W
Reset			–	–	–	–	–	–

Bit 5 – RLDERR Reload Command Error Interrupt Disable

Value	Description
0	No effect.
1	Disables the interrupt when there is an attempt to do a Watchdog 0 reload with an incorrect key (WDT0_CR.KEY).

Bit 4 – W1RPTHINT Watchdog 1 Repeat Threshold Interrupt Disable

Value	Description
0	No effect.
1	Disables the interrupt when Watchdog 1 is reloaded before the period configured in WDT1_WL.RPTH.

Bit 3 – W1PERINT Watchdog 1 Overflow Interrupt Disable

Value	Description
0	No effect.
1	Disables the interrupt when Watchdog 1 overflows.

Bit 2 – LVLINT Interrupt Level Threshold Interrupt Disable

Value	Description
0	No effect.
1	Disables the interrupt when the Watchdog 0 counter reaches the interrupt period defined in WDT0_IL.LVLTH.

Bit 1 – RPTHINT Reload Repeat Period Interrupt Disable

Value	Description
0	No effect.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
1	Disables the interrupt when the Watchdog 0 is reloaded before the period configured in WDT0_WL.RPTH.

Bit 0 – PERINT Overflow Period Interrupt Disable

Value	Description
0	No effect.
1	Disables the interrupt when the Watchdog 0 overflow occurs (period configured in WDT0_WL.PERIOD).

© 2026 Microchip Technology Inc. and its subsidiaries

18.5.17. DWDT Watchdog 0 Interrupt Status Register

Name: WDT0_ISR
Offset: 0x122C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			RLDERR	W1RPTHINT	W1PERINT	LVLINT	RPTHINT	PERINT
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit 5 – RLDERR Reload Command Error Status (cleared on read)

Value	Description
0	No attempt to do a Watchdog 0 reload with an incorrect key (WDT0_CR.KEY) since the last read of WDT0_ISR.
1	At least one attempt to do a Watchdog 0 reload with an incorrect key (WDT0_CR.KEY) since the last read of WDT0_ISR.

Bit 4 – W1RPTHINT Watchdog 1 Repeat Threshold Interrupt Status

Value	Description
0	No Watchdog 1 repeat threshold failure since the last read of WDT0_ISR.
1	At least one Watchdog 1 repeat threshold failure since the last read of WDT0_ISR.

Bit 3 – W1PERINT Watchdog 1 Overflow Interrupt Status

Value	Description
0	No Watchdog 1 overflow has occurred in the Watchdog 0 since the last read of WDT0_ISR.
1	At least one Watchdog 1 overflow occurred in the Watchdog 0 since the last read of WDT0_ISR.

Bit 2 – LVLINT Interrupt Level Threshold Status (cleared on read)

Value	Description
0	The Watchdog 0 counter did not reach the period defined in WDT0_IL.LVLTH since the last read of WDT0_ISR.
1	The Watchdog 0 counter reached the period defined in WDT0_IL.LVLTH since the last read of WDT0_ISR.

Bit 1 – RPTHINT Reload Repeat Period Status (cleared on read)

Value	Description
0	No reload of the Watchdog 0 before the period configured in WDT0_WL.RPTH since the last read of WDT0_ISR.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
1	At least one reload of the Watchdog 0 before the period configured in WDT0_WL.RPTH since the last read of WDT0_ISR.

Bit 0 – PERINT Overflow Period Status (cleared on read)

Value	Description
0	No Watchdog 0 overflow has occurred since the last read of WDT0_ISR.
1	At least one Watchdog 0 overflow has occurred since the last read of WDT0_ISR.

© 2026 Microchip Technology Inc. and its subsidiaries

18.5.18. DWDT Watchdog 0 Interrupt Mask Register

Name: WDT0_IMR
Offset: 0x1230
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			RLDERR	W1RPTHINT	W1PERINT	LVLINT	RPTHINT	PERINT
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit 5 – RLDERR Reload Command Error Interrupt Mask

Value	Description
0	The interrupt on Watchdog 0 attempt to reload with an incorrect key (WDT0_CR.KEY) is disabled.
1	The interrupt on Watchdog 0 attempt to reload with an incorrect key (WDT0_CR.KEY) is enabled.

Bit 4 – W1RPTHINT Watchdog 1 Repeat Threshold Interrupt Mask

Value	Description
0	The interrupt is disabled when Watchdog 1 is reloaded before the period configured in WDT1_WL.RPTH.
1	The interrupt is enabled when Watchdog 1 is reloaded before the period configured in WDT1_WL.RPTH.

Bit 3 – W1PERINT Watchdog 1 Overflow Interrupt Mask

Value	Description
0	The interrupt is disabled when Watchdog 1 overflow occurs.
1	The interrupt is enabled when Watchdog 1 overflow occurs.

Bit 2 – LVLINT Interrupt Level Threshold Interrupt Mask

Value	Description
0	The interrupt is disabled when the Watchdog 0 counter reaches the period defined in WDT0_IL.LVLTH.
1	The interrupt is enabled when the Watchdog 0 counter reaches the period defined in WDT0_IL.LVLTH.

Bit 1 – RPTHINT Reload Repeat Period Interrupt Mask

Value	Description
0	The interrupt is disabled when the Watchdog 0 is reloaded before the period configured in WDT0_WL.RPTH.
1	The interrupt is enabled when the Watchdog 0 is reloaded before the period configured in WDT0_WL.RPTH.

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 0 – PERINT Overflow Period Interrupt Mask

Value	Description
0	The interrupt is disabled when Watchdog 0 overflow occurs.
1	The interrupt is enabled when Watchdog 0 overflow occurs.

© 2026 Microchip Technology Inc. and its subsidiaries

18.5.19. DWDT Watchdog 1 Level Limit Register

Name: WDT1_LVLLIM
Offset: 0x1234
Reset: 0x0FFF0000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
					LVLMAX[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					1	1	1	1

Bit	23	22	21	20	19	18	17	16
	LVLMAX[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8
					LVLMIN[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	LVLMIN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 27:16 – LVLMAX[11:0] Maximum Level
Defines the maximum value that can be applied on WDT1_IL.LVLTH.
If the currently defined LVLTH value is higher than the LVLMAX to be configured, LVLMAX value is applied to LVLTH.

Bits 11:0 – LVLMIN[11:0] Minimum Level
Defines the minimum value that can be applied on WDT1_IL.LVLTH.
If the currently defined LVLTH value is lower than the LVLMIN to be configured, LVLMIN value is applied to LVLTH.

© 2026 Microchip Technology Inc. and its subsidiaries

18.5.20. DWDT Watchdog 1 Repeat Threshold Limit Register

Name: WDT1_RLIM
Offset: 0x1238
Reset: 0x0FFF0000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
					RPTHMAX[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					1	1	1	1

Bit	23	22	21	20	19	18	17	16
	RPTHMAX[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8
					RPTHMIN[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RPTHMIN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 27:16 – RPTHMAX[11:0] Maximum Repeat Threshold
Defines the maximum value that can be applied to WDT1_WL.RPTH.
If the currently defined RPTH value is higher than the RPTHMAX to be configured, RPTHMAX value is applied to RPTH.

Bits 11:0 – RPTHMIN[11:0] Minimum Repeat Threshold
Defines the minimum value that can be applied to WDT1_WL.RPTH.
If the currently defined RPTH value is lower than the RPTHMIN to be configured, RPTHMIN value is applied to RPTH.

© 2026 Microchip Technology Inc. and its subsidiaries

18.5.21. DWDT Watchdog 1 Period Limit Register

Name: WDT1_PLIM
Offset: 0x123C
Reset: 0x0FFF0000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
					PERMAX[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					1	1	1	1

Bit	23	22	21	20	19	18	17	16
	PERMAX[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8
					PERMIN[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PERMIN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 27:16 – PERMAX[11:0] Period Maximum
Defines the maximum value that can be applied to WDT1_WL.PERIOD.
If the currently defined PERIOD value of the WDT1 is higher than the PERMAX to be configured, PERMAX value is applied to PERIOD.

Bits 11:0 – PERMIN[11:0] Period Minimum
Defines the minimum value that can be applied to WDT1_WL.PERIOD.
If the currently defined PERIOD value of the WDT1 is lower than the PERMIN to be configured, PERMIN value is applied to PERIOD.

© 2026 Microchip Technology Inc. and its subsidiaries

19. Clock Generator

19.1. Description

The Clock Generator user interface is embedded within the Power Management Controller and is described in the Register Summary. However, the Clock Generator registers are named CKGR_.

19.2. Embedded Characteristics

The Clock Generator is made up of:

- Oscillators
  - A low-power 32.768 kHz oscillator supporting crystals, MEMs, resonators and Bypass mode
    Note: This oscillator is referred to as 32.768 kHz Crystal Oscillator throughout the document.
  - An embedded always-on, slow RC oscillator generating a typical 32 kHz clock
  - A 12 to 48 MHz oscillator supporting crystals, MEMs, resonators and Bypass mode
    Note: This oscillator is referred to as Main Crystal Oscillator throughout the document.
  - A Main RC oscillator generating a typical 12 MHz clock
- Three fractional-N PLLs

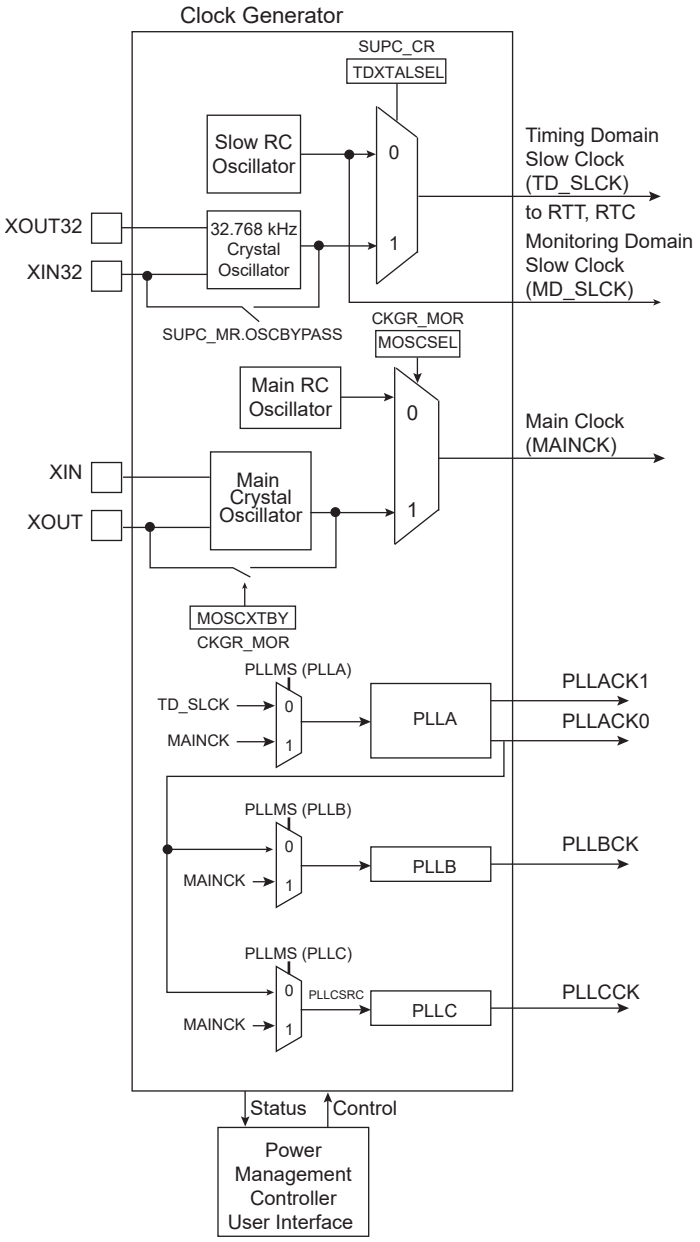
It provides the following clocks:

- MD_SLCK—Monitoring Domain Slow clock. This clock, sourced from the always-on Slow RC oscillator only, is the only permanent clock of the system and feeds safety-critical functions of the device (DWDT, RSTC, SUPC, frequency monitors and detectors, PMC start-up time counters).
- TD_SLCK—Timing Domain Slow clock. This clock, sourced from the 32.768 kHz crystal oscillator or the always-on Slow RC oscillator, is routed to the RTC and RTT peripherals.
- MAINCK—Output of the Main clock oscillator selection. This clock is either the Main crystal oscillator or Main RC oscillator.
- PLL Clocks—Outputs of embedded PLLs
- One SysTick external clock for each Processor core

© 2026 Microchip Technology Inc. and its subsidiaries

19.3. Block Diagram

Figure 19-1. Clock Generator Block Diagram



19.4. Slow Clock

The PMC does not control the Slow clock generation. The control of the Slow clock is performed by the Supply Controller (SUPC) which embeds a slow clock generator that is supplied by the output of the power switch (VBAT or VDD3V3). As soon as the SUPC is supplied, both the 32.768 kHz crystal oscillator and the Slow RC oscillator are powered, but only the Slow RC oscillator is enabled.

MD_SLCK is always generated by the Slow RC oscillator.

TD_SLCK is generated either by the 32.768 kHz crystal oscillator or by the Slow RC oscillator.

The selection is made via the TDXTALSEL bit in the SUPC Control register (SUPC_CR).

© 2026 Microchip Technology Inc. and its subsidiaries

19.4.1. Slow RC Oscillator (32 kHz typical)

The Slow RC oscillator is a permanent 32 kHz clock that is the source clock of MD_SLCK and the default source clock of TD_SLCK.

Compared to the 32.768 kHz crystal oscillator, this oscillator offers a faster start-up time and is less exposed to the external environment, as it is fully integrated. However, its output frequency is subject to larger variations with supply voltage, temperature and manufacturing process. Therefore, the user must take these variations into account when this oscillator is used as a time base (start-up counter, frequency monitor, etc.). Refer to the section “Electrical Characteristics”.

19.4.2. 32.768 kHz Crystal Oscillator

By default, the 32.768 kHz oscillator is disabled. To use this oscillator, the XIN32 and XOUT32 pins must be connected to a 32.768 kHz crystal, to a MEMS oscillator, or to a ceramic resonator. Refer to the section “Electrical Characteristics” for appropriate loading capacitors selection on XIN32 and XOUT32.

To select the 32.768 kHz crystal oscillator as the source of TD_SLCK, SUPC_CR.TDXTALSEL must be set. The switch of TD_SLCK source is glitch-free.

Reverting to the Slow RC oscillator is only possible by shutting down the Backup power supply.

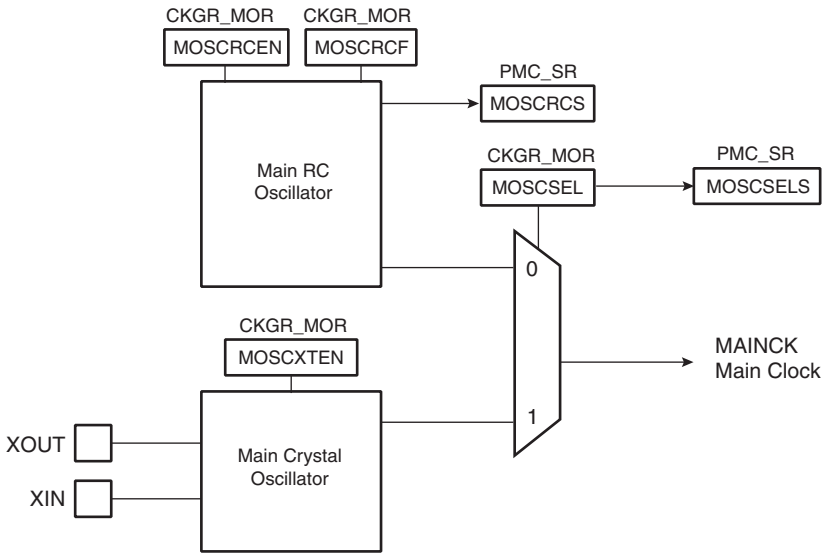
The user can also set the 32.768 kHz crystal oscillator in Bypass mode instead of connecting a crystal. In this case, the user must provide the external clock signal on XIN32. For input characteristics of the XIN32 pin, refer to the section “Electrical Characteristics”. To enter Bypass mode, the OSCBYPASS bit of the Supply Controller Mode register (SUPC_MR) must be set prior to setting SUPC_CR.TDXTALSEL.

19.5. Main Clock

The Main clock (MAINCK) has two sources:

- A Main RC oscillator with a fast start-up time and that is selected by default to start the system
- A Main crystal oscillator with Bypass mode

Figure 19-2. Main Clock (MAINCK) Block Diagram



19.5.1. Main RC Oscillator

After reset, the Main RC oscillator is enabled. This oscillator is selected as the source of MAINCK. MAINCK is the default clock selected to start the system.

© 2026 Microchip Technology Inc. and its subsidiaries

The Main RC oscillator is calibrated in production. For output frequency specifications, refer to the section “Electrical Characteristics”.

The software can disable or enable the Main RC oscillator with the MOSCRCEN bit in the Clock Generator Main Oscillator register (CKGR_MOR).

When disabling the Main RC oscillator by clearing CKGR_MOR.MOSCRCEN, PMC_SR.MOSCRCS is automatically cleared, indicating that the oscillator is OFF.

Setting MOSCRCS in the Interrupt Enable register (PMC_IER) triggers an interrupt to the processor.

19.5.2. Main RC Oscillator Frequency Adjustment

The user can adjust the value of the Main RC oscillator frequency by modifying the trimming values set in production by Microchip. This may be used to compensate frequency drifts due to temperature or voltage. The values stored in the Flash cannot be erased by a Flash erase command or by the ERASE pin. Values written by the user application in the Oscillator Calibration register (PMC_OCR) are reset after each power-up or peripheral reset.

By default, PMC_OCR.SEL12 is cleared, so the Main RC oscillator is driven with the factory-defined calibration bits which are programmed during chip production.

In order to adjust the oscillator frequency, PMC_OCR.SEL12 must be set to ‘1’ and a valid value must be configured in PMC_OCR.CAL12.

It is possible to adjust the oscillator frequency while operating from this oscillator.

When reading PMC_OCR, CAL12 contains the value of the trimming that is currently sent to the Main RC oscillator. This means that the read value is either the factory-defined value (PMC_OCR.SEL12=’0’) or the value written in the register by the user (PMC_OCR.SEL12=’1’).

At any time, the user can measure the main RC oscillator output frequency by means of the Main Frequency Counter (refer to Main Frequency Counter). Once the frequency measurement is done, the main RC oscillator calibration field CAL12 can be adjusted accordingly to correct this oscillator output frequency.

19.5.3. Main Crystal Oscillator

After reset, the Main crystal oscillator is disabled and is not selected as the source of MAINCK.

The software enables or disables this oscillator in order to reduce power consumption via CKGR_MOR.MOSCXTEN.

When disabling this oscillator by clearing the CKGR_MOR.MOSCXTEN, the PMC_SR.MOSCXTS status bit is automatically cleared, indicating the oscillator is off. To activate the Main Crystal oscillator bypass mode, refer to Bypassing the Main Crystal Oscillator.

When enabling this oscillator, the user must initiate the start-up time counter. The start-up time depends on the characteristics of the external device connected to this oscillator.

When CKGR_MOR.MOSCXTEN and CKGR_MOR.MOSCXTST are written to enable this oscillator, XIN and XOUT are driven by the Main crystal oscillator. PMC_SR.MOSCXTS is cleared and the counter starts counting down on MD_SLCK divided by 8 from the CKGR_MOR.MOSCXTST value. Since the CKGR_MOR.MOSCXTST value is coded with 8 bits, the start-up time can be programmed up to 2048 MD_SLCK periods, corresponding to about 62 ms when running at 32.768 kHz.

When the start-up time counter reaches ‘0’, PMC_SR.MOSCXTS is set, indicating that the oscillator is stabilized. Setting MOSCXTS in the Interrupt Mask register (PMC_IMR) can trigger an interrupt to the processor.

19.5.4. Main Clock Source Selection

The source of MAINCK can be selected from the following:

© 2026 Microchip Technology Inc. and its subsidiaries

- the Main RC oscillator
- the Main crystal oscillator
- an external clock signal provided on the XIN input (Bypass mode of the Main crystal oscillator)

The advantage of the Main RC oscillator is its fast start-up time. By default, this oscillator is selected to start the system and it must be selected prior to entering Wait mode.

The advantage of the Main crystal oscillator is its high level of accuracy.

The selection of the oscillator is made by configuring CKGR_MOR.MOSCSEL. The switchover of the MAINCK source is glitch-free, thus the switchover can be performed even if MCK0 and MCK1 are fed by MAINCK. PMC_SR.MOSCSELS indicates when the switch sequence is done.

Setting PMC_IMR.MOSCSELS triggers an interrupt to the processor.

19.5.5. Bypassing the Main Crystal Oscillator

Prior to bypassing the Main crystal oscillator (CKGR_MOR.MOSCXTBY='1'), an external clock frequency provided on the XIN pin must be stable and within the values specified in the XIN Clock characteristics in the section "Electrical Characteristics". Then the Main crystal oscillator must be enabled by setting CKGR_MOR.MOSCXTEN to 1.

19.5.6. Main Frequency Counter

The Main frequency counter measures the Main RC oscillator or the Main crystal oscillator against the MD_SLCK and is managed by CKGR_MCFR.

During the measurement period, the Main frequency counter increments at the speed of the clock defined by CKGR_MCFR.CCSS.

A measurement is started in the following cases:

- When CKGR_MCFR.RCMEAS is written to '1'
- When the Main RC oscillator is selected as the source of MAINCK and when this oscillator is stable (i.e., when MOSCRCS is set)
- When the Main crystal oscillator is selected as the source of MAINCK and when this oscillator is stable (i.e., when MOSCXTS is set)
- When MAINCK source selection is modified

The measurement period ends at the 16th falling edge of MD_SLCK, MAINFRDY in CKGR_MCFR is set and the counter stops counting. Its value can be read in CKGR_MCFR.MAINF and gives the number of clock cycles during 16 periods of MD_SLCK, so that the frequency of the Main RC oscillator or Main crystal oscillator can be determined.

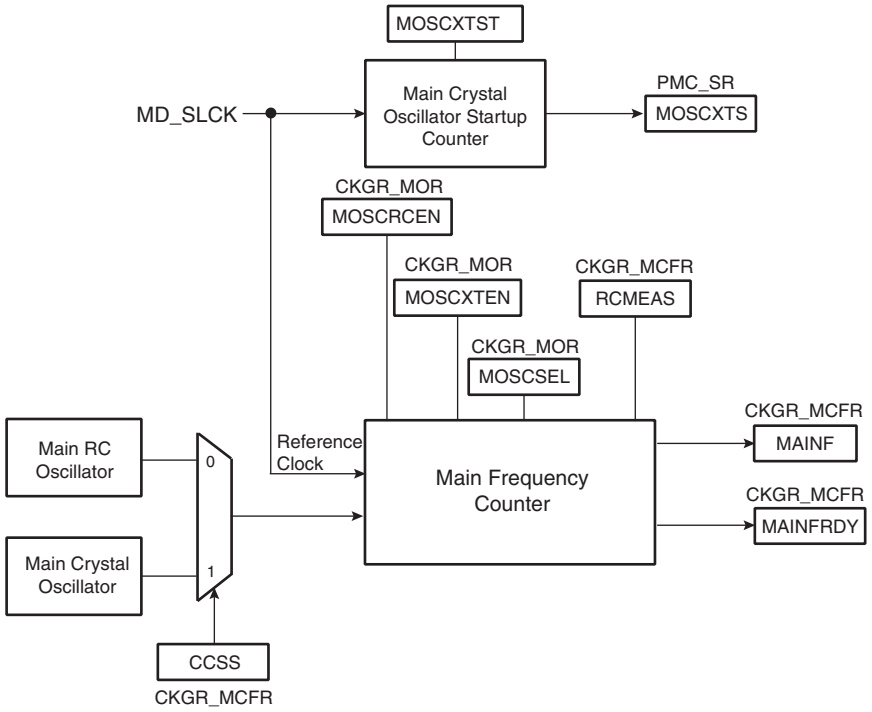
When switching the source of MAINCK from the Main RC oscillator to the Main crystal oscillator, follow the programming sequence below to ensure that the oscillator is present and that its frequency is valid:

1. Enable the Main crystal oscillator by setting CKGR_MOR.MOSCXTEN. Configure CKGR_MOR. MOSCXTST with the Main crystal oscillator start-up time as defined in the section "Electrical Characteristics".
2. Wait for PMC_SR.MOSCXTS flag to rise, indicating the end of a start-up period of the Main crystal oscillator.
3. Select the Main crystal oscillator as the source clock of the Main frequency counter by setting CKGR_MCFR.CCSS.
4. Initiate a frequency measurement by setting CKGR_MCFR.RCMEAS.
5. Read CKGR_MCFR.MAINFRDY until its value equals 1.
6. Read CKGR_MCFR.MAINF and compute the value of the Main crystal frequency.

© 2026 Microchip Technology Inc. and its subsidiaries

If the MAINF value is valid, software can switch MAINCK to the Main crystal oscillator. Refer to Main Clock Source Selection.

Figure 19-3. Main Frequency Counter Block Diagram



19.6. PLL Controls

The PMC embeds 3 PLLs that are controlled by the registers PMC_PLL_CTRL0, PMC_PLL_CTRL1, PMC_PLL_CTRL2, PMC_PLL_SSR, PMC_PLL_ACR and PMC_PLL_UPDATE.

Although these PLLs are similar, their implementations differ in terms of input clock signal, maximum output clock frequency or availability of a dedicated output line.

Each PLL has a constraint on the frequency it can generate on its clock output. Refer to the section “Electrical Characteristics”.

The table below describes all PLLs with their names and source clocks. For maximum frequency, refer to the section “Electrical Characteristics”.

Table 19-1. PLL List

Index	PLL Name	Clock Name	PLL Clock Source	Usage Example
0	PLLA	PLLACK0	MAINCK/TD_SLOW_CLOCK	See General Clock Distribution Block Diagram
		PLLACK1		
1	PLLB	PLLBCK	PLLACK0/MAINCK	
2	PLLC	PLLCCK	PLLACK0/MAINCK	

19.6.1. Divider and Phase Lock Loop Programming

Each PLL is controlled the same way. The internal clock frequency is configured by setting PMC_PLL_CTRL1.MUL and PMC_PLL_CTRL2.FRACR.

The COREPLLCK operating frequency for PLLA, PLLB and PLLC is defined as:

$$f_{\mathrm{COREPLLCK}} = f_{\mathrm{ref}}\left(\mathrm{MUL} + 1 + \frac{\mathrm{FRACR}}{2^{22}}\right)$$

© 2026 Microchip Technology Inc. and its subsidiaries

where f_{ref} = Selected reference clock in the figure PLL Controls.

PLLACK0, PLLBCK and PLLCCK clock frequencies, noted as f_{PLL} in the formula below, are defined by the following formula:

$$f_{\mathrm{PLL}} = \frac{f_{\mathrm{COREPLLCK}}}{(\mathrm{DIVPMC0} + 1)}$$

The PLLACK1 frequency is defined by the following formula:

$$f_{\mathrm{PLLACK1}} = \frac{f_{\mathrm{COREPLLCK}}}{(\mathrm{DIVPMC1} + 1)}$$

Each PLL sends a lock signal to the PMC to indicate its lock status. Once the lock signal has risen, the clock generated by the PLL is stable and can be sent to the PMC and/or its corresponding I/O if available.

This signal reports the lock status of the PLL by setting the corresponding PMC_PLL_CTRL0.ENLOCK to '1'.

If the lock status is disabled, a start-up time can be used instead in the PMC_PLL_UPDT register. The start-up time is expressed as a number of MD_SLCK cycles. Once the counter has reached the specified value, a flag rises. The start-up time field can only be written while all PLLs are disabled (i.e., their PLLEN fields are null).

If both a start-up time and the lock are enabled, the lock sent by the PLL is read once the start-up time has elapsed.

If neither the start-up time nor the lock are enabled, there is no way to know the lock status of the PLL.

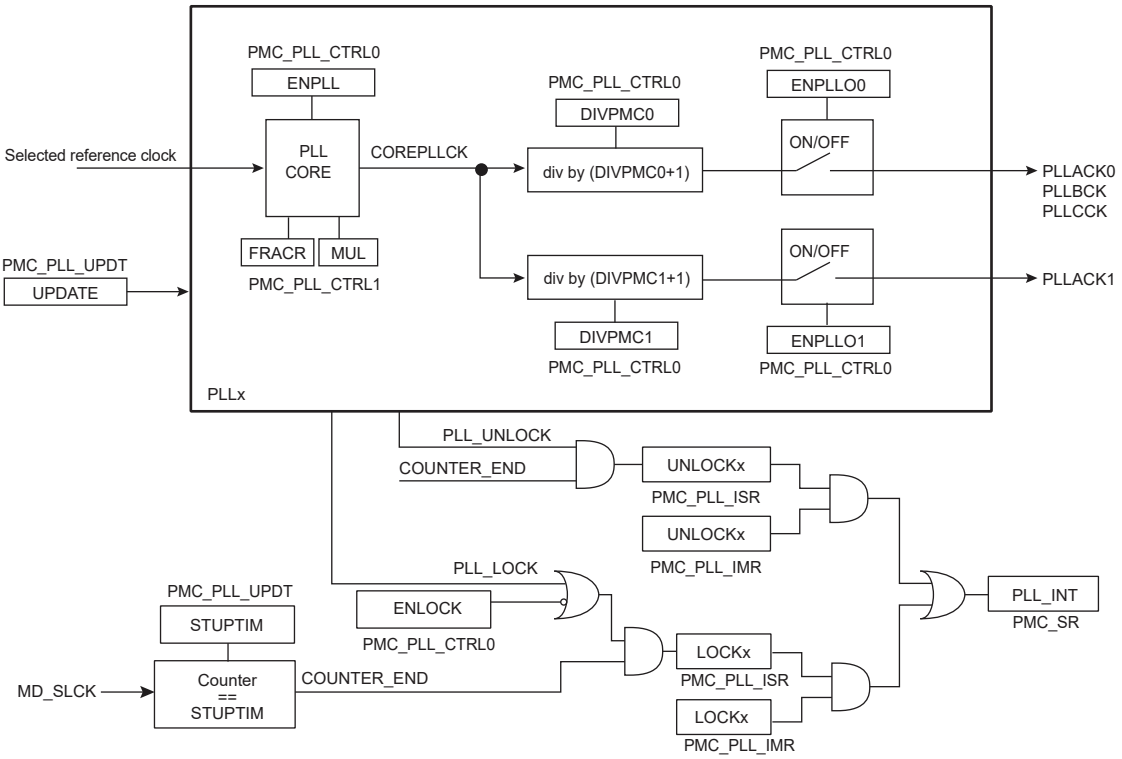
The PLL also embeds an unlock status that informs when the PLL lock is lost. When enabled, this status is read once the start-up time (if defined) has elapsed.

The lock and unlock status can be used as interrupts.

Refer to the figure below.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 19-4. PLL Controls



Follow the steps below to initialize and start a PLL:

Note: Steps referring to ENPLLO1 and DIVPMC1 apply to PLLA only.

1. Define the ID (ID=n) and start-up time by configuring the fields PMC_PLL_UPDT.ID and PMC_PLL_UPDT.STUPTIM. Set PMC_PLL_UPDT.UPDATE to '0'.
2. Configure PMC_PLL_ACR with the following values:
   - PLLA = 0x0F000038
   - PLLB = 0x28000058
   - PLLC = 0x28000058
3. Define the MUL and FRACR to be applied to PLL(n) in PMC_PLL_CTRL1 and PMC_PLL_CTRL2, respectively.
4. Set PMC_PLL_UPDT.UPDATE to '1'. PMC_PLL_UPDT.ID must equal the one written during Step 1. else the update is cancelled.
5. In PMC_PLL_CTRL0, write a '1' to ENLOCK and to ENPLL and configure DIVPMC0, DIVPMC1, ENPLLO0 and ENPLLO1.
6. Set PMC_PLL_UPDT.UPDATE to '1'. PMC_PLL_UPDT.ID must equal the one written during Step 1. else the update is cancelled.
7. Wait for the lock bit to rise by polling the PMC_PLL_ISR0 or by enabling the corresponding interrupt in PMC_PLL_IER.
8. Disable the interrupt (if enabled)
9. Enable the unlock interrupt to quickly detect a failure on the generation of the clock of the PLL.

Once enabled (PMC_PLL_CTRL0.ENPLL=1), the PLL core generates its core clock (COREPLLCK).

Once the PLL has been enabled and has locked, the configuration of the PLL can be modified without switching off the cell.

© 2026 Microchip Technology Inc. and its subsidiaries

The clock generated by the PLL (COREPLLCK) is sent to the PMC if ENPLLO0 has been set to '1' and PMC_PLL_UPDT.UPDATE has then been written to '1'.

The first clock generated by the PLLA is sent to the PLLB and PLLC.

To stop a PLL, the following sequence must be applied:

1. If the PLL drives a section of the system that is active, modify the source clock of the system.
2. Define the ID (ID=n) of the PLL to be switched off in PMC_UPDT. Set PMC_UPDT.UPDATE to '0' in this step.
3. In PMC_PLL_CTRL0, clear ENPLLO0 and leave ENPLL at '1'.
4. Set PMC_PLL_UPDT.UPDATE to '1'. PMC_PLL_UPDT.ID must equal the one written during Step 2. else the update is cancelled.
5. Write a '0' to PMC_PLL_CTRL0.ENPLL.

19.6.2. Spread Spectrum

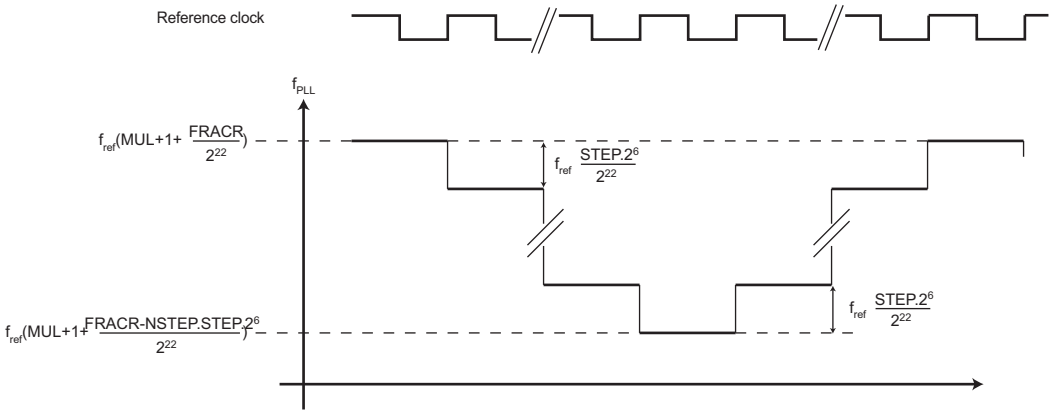
Spread spectrum is performed by modifying the target frequency of the PLL at its input clock rate. Two parameters are used to configure the spread spectrum:

- STEP—the size of the step that will be applied to the FRACR input of the PLL
- NSTEP—the number of times the STEP will be applied to the FRACR input of the PLL

The spread spectrum can be applied only if the PLL is already enabled and locked. Once the spread spectrum has been enabled, it is no longer possible to modify the target frequency of the PLL. The spread spectrum must first be disabled and (2 x NSTEP) cycles of the source clock of the PLL must be waited.

Starting from the base frequency of the PLL configured in PMC_PLL_CTRL1, the spread spectrum mechanism reduces FRACR by (8 x STEP) units at each rising edge of the source clock of the PLL. If FRACR is too low, MUL is automatically modified to match the new PLL frequency. This subtraction is performed NSTEP times. Then the FSM increments FRACR by (8 x STEP) units until it reaches the initial PLL frequency value (as configured in the PMC_PLL_CTRL1). If FRACR is too high, MUL is automatically modified to match the new PLL frequency.

Figure 19-5. Spread Spectrum Mechanism



© 2026 Microchip Technology Inc. and its subsidiaries

20. Power Management Controller (PMC)

20.1. Description

The Power Management Controller (PMC) optimizes power consumption by controlling all system and user peripheral clocks. The PMC enables/disables the clock inputs to many of the peripherals and to the processor.

The Supply Controller (SUPC) selects the source of TD_SLCK (drives the real-time part (RTT/RTC)). The source of MD_SLCK (drives the rest of the System Controller: wake-up logic, watchdog, PMC, etc.) is always the Slow RC Oscillator.

By default, at start-up, the chip runs out of the Main RC oscillator.

20.2. Embedded Characteristics

The Power Management Controller provides the following clocks:

- Processor Clock (CPU_CLK0)
- Free-running Processor Clock (FCLK0)
- Peripheral Clocks with independent ON/OFF control, provided to the peripherals. Each peripheral clock is inherited from one of the MCKx clocks.
- Programmable Clock Outputs (PCKx), selected from the clock generator outputs to drive the device PCKx pins.
- Generic Clock (GCLK) with controllable division and ON/OFF control, independent of MCKx and CPU_CLK0. Provided to selected peripherals. Refer to the table “Peripheral Identifiers” for more details on GCLK availability per peripheral.

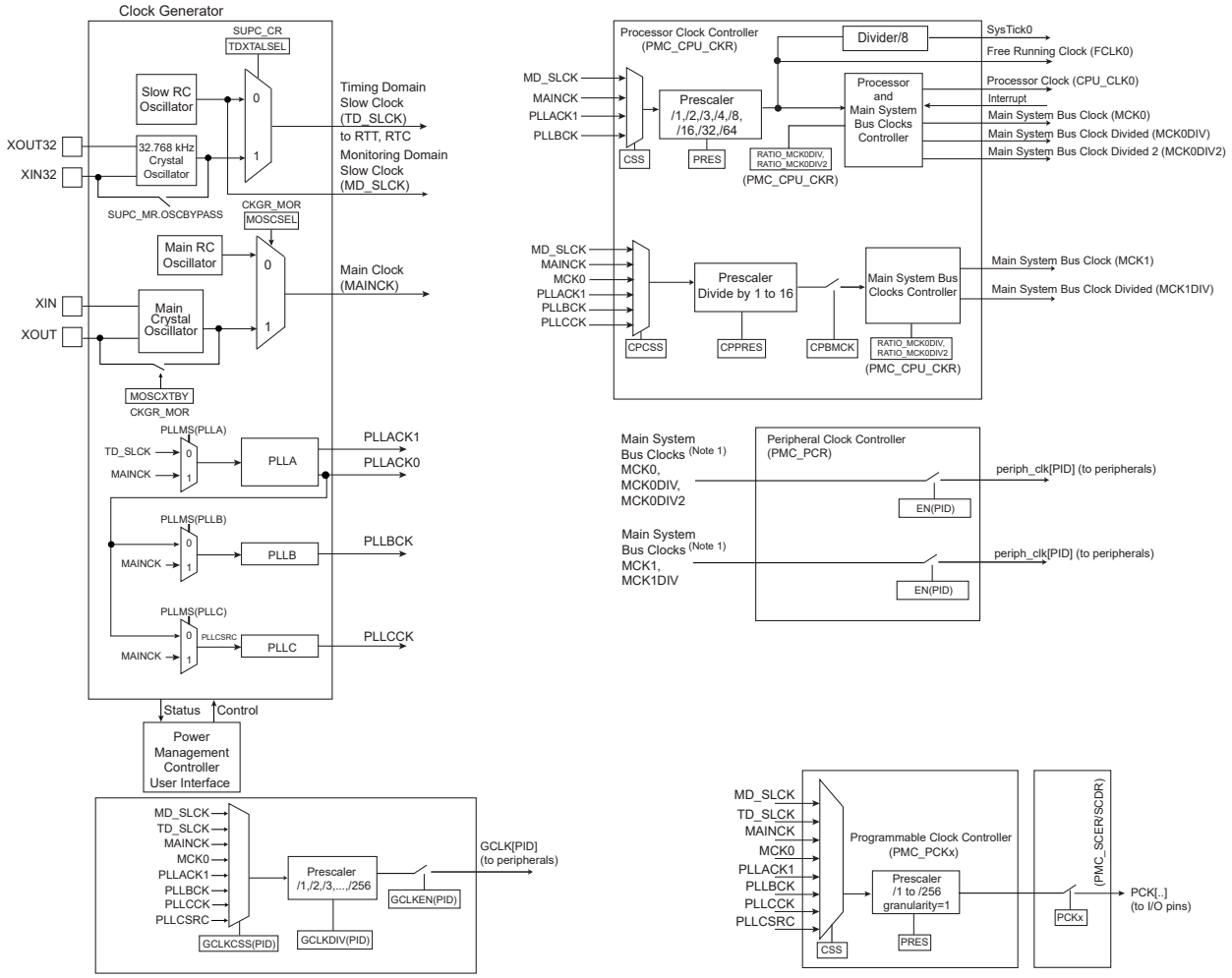
The Power Management Controller also provides the following features on clocks:

- A Main crystal oscillator failure detector
- A 32.768 kHz crystal oscillator frequency monitor
- A frequency counter on Main crystal oscillator or Main RC oscillator
- An MCK0 failure detector

© 2026 Microchip Technology Inc. and its subsidiaries

20.3. Block Diagram

Figure 20-1. General Clock Distribution Block Diagram



Note: (1) See the table “Peripheral Identifiers” for Main System Bus Clocks used versus peripherals.

20.4. Processor Clock Controller

The PMC features a Processor Clock (CPU_CLK0) controller that implements the processor Sleep mode. CPU_CLK0 can be disabled by executing the WFI (WaitForInterrupt) or the WFE (WaitForEvent) processor instruction while LPM is at ‘0’ in the Fast Start-up Mode register (PMC_FSMR).

CPU_CLK0 is enabled after a reset and is automatically re-enabled by any enabled interrupt. The processor Sleep mode is entered by disabling CPU_CLK0, which is automatically re-enabled by any enabled interrupt, or by the reset of the product.

When processor Sleep mode is entered, the current instruction is finished before CPU_CLK0 is stopped, but this does not prevent data transfers from other hosts of the system bus.

The clock selection is done in PMC_CPU_CKR.CSS and PMC_CPU_CKR.CPCSS.

The prescaler is configured in PMC_CPU_CKR.PRES and PMC_CPU_CKR.CPPRES.

The Processor Clock Controller also generates the Main System Bus Clocks, MCKx, and a sub-division of this MCKx, MCKxDIVx. These clocks are distributed to peripherals, to the system bus and to matrices. See the table “Peripheral Identifiers” for Main System Bus Clocks used versus peripherals.

© 2026 Microchip Technology Inc. and its subsidiaries

Only one of CSS/CPCSS and PRES/CPPRES fields can be modified at a time. When one of these parameters is modified, no other modification can be performed on these fields as long as the MCKRDY/CPMCKRDY status flags are low.

Any modification in CSS/CPCSS and PRES/CPPRES fields must never lead to generate a MCK frequency that is greater than the maximum allowed system frequency. When changing the source clock of the system to a faster clock, the fields must be modified using the following order: PRES/CPPRES and then CSS/CPCSS. When changing the source clock of the system to a slower clock, the fields must be modified using the following order: CSS/CPCSS and then PRES/CPPRES.

If the destination clock does not exist, the switching is not performed. The CPU_CLK0 keeps running with the previous clock and the system must be reset to run correctly again.

20.5. Free-running Processor Clock

The Free-running Processor Clock (FCLK0) is used for sampling interrupts and clocking debug blocks. This clock ensures that interrupts can be sampled, and sleep events can be traced, while the processor is sleeping.

20.6. Peripheral and Generic Clock Controller

The PMC controls the clocks of the embedded peripherals by means of the Peripheral Control register (PMC_PCR). With this register, the user can enable and disable the different clocks used by the peripherals:

- Peripheral clocks (periph_clk[PID]), routed to every peripheral and derived from the corresponding MCKx. It is mandatory to enable this clock before using a peripheral.
- Generic clocks (GCLK[PID]), routed to selected peripherals only (refer to the Peripheral Identifiers table in section Peripherals). These clocks are independent of the core and bus clocks (CPU_CLK0, MCKx and periph_clk[PID]). They are generated by selection and division of available sources. The list of available source clocks depends on the peripheral. Refer to the description of each peripheral to know available sources and limitations to be applied to GCLK[PID] compared to periph_clk[PID].

To configure a peripheral's clocks, PMC_PCR.CMD must be written to '1' and PMC_PCR.PID must be written with the index of the corresponding peripheral. All other configuration fields must be correctly set.

To read the current clock configuration of a peripheral, PMC_PCR must be first accessed with PMC_PCR.CMD written to '0' and PMC_PCR.PID written with the index of the corresponding peripheral. This write does not modify the configuration of the peripheral. PMC_PCR can then be read to know the configuration status of the corresponding PID.

The status of the peripheral clock activity can be read in the Peripheral Clock Status registers (PMC_CSRx).

The status of the peripheral generic clock activity can be read in the Generic Clock Status registers (PMC_GCSRx).

When a peripheral or a generic clock is disabled, it is immediately stopped. These clocks are disabled after a reset. The source and the division ratio of generic clocks must not be modified while the peripheral is enabled. The generic clock configuration must be set before the peripheral is enabled.

To stop a peripheral clock, it is recommended that the system software wait until the peripheral has executed its last programmed operation before disabling the clock. This is to avoid data corruption or erroneous behavior of the system.

Note: The QSPI Controller generic clock must be set to twice the desired QSPI Controller frequency. Refer to the section "Peripheral Identifiers" for the QSPI generic clock ID.

© 2026 Microchip Technology Inc. and its subsidiaries

20.7. SysTick Clock

The SysTick calibration value is fixed to 25000, which allows the generation of a time base of 1 ms with SysTick clock to the maximum frequency divided by 8.

20.8. Programmable Clock Output Controller

The PMC controls three signals to be output on the external pins PCKx. Each signal can be independently programmed via the Programmable Clock registers (PMC_PCKx).

PCKx can be independently selected between MD_SLCK, TD_SLCK, MAINCK, MCK0 and most PLLxCKy by configuring PMC_PCKx.CSS. Each output signal can also be divided by 1 to 256 by configuring PMC_PCKx.PRES.

Each output signal can be enabled and disabled by writing a '1' to the corresponding bits PMC_SCER.PCKx and PMC_SCDR.PCKx, respectively. The status of the active programmable output clocks is given in PMC_SCSR.PCKx.

The status flag PMC_SR.PCKRDYx indicates that the clock configured through the PMC_PCKx registers is correctly established.

As the Programmable Clock Controller does not manage with glitch prevention when switching clocks, it is strongly recommended to disable PCKx before any configuration change and to re-enable it once the change is performed.

20.9. Ultra Low Power Modes and Fast Start-up

20.9.1. Wait Mode

When the system is in Wait mode, all clocks of the system except MD_SLCK are stopped. The source clock of all MCKx must be set to the main clock, and the source of the main clock must be set to main RC.

Prior to instructing the device to enter Wait mode:

Note: The sequence below only relates to the PMC. For the complete sequence to enter Wait mode, see "Wait Mode" in the section "Power Supply and Power Control".

1. Select Main RC as the source of MAINCK by configuring CKGR_MOR.MOSCSEL to '0'
2. Select MAINCK as the source of all MCKx by configuring PMC_CPU_CKR.CSS to '1' and PMC_CPU_CKR.CPCSS to '1'.
3. Disable the PLL if enabled and disable the main crystal oscillator by setting CKGR_MOR.MOSCXTEN to '0'.
4. Wait for two SLCK clock cycles.
5. Clear the internal wake-up sources.
6. Verify that none of the enabled external wake-up inputs (WKUP) hold an active polarity.

The system enters Wait mode by setting CKGR_MOR.WAITMODE. The PMC registers must not be accessed immediately after this access.

20.9.2. Fast Start-up

At exit from Wait mode, the device allows the processor to restart in several microseconds only if the C-code function that manages the Wait mode entry and exit is linked to and executed from on-chip SRAM.

The fast start-up time cannot be achieved if the first instruction after an exit is located in the embedded Flash.

If fast start-up is not required, or if the first instruction after exit from Wait mode is located in embedded Flash, see Start-up from Embedded Flash.

© 2026 Microchip Technology Inc. and its subsidiaries

A fast start-up occurs upon the detection of a programmed level on one of the wake-up inputs (WKUP) or upon an active alarm from the RTC and the RTT. The polarity of each of the wake-up inputs is programmable in the Wake-up Control register (PMC_WCR).

WARNING: The duration of the WKUPx pins active level must be greater than four MAINCK cycles.

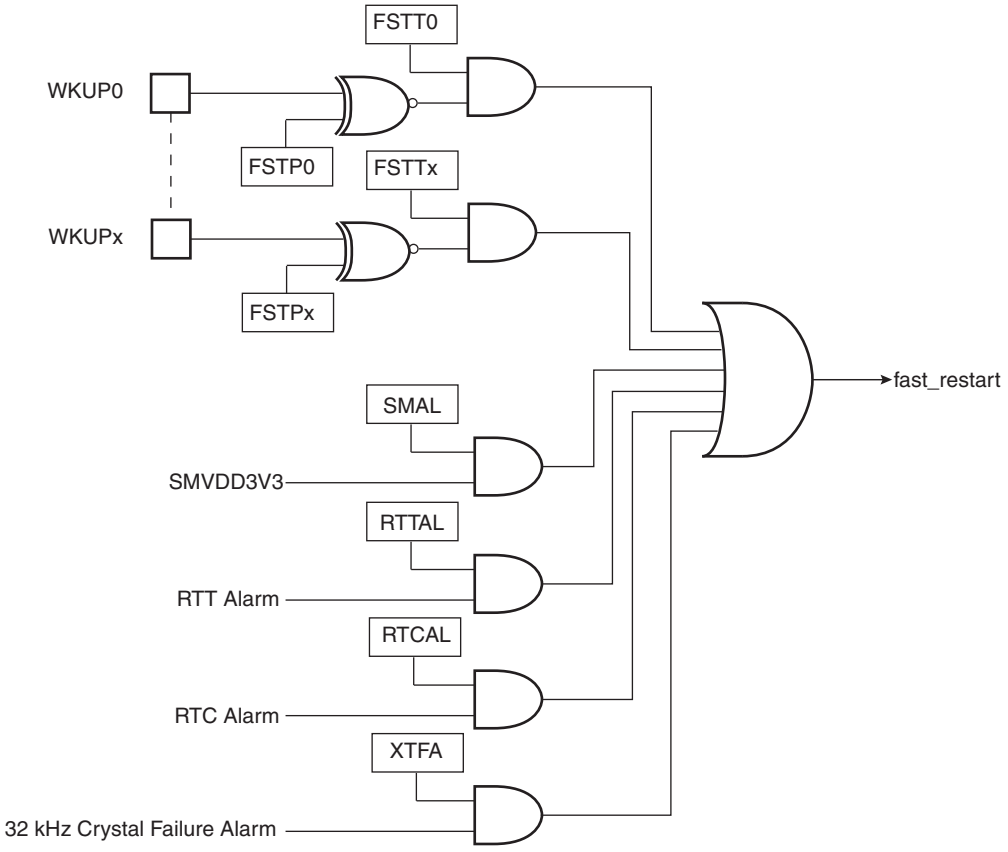
The fast start-up circuitry, as shown in the figure below, is fully asynchronous and provides a fast start-up signal to the PMC. As soon as the fast start-up signal is asserted, the Main RC oscillator restarts automatically.

When entering Wait mode, the embedded Flash can be placed in one of the low-power modes (Deep-Power-Down or Standby mode) with PMC_FSMR.FLPM. FLPM can be configured at any time and its value will be applied to the next Wait mode period.

The power consumption reduction is optimal when PMC_FSMR.FLPM is configured to '1' (Deep-Power-Down mode). If the field is configured to '0' (Standby mode), the power consumption is slightly higher than in Deep-power-down mode.

When PMC_FSMR.FLPM is configured to '2', the Wait mode Flash power consumption is equivalent to that of the Active mode when there is no read access on the Flash.

Figure 20-2. Fast Start-up Circuitry



Each wake-up input pin can be configured to generate a fast start-up event by setting the corresponding bits in PMC_WCR.

To configure a wake-up pin, a write access must be performed in PMC_WCR (CMD='1'). PID must be written with the ID of the wake-up pin, FSTP set to the polarity of the wake-up pin and EN set to enable/disable the wake-up pin.

© 2026 Microchip Technology Inc. and its subsidiaries

To read the configuration status of a wake-up pin, PMC_WCR.PID must be written with the ID of the wake-up pin and CMD set to '0'. Then the next read access to PMC_WCR sends the configuration status of the wake-up pin specified in PID.

Each alarm can be enabled to generate a fast start-up event by setting the corresponding bit in PMC_FSMR.

The user interface does not provide any status for fast start-up. The status can be read in the PIO Controller and the status registers of the RTC and the RTT.

20.10. Asynchronous Partial Wake-Up

20.10.1. Description

The asynchronous partial wake-up wakes up a peripheral in a fully asynchronous way when activity is detected on the external communication line. The asynchronous partial wake-up function automatically manages the peripheral clock. It reduces overall power consumption of the system by clocking peripherals only when needed.

Asynchronous partial wake-up can be enabled in Wait mode or in Active mode.

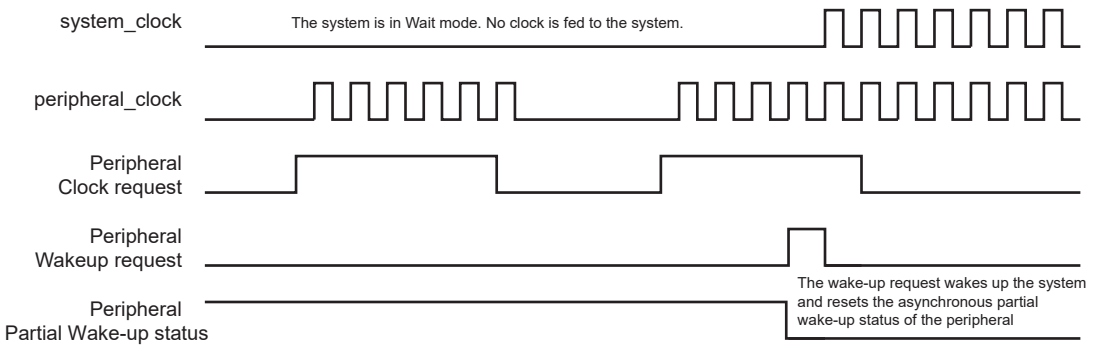
Only the following peripherals can be configured with asynchronous partial wake-up: FLEXCOM and ADC.

The peripheral selected for asynchronous partial wake-up must first be configured so that its clock is enabled. To do so, configure PMC_PCR.PID and write a '1 to PMC_PCR.EN.

20.10.2. Asynchronous Partial Wake-Up in Wait Mode

When an asynchronous clock request from a peripheral occurs, the PMC partially wakes up the system to feed the clock only to this peripheral. The rest of the system is not fed with the clock, thus optimizing power consumption. Finally, depending on user-configurable conditions, the peripheral either wakes up the whole system if these conditions are met or stops the peripheral clock until the next clock request. If a wake-up request occurs, asynchronous partial wake-up is automatically disabled until the user instructs the PMC to enable asynchronous partial wake-up. This is done by writing a '1' to SLPWKSR in the Partial Wake-Up Control register (PMC_SLPWKCR).

Figure 20-3. Asynchronous Partial Wake-Up Waveforms



20.10.2.1. Configuration Procedure

Before configuring asynchronous partial wake-up for a peripheral, check that PIDx in PMC_CSR is set. This ensures that the peripheral clock is enabled.

The steps to enable asynchronous partial wake-up for a peripheral are the following:

1. Check that PMC_SLPWKCR.ASR is set to '0' for the corresponding peripheral. This ensures that the peripheral has no activity in progress.
2. Enable asynchronous partial wake-up for the peripheral by writing a '1' to PMC_SLPWKCR.SLPWKSR.

© 2026 Microchip Technology Inc. and its subsidiaries

3. Check that PMC_SLPWKCR.ASR is set to '0'. This ensures that no activity has started during the enable phase.
4. In the PMC_SLPWKCR, asynchronous partial wake-up must be immediately disabled by writing a '0' to SLPWKSR. Wait for the end of peripheral activity before reinitializing the procedure. If the corresponding PIDx bit is set to '0', then the peripheral clock is disabled and the system can then be placed in Wait mode.

Before entering Wait mode, check that AIP in the Partial Wake-Up Activity In Progress register (PMC_SLPWK_AIPR) is cleared. This ensures that none of the peripherals is currently active.

Note: When asynchronous partial wake-up is enabled for a peripheral and the core is running (system not in Wait mode), the peripheral must not be accessed before a wake-up of the peripheral is performed.

20.10.3. Asynchronous Partial Wake-up in Active Mode

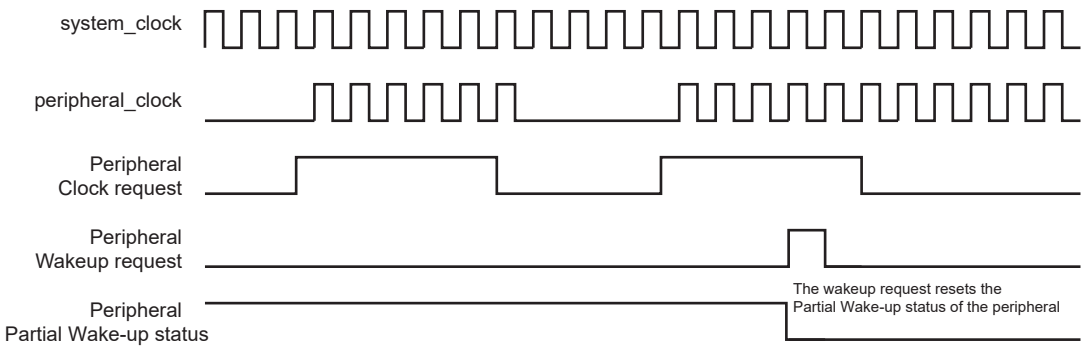
When the system is in Active mode, peripherals enabled for asynchronous partial wake-up have their respective clocks stopped until the peripherals request a clock. When a peripheral requests the clock, the PMC provides the clock without processor intervention.

The triggering of the peripheral clock request depends on conditions which can be configured for each peripheral. If these conditions are met, the peripheral asserts a request to the PMC. The PMC disables the Asynchronous Partial Wake-up mode of the peripheral and provides the clock to the peripheral until the user instructs the PMC to re-enable partial wake-up on the peripheral. This is done by setting PMC_SLPWKCR.SLPWKSR.

If the conditions are not met, the peripheral clears the clock request and the PMC stops the peripheral clock until the clock request is reasserted by the peripheral.

Note: Configuring Asynchronous Partial Wake-up in Active mode requires the same registers as in Wait mode.

Figure 20-4. Asynchronous Partial Wake-Up in Active Mode



20.10.3.1. Configuration Procedure

Before configuring the asynchronous partial wake-up function of a peripheral, check that PIDx in PMC_CSR is set. This ensures that the peripheral clock is enabled.

The steps to enable the asynchronous partial wake-up function of a peripheral are the following:

1. Check that PMC_SLPWKCR.ASR is set to '0'. This ensures that the peripheral has no activity in progress.
2. Enable the asynchronous partial wake-up function of the peripheral by writing a '1' to PMC_SLPWK_ER.SLPWKSR.
3. Check that PMC_SLPWKCR.ASR is set to '0'. This ensures that no activity has started during the enable phase.

© 2026 Microchip Technology Inc. and its subsidiaries

If an activity has started during the enable phase, the asynchronous partial wake-up function must be immediately disabled by writing a '0' to PMC_SLPWKCR.SLPWKSR. Wait for the end of peripheral activity before reinitializing the procedure.

20.11. Start-up from Embedded Flash

The inherent start-up time of the embedded Flash cannot provide a fast start-up of the system.

If system fast start-up time is not required, the first instruction after a Wait mode exit can be located in the embedded Flash. Under these conditions, prior to entering Wait mode, the Flash controller must be programmed to perform access in 0 wait-state (refer to the embedded Flash Controller section).

The procedure and conditions to enter Wait mode and the circuitry to exit Wait mode are strictly the same as fast start-up (see Fast Start-up).

20.12. Main Crystal Oscillator Failure Detection

The Main crystal oscillator failure detector monitors the Main crystal oscillator against the Slow RC oscillator and provides an automatic switchover of the MAINCK source to the Main RC oscillator in case of failure detection.

The failure detector can be enabled or disabled by configuring CKGR_MOR.CFDEN. It cannot be enabled if the Main crystal oscillator is disabled. It must be disabled before disabling the Main crystal oscillator.

It is also disabled in either of the following cases:

- after a VDDCORE reset
- when the Main crystal oscillator is disabled (MOSCXTEN = 0)

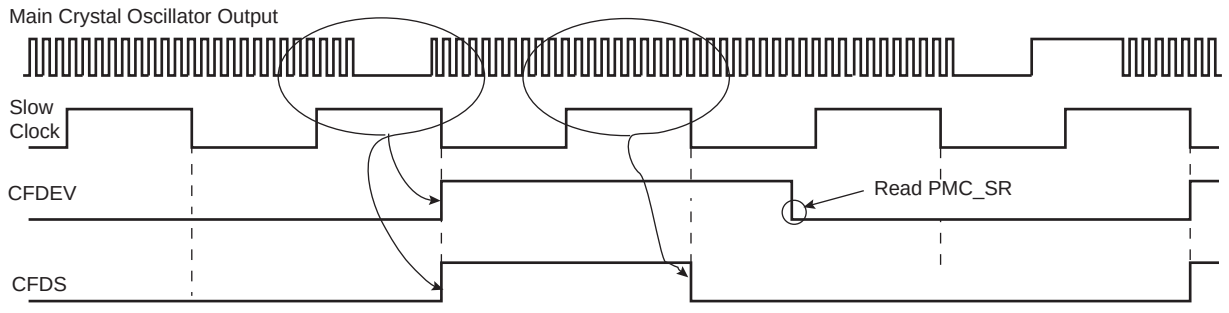
A failure is detected by means of a counter incrementing on the Main crystal oscillator output and detection logic is triggered by the Slow RC oscillator which is automatically enabled when CFDEN = 1.

The counter is cleared when the Slow RC oscillator clock signal is low and enabled when the signal is high. Thus, the failure detection time is one Slow RC oscillator period. If, during the high level period of the Slow RC oscillator clock signal, less than eight Main crystal oscillator clock periods have been counted, then a failure is reported. Note that when enabling the failure detector, up to two cycles of the Slow RC oscillator are needed to detect a failure of the Main crystal oscillator.

If a failure of Main crystal oscillator is detected, PMC_SR.CFDEV and PMC_SR.FOS both indicate a failure event. PMC_SR.CFDEV is cleared on read of PMC_SR, and PMC_SR.FOS is cleared by writing a '1' to the FOCLR bit in the Fault Output Clear register (PMC_FOCR).

Only PMC_SR.CFDEV can generate an interrupt if the corresponding interrupt source is enabled in PMC_IER. The current status of the clock failure detection can be read at any time from PMC_SR.CFDS.

Figure 20-5. Clock Failure Detection Example



Note: Ratio of clock periods is for illustration purposes only.

© 2026 Microchip Technology Inc. and its subsidiaries

If the Main crystal oscillator is selected as the source clock of MAINCK (CKGR_MOR.MOSCSEL = 1), and if the MCK0 source is a PLL, a clock failure detection automatically forces MAINCK to be the source clock for MCK0. Then, regardless of the PMC configuration, a clock failure detection automatically forces the Main RC oscillator to be the source clock for MAINCK. If the Main RC oscillator is disabled when a clock failure detection occurs, it is automatically re-enabled by the clock failure detection mechanism.

Two Slow RC oscillator clock cycles are necessary to detect and switch from the Main crystal oscillator to the Main RC oscillator if the source of MCK0 is MAINCK, or three Slow RC oscillator clock cycles if the source of MCK0 is a PLL.

A clock failure detection activates a fault output that is connected to the Pulse Width Modulator (PWM) Controller. With this connection, the PWM controller is able to force its outputs and to protect the driven device, if a clock failure is detected.

20.13. 32.768 kHz Crystal Oscillator Frequency Monitor

The frequency of the 32.768 kHz crystal oscillator can be monitored by configuring CKGR_MOR.XT32KFME. Prior to enabling the monitoring, the 32.768 kHz crystal oscillator must be started and its start-up time be elapsed. Refer to details on the Slow clock generator in the section “Supply Controller (SUPC)”.

An error flag (PMC_SR.XT32KERR) is asserted when the 32.768 kHz crystal oscillator frequency is out of its nominal frequency value (i.e., 32.768 kHz). The error flag can be cleared only if the monitoring is disabled.

The frequency drift is computed with the Main RC oscillator. The permitted drift of the crystal is 10000 ppm (1%), which allows any standard crystal to be used.

The monitored clock frequency is declared invalid if at least four consecutive 32.768 kHz crystal oscillator clock period measurement results are over the nominal period. Note that modifying the trimming values of the Main RC oscillator (PMC_OCR) may impact the monitor accuracy and lead to inappropriate failure detection.

The error flag can be defined as an interrupt source of the PMC by setting PMC_IER.XT32KERR. This flag is also routed to the Reset Controller (RSTC) and may generate a reset of the device.

A second frequency monitor is also available in the SUPC backup domain, please refer to SUPC section for more details.

20.14. MCK0 Frequency Monitor

The frequency of MCK0 can be monitored with the Main RC oscillator. This monitoring can only be performed if the MCK0 frequency is at least 3 times faster than the embedded Main RC oscillator. This function is enabled by writing a ‘1’ to CKGR_MOR.BMCKRST or PMC_IER.MCKMON.

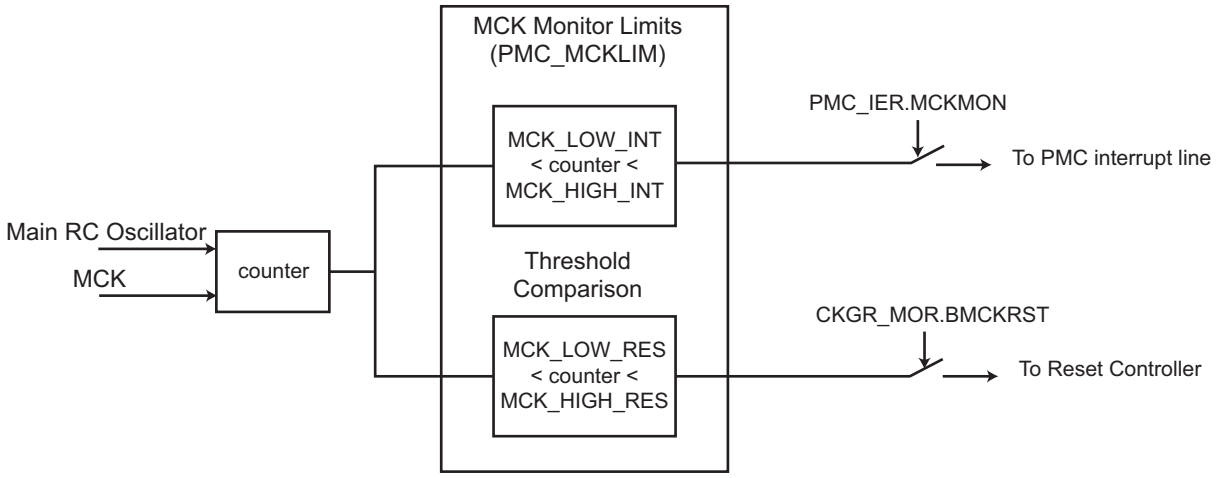
An error on the MCK0 frequency can lead to a PMC interrupt or a reset of the system (refer to section “Reset Controller (RSTC)”).

When the corresponding PMC interrupt is enabled, the status of the MCK0 monitoring can be read on the PMC_SR.MCKMON. This status is cleared on read.

Once enabled, the monitor continuously counts the number of MCK0 cycles within 15 cycles of the embedded Main RC oscillator. The result is then compared to threshold values defined in the PMC_MCKLIM register. Two levels of thresholds can be defined: a first couple of values defines thresholds to generate an interrupt and a second couple of values defines thresholds to generate a reset of the system.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 20-6. MCK0 Frequency Monitor



20.15. Recommended Programming Sequence

Steps given below are a high-level PMC programming sequence to achieve CPU and Bus clock configuration. This sequence does not provide any specific use cases regarding clock source configuration, clock division, etc.

1. If the Main crystal oscillator is not required, the PLL can be directly configured (Step 5.) else this oscillator must be started (Step 2.).
2. Verify the existence and frequency value of the Main crystal oscillator following the sequence defined in Main Frequency Counter
3. If the Main crystal oscillator is enabled and valid, the source of MAINCK can be switched to the Main crystal oscillator by writing CKGR_MOR.MOSCSEL to 1 else the PLL can be directly configured.
4. Wait for the end of the MAINCK source switching by either polling the MOSCSELS or setting the corresponding interrupt
5. Configure the PLLs by following the setup defined in Divider and Phase Lock Loop Programming (if not required, proceed to Step 6.):
6. Configure the MCK0DIV, MCK0DIV2 and MCK1DIV division ratio by setting PMC_CPU_CKR.RATIO_MCK0DIV, PMC_CPU_CKR.RATIO_MCK0DIV2 and PMC_CPU_CKR.RATIO_MCK1DIV.
7. Select the division ratio of CPU_CLK0 by setting PMC_CPU_CKR.PRES.
   PRES is used to define the CPU_CLK0 and MCK0 prescaler. The user can choose between different values (1, 2, 3, 4, 8, 16, 32, 64). Prescaler output is the selected clock source frequency divided by the PRES value.
8. Wait for the end of the CPU_CLK0 ratio switching by either polling the MCKRDY or setting the corresponding interrupt.
9. Select the source clock of CPU_CLK0 by setting PMC_CPU_CKR.CSS.
   CSS is used to select the clock source of MCK0 and CPU_CLK0. By default, the selected clock source is MAINCK.
10. Wait for the end of the CPU_CLK0 source switching by either polling the MCKRDY or setting the corresponding interrupt.
    PMC_CPU_CKR must not be programmed in a single write operation.
    Reconfiguring PRES and CSS fields must always be done by following the right order of operation described above (steps 6 to 11)

© 2026 Microchip Technology Inc. and its subsidiaries

11. Configure the Programmable clocks (PCKx):
    PCKx are controlled via registers PMC_SCER, PMC_SCDR and PMC_SCSR.

    PCKx can be enabled and/or disabled via PMC_SCER and PMC_SCDR. Three PCKx can be used. PMC_SCSR indicates which PCKx is enabled. By default all PCKx are disabled.

    PMC_PCKx registers are used to configure PCKx as described in SysTick Clock.
12. Enable the peripheral and generic clocks:
    Once all of the previous steps have been completed, the peripheral and generic clocks can be configured via PMC_PCR as described in Peripheral and Generic Clock Controller.

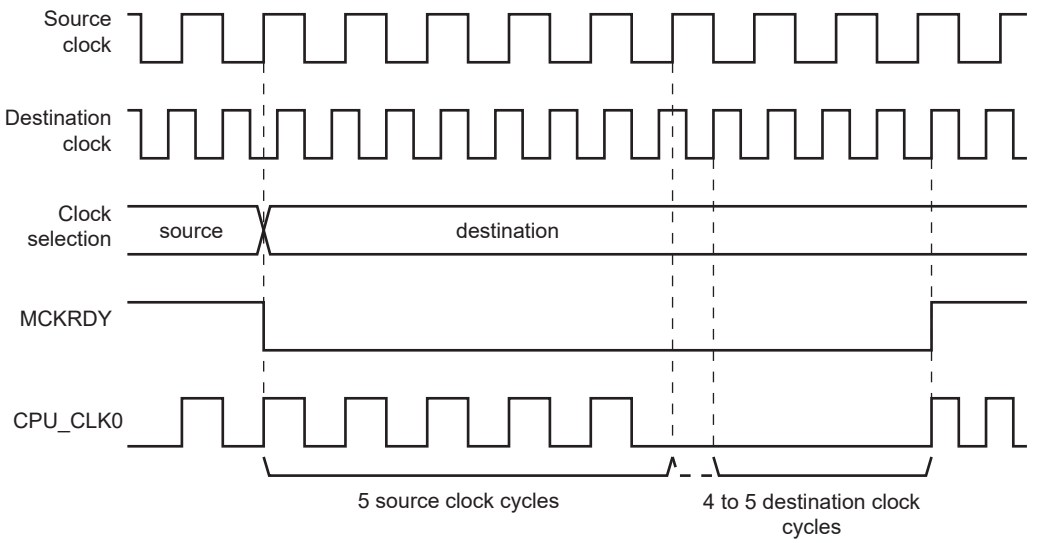
20.16. Clock Switching Details

20.16.1. CPU Clock Switching Timings

The glitch-free clock switcher implemented to control the sources of CPU_CLK0 performs clock switching in 5 clock cycles of the currently used clock plus 5 cycles of the target clock.

The clock switching is effective once MCKRDY rises. Refer to the figure below.

Figure 20-7. Switch CPU Clock (CPU_CLK0) from Source Clock to Destination Clock



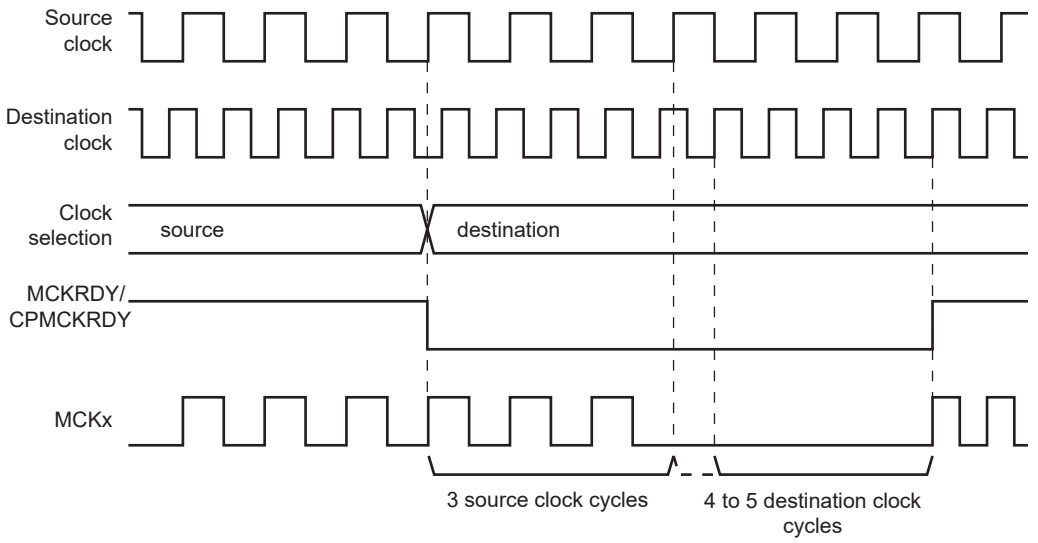
20.16.2. Main System Bus Clocks Switching Timings

The glitch-free clock switcher implemented to control the sources of MCKx performs the clock switching in 3 clock cycles of the currently used clock plus 3 cycles of the target clock.

The clock switching is effective once MCKXRDY/CPMCKRDY rise. Refer to the figure below.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 20-8. Switch Domain Clock (MCKx) from Source Clock to Destination Clock



20.17. Register Write Protection

To prevent any single software error from corrupting PMC behavior, certain registers in the address space can be write-protected by setting the WPEN bit or the WPITEN bit in the PMC Write Protection Mode Register (PMC_WPMR).

If a write access to a write-protected register is detected, the WPVS flag in the PMC Write Protection Status Register (PMC_WPSR) is set and the field WPVSRC indicates the register in which the write access has been attempted.

The WPVS bit is automatically cleared after reading the PMC_WPSR.

The following registers are write-protected when the WPEN bit is set in PMC_WPMR:

- PMC System Clock Enable Register
- PMC System Clock Disable Register
- PMC PLL Control Register 0
- PMC PLL Control Register 1
- PMC PLL Spread Spectrum Register
- PMC PLL Update Register
- PMC Clock Generator Main Oscillator Register
- PMC Clock Generator Main Clock Frequency Register
- PMC CPU Clock Register
- PMC Programmable Clock Register
- PMC Programmable Clock Register
- PMC Fast Start-Up Mode Register
- PMC Wake-Up Control Register
- PMC Peripheral Control Register
- PMC Oscillator Calibration Register
- PMC Partial Wake-Up Control Register
- PMC MCK0 Monitor Limits Register

The following interrupt registers are write-protected when the WPITEN bit is set in PMC_WPMR:

© 2026 Microchip Technology Inc. and its subsidiaries

- PMC Interrupt Enable Register
- PMC Interrupt Disable Register
- PMC PLL Interrupt Enable Register
- PMC PLL Interrupt Disable Register

© 2026 Microchip Technology Inc. and its subsidiaries

20.18. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	PMC_SCER	31:24								
		23:16	CPKEY[3:0]						CPBMCK	
		15:8					PCK3	PCK2	PCK1	PCK0
		7:0								
0x04	PMC_SCDR	31:24								
		23:16	CPKEY[3:0]						CPBMCK	
		15:8					PCK3	PCK2	PCK1	PCK0
		7:0								
0x08	PMC_SCSR	31:24								
		23:16							CPBMCK	
		15:8					PCK3	PCK2	PCK1	PCK0
		7:0								CPU_CLK0S
0x0C	PMC_PLL_CTRL0	31:24	ENLOCK	ENPLLO1	ENPLLO0	ENPLL	PLLMS			
		23:16					DIVPMC1[7:4]			
		15:8	DIVPMC1[3:0]							
		7:0	DIVPMC0[7:0]							
0x10	PMC_PLL_CTRL1	31:24								
		23:16								
		15:8		MUL[14:8]						
		7:0	MUL[7:0]							
0x14	PMC_PLL_CTRL2	31:24								
		23:16			FRACR[21:16]					
		15:8	FRACR[15:8]							
		7:0	FRACR[7:0]							
0x18	PMC_PLL_SSR	31:24				ENSPREAD				
		23:16	NSTEP[7:0]							
		15:8	STEP[15:8]							
		7:0	STEP[7:0]							
0x1C	PMC_PLL_ACR	31:24	LOOP_FILTER[7:0]							
		23:16	LOCK_THR[7:0]							
		15:8					CONTROL[11:8]			
		7:0	CONTROL[7:0]							
0x20	PMC_PLL_UPDT	31:24								
		23:16	STUPTIM[7:0]							
		15:8								UPDATE
		7:0					ID[3:0]			
0x24	CKGR_MOR	31:24					BMCKRST	XT32KFME	CFDEN	MOSCSEL
		23:16	KEY[7:0]							
		15:8	MOSCXTST[7:0]							
		7:0					MOSCRCEN	WAITMODE	MOSCXTBY	MOSCXTEN
0x28	CKGR_MCFR	31:24								CCSS
		23:16				RCMEAS				MAINFRDY
		15:8	MAINF[15:8]							
		7:0	MAINF[7:0]							
0x2C	PMC_CPU_CKR	31:24						RATIO_MCK0 DIV2	RATIO_MCK1 DIV	RATIO_MCK0 DIV
		23:16	CPPRES[3:0]					CPCSS[2:0]		
		15:8								
		7:0		PRES[2:0]					CSS[1:0]	
0x30 ... 0x43	Reserved									
0x44	PMC_PCK0	31:24								
		23:16								
		15:8	PRES[7:0]							
		7:0				CSS[4:0]				

© 2026 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x48	PMC_PCK1	31:24								
		23:16								
		15:8	PRES[7:0]							
		7:0				CSS[4:0]				
0x4C	PMC_PCK2	31:24								
		23:16								
		15:8	PRES[7:0]							
		7:0				CSS[4:0]				
0x50 ... 0x63	Reserved									
0x64	PMC_IER	31:24								
		23:16	MCKMON		XT32KERR			CFDEV	MOSCRCS	MOSCSELS
		15:8					PCKRDY3	PCKRDY2	PCKRDY1	PCKRDY0
		7:0				CPMCKRDY	MCKRDY			MOSCXTS
0x68	PMC_IDR	31:24								
		23:16	MCKMON		XT32KERR			CFDEV	MOSCRCS	MOSCSELS
		15:8					PCKRDY3	PCKRDY2	PCKRDY1	PCKRDY0
		7:0				CPMCKRDY	MCKRDY			MOSCXTS
0x6C	PMC_SR	31:24							PLL_INT	GCLKRDY
		23:16	MCKMON		XT32KERR	FOS	CFDS	CFDEV	MOSCRCS	MOSCSELS
		15:8					PCKRDY3	PCKRDY2	PCKRDY1	PCKRDY0
		7:0	OSCSELS			CPMCKRDY	MCKRDY			MOSCXTS
0x70	PMC_IMR	31:24								
		23:16	MCKMON		XT32KERR			CFDEV	MOSCRCS	MOSCSELS
		15:8					PCKRDY3	PCKRDY2	PCKRDY1	PCKRDY0
		7:0				CPMCKRDY	MCKRDY			MOSCXTS
0x74	PMC_FSMR	31:24								
		23:16	FFLPM	FLPM[1:0]		LPM	SMAL	XTFA	RTCAL	RTTAL
		15:8								
		7:0								
0x78	PMC_WCR	31:24								CMD
		23:16							WIP	EN
		15:8								
		7:0					WKPIONB[3:0]			
0x7C	PMC_FOCR	31:24								
		23:16								
		15:8								
		7:0								FOCLR
0x80 ... 0x83	Reserved									
0x84	PMC_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0							WPITEN	WPEN
0x88	PMC_WPSR	31:24								
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0								WPVS
0x8C	PMC_PCR	31:24	CMD		GCLKEN	EN	GCLKDIV[7:4]			
		23:16	GCLKDIV[3:0]							
		15:8				GCLKCSS[4:0]				
		7:0		PID[6:0]						
0x90	PMC_OCR	31:24								
		23:16	SEL12	CAL12[6:0]						
		15:8								
		7:0								

© 2026 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x94	PMC_SLPWK_AIPR	31:24								
		23:16								
		15:8								
		7:0								AIP
0x98	PMC_SLPWKCR	31:24				SLPWKSR				
		23:16								ASR
		15:8				CMD				
		7:0		PID[6:0]						
0x9C ... 0x9F	Reserved									
0xA0	PMC_MCKLIM	31:24	MCK_HIGH_RES[7:0]							
		23:16	MCK_LOW_RES[7:0]							
		15:8	MCK_HIGH_IT[7:0]							
		7:0	MCK_LOW_IT[7:0]							
0xA4	PMC_CSR0	31:24	PID31	PID30		PID28			PID25	PID24
		23:16	PID23						PID17	PID16
		15:8	PID15	PID14	PID13	PID12	PID11	PID10	PID9	
		7:0								
0xA8	PMC_CSR1	31:24					PID59		PID57	
		23:16	PID55		PID53		PID51		PID49	
		15:8								
		7:0	PID39	PID38	PID37	PID36	PID35	PID34	PID33	PID32
0xAC	PMC_CSR2	31:24					PID91	PID90	PID89	PID88
		23:16	PID87		PID85				PID81	PID80
		15:8	PID79	PID78			PID75			
		7:0								
0xB0	PMC_CSR3	31:24								
		23:16								
		15:8								
		7:0								PID96
0xB4 ... 0xC3	Reserved									
0xC4	PMC_GCSR0	31:24	GPID31							GPID24
		23:16	GPID23							GPID16
		15:8	GPID15	GPID14	GPID13	GPID12	GPID11	GPID10	GPID9	
		7:0								
0xC8	PMC_GCSR1	31:24								
		23:16								
		15:8								
		7:0			GPID37			GPID34		
0xCC	PMC_GCSR2	31:24						GPID90	GPID89	
		23:16	GPID87							
		15:8	GPID79				GPID75			
		7:0								
0xD0	PMC_GCSR3	31:24								
		23:16								
		15:8								
		7:0								GPID96
0xD4 ... 0xE3	Reserved									
0xE4	PMC_PLL_IER	31:24								
		23:16						UNLOCK2	UNLOCK1	UNLOCK0
		15:8								
		7:0						LOCK2	LOCK1	LOCK0

© 2026 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xE8	PMC_PLL_IDR	31:24								
		23:16						UNLOCK2	UNLOCK1	UNLOCK0
		15:8								
		7:0						LOCK2	LOCK1	LOCK0
0xEC	PMC_PLL_IMR	31:24								
		23:16						UNLOCK2	UNLOCK1	UNLOCK0
		15:8								
		7:0						LOCK2	LOCK1	LOCK0
0xF0	PMC_PLL_ISR0	31:24								
		23:16						UNLOCK2	UNLOCK1	UNLOCK0
		15:8								
		7:0						LOCK2	LOCK1	LOCK0
0xF4	PMC_PLL_ISR1	31:24								
		23:16						OVR2	OVR1	OVR0
		15:8								
		7:0						UDR2	UDR1	UDR0

© 2026 Microchip Technology Inc. and its subsidiaries

20.18.1. PMC System Clock Enable Register

Name: PMC_SCER
Offset: 0x0000
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the PMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	CPKEY[3:0]						CPBMCK	
Access	W	W	W	W			W	
Reset	–	–	–	–			–	

Bit	15	14	13	12	11	10	9	8
					PCK3	PCK2	PCK1	PCK0
Access					W	W	W	W
Reset					–	–	–	–

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 23:20 – CPKEY[3:0] Clocks Enable Key

Value	Name	Description
0xA	PASSWD	This field must be written to 0xA to validate CPBMCK.

Bit 17 – CPBMCK Main System Bus Clocks Enable

Value	Description
0	No effect.
1	Enables the Main System Bus Clocks if CPKEY = 0xA.

Bits 8, 9, 10, 11 – PCKx Programmable Clock x Output Enable

Value	Description
0	No effect.
1	Enables the corresponding Programmable Clock output.

© 2026 Microchip Technology Inc. and its subsidiaries

20.18.2. PMC System Clock Disable Register

Name: PMC_SCDR
Offset: 0x0004
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the PMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	CPKEY[3:0]						CPBMCK	
Access	W	W	W	W			W	
Reset	–	–	–	–			–	

Bit	15	14	13	12	11	10	9	8
					PCK3	PCK2	PCK1	PCK0
Access					W	W	W	W
Reset					–	–	–	–

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 23:20 – CPKEY[3:0] Clocks Enable Key

Value	Name	Description
0xA	PASSWD	This field must be written to 0xA to validate CPBMCK.

Bit 17 – CPBMCK Main System Bus Clocks Disable

Value	Description
0	No effect.
1	Disables the Main System Bus Clocks if CPKEY = 0xA.

Bits 8, 9, 10, 11 – PCKx Programmable Clock x Output Disable

Value	Description
0	No effect.
1	Disables the corresponding Programmable Clock output.

© 2026 Microchip Technology Inc. and its subsidiaries

20.18.3. PMC System Clock Status Register

Name: PMC_SCSR
Offset: 0x0008
Reset: 0x00000003
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							CPBMCK	
Access							R	
Reset							0	

Bit	15	14	13	12	11	10	9	8
					PCK3	PCK2	PCK1	PCK0
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
								CPU_CLK0S
Access								R
Reset								1

Bit 17 – CPBMCK Main System Bus Clocks Status

Value	Description
0	The Main System Bus Clocks are disabled.
1	The Main System Bus Clocks are enabled.

Bits 8, 9, 10, 11 – PCKx Programmable Clock x Output Status

Value	Description
0	The corresponding Programmable Clock output is disabled.
1	The corresponding Programmable Clock output is enabled.

Bit 0 – CPU_CLK0S CPU_CLK Status

© 2026 Microchip Technology Inc. and its subsidiaries

20.18.4. PMC PLL Control Register 0

Name: PMC_PLL_CTRL0
Offset: 0x000C
Reset: 0x00000000
Property: Read/Write

All fields defined here are applied to the PLL defined by the last ID field written in PMC_PLL_UPDT.

Bit	31	30	29	28	27	26	25	24
	ENLOCK	ENPLLO1	ENPLLO0	ENPLL	PLLMS			
Access	R/W	R/W	R/W	R/W	R/W			
Reset	0	0	0	0	0			

Bit	23	22	21	20	19	18	17	16
					DIVPMC1[7:4]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DIVPMC1[3:0]							
Access	R/W	R/W	R/W	R/W				
Reset	0	0	0	0				

Bit	7	6	5	4	3	2	1	0
	DIVPMC0[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 31 – ENLOCK Enable PLL Lock

Value	Description
0	The lock signal sent by the PLL is ignored. The PLL is considered as locked once the start-up time defined by PMC_PLL_UPDT.STUPTIM has elapsed.
1	The PLL is considered as locked once the start-up time defined by PMC_PLL_UPDT.STUPTIM has elapsed and the lock signal sent by the PLL has risen.

Bit 30 – ENPLLO1 Enable PLL Clock Output 1 (PLLA only)

Value	Description
0	The output clock 1 generated by the PLL is disabled (DIVPMC1 must be set to 0 to get the PLL output clock 1 disabled).
1	The output clock 1 generated by the PLL is active.

Bit 29 – ENPLLO0 Enable PLL Clock Output 0

Value	Description
0	The output clock 0 generated by the PLL is disabled (DIVPMC0 must be set to 0 to get the PLL output clock 0 disabled).
1	The output clock 0 generated by the PLL is active.

Bit 28 – ENPLL Enable PLL

Value	Description
0	The PLL is off.
1	The PLL is on.

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 27 – PLLMS PLL Multiplexer Select

Value	Description
0	PLL Multiplexer select input is set to '0'.
1	PLL Multiplexer select input is set to '1'.

Bits 19:12 – DIVPMC1[7:0] Divider for PMC Output 1 (PLLA only)
Specifies the division ratio applied to the internal PLL clock to generate PLLACK1.

Bits 7:0 – DIVPMC0[7:0] Divider for PMC Output 0
Specifies the division ratio applied to the internal PLL clock to generate PLLACK0, PLLBCK and PLLCCK.

© 2026 Microchip Technology Inc. and its subsidiaries

20.18.5. PMC PLL Control Register 1

Name: PMC_PLL_CTRL1
Offset: 0x0010
Reset: 0x00000000
Property: Read/Write

All fields defined here are applied to the PLL defined by the last ID field written in PMC_PLL_UPDT.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		MUL[14:8]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MUL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 14:0 – MUL[14:0] Multiplier Factor Value for PLLA, B and C
For PLLA, bits [14:0] are active.
For PLLB and C, only bits [7:0] are active.

© 2026 Microchip Technology Inc. and its subsidiaries

20.18.6. PMC PLL Control Register 2

Name: PMC_PLL_CTRL2
Offset: 0x0014
Reset: 0x00000000
Property: Read/Write

All fields defined here are applied to the PLL defined by the last ID field written in PMC_PLL_UPDT.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			FRACR[21:16]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	FRACR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FRACR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 21:0 – FRACR[21:0] Fractional Loop Divider Setting

© 2026 Microchip Technology Inc. and its subsidiaries

20.18.7. PMC PLL Spread Spectrum Register

Name: PMC_PLL_SSR
Offset: 0x0018
Reset: 0x00000000
Property: Read/Write

All fields defined here are applied to the PLL defined by the last ID field written in the PMC_PLL_UPDT register.

Bit	31	30	29	28	27	26	25	24
				ENSPREAD				
Access				R/W				
Reset				0				

Bit	23	22	21	20	19	18	17	16
	NSTEP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	STEP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	STEP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 28 – ENSPREAD Spread Spectrum Enable

Value	Description
0	The spread spectrum is not applied to the PLL.
1	The spread spectrum is applied to the PLL.

Bits 23:16 – NSTEP[7:0] Spread Spectrum Number of Step
Specifies how many times STEP is applied to the ratio of the PLL.

Bits 15:0 – STEP[15:0] Spread Spectrum Step Size
When the spread spectrum is active, this field defines the step size that will be applied the PMC_PLL_CTRL2.FRACR factor. The step is applied on the LSB of PMC_PLL_CTRL2.FRACR.

© 2026 Microchip Technology Inc. and its subsidiaries

20.18.8. PMC PLL Analog Control Register

Name: PMC_PLL_ACR
Offset: 0x001C
Reset: See the table below.
Property: Read/Write

All fields defined here are applied to the PLL defined by the last ID field written in PMC_PLL_UPDT.

Register reset values are:

PLL Name	PMC_PLL_ACR Reset Value
PLLA	0x00020058
PLLB	0x35020058
PLLC	0x35020058

Bit	31	30	29	28	27	26	25	24
	LOOP_FILTER[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								

Bit	23	22	21	20	19	18	17	16
	LOCK_THR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								

Bit	15	14	13	12	11	10	9	8
					CONTROL[11:8]			
Access					R/W	R/W	R/W	R/W
Reset								

Bit	7	6	5	4	3	2	1	0
	CONTROL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								

Bits 31:24 – LOOP_FILTER[7:0] LOOP Filter Selection

Bits 23:16 – LOCK_THR[7:0] PLL Lock Threshold Value Selection

Bits 11:0 – CONTROL[11:0] PLL CONTROL Value Selection

© 2026 Microchip Technology Inc. and its subsidiaries

20.18.9. PMC PLL Update Register

Name: PMC_PLL_UPDT
Offset: 0x0020
Reset: 0x00030000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	STUPTIM[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	1	1

Bit	15	14	13	12	11	10	9	8
								UPDATE
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
					ID[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 23:16 – STUPTIM[7:0] Start-up Time
The start-up time is defined as a number of MD_SLCK cycles and is the same for all PLLs.

Value	Description
0	Only the lock of the PLL is considered to know the lock status of the PLL. If the lock of the PLL is not enabled, the lock never rises.
Other values	If PMC_PLL_CTRL0.ENLOCK is low, specifies the start-up time of the PLL. If PMC_PLL_CTRL0.ENLOCK is high, specifies how long the LOCK signal of the PLL is masked before being read.

Bit 8 – UPDATE PLL Setting Update (write-only)

Value	Description
0	No effect.
1	The PLL configuration written in PMC_PLL_CTRL0 and PMC_PLL_CTRL1 are applied to the PLL defined by the last ID written in the PMC_PLL_UPDT register.

Bits 3:0 – ID[3:0] PLL ID
When writing a PLL control register (PMC_PLL_CTRLx), this ID specifies which PLL is impacted by written fields.
When reading a PLL control register (PMC_PLL_CTRLx), this ID specifies which PLL fields are read.

© 2026 Microchip Technology Inc. and its subsidiaries

20.18.10. PMC Clock Generator Main Oscillator Register

Name: CKGR_MOR
Offset: 0x0024
Reset: 0x00000008
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the PMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
					BMCKRST	XT32KFME	CFDEN	MOSCSEL
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0
Bit	23	22	21	20	19	18	17	16
	KEY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	MOSCXTST[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
					MOSCRCEN	WAITMODE	MOSCXTBY	MOSCXTEN
Access					R/W	R/W	R/W	R/W
Reset					1	0	0	0

Bit 27 – BMCKRST Bad MCK0 Clock Reset Enable

Value	Description
0	An MCK0 clock failure detection cannot reset the system.
1	An MCK0 clock failure detection can reset the system if the reset controller is configured accordingly.

Bit 26 – XT32KFME 32.768 kHz Crystal Oscillator Frequency Monitoring Enable

Value	Description
0	The 32.768 kHz crystal oscillator frequency monitoring is disabled.
1	The 32.768 kHz crystal oscillator frequency monitoring is enabled.

Bit 25 – CFDEN Clock Failure Detector Enable

Value	Description
0	The clock failure detector is disabled.
1	The clock failure detector is enabled.

Bit 24 – MOSCSEL Main Clock Oscillator Selection

Value	Description
0	The Main RC oscillator is selected.
1	The Main crystal oscillator is selected.

Bits 23:16 – KEY[7:0] Write Access Password

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Name	Description
0x37	PASSWD	Writing any other value in this field aborts the write operation. Always reads as 0.

Bits 15:8 – MOSCXTST[7:0] Main Crystal Oscillator Start-up Time
Specifies the number of MD_SLCK cycles multiplied by 8 for the main crystal oscillator start-up time.

Bit 3 – MOSCRCEN Main RC Oscillator Enable
When MOSCRCEN is set, the MOSCRCS flag is set once the Main RC oscillator start-up time is achieved.

Value	Description
0	The Main RC oscillator is disabled.
1	The Main RC oscillator is enabled.

Bit 2 – WAITMODE Wait Mode Command (write-only)

Value	Description
0	No effect.
1	Puts the device in Wait mode.

Bit 1 – MOSCXTBY 12 to 48 MHz Crystal Oscillator Bypass
When MOSCXTBY = 0, the MOSCXTS flag must be read at 0 in PMC_SR before enabling the crystal oscillator (MOSCXTEN = 1).
When MOSCXTBY = 1, the MOSCXTS flag in PMC_SR is automatically set to 1.
Clearing MOSCXTEN and MOSCXTBY bits resets the MOSCXTS flag.

Value	Description
0	No effect.
1	The 12 to 48 MHz crystal oscillator is bypassed. MOSCXTEN must be cleared. An external clock must be connected on XIN.

Bit 0 – MOSCXTEN Main Crystal Oscillator Enable
A crystal must be connected between XIN and XOUT.
When MOSCXTEN is set, the MOSCXTS flag is set once the Main crystal oscillator start-up time is achieved.

Value	Description
0	The Main crystal oscillator is disabled.
1	The Main crystal oscillator is enabled. MOSCXTBY must be cleared.

© 2026 Microchip Technology Inc. and its subsidiaries

20.18.11. PMC Clock Generator Main Clock Frequency Register

Name: CKGR_MCFR
Offset: 0x0028
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the PMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
								CCSS
Access								R/W
Reset								0

Bit	23	22	21	20	19	18	17	16
				RCMEAS				MAINFRDY
Access				R/W				R/W
Reset				0				0

Bit	15	14	13	12	11	10	9	8
	MAINF[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MAINF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 24 – CCSS Counter Clock Source Selection

Value	Description
0	The measured clock of the MAINF counter is the Main RC oscillator.
1	The measured clock of the MAINF counter is the Main crystal oscillator.

Bit 20 – RCMEAS RC Oscillator Frequency Measure (write-only)
The measurement is performed on the main frequency (i.e., not limited to the Main RC oscillator only). If the source of MAINCK is the Main crystal oscillator, the restart of measurement may not be required because of the stability of crystal oscillators.

Value	Description
0	No effect.
1	Restarts measuring of the frequency of MAINCK. MAINF carries the new frequency as soon as a low-to-high transition occurs on the MAINFRDY flag.

Bit 16 – MAINFRDY Main Clock Frequency Measure Ready
To ensure that a correct value is read on the MAINF field, the MAINFRDY flag must be read at '1' then another read access must be performed on the register to get a stable value on the MAINF field.

Value	Description
0	MAINF value is not valid or the measured oscillator is disabled or a measure has just been started by means of RCMEAS.
1	The measured oscillator has been enabled previously and MAINF value is available.

© 2026 Microchip Technology Inc. and its subsidiaries

Bits 15:0 – MAINF[15:0] Main Clock Frequency

Gives the number of cycles of the clock selected by the bit CCSS within 16 MD_SLCK periods. To calculate the frequency of the measured clock:

$f_{SELCLK} = (MAINF \times f_{MD\_SLCK})/16$

where frequency is in MHz.

© 2026 Microchip Technology Inc. and its subsidiaries

20.18.12. PMC CPU Clock Register

Name: PMC_CPU_CKR
Offset: 0x002C
Reset: 0x00000001
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the PMC Write Protection Mode Register.

Only one of CSS and PRES fields can be modified at a time. When one of these parameters is modified, no other modification can be performed on these fields as long as the MCKRDY/ CPMCKRDY status flags are low.

Bit	31	30	29	28	27	26	25	24
						RATIO_MCK0DIV2	RATIO_MCK1DIV	RATIO_MCK0DIV
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	CPPRES[3:0]					CPCSS[2:0]		
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		PRES[2:0]					CSS[1:0]	
Access		R/W	R/W	R/W			R/W	R/W
Reset		0	0	0			0	1

Bit 26 – RATIO_MCK0DIV2 MCK0 Clock Frequency Division for MCK0DIV2 Clock

Value	Description
0	MCK0DIV2 frequency is the same as MCK0.
1	MCK0DIV2 frequency is MCK0 frequency divided by 2.

Bit 25 – RATIO_MCK1DIV MCK1 Clock Frequency Division for MCK1DIV Clock

Value	Description
0	MCK1DIV frequency is the same as MCK1.
1	MCK1DIV frequency is MCK1 frequency divided by 2.

Bit 24 – RATIO_MCK0DIV MCK0 Clock Frequency Division for MCK0DIV Clock

Value	Description
0	MCK0DIV frequency is the same as MCK0.
1	MCK0DIV frequency is MCK0 frequency divided by 2.

Bits 23:20 – CPPRES[3:0] Clock Prescaler
Selected clock is divided by (CPPRES + 1).

© 2026 Microchip Technology Inc. and its subsidiaries

Bits 18:16 – CPCSS[2:0] MCK1 Source Selection

Value	Name	Description
0	SLOW_CLK	MD_SLCK is selected
1	MAIN_CLK	MAINCK is selected
2	MCK0	MCK0 is selected
3	PLLACK1	PLLACK1 is selected
4	PLLBCK	PLLBCK is selected
5	PLLCCK	PLLCCK is selected

Bits 6:4 – PRES[2:0] Processor (CPU_CLK0) and Main System Bus Clock (MCK0) Prescaler

Value	Name	Description
0	CLK_1	Selected clock
1	CLK_2	Selected clock divided by 2
2	CLK_4	Selected clock divided by 4
3	CLK_8	Selected clock divided by 8
4	CLK_16	Selected clock divided by 16
5	CLK_32	Selected clock divided by 32
6	CLK_64	Selected clock divided by 64
7	CLK_3	Selected clock divided by 3

Bits 1:0 – CSS[1:0] Processor (CPU_CLK0) and Main System Bus Clock (MCK0) Source Selection

Value	Name	Description
0	SLOW_CLK	MD_SLCK is selected
1	MAIN_CLK	MAINCK is selected
2	PLLACK1	PLLACK1 is selected
3	PLLBCK	PLLBCK is selected

© 2026 Microchip Technology Inc. and its subsidiaries

20.18.13. PMC Programmable Clock Register

Name: PMC_PCKx
Offset: 0x44 + x*0x04 [x=0..2]
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the PMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	PRES[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
				CSS[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bits 15:8 – PRES[7:0] Programmable Clock Prescaler

Value	Description
0–255	Selected clock is divided by PRES+1.

Bits 4:0 – CSS[4:0] Programmable Clock Source Selection
Values not listed are considered 'reserved'.

Value	Name	Description
0	MD_SLOW_CLK	MD_SLCK is selected
1	TD_SLOW_CLOCK	TD_SLCK is selected
2	MAINCK	MAINCK is selected
3	MCK0	MCK0 is selected
4	PLLACK1	PLLACK1 is selected.
5	PLLBCK	PLLBCK is selected.
6	PLLCCK	PLLCCK is selected.
7	PLLCSRC	PLLCSRC is selected.

© 2026 Microchip Technology Inc. and its subsidiaries

20.18.14. PMC Interrupt Enable Register

Name: PMC_IER
Offset: 0x0064
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the PMC Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	MCKMON		XT32KERR			CFDEV	MOSCRCS	MOSCSELS
Access	W		W			W	W	W
Reset	–		–			–	–	–

Bit	15	14	13	12	11	10	9	8
					PCKRDY3	PCKRDY2	PCKRDY1	PCKRDY0
Access					W	W	W	W
Reset					–	–	–	–

Bit	7	6	5	4	3	2	1	0
				CPMCKRDY	MCKRDY			MOSCXTS
Access				W	W			W
Reset				–	–			–

Bit 23 – MCKMON Main System Bus Clock (MCK0) Monitor Interrupt Enable

Bit 21 – XT32KERR 32.768 kHz Crystal Oscillator Error Interrupt Enable

Bit 18 – CFDEV Clock Failure Detector Event Interrupt Enable

Bit 17 – MOSCRCS Main RC Oscillator Status Interrupt Enable

Bit 16 – MOSCSELS Main Clock Source Oscillator Selection Status Interrupt Enable

Bits 8, 9, 10, 11 – PCKRDYx Programmable Clock Ready x Interrupt Enable

Bit 4 – CPMCKRDY Main System Bus Clock (MCK1) Ready Interrupt Enable

Bit 3 – MCKRDY Processor Clock (CPU_CLK0) and Main System Bus Clock (MCK0) Ready Interrupt Enable

Bit 0 – MOSCXTS Main Crystal Oscillator Status Interrupt Enable

© 2026 Microchip Technology Inc. and its subsidiaries

20.18.15. PMC Interrupt Disable Register

Name: PMC_IDR
Offset: 0x0068
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the PMC Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	MCKMON		XT32KERR			CFDEV	MOSCRCS	MOSCSELS
Access	W		W			W	W	W
Reset	–		–			–	–	–

Bit	15	14	13	12	11	10	9	8
					PCKRDY3	PCKRDY2	PCKRDY1	PCKRDY0
Access					W	W	W	W
Reset					–	–	–	–

Bit	7	6	5	4	3	2	1	0
				CPMCKRDY	MCKRDY			MOSCXTS
Access				W	W			W
Reset				–	–			–

Bit 23 – MCKMON Main System Bus Clock (MCK0) Monitor Interrupt Disable

Bit 21 – XT32KERR 32.768 kHz Crystal Oscillator Error Interrupt Disable

Bit 18 – CFDEV Clock Failure Detector Event Interrupt Disable

Bit 17 – MOSCRCS Main RC Status Interrupt Disable

Bit 16 – MOSCSELS Main Clock Source Oscillator Selection Status Interrupt Disable

Bits 8, 9, 10, 11 – PCKRDYx Programmable Clock Ready x Interrupt Disable

Bit 4 – CPMCKRDY Main System Bus Clock (MCK1) Ready Interrupt Disable

Bit 3 – MCKRDY Processor Clock (CPU_CLK0) and Main System Bus Clock (MCK0) Ready Interrupt Disable

Bit 0 – MOSCXTS Main Crystal Oscillator Status Interrupt Disable

© 2026 Microchip Technology Inc. and its subsidiaries

20.18.16. PMC Status Register

Name: PMC_SR
Offset: 0x006C
Reset: 0x00030008
Property: Read-only

Bit	31	30	29	28	27	26	25	24
							PLL_INT	GCLKRDY
Access							R	R
Reset							0	0

Bit	23	22	21	20	19	18	17	16
	MCKMON		XT32KERR	FOS	CFDS	CFDEV	MOSCRCS	MOSCSELS
Access	R		R	R	R	R	R	R
Reset	0		0	0	0	0	1	1

Bit	15	14	13	12	11	10	9	8
					PCKRDY3	PCKRDY2	PCKRDY1	PCKRDY0
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	OSCSELS			CPMCKRDY	MCKRDY			MOSCXTS
Access	R			R	R			R
Reset	0			0	1			0

Bit 25 – PLL_INT PLL Interrupt Status

Value	Description
0	No PLL interrupt has occurred
1	A PLL interrupt has occurred. PLL interrupt is defined by the configuration of the PMC_PIMR register.

Bit 24 – GCLKRDY GCLK Ready

Value	Description
0	A GCLK is not ready to use (clock switching in progress).
1	All GCLK are switched on their selected source clock and ready to use.

Bit 23 – MCKMON Main System Bus Clock (MCK0) Monitor Error
This status is cleared on read.

Value	Description
0	The MCK0 clock is correct or the clock monitor is disabled.
1	The MCK0 clock is incorrect or has been incorrect for an elapsed period of time since the clock monitor has been enabled.

Bit 21 – XT32KERR Slow Crystal Oscillator Error

Value	Description
0	The frequency of the 32.768 kHz crystal oscillator is correct (32.768 kHz ±1%) or the monitoring is disabled.
1	The frequency of the 32.768 kHz crystal oscillator is incorrect or has been incorrect for an elapsed period of time since the monitoring has been enabled.

Bit 20 – FOS Clock Failure Detector Fault Output Status

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	The fault output of the clock failure detector is inactive.
1	The fault output of the clock failure detector is active. This status is cleared by writing a '1' to FOCLR in PMC_FOCR.

Bit 19 – CFDS Clock Failure Detector Status

Value	Description
0	A clock failure of the Main crystal oscillator clock is not detected.
1	A clock failure of the Main crystal oscillator clock is detected.

Bit 18 – CFDEV Clock Failure Detector Event

Value	Description
0	No clock failure detection of the Main crystal oscillator clock has occurred since the last read of PMC_SR.
1	At least one clock failure detection of the Main crystal oscillator clock has occurred since the last read of PMC_SR.

Bit 17 – MOSCRCS Main RC Oscillator Status

Value	Description
0	Main RC oscillator is not stabilized.
1	Main RC oscillator is stabilized.

Bit 16 – MOSCSELS Main Clock Source Oscillator Selection Status

Value	Description
0	Selection is in progress.
1	Selection is done.

Bits 8, 9, 10, 11 – PCKRDYx Programmable Clock Ready Status

Value	Description
0	Programmable Clock x is not ready.
1	Programmable Clock x is ready.

Bit 7 – OSCSELS Monitoring Domain Slow Clock Source Oscillator Selection

Value	Description
0	Slow RC oscillator is selected.
1	32.768 kHz crystal oscillator is selected.

Bit 4 – CPMCKRDY Main System Bus Clock (MCK1) Ready Status

Value	Description
0	Not ready.
1	Ready.

Bit 3 – MCKRDY Processor Clock (CPU_CLK0) and Main System Bus Clock (MCK0) Ready Status

Value	Description
0	Not ready.
1	Ready.

Bit 0 – MOSCXTS Main Crystal Oscillator Status

Value	Description
0	Main crystal oscillator is not stabilized.
1	Main crystal oscillator is stabilized.

© 2026 Microchip Technology Inc. and its subsidiaries

20.18.17. PMC Interrupt Mask Register

Name: PMC_IMR
Offset: 0x0070
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	MCKMON		XT32KERR			CFDEV	MOSCRCS	MOSCSELS
Access	R		R			R	R	R
Reset	0		0			0	0	0

Bit	15	14	13	12	11	10	9	8
					PCKRDY3	PCKRDY2	PCKRDY1	PCKRDY0
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
				CPMCKRDY	MCKRDY			MOSCXTS
Access				R	R			R
Reset				0	0			0

Bit 23 – MCKMON Main System Bus Clock (MCK0) Monitor Interrupt Mask

Bit 21 – XT32KERR 32.768 kHz Crystal Oscillator Error Interrupt Mask

Bit 18 – CFDEV Clock Failure Detector Event Interrupt Mask

Bit 17 – MOSCRCS Main RC Status Interrupt Mask

Bit 16 – MOSCSELS Main Clock Source Oscillator Selection Status Interrupt Mask

Bits 8, 9, 10, 11 – PCKRDYx Programmable Clock Ready x Interrupt Mask

Bit 4 – CPMCKRDY Main System Bus Clock (MCK1) Ready Interrupt Mask

Bit 3 – MCKRDY Processor Clock (CPU_CLK0) and Main System Bus Clock (MCK0) Ready Interrupt Mask

Bit 0 – MOSCXTS Main Crystal Oscillator Status Interrupt Mask

© 2026 Microchip Technology Inc. and its subsidiaries

20.18.18. PMC Fast Start-up Mode Register

Name: PMC_FSMR
Offset: 0x0074
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the PMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	FFLPM	FLPM[1:0]		LPM	SMAL	XTFA	RTCAL	RTTAL
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 23 – FFLPM Force Flash Low-power Mode

Value	Description
0	The Flash Low-power mode, defined in the FLPM field, is automatically applied when in Wait mode and released when going back to Active mode.
1	The Flash Low-power mode is user defined by the FLPM field and immediately applied.

Bits 22:21 – FLPM[1:0] Flash Low-power Mode

Value	Name	Description
0	FLASH_STANDBY	Flash is in Standby mode when system enters Wait mode
1	FLASH_DEEP_POWERDOWN	Flash is in Deep-Powerdown mode when system enters Wait mode
2	FLASH_IDLE	Idle mode

Bit 20 – LPM Low Power Mode

Value	Description
0	The WaitForInterrupt (WFI) or the WaitForEvent (WFE) instruction of the processor makes the processor enter Sleep mode.
1	The WaitForEvent (WFE) instruction of the processor makes the system enter Wait mode.

Bit 19 – SMAL Supply Monitor Alarm Enable

Value	Description
0	The supply monitor alarm has no effect on the PMC.
1	The supply monitor alarm enables a fast restart signal to the PMC.

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 18 – XTFA 32KHz Crystal Failure Alarm Enable

Value	Description
0	The 32KHz Crystal Failure alarm as no effect on the PMC.
1	The 32KHz Crystal Failure alarm enables a fast restart signal to the PMC.

Bit 17 – RTCAL RTC Alarm Enable

Value	Description
0	The RTC alarm has no effect on the PMC.
1	The RTC alarm enables a fast restart signal to the PMC.

Bit 16 – RTTAL RTT Alarm Enable

Value	Description
0	The RTT alarm has no effect on the PMC.
1	The RTT alarm enables a fast restart signal to the PMC.

© 2026 Microchip Technology Inc. and its subsidiaries

20.18.19. PMC Wake-Up Control Register

Name: PMC_WCR
Offset: 0x0078
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
								CMD
Access								R/W
Reset								0

Bit	23	22	21	20	19	18	17	16
							WIP	EN
Access							R/W	R/W
Reset							0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					WKPIONB[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 24 – CMD Command

Value	Description
0	Read mode.
1	Write mode.

Bit 17 – WIP Wake-up Input Polarity
Defines the active polarity of the selected wake-up input. If the corresponding wake-up input is enabled at the FSTP level, it enables a fast restart signal.

Bit 16 – EN Wake-up Input Enable

Value	Description
0	The selected wake-up input has no effect on the PMC.
1	The selected wake-up input enables a fast restart signal to the PMC.

Bits 3:0 – WKPIONB[3:0] Wake-up Input Number
Defines which wake-up source is to be modified during a write access (CMD is set to '1') or which wake-up source status is read on the next read access to this register (CMD is set to '0').,

Primary Signal Name	WKPIONB
WKUP0	0
WKUP1	1
WKUP2	2
PA2	3
PA3	4
PA5	5
PA9	6

© 2026 Microchip Technology Inc. and its subsidiaries

WKPIONB (continued)	
Primary Signal Name	**WKPIONB**
PA13	7
PB25	8
PB9	9
PB12	10
PC7	11
PC9	12
PC14	13
PC21	14
WOD (Wake-up from debug)	15

© 2026 Microchip Technology Inc. and its subsidiaries

20.18.20. PMC Fault Output Clear Register

Name: PMC_FOCR
Offset: 0x007C
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								FOCLR
Access								W
Reset								–

Bit 0 – FOCLR Fault Output Clear
Clears the clock failure detector fault output.

© 2026 Microchip Technology Inc. and its subsidiaries

20.18.21. PMC Write Protection Mode Register

Name: PMC_WPMR
Offset: 0x0084
Reset: 0x00000000
Property: Read/Write

See the section Register Write Protection for the list of registers that can be write-protected.

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
							WPITEN	WPEN
Access							R/W	R/W
Reset							0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x504D43	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0.

Bit 1 – WPITEN Write Protection Interrupt Enable

Value	Description
0	Disables the write protection on interrupt registers if WPKEY corresponds to 0x504D43 (“PMC” in ASCII).
1	Enables the write protection on interrupt registers if WPKEY corresponds to 0x504D43 (“PMC” in ASCII).

Bit 0 – WPEN Write Protection Enable

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x504D43 (“PMC” in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x504D43 (“PMC” in ASCII).

© 2026 Microchip Technology Inc. and its subsidiaries

20.18.22. PMC Write Protection Status Register

Name: PMC_WPSR
Offset: 0x0088
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	WPVSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPVS
Access								R
Reset								0

Bits 23:8 – WPVSRC[15:0] Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protection Violation Status

Value	Description
0	No write protection violation has occurred since the last read of the PMC_WPSR.
1	A write protection violation has occurred since the last read of the PMC_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

© 2026 Microchip Technology Inc. and its subsidiaries

20.18.23. PMC Peripheral Control Register

Name: PMC_PCR
Offset: 0x008C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	CMD		GCLKEN	EN	GCLKDIV[7:4]			
Access	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset	0		0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	GCLKDIV[3:0]							
Access	R/W	R/W	R/W	R/W				
Reset	0	0	0	0				

Bit	15	14	13	12	11	10	9	8
				GCLKCSS[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		PID[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit 31 – CMD Command

Value	Description
0	Read mode.
1	Write mode.

Bit 29 – GCLKEN Generic Clock Enable

Value	Description
0	The selected generic clock is disabled.
1	The selected generic clock is enabled.

Bit 28 – EN Enable

Value	Description
0	Selected Peripheral clock is disabled.
1	Selected Peripheral clock is enabled.

Bits 27:20 – GCLKDIV[7:0] Generic Clock Division Ratio
Generic clock is the selected clock period divided by GCLKDIV + 1.
GCLKDIV must not be changed while the peripheral selects GCLKx (e.g., bit rate, etc.).

Bits 12:8 – GCLKCSS[4:0] Generic Clock Source Selection

Value	Name	Description
0	MD_SLOW_CLK	MD_SLCK is selected
1	TD_SLOW_CLOCK	TD_SLCK is selected
2	MAINCK	MAINCK is selected

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Name	Description
3	MCK0	MCK0 is selected
4	PLLACK1	PLLACK1 is selected.
5	PLLBCK	PLLBCK is selected.
6	PLLCCK	PLLCCK is selected.
7	PLLCSRC	PLLCSRC is selected.

Bits 6:0 – PID[6:0] Peripheral ID
Peripheral ID selection.
Refer to the identifiers as defined in the section “Peripheral Identifiers”.

© 2026 Microchip Technology Inc. and its subsidiaries

20.18.24. PMC Oscillator Calibration Register

Name: PMC_OCR
Offset: 0x0090
Reset: 0x00404040
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the PMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	SEL12	CAL12[6:0]						
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 23 – SEL12 Selection of Main RC Oscillator Calibration Bits

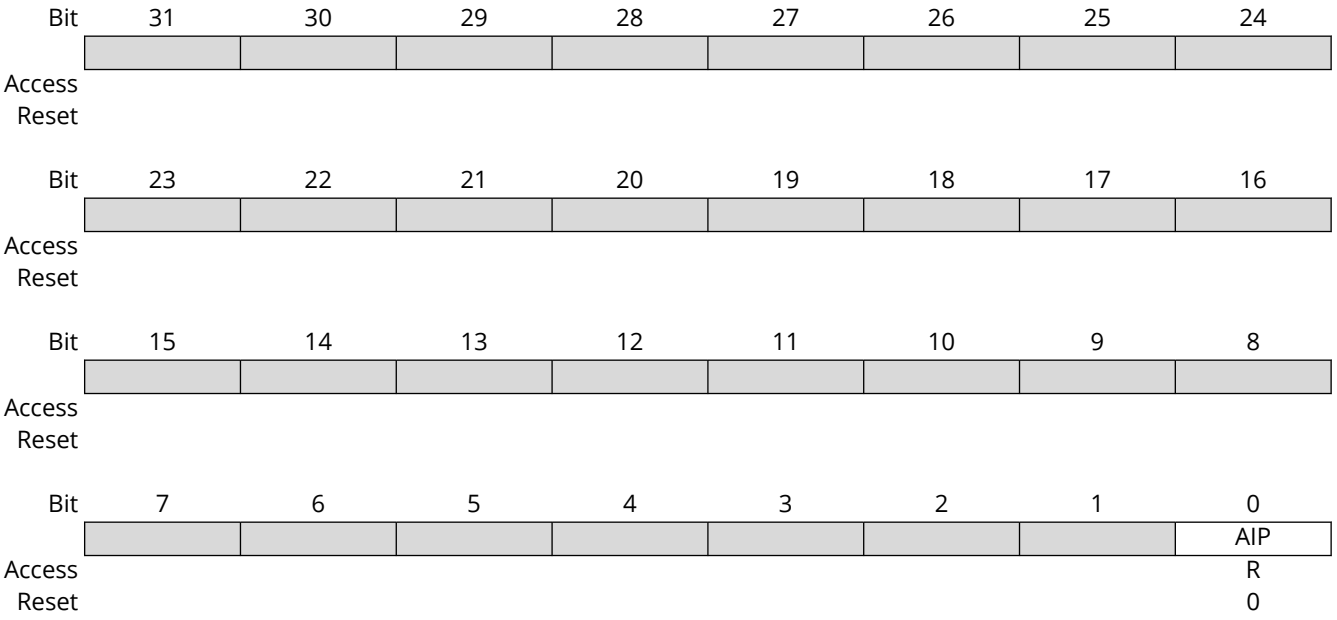
Value	Description
0	Factory-defined value.
1	Value written by user in CAL12 field of this register.

Bits 22:16 – CAL12[6:0] Main RC Oscillator Calibration Bits
Calibration bits applied to the RC Oscillator. Refer to the section " Electrical Characteristics".

© 2026 Microchip Technology Inc. and its subsidiaries

20.18.25. PMC Partial Wake-Up Activity In Progress Register

Name: PMC_SLPWK_AIPR
Offset: 0x0094
Reset: 0x00000000
Property: Read-only



Bit 0 – AIP Activity In Progress
Only the following PIDs can be configured with asynchronous partial wake-up: FLEXCOM.

Value	Description
0	There is no activity on peripherals. The asynchronous partial wake-up function can be activated on one or more peripherals. The device can enter Wait Mode.
1	One or more peripherals are currently active. The device must not enter Wait Mode if the asynchronous partial wake-up is enabled for one of the following PIDs: FLEXCOM.

© 2026 Microchip Technology Inc. and its subsidiaries

20.18.26. PMC Partial Wake-Up Control Register

Name: PMC_SLPWKCR
Offset: 0x0098
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the PMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
				SLPWKSR				
Access				R/W				
Reset				0				

Bit	23	22	21	20	19	18	17	16
								ASR
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
				CMD				
Access				R/W				
Reset				0				

Bit	7	6	5	4	3	2	1	0
		PID[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit 28 – SLPWKSR Partial Wake-Up Sleep Register
Not all PIDs can be configured with asynchronous partial wake-up.
Only the following PIDs can be configured with asynchronous partial wake-up: FLEXCOM.

Value	Description
0	The asynchronous partial wake-up function of the peripheral is disabled.
1	The asynchronous partial wake-up function of the peripheral is enabled.

Bit 16 – ASR Activity Status Register
Not all PIDs can be configured with asynchronous partial wake-up.
Only the following PIDs can be configured with asynchronous partial wake-up: FLEXCOM.

Value	Description
0	The peripheral x is not currently active; the asynchronous partial wake-up function can be activated.
1	The peripheral x is currently active; the asynchronous partial wake-up function must not be activated.

Bit 12 – CMD Command

Value	Description
0	Read mode.
1	Write mode.

Bits 6:0 – PID[6:0] Peripheral ID
Peripheral ID selection from PID2 to the maximum PID number. This refers to identifiers as defined in the section “Peripheral Identifiers”.

© 2026 Microchip Technology Inc. and its subsidiaries

20.18.27. PMC MCK0 Monitor Limits Register

Name: PMC_MCKLIM
Offset: 0x00A0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	MCK_HIGH_RES[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	MCK_LOW_RES[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	MCK_HIGH_IT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MCK_LOW_IT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – MCK_HIGH_RES[7:0] MCK0 Monitoring High Reset Limit
Beyond this limit, the MCK0 frequency monitor generates a reset.

Bits 23:16 – MCK_LOW_RES[7:0] MCK0 Monitoring Low RESET Limit
Below this limit, the MCK0 frequency monitor generates a reset.

Bits 15:8 – MCK_HIGH_IT[7:0] MCK0 Monitoring High IT Limit
Beyond this limit, the MCK0 frequency monitor generates an interrupt.

Bits 7:0 – MCK_LOW_IT[7:0] MCK0 Monitoring Low IT Limit
Below this limit, the MCK0 frequency monitor generates an interrupt.

© 2026 Microchip Technology Inc. and its subsidiaries

20.18.28. PMC Peripheral Clock Status Register 0

Name: PMC_CSR0
Offset: 0x00A4
Reset: 0x00000000
Property: Read-only

"PIDx" refers to identifiers as defined in the section "Peripheral Identifiers".
The following configuration values are valid for all listed bit names of this register:

0: The corresponding peripheral clock is disabled.

1: The corresponding peripheral clock is enabled.

Bit	31	30	29	28	27	26	25	24
	PID31	PID30		PID28			PID25	PID24
Access	R	R		R			R	R
Reset	0	0		0			0	0

Bit	23	22	21	20	19	18	17	16
	PID23						PID17	PID16
Access	R						R	R
Reset	0						0	0

Bit	15	14	13	12	11	10	9	8
	PID15	PID14	PID13	PID12	PID11	PID10	PID9	
Access	R	R	R	R	R	R	R	
Reset	0	0	0	0	0	0	0	

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 30, 31 – PID30, PID31 Peripheral Clock x Status

Bit 28 – PID28 Peripheral Clock x Status

Bits 23, 24, 25 – PID23, PID24, PID25 Peripheral Clock x Status

Bits 9, 10, 11, 12, 13, 14, 15, 16, 17 – PID9, PID10, PID11, PID12, PID13, PID14, PID15, PID16, PID17 Peripheral Clock x Status

© 2026 Microchip Technology Inc. and its subsidiaries

20.18.29. PMC Peripheral Clock Status Register 1

Name: PMC_CSR1
Offset: 0x00A8
Reset: 0x00000000
Property: Read-only

"PIDx" refers to identifiers as defined in the section "Peripheral Identifiers".
The following configuration values are valid for all listed bit names of this register:

0: The corresponding peripheral clock is disabled.

1: The corresponding peripheral clock is enabled.

Bit	31	30	29	28	27	26	25	24
					PID59		PID57	
Access					R		R	
Reset					0		0	

Bit	23	22	21	20	19	18	17	16
	PID55		PID53		PID51		PID49	
Access	R		R		R		R	
Reset	0		0		0		0	

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	PID39	PID38	PID37	PID36	PID35	PID34	PID33	PID32
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 27 – PID59 Peripheral Clock x Status

Bit 25 – PID57 Peripheral Clock x Status

Bit 23 – PID55 Peripheral Clock x Status

Bit 21 – PID53 Peripheral Clock x Status

Bit 19 – PID51 Peripheral Clock x Status

Bit 17 – PID49 Peripheral Clock x Status

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PID32, PID33, PID34, PID35, PID36, PID37, PID38, PID39 Peripheral Clock x Status

© 2026 Microchip Technology Inc. and its subsidiaries

20.18.30. PMC Peripheral Clock Status Register 2

Name: PMC_CSR2
Offset: 0x00AC
Reset: 0x00000000
Property: Read-only

"PIDx" refers to identifiers as defined in the section "Peripheral Identifiers".
The following configuration values are valid for all listed bit names of this register:

0: The corresponding peripheral clock is disabled.

1: The corresponding peripheral clock is enabled.

Bit	31	30	29	28	27	26	25	24
					PID91	PID90	PID89	PID88
Access					R	R	R	R
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
	PID87		PID85				PID81	PID80
Access	R		R				R	R
Reset	0		0				0	0

Bit	15	14	13	12	11	10	9	8
	PID79	PID78			PID75			
Access	R	R			R			
Reset	0	0			0			

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 23, 24, 25, 26, 27 – PID87, PID88, PID89, PID90, PID91 Peripheral Clock x Status

Bit 21 – PID85 Peripheral Clock x Status

Bits 14, 15, 16, 17 – PID78, PID79, PID80, PID81 Peripheral Clock x Status

Bit 11 – PID75 Peripheral Clock x Status

© 2026 Microchip Technology Inc. and its subsidiaries

20.18.31. PMC Peripheral Clock Status Register 3

Name: PMC_CSR3
Offset: 0x00B0
Reset: 0x00000000
Property: Read-only

"PIDx" refers to identifiers as defined in the section "Peripheral Identifiers".
The following configuration values are valid for all listed bit names of this register:

0: The corresponding peripheral clock is disabled.

1: The corresponding peripheral clock is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								PID96
Access								R
Reset								0

Bit 0 – PID96 Peripheral Clock x Status

© 2026 Microchip Technology Inc. and its subsidiaries

20.18.32. PMC Generic Clock Status Register 0

Name: PMC_GCSR0
Offset: 0x00C4
Reset: 0x00000000
Property: Read-only

"GPIDx" refers to identifiers as defined in the section "Peripheral Identifiers".
The following configuration values are valid for all listed bit names of this register:

0: The corresponding Generic clock is disabled.

1: The corresponding Generic clock is enabled.

Bit	31	30	29	28	27	26	25	24
	GPID31							GPID24
Access	R							R
Reset	0							0

Bit	23	22	21	20	19	18	17	16
	GPID23							GPID16
Access	R							R
Reset	0							0

Bit	15	14	13	12	11	10	9	8
	GPID15	GPID14	GPID13	GPID12	GPID11	GPID10	GPID9	
Access	R	R	R	R	R	R	R	
Reset	0	0	0	0	0	0	0	

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 31 – GPID31 Generic Clock x Status

Bits 23, 24 – GPID23, GPID24 Generic Clock x Status

Bits 9, 10, 11, 12, 13, 14, 15, 16 – GPID9, GPID10, GPID11, GPID12, GPID13, GPID14, GPID15, GPID16
Generic Clock x Status

© 2026 Microchip Technology Inc. and its subsidiaries

20.18.33. PMC Generic Clock Status Register 1

Name: PMC_GCSR1
Offset: 0x00C8
Reset: 0x00000000
Property: Read-only

"GPIDx" refers to identifiers as defined in the section "Peripheral Identifiers".
The following configuration values are valid for all listed bit names of this register:

0: The corresponding Generic clock is disabled.

1: The corresponding Generic clock is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			GPID37			GPID34		
Access			R			R		
Reset			0			0		

Bit 5 – GPID37 Generic Clock x Status

Bit 2 – GPID34 Generic Clock x Status

© 2026 Microchip Technology Inc. and its subsidiaries

20.18.34. PMC Generic Clock Status Register 2

Name: PMC_GCSR2
Offset: 0x00CC
Reset: 0x00000000
Property: Read-only

"GPIDx" refers to identifiers as defined in the section "Peripheral Identifiers".
The following configuration values are valid for all listed bit names of this register:

0: The corresponding Generic clock is disabled.

1: The corresponding Generic clock is enabled.

Bit	31	30	29	28	27	26	25	24
						GPID90	GPID89	
Access						R	R	
Reset						0	0	

Bit	23	22	21	20	19	18	17	16
	GPID87							
Access	R							
Reset	0							

Bit	15	14	13	12	11	10	9	8
	GPID79				GPID75			
Access	R				R			
Reset	0				0			

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 25, 26 – GPID89, GPID90 Generic Clock x Status

Bit 23 – GPID87 Generic Clock x Status

Bit 15 – GPID79 Generic Clock x Status

Bit 11 – GPID75 Generic Clock x Status

© 2026 Microchip Technology Inc. and its subsidiaries

20.18.35. PMC Generic Clock Status Register 3

Name: PMC_GCSR3
Offset: 0x00D0
Reset: 0x00000000
Property: Read-only

"GPIDx" refers to identifiers as defined in the section "Peripheral Identifiers".
The following configuration values are valid for all listed bit names of this register:

0: The corresponding Generic clock is disabled.

1: The corresponding Generic clock is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								GPID96
Access								R
Reset								0

Bit 0 – GPID96 Generic Clock x Status

© 2026 Microchip Technology Inc. and its subsidiaries

20.18.36. PMC PLL Interrupt Enable Register

Name: PMC_PLL_IER
Offset: 0x00E4
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the PMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						UNLOCK2	UNLOCK1	UNLOCK0
Access						W	W	W
Reset						–	–	–

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						LOCK2	LOCK1	LOCK0
Access						W	W	W
Reset						–	–	–

Bits 16, 17, 18 – UNLOCKx PLL of Index x Unlock Interrupt Enable

Bits 0, 1, 2 – LOCKx PLL of Index x Lock Interrupt Enable

© 2026 Microchip Technology Inc. and its subsidiaries

20.18.37. PMC PLL Interrupt Disable Register

Name: PMC_PLL_IDR
Offset: 0x00E8
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the PMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						UNLOCK2	UNLOCK1	UNLOCK0
Access						W	W	W
Reset						–	–	–

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						LOCK2	LOCK1	LOCK0
Access						W	W	W
Reset						–	–	–

Bits 16, 17, 18 – UNLOCKx PLL of Index x Unlock Interrupt Disable

Bits 0, 1, 2 – LOCKx PLL of Index x Lock Interrupt Disable

© 2026 Microchip Technology Inc. and its subsidiaries

20.18.38. PMC PLL Interrupt Mask Register

Name: PMC_PLL_IMR
Offset: 0x00EC
Reset: 0x00000000
Property: Read-only

This register can only be written if the WPITEN bit is cleared in the PMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						UNLOCK2	UNLOCK1	UNLOCK0
Access						R	R	R
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						LOCK2	LOCK1	LOCK0
Access						R	R	R
Reset						0	0	0

Bits 16, 17, 18 – UNLOCKx PLL of Index x Unlock Interrupt Mask

Bits 0, 1, 2 – LOCKx PLL of Index x Lock Interrupt Mask

© 2026 Microchip Technology Inc. and its subsidiaries

20.18.39. PMC PLL Interrupt Status Register 0

Name: PMC_PLL_ISR0
Offset: 0x00F0
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						UNLOCK2	UNLOCK1	UNLOCK0
Access						R	R	R
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						LOCK2	LOCK1	LOCK0
Access						R	R	R
Reset						0	0	0

Bits 16, 17, 18 – UNLOCKx PLL of Index x Unlock Interrupt Status

Value	Description
0	PLL is not unlocked.
1	PLLx is unlocked. To know the unlock type, the PMC_PISR1 register can be read.

Bits 0, 1, 2 – LOCKx PLL of Index x Lock Interrupt Status

Value	Description
0	PLLx is not locked.
1	PLLx is locked.

© 2026 Microchip Technology Inc. and its subsidiaries

20.18.40. PMC PLL Interrupt Status Register 1

Name: PMC_PLL_ISR1
Offset: 0x00F4
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						OVR2	OVR1	OVR0
Access						R	R	R
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						UDR2	UDR1	UDR0
Access						R	R	R
Reset						0	0	0

Bits 16, 17, 18 – OVRx PLLx Overflow

Value	Description
0	PLL is not in overflow state.
1	PLL has encountered an overflow.

Bits 0, 1, 2 – UDRx PLLx Underflow

Value	Description
0	PLL is not in underflow state.
1	PLL has encountered an underflow.

© 2026 Microchip Technology Inc. and its subsidiaries

21. Parallel Input/Output Controller (PIO)

21.1. Description

The Parallel Input/Output Controller (PIO) manages up to 74 fully programmable input/output lines. Each I/O line may be dedicated as a general purpose I/O or be assigned to a function of an embedded peripheral. This ensures effective optimization of the pins of the product.

The PIO Controller features a synchronous output providing up to 32 bits of data output in a single write operation.

The PIO embeds safety and security features.

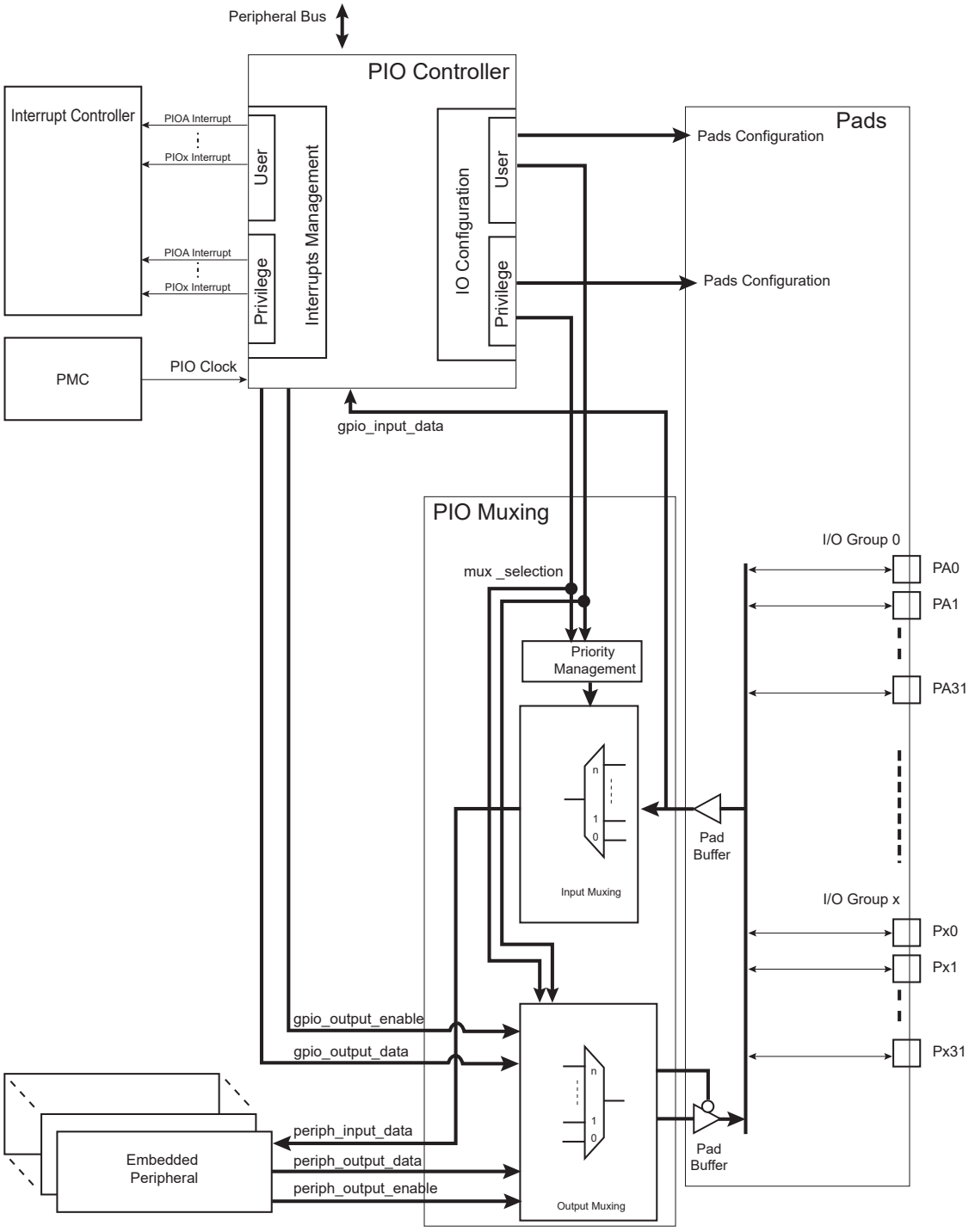
21.2. Embedded Characteristics

- Up to 74 Programmable I/O Lines
- Multiplexing of Up to 4 Peripheral Functions per I/O Line
- For Each I/O Line (whether assigned to a peripheral or used as general purpose I/O):
  - Input change interrupt
  - Programmable glitch filter
  - Programmable debouncing filter
  - Multi-drive option enables driving in open drain
  - Programmable pull-up/pull-down
  - Pin data status register, supplies visibility of the level on the pin at any time
  - Programmable event: rising edge, falling edge, both edges, low-level or high-level
  - Configuration lock by the connected peripheral
  - Privileged-Access or User-Access mode management
  - Programmable configuration lock (active until next V_{DDCORE} reset) to protect against further software modifications (intentional or unintentional)
- Separate Interrupt Lines: One Group Driven by Privileged Access I/O and the Second Group Driven by User Access I/O
- Register Write Protection against Unintentional Software Modifications:
  - One configuration bit to enable or disable protection of I/O line settings
  - One configuration bit to enable or disable protection of interrupt settings
- Synchronous Output, Possibility to Set or Clear Simultaneously Up to 32 I/O Lines in a Single Write
- Programmable Schmitt Trigger Inputs
- Programmable Slew Rate

© 2026 Microchip Technology Inc. and its subsidiaries

21.3. Block Diagram

Figure 21-1. PIO Controller Block Diagram



© 2026 Microchip Technology Inc. and its subsidiaries

Notes:

1. x = 2 (the number of I/O groups is 3).
2. n depends on the number of I/O lines affected to the IP input.

21.4. Product Dependencies

21.4.1. Pin Multiplexing

Each pin is configurable, depending on the product, as either a general purpose I/O line only, or as an I/O line multiplexed with up to 4 peripheral I/Os. As the multiplexing is hardware defined and thus product-dependent, the hardware designer and programmer must carefully determine the configuration of the PIO Controllers required by their application. When an I/O line is general purpose only, i.e., not multiplexed with any peripheral I/O, programming of the PIO Controller regarding the assignment to a peripheral has no effect and only the PIO Controller can control how the pin is driven by the product.

21.4.2. External Interrupt Lines

When the WKUPx input pins must be used as external interrupt lines, the PIO Controller must be configured to disable the peripheral control on these IOs, and the corresponding IO lines must be set to Input mode.

21.4.3. Power Management

The Power Management Controller (PMC) controls the PIO Controller clock in order to save power. Writing any of the registers of the user interface does not require the PIO Controller clock to be enabled. This means that the configuration of the I/O lines does not require the PIO Controller clock to be enabled.

However, when the clock is disabled, not all of the features of the PIO Controller are available, including glitch filtering. Note that the input change interrupt, the interrupt modes on a programmable event and the read of the pin level require the clock to be validated.

After a hardware reset, the PIO clock is disabled by default.

The user must configure the PMC before any access to the input line information.

21.4.4. Interrupt Generation

For interrupt handling, the PIO Controllers are considered as user peripherals. This means that the PIO Controller interrupt lines are connected among the interrupt sources. The PIO Controller supplies one interrupt signal per I/O group. Refer to the PIO Controller peripheral identifier in the product description to identify the interrupt sources dedicated to the PIO Controller. The PIO Controller provides two groups of interrupt lines to ease management of interrupt priorities related to Privileged-Access and User-Access mode I/O lines. Using the PIO Controller requires the Interrupt Controller to be programmed first.

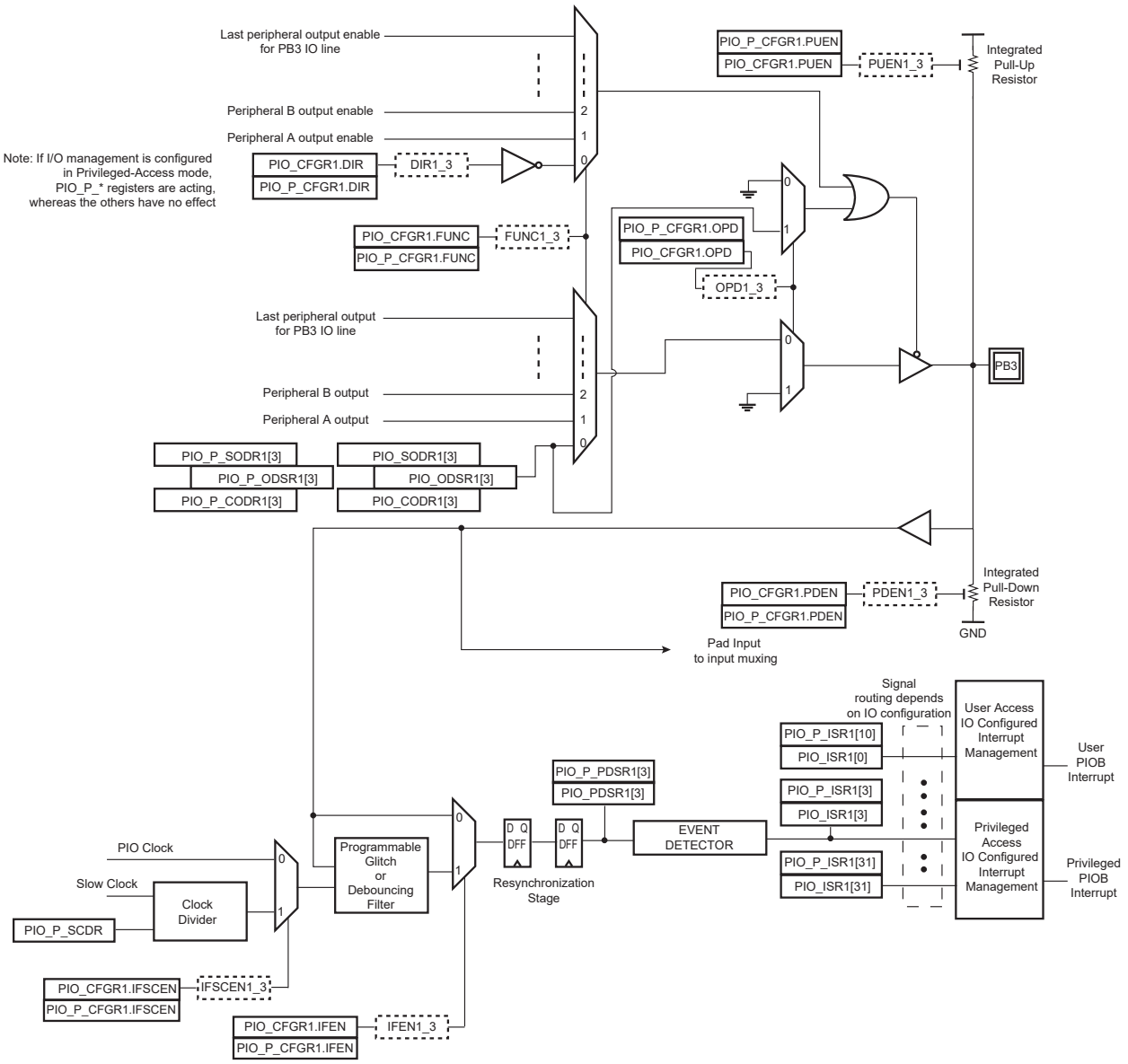
The PIO Controller interrupt can be generated only if the PIO Controller clock is enabled.

21.5. Functional Description

The PIO Controller features up to 512 fully-programmable I/O lines. Most of the control logic associated to each I/O is represented in the following figure, where the I/O line 3 of the PIOB (PB3) is described as an example. In this description each signal shown represents one of up to 512 possible indexes.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 21-2. I/O Line Control Logic



21.5.1. I/O Line Configuration Method

The user interface of the PIO Controller provides several sets of registers. Each set of registers interfaces with one I/O group.

Table 21-1. I/O Group List

I/O Group Number	PIO
0	PIOA
1	PIOB
...	...
2	PIOC

21.5.1.1. Privileged/User IO Access Management

Safety/security can be reinforced for the configuration and access of I/O lines. This safety/security enhancement is related to the Privileged-/User Access modes offered by the CPU core. This Privileged-Access mode prevents an unexpected access performed by a non-privileged software

© 2026 Microchip Technology Inc. and its subsidiaries

task/process to modify the configuration of any I/O considered relevant for safety/security. The Privileged-Access mode does not protect against an abnormal access resulting from a Single-event upset that may corrupt the value of 1 bit on the system bus. To prevent these events, see section I/O Line Configuration Freeze and section Register Write Protection.

The user must first configure the configuration and control access level for the I/O line. Each I/O line of each I/O group must be configured to be accessed either in Privileged-Access mode or User-Access mode.

Each I/O line of the I/O group x can be set as User-Access mode I/O line by writing a 1 to the corresponding bit P0–P31 of the PIO Privilege Set I/O User Access Register (PIO_P_SIO_UARx) of the I/O group x.

To define an I/O line of I/O group x as a Privileged-Access mode I/O line, write a 1 to the corresponding bit P0–P31 of the PIO Privilege Set I/O Privilege Access Register (PIO_P_SIO_PARx) of the I/O group x.

Examples:

To set the I/O line PA4 as User access line:

- Write the value 16 (bit 4 at 1) at address 0x10B0 (PIO_P_SIO_UAR0)

To set the I/O line PB3 as Privilege access line:

- Write the value 8 (bit 3 at 1) at address 0x1074 (PIO_P_SIO_PAR1)

The access level of each I/O line is reported by the PIO Privilege I/O Security Status register (PIO_P_IOSSRx) of the corresponding I/O group. Reading 0 at the corresponding bit P0–P31 means that the corresponding I/O line of the I/O group is configured in Privileged-Access mode. Reading 1 means that this I/O line of the I/O group is configured in User-Access mode.

The PIO Controller user interface is divided into two register mapping areas:

- The User-Access area, located from address 0x0 to 0x1000, can be accessed by the CPU core. This area interfaces with all the I/O lines defined as User-Access. Trying to access to I/O line configured in Privileged-Access mode through this area will have no effect on the I/O line and read values will be 0.
- The Privileged-Access area, located above address 0x1000, can only be accessed by the CPU core configured in Privileged-Access mode (if the PIO Controller is defined as a privileged-access client peripheral at the system bus matrix level). This area interfaces with all the I/O lines configured to be accessed in Privileged-Access mode. Trying to access an I/O line configured to be accessed in User-Access mode through this area will have no effect on the I/O line and read values will be 0.

21.5.1.2. Programming I/O Line Configuration

The user must first define which I/O line in the group will be targeted by writing a 1 to the corresponding bit in the PIO Mask Register (PIO_MSKRx). Several I/O lines in an I/O group can be configured at the same time by setting the corresponding bits in PIO_MSKRx. Then, writing the PIO Configuration Register (PIO_CFGRx) apply the configuration to the I/O line(s) defined in PIO_MSKRx. All the I/O lines defined as Privileged-Access mode in the PIO_P_SIO_PARx must be configured by writing the PIO_P_CFGRx and PIO_P_MSKRx registers.

For more details concerning the I/O line configuration using PIO_MSKRx and PIO_CFGRx, see section I/O Lines Programming Example.

21.5.1.3. Reading the I/O Line Configuration

As for programming operation, reading configuration requires the user to first define which I/O line in the group x will be targeted by writing a 1 to the corresponding bit in the PIO_MSKRx. The value of the targeted I/O line is read in PIO_CFGRx.

If several bits are set in PIO_MSKRx, then the read configuration in PIO_CFGRx is the configuration of the I/O line with the lowest index.

© 2026 Microchip Technology Inc. and its subsidiaries

Note that PIO_P_MSKRx and PIO_P_CFGRx must be used to read the configuration of a Privileged-Access mode I/O line.

21.5.2. Pull-Up and Pull-Down Resistor Control

Each I/O line is designed with an embedded pull-up resistor and an embedded pull-down resistor.

The pull-up resistor on the I/O line(s) defined in PIO_MSKRx can be enabled by setting the PUEN bit in PIO_CFGRx. Clearing the PUEN bit in PIO_CFGRx disables the pull-up resistor of I/O lines defined in PIO_MSKRx.

The pull-down resistor on the I/O line(s) defined in PIO_MSKRx can be enabled by setting the PDEN bit in PIO_CFGRx. Clearing the PDEN bit in PIO_CFGRx disables the pull-down resistor of I/O lines defined in PIO_MSKRx.

If both PUEN and PDEN bits are set in PIO_CFGRx, only the pull-up resistor is enabled for I/O line(s) defined in PIO_MSKRx and the PDEN bit is discarded.

Control of the pull-up resistor is possible regardless of the configuration of the I/O line (Input, Output, Open-drain).

Note that PIO_P_MSKRx and PIO_P_CFGRx must be used to program the pull-up or pull-down configuration of a Privileged-Access mode I/O line.

For more details concerning Pull-up and Pull-down configuration, see PIO_CFGRx or PIO_P_CFGRx for Privileged-Access mode I/O line configuration.

The reset value of PUEN and PDEN bits of each I/O line is defined at the product level and depends on the multiplexing of the device.

21.5.3. General Purpose or Peripheral Function Selection

The PIO Controller provides multiplexing of up to 4 peripheral functions on a single pin. The selection is performed by writing the FUNC field in PIO_CFGRx. The selected function is applied to the I/O line(s) defined in PIO_MSKRx.

When FUNC is 0, no peripheral is selected and the General Purpose PIO (GPIO) mode is selected (in this mode, the I/O line is controlled by the PIO Controller).

When the value configured in PIO_CFGRx.FUNC is greater than 0, the software cannot drive the I/O line anymore and the value configured in PIO_CFGRx.FUNC defines which embedded peripheral drives the I/O lineFor more details, refer to the table Pin Description in section Package and Pinout.

Note that PIO_P_MSKRx and PIO_P_CFGRx must be used to program the FUNC field of a Privileged-Access mode I/O line.

For more details, see PIO_CFGRx or PIO_P_CFGRx for Privileged-Access mode I/O line configuration.

Note that multiplexing of peripheral lines affects both input and output peripheral lines. When a peripheral is not selected on any I/O line, its inputs are assigned with constant default values defined at the product level. The user must ensure that only one I/O line is affected to a peripheral input at a time.

The reset value of the FUNC field of each I/O line is defined at the product level and depends on the multiplexing of the device.

21.5.4. Output Control

When the I/O line is assigned to a peripheral function, i.e., the corresponding FUNC field of the line configuration is 1, the drive of the I/O line (direction, output value) is controlled by the peripheral.

When the FUNC field of a I/O line is 0, then the I/O line is set in General Purpose mode and the I/O line can be configured to be driven by the PIO Controller (software) instead of the peripheral.

© 2026 Microchip Technology Inc. and its subsidiaries

If PIO_CFGRx/PIO_P_CFGRx.DIR is configured in Output mode and PIO_CFGRx/PIO_P_CFGRx.FUNC=0, then the I/O line can be driven by the PIO Controller. The level driven on an I/O line can be determined by writing in the PIO Set Output Data Register (PIO_SODRx)/PIO Privilege Set Output Data Register (PIO_P_SODRx) and the PIO Clear Output Data Register (PIO_CODRx)PIO Privilege Clear Output Data Register (PIO_P_CODRx). These write operations, respectively, set and clear the PIO Output Data Status Register (PIO_ODSRx)/PIO Privilege Output Data Status Register (PIO_P_ODSRx), which represents the data driven on the I/O lines. Writing PIO_ODSRx/PIO_P_ODSRx directly is possible and only affects the I/O line set to 1 in PIO_MSKRx/PIO_P_MSKRx (see Synchronous Data Output).

When DIR of the I/O line configuration is at zero, the corresponding I/O line is used as an input only.

DIR has no effect if the corresponding line is assigned to a peripheral function, but writing DIR is managed whether the pin is configured to be controlled by the PIO Controller or assigned to a peripheral function. This enables configuration of the I/O line prior to setting it to be managed by the PIO Controller.

Similarly, writing in PIO_SODRx/PIO_P_SODRx and PIO_CODRx/PIO_P_CODRx affects PIO_ODSRx/PIO_P_ODSRx. This is important as it defines the first level driven on the I/O line.

21.5.5. Synchronous Data Output

Clearing one or more PIO line(s) and setting another one or more PIO line(s) synchronously cannot be done by using PIO_SODRx/PIO_P_SODRx and PIO_CODRx/PIO_P_CODRx. It requires two successive write operations into two different registers. To overcome this, the PIO Controller offers a direct control of PIO outputs by single write access to PIO_ODSRx/PIO_P_ODSRx. Only I/O lines set to 1 in PIO_MSKRx/PIO_P_MSKRx are written.

21.5.6. Open-Drain Mode

Each I/O can be independently programmed in Open-Drain mode. This feature permits several drivers to be connected on the I/O line which is driven low only by each device. An external pull-up resistor (or enabling of the internal one) is generally required to ensure a high level on the line.

The Open-Drain mode is controlled by the OPD bit in the I/O line configuration (PIO_CFGRx or PIO_P_CFGRx). An I/O line is switched in Open-Drain mode by setting the PIO_CFGRx/PIO_P_CFGRx.OPD bit. The Open-Drain mode can be selected if the I/O line is not controlled by a peripheral (the FUNC field must be cleared in PIO_CFGRx/PIO_P_CFGRx).

For more details concerning the Open-Drain mode, see PIO_CFGRx or PIO_P_CFGRx for Privileged-Access mode I/O line configuration.

After reset, the OPD bit of each I/O line is defined at the product level and depends on the multiplexing of the device.

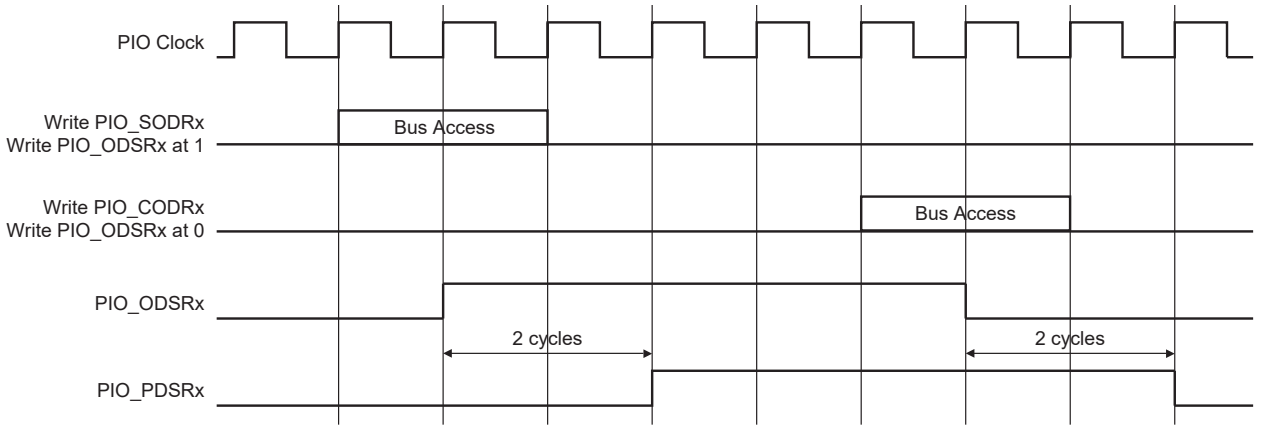
Note: Open-drain capability is not possible when the I/O line is driven by a peripheral. Only software control (GPIO mode) is able to manage the open-drain for an I/O line. TWI is able to manage open-drain because this peripheral does not require the PIO to be configured in Open-drain mode.

21.5.7. Output Line Timings

The figure below shows how the outputs are driven either by writing PIO_SODRx/PIO_P_SODRx or PIO_CODRx/PIO_P_CODRx, or by directly writing PIO_ODSRx/PIO_P_ODSRx. This last case is valid only if the corresponding bit in PIO_MSKRx/PIO_P_MSKRx is set. The figure also shows when the feedback in the Pin Data Status register (PIO_PDSRx/PIO_P_PDSRx) is available.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 21-3. Output Line Timings



21.5.8. Inputs

The level on each I/O line of the I/O group x can be read through PIO_PDSRx/PIO_P_PDSRx. This register indicates the level of the I/O lines regardless of their configuration, whether uniquely as an input, or driven by the PIO Controller, or driven by a peripheral.

Reading the I/O line levels requires the clock of the PIO Controller to be enabled, otherwise PIO_PDSRx/PIO_P_PDSRx reads the levels present on the I/O line at the time the clock was disabled.

21.5.9. Input Glitch and Debouncing Filters

Optional input glitch and debouncing filters are independently programmable on each I/O line.

The glitch filter can filter a glitch with a duration of less than 1 peripheral clock and the debouncing filter can filter a pulse of less than 1 period of a programmable divided slow clock.

The selection between glitch filtering or debounce filtering is done by writing the PIO_CFGR.IFSCEN. The selected filtering mode is applied to the I/O line(s) defined in PIO_MSKRx.

- If IFSCEN = 0: The glitch filter can filter a glitch with a duration of less than 1 peripheral clock period.
- If IFSCEN = 1: The debouncing filter can filter a pulse with a duration of less than 1 programmable divided slow clock period.

For the debouncing filter, the period of the divided slow clock is performed by writing in the DIV field of the PIO Privilege Slow Clock Divider Debouncing Register (PIO_P_SCDR): $t_{div\_slck} = ((DIV + 1) \times 2) \times t_{slck}$.

When the glitch or debouncing filter is enabled, a glitch or pulse with a duration of less than 1 selected clock cycle (selected clock represents PIO clock or divided slow clock depending on IFSCEN configuration) is automatically rejected. A pulse of a duration equal to 1 clock cycle may be filtered or not depending on the clock jitter.

The filters also introduce some latencies, illustrated in the figures below .

The glitch filter of each I/O line is controlled by PIO_CFGR.IFEN. Setting PIO_CFGRx.IFEN enables the glitch filter of the I/O line(s) defined in PIO_MSKRx.

When the glitch and/or debouncing filter is enabled, it does not modify the behavior of the inputs on the peripherals. It acts only on the value read in PIO_PDSRx and on the input change interrupt detection. The glitch and debouncing filters require that the PIO Controller clock is enabled.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 21-4. Input Glitch Filter Timing

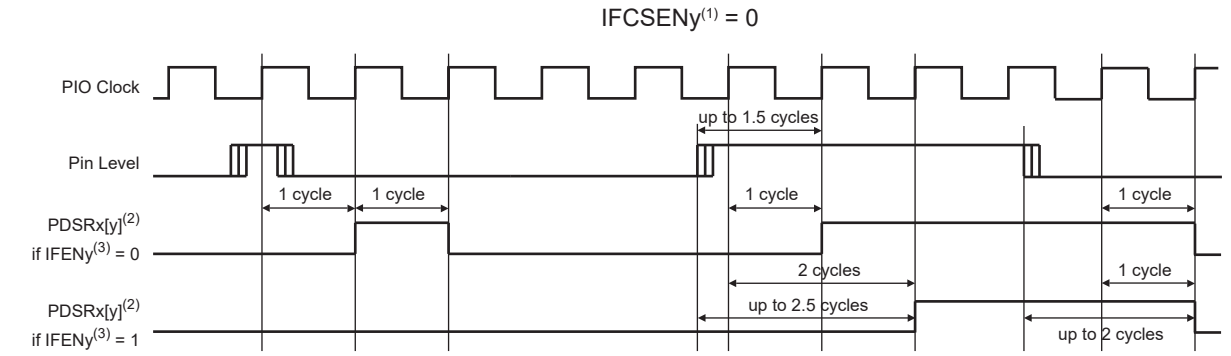
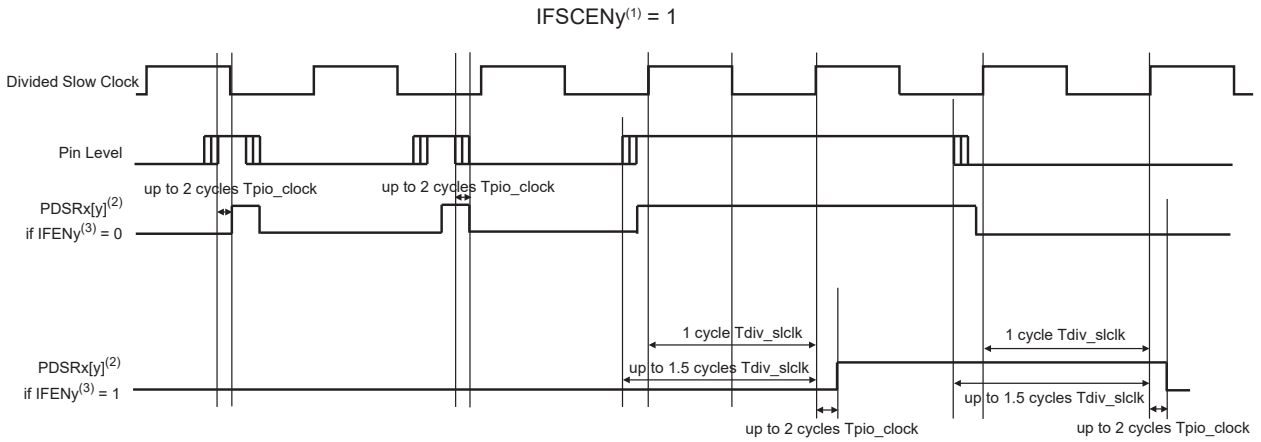


Figure 21-5. Input Debouncing Filter Timing



Note:

1. Means IFCSEN of the I/O line y of the I/O group x.
2. Means PIO Data Status value of the I/O line y of the I/O group x.
3. Means IFEN of the I/O line y of the I/O group x.

21.5.10. Input Edge/Level Interrupt

Each I/O group can be programmed to generate an interrupt when it detects an edge or a level on an I/O line. The Input Edge/Level interrupts are controlled by writing the PIO Interrupt Enable Register (PIO_IERx) and the PIO Interrupt Disable Register (PIO_IDRx), which enable and disable the input change interrupt respectively by setting and clearing the corresponding bit in the PIO Interrupt Mask Register (PIO_IMRx). For the Privileged-Access mode I/O lines, the Input Edge/Level interrupts are controlled by writing PIO_P_IERx and PIO_P_IDRx, which enable and disable input change interrupts, respectively, by setting and clearing the corresponding bit in the PIO_P_IMRx. As input change detection is possible only by comparing two successive samplings of the input of the I/O line, the PIO Controller clock must be enabled. The Input Change interrupt is available regardless of the configuration of the I/O line, i.e., configured as an input only, controlled by the PIO Controller or assigned to a peripheral function.

Each I/O group can generate a User-Access IO interrupt and a Privileged-Access IO interrupt according to the access level of the I/O line which triggers the interrupt.

According to the EVTSEL field value in PIO_CFGRx or PIO_P_CFGRx in case of a Privileged-Access mode I/O line, the interrupt signal of the I/O group x can be generated on the following occurrence:

- PIO_(P_)CFGRx.EVTSELy = 0: The interrupt signal of the I/O group x is generated on the I/O line y falling edge detection (assuming that PIO_(P_)IMRx[y] = 1).

© 2026 Microchip Technology Inc. and its subsidiaries

- PIO_(P_)CFGRx.EVTSELy = 1: The interrupt signal of the I/O group x is generated on the I/O line y rising edge detection (assuming that PIO_(P_)IMRx[y] = 1).
- PIO_(P_)CFGRx.EVTSELy = 2: The interrupt signal of the I/O group x is generated on the I/O line y both rising and falling edge detection (assuming that PIO_(P_)IMRx[y] = 1).
- PIO_(P_)CFGRx.EVTSELy = 3: The interrupt signal of the I/O group x is generated on the I/O line y low level detection (assuming that PIO_(P_)IMRx[y] = 1).
- PIO_(P_)CFGRx.EVTSELy = 4: The interrupt signal of the I/O group x is generated on the I/O line y high level detection (assuming that PIO_(P_)IMRx[y] = 1).

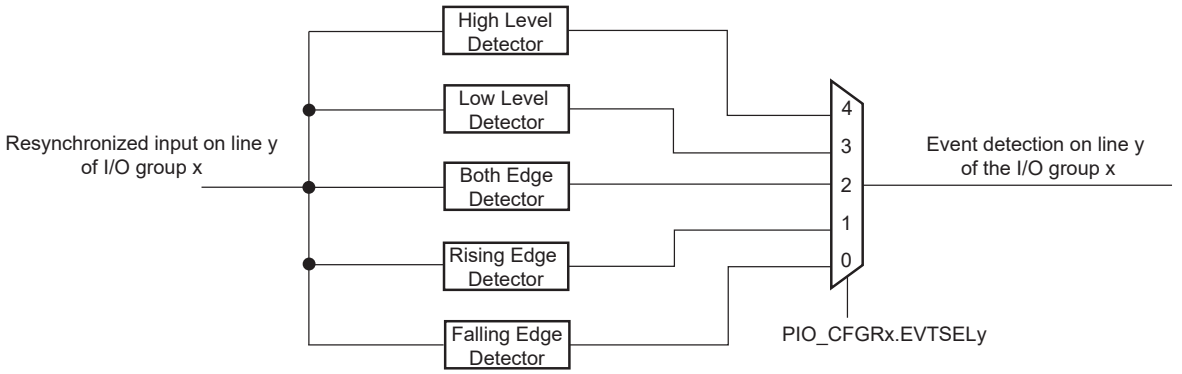
By default, the interrupt can be generated at any time a falling edge is detected on the input.

When an input edge or level is detected on an I/O line, the corresponding bit in the PIO Interrupt Status Register (PIO_ISRx), or in the PIO Privilege Interrupt Status Register (PIO_P_ISRx) if the I/O line access is privileged, is set.

For a User-Access mode I/O line, if the corresponding bit in PIO_IMRx is set, the User-Access mode I/O interrupt line of the I/O group x is asserted. For a Privileged-Access mode I/O line, if the corresponding bit in PIO_P_IMRx is set, the Privileged-Access mode I/O interrupt line of the I/O group x is asserted.

When the software reads PIO_ISRx, all the User-Access mode interrupts of the I/O group x are automatically cleared. When the software reads PIO_P_ISRx, all the interrupt sources related to Privileged-Access mode I/O lines of the I/O group x are automatically cleared. This signifies that all the interrupts that are pending when PIO_ISRx or PIO_P_ISRx are read must be handled. When an interrupt is enabled on a “level”, the interrupt is generated as long as the interrupt source is not cleared, even if some read accesses in PIO_ISRx or PIO_P_ISRx are performed.

Figure 21-6. Event Detector on Input Lines



Example of interrupt generation on following lines:

- Rising edge on the Privileged-Access PIO line 0 of the I/O group 0 (PIOA)
- Low-level edge on the Privileged-Access PIO line 1of the I/O group 0 (PIOA)
- Rising edge on the Privileged-Access PIO line 2 of the I/O group 0 (PIOA)
- High-level on the Privileged-Access PIO line 3 of the I/O group 0 (PIOA)
- Low-level on the User-Access PIO line 4 of the I/O group 0 (PIOA)
- High-level on the Privileged-Access PIO line 0 of the I/O group 1 (PIOB)
- Falling edge on the Privileged-Access PIO line 1 of the I/O group 1 (PIOB)
- Rising edge on the Privileged-Access PIO line 2 of the I/O group 1 (PIOB)
- Any edge on the other User-Access lines of the I/O group 1 (PIOB)

The table below details the required configuration for this example.

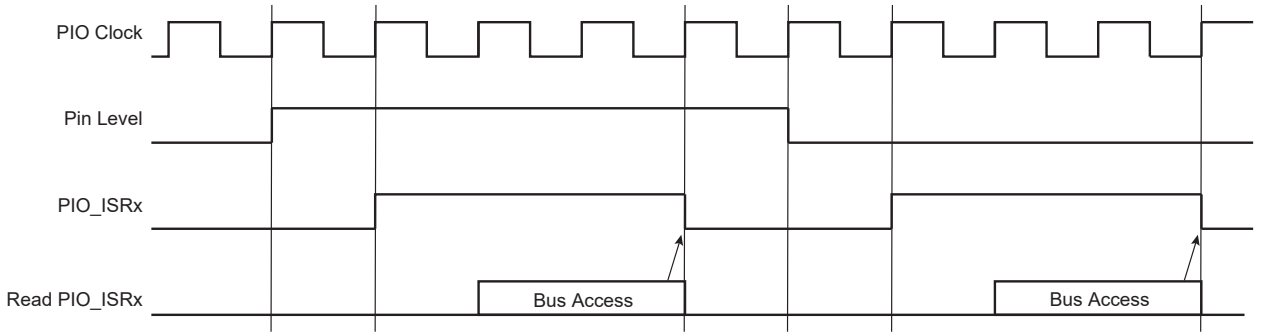
© 2026 Microchip Technology Inc. and its subsidiaries

Table 21-2. Configuration for Example Interrupt Generation

Configuration	Name
PIOA: I/O Line Security Level	Define the I/O lines 0 to 3 of the PIOA as Privileged-Access by writing 32'h0000_000F in the PIO_P_SIO_PAR0 (offset 0x1034) Define the I/O lines 4 of the PIOA as User-Access by writing 32'h0000_0010 in the PIO_P_SIO_UAR0 (offset 0x1030)
PIOA: Interrupt Mode	Enable interrupt sources for lines 0 to 3 of PIOA by writing 32'h0000_000F in PIO_P_IER0 (offset 0x1020) Enable interrupt source for the line 4 of PIOA by writing 32'h0000_0010 in PIO_IER0 (offset 0x20)
PIOA: Event Selection	Configure Rising Edge detection for Privileged-Access lines 0 and 2: Write 32'h0000_0005 in PIO_P_MSKR0 (offset 0x1000) Write 32'h0100_0000 in PIO_P_CFGR0 (offset 0x1004)
	Configure Low Level detection for Privileged-Access line 1: Write 32'h0000_0002 in PIO_P_MSKR0 (offset 0x1000) Write 32'h0300_0000 in PIO_P_CFGR0 (offset 0x1004)
	Configure High Level detection for Privileged-Access line 3: Write 32'h0000_0008 in PIO_P_MSKR0 (offset 0x1000) Write 32'h0400_0000 in PIO_P_CFGR0 (offset 0x1004)
	Configure Low Level detection for User-Access line 4: Write 32'h0000_0010 in PIO_MSKR0 (offset 0x0) Write 32'h0300_0000 in PIO_CFGR0 (offset 0x4)
PIOB: I/O Line Security Level	Define the I/O lines 0 to 2 of the PIOB as Privileged-Access by writing 32'h0000_0007 in the PIO_P_SIO_PAR1 (offset 0x1074) Define the other I/O lines of the PIOB as User-Access by writing 32'hFFFF_FFF8 in the PIO_P_SIO_UAR1 (offset 0x1070)
PIOB: Interrupt Mode	Enable interrupt sources for lines 0 to 2 of PIOB by writing 32'h0000_0007 in PIO_P_IER1 (offset 0x1060) Enable interrupt sources for all other lines of PIOB by writing 32'hFFFF_FFF8 in PIO_IER1 (offset 0x60)
PIOB: Event Selection	Configure High Level detection for Privileged-Access line 0: Write 32'h0000_0001 in PIO_P_MSKR1 (offset 0x1040) Write 32'h0400_0000 in PIO_P_CFGR1 (offset 0x1044)
	Configure Falling Edge detection for Privileged-Access line 1: Write 32'h0000_0002 in PIO_P_MSKR1 (offset 0x1040) Write 32'h0000_0000 in PIO_P_CFGR1 (offset 0x1044)
	Configure Rising Edge detection for Privileged-Access line 2: Write 32'h0000_0004 in PIO_P_MSKR1 (offset 0x1040) Write 32'h0100_000 in PIO_P_CFGR1 (offset 0x1044)
	Configure Low Level detection for User-Access lines: Write 32'hFFFF_FFF8 in PIO_MSKR1 (offset 0x40) Write 32'h0200_000 in PIO_CFGR1 (offset 0x44)

© 2026 Microchip Technology Inc. and its subsidiaries

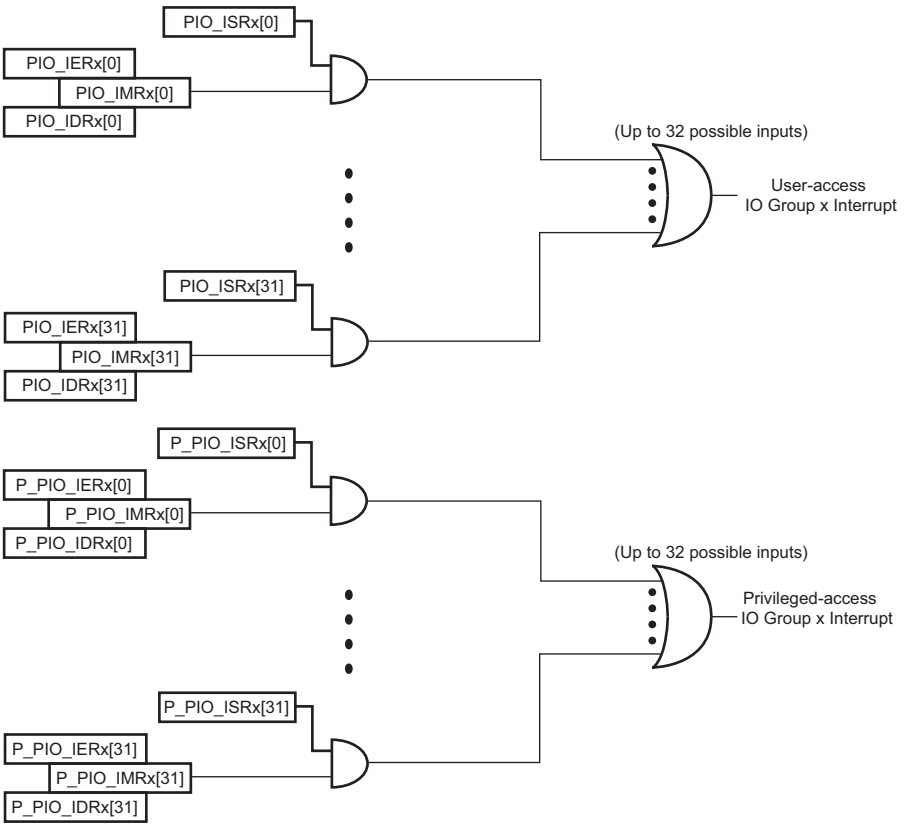
Figure 21-7. Input Change Interrupt Timings When No Additional Interrupt Modes



21.5.11. Interrupt Management

The PIO Controller drives one interrupt line reserved for Privileged-Access mode I/O lines and a second interrupt line reserved for User-Access mode I/O lines per I/O group (refer to the Block Diagram). Separate lines ease management of interrupt priorities related to Privileged-Access mode I/O lines.

Figure 21-8. PIO Interrupt Management



21.5.12. I/O Lines Lock

When an I/O line is controlled by a peripheral (particularly the Pulse Width Modulation Controller PWM), it can become locked by the action of this peripheral via an input of the PIO Controller. When an I/O line is locked, the following fields in PIO_CFGRx/PIO_P_CFGRx are locked and cannot be modified:

- FUNC: Peripheral selection cannot be changed when the corresponding I/O line is locked.

© 2026 Microchip Technology Inc. and its subsidiaries

- PUEN: Pull-Up configuration cannot be changed when the corresponding I/O line is locked.
- PDEN: Pull-Down configuration cannot be changed when the corresponding I/O line is locked.
- OPD: Open Drain configuration cannot be changed when the corresponding I/O line is locked.

Writing to one of these fields while the corresponding I/O line is locked will have no effect.

The user can know at anytime which I/O line is locked by reading the PIO Lock Status Register (PIO_LOCKSR) or PIO Privilege Lock Status Register (PIO_P_LOCKSR) for locked Privileged-Access mode I/O lines. Once an I/O line is locked, the only way to unlock it is to apply a hardware reset to the PIO Controller.

21.5.13. Programmable Schmitt Trigger

It is possible to configure each input for the Schmitt trigger. The Schmitt trigger can be enabled by setting PIO_CFGRx.SCHMITT if the corresponding line is configured in User-Access mode or PIO_P_CFGRx for Privileged-Access mode I/O lines. By default, the Schmitt trigger is active.

21.5.14. Programmable Slew Rate

Each input can be configured for slew rate control. Slew rate control can be configured in the field SLEWRATE of PIO_CFGRx if the corresponding line is configured in User-Access mode or PIO_P_CFGRx for Privileged-Access mode I/O lines. For details, refer to the section “Electrical Characteristics”.

21.5.15. I/O Line Configuration Freeze

21.5.15.1. Introduction

The I/O line configuration freeze function can reinforce the protection against the effects of an abnormal access resulting from a Single-event upset that may corrupt the value of one bit on the system bus during an access to the PIO or any other peripheral. Freezing the configuration of an I/O line prevents an unexpected access from modifying the configuration of the I/O line. Once the freeze is done, the I/O line configuration cannot be modified whatever software sequence is performed on the PIO.

21.5.15.2. Software Freeze

Once the I/O line configuration is done, it can be frozen by using the PIO I/O Freeze Configuration Register (PIO_IOFRx) of the corresponding group if the corresponding line is configured in User-Access mode or the PIO Privilege I/O Freeze Configuration Register (PIO_P_IOFRx) for Privileged-Access mode I/O lines.

21.5.15.2.1. Physical Freeze

Setting PIO_IOFR.FPHY freezes the following fields (configured in PIO_CFGRx) of the I/O lines if the corresponding MSKx bit is set in PIO_MSKRx:

- FUNC: I/O Line Function
- DIR: Direction
- PUEN: Pull-Up Enable
- PDEN: Pull-Down Enable
- OPD: Open-Drain
- SCHMITT: Schmitt Trigger
- SLEWRATE: Slew Rate

For Privileged-Access mode I/O lines, use PIO_P_IOFRx.FPHY and the PIO_P_MSKRx to freeze the fields above.

When the physical freeze is currently active on an I/O line, the PCFS flag is set when reading the PIO_CFGRx of the I/O line if the corresponding line is configured in User-Access mode or the PIO_P_CFGRx for the Privileged-Access mode I/O line .

© 2026 Microchip Technology Inc. and its subsidiaries

Only a hardware reset can release fields listed above.

21.5.15.2.2. Interrupt Freeze

Setting PIO_IOFRx.FINT freezes the following fields (configured in PIO_CFGRx) of the I/O lines if the corresponding MSKx bit is set in PIO_MSKRx:

- IFEN: Input Filter Enable
- IFSCEN: Input Filter Slow Clock Enable
- EVTSEL: Event Selection

For Privileged-Access mode I/O lines, use PIO_P_IOFRx.FINT and the PIO_P_MSKRx to freeze the fields above.

When the “Interrupt Freeze” is currently active on an I/O line, the ICFS flag is set when reading the PIO_CFGRx of the I/O line if the corresponding line is configured in User-Access mode or the PIO_P_CFGRx if the I/O line access is configured in Privileged-Acces mode .

Only a hardware reset can release fields listed above.

21.5.16. Register Write Protection

To prevent any single software error from corrupting PIO behavior, certain registers in the address space can be write-protected by setting WPEN and WPITEN in the PIO Write Protection Mode Register (PIO_WPMR) or the PIO Privilege Write Protection Mode Register (PIO_P_WPMR).

If a write access to a User-Accesswrite-protected register is detected, the WPVS flag in the PIO Write Protection Status Register (PIO_WPSR) is set and the field WPVSRC indicates the register in which the write access has been attempted.

If a write access to a privileged-access write-protected register is detected, the WPVS flag in the PIO Privilege Write Protection Status Register (PIO_P_WPSR) is set and the field WPVSRC indicates the register in which the write access has been attempted.

The respective WPVS bit is automatically cleared after reading the PIO_WPSR or PIO_P_WPSR.

The following registers are write-protected when WPEN is set in PIO_WPMR:

- PIO Mask Register
- PIO Configuration Register

The following registers are write-protected when WPEN is set in PIO_P_WPMR:

- PIO Privilege Mask Register
- PIO Privilege Configuration Register
- PIO Privilege Slow Clock Divider Debouncing Register

The following registers are write-protected when WPITEN is set in PIO_WPMR:

- PIO Interrupt Enable Register
- PIO Interrupt Disable Register

The following registers are write-protected when WPITEN is set in PIO_P_WPMR:

- PIO Privilege Interrupt Enable Register
- PIO Privilege Interrupt Disable Register

21.6. I/O Lines Programming Example

The programming example shown in the table below is used to obtain the following configurations:

- PIOA Configuration:
  - 4-bit output port on Privileged-Access mode I/O lines 0 to 3, open-drain, with pull-up resistor

© 2026 Microchip Technology Inc. and its subsidiaries

  - Four output signals on User-Access mode I/O lines 4 to 7 (to drive LEDs for example), driven high and low, no pull-up resistor, no pull-down resistor
  - Privileged-Access modeI/O lines 16 to 19 assigned to peripheral A functions with pull-up resistor
  - User-Access modeI/O lines 20 to 23 assigned to peripheral B functions with pull-down resistor
- PIOB Configuration:
  - Four input signals on Privileged-Access mode I/O lines 0 to 3 (to read push-button states for example), with pull-up resistors, glitch filters and input change interrupts
  - Four input signals on User-Access mode I/O lines 12 to 15 to read an external device status (polled, thus no input change interrupt), no pull-up resistor, no glitch filter
  - Privileged-Access modeI/O lines 16 to 23 assigned to peripheral B functions with pull-down resistor
  - User-Access modeI/O lines 24 to 27 assigned to peripheral D with Input Change Interrupt, no pull-up resistor and no pull-down resistor

Table 21-3. Programming Example

Action	Register	Value to be Written
PIOA: Set I/O lines 0 to 3 and 16 to 19 in Privileged-Access mode	PIO_P_SIO_PAR0 (offset 0x1034)	0x000F000F
PIOA: Set I/O lines 4 to 7 and 20 to 23 in User-Access mode	PIO_P_SIO_UAR0 (offset 0x1030)	0x00F000F0
PIOA: 4-bit output port on Privileged-Access mode I/O lines 0 to 3, open-drain, with pull-up resistor	PIO_P_MSKR0 (offset 0x1000)	0x0000000F
	PIO_P_CFGR0 (offset 0x1004)	0x00004300
PIOA: Four output signals on User-Access mode I/O lines 4 to 7 (to drive LEDs for example), driven high and low, no pull-up resistor, no pull-down resistor	PIO_MSKR0 (offset 0x0)	0x000000F0
	PIO_CFGR0 (offset 0x4)	0x00000100
PIOA: Privileged-Access mode I/O lines 16 to 19 assigned to peripheral A functions with pull-up resistor	PIO_P_MSKR0 (offset 0x1000)	0x000F0000
	PIO_P_CFGR0 (offset 0x1004)	0x00000201
PIOA: User-Access mode I/O lines 20 to 23 assigned to peripheral B functions with pull-down resistor	PIO_MSKR0 (offset 0x0)	0x00F00000
	PIO_CFGR0 (offset 0x4)	0x00000402
PIOB: Set I/O lines 0 to 3 and 16 to 23 in Privileged-Access mode	PIO_P_SIO_PAR1 (offset 0x1074)	0x00FF000F
PIOB: Set I/O lines 12 to 15 and 24 to 27 in User-Access mode	PIO_P_SIO_UAR1 (offset 0x1070)	0x0F00F000
PIOB: Four input signals on Privileged-Access mode I/O lines 0 to 3 (to read push-button states for example), with pull-up resistors, glitch filters and interrupts on rising edge	PIO_P_MSKR1 (offset 0x1040)	0x0000000F
	PIO_P_CFGR1 (offset 0x1044)	0x01001200

© 2026 Microchip Technology Inc. and its subsidiaries

Table 21-3. Programming Example (continued)

Action	Register	Value to be Written
PIOB: Four input signals on User-Access mode I/O line 12 to 15 to read an external device status (polled, thus no input change interrupt), no pull-up resistor, no glitch filter	PIO_MSKR1 (offset 0x40)	0x0000F000
	PIO_CFGR1 (offset 0x44)	0x01001200
PIOB: Privileged-Access mode I/O lines 16 to 23 assigned to peripheral B functions with pull-down resistor	PIO_P_MSKR1 (offset 0x1040)	0x00FF0000
	PIO_P_CFGR1 (offset 0x1044)	0x00000402
PIOB: User-Access mode I/O line 24 to 27 assigned to peripheral D with Input Interrupt on both edges, no pull-up resistor and no pull-down resistor	PIO_MSKR1 (offset 0x40)	0x0F000000
	PIO_CFGR1 (offset 0x44)	0x02000004
PIOB: Enable interrupt	PIO_P_IER1 (offset 0x1060)	0x0000000F
	PIO_IER1 (offset 0x60)	0x0F000000

© 2026 Microchip Technology Inc. and its subsidiaries

21.7. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	PIO_MSKR0	31:24	MSK31	MSK30	MSK29	MSK28	MSK27	MSK26	MSK25	MSK24
		23:16	MSK23	MSK22	MSK21	MSK20	MSK19	MSK18	MSK17	MSK16
		15:8	MSK15	MSK14	MSK13	MSK12	MSK11	MSK10	MSK9	MSK8
		7:0	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0
0x04	PIO_CFGR0	31:24		ICFS	PCFS			EVTSEL[2:0]		
		23:16							SLEWRATE[1:0]	
		15:8	SCHMITT	OPD	IFSCEN	IFEN		PDEN	PUEN	DIR
		7:0						FUNC[2:0]		
0x08	PIO_PDSR0	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x0C	PIO_LOCKSR0	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x10	PIO_SODR0	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x14	PIO_CODR0	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x18	PIO_ODSR0	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x1C ... 0x1F	Reserved									
0x20	PIO_IER0	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x24	PIO_IDR0	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x28	PIO_IMR0	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x2C	PIO_ISR0	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x30 ... 0x3B	Reserved									
0x3C	PIO_IOFR0	31:24	FRZKEY[23:16]							
		23:16	FRZKEY[15:8]							
		15:8	FRZKEY[7:0]							
		7:0							FINT	FPHY
0x40	PIO_MSKR1	31:24	MSK31	MSK30	MSK29	MSK28	MSK27	MSK26	MSK25	MSK24
		23:16	MSK23	MSK22	MSK21	MSK20	MSK19	MSK18	MSK17	MSK16
		15:8	MSK15	MSK14	MSK13	MSK12	MSK11	MSK10	MSK9	MSK8
		7:0	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0

© 2026 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x44	PIO_CFGR1	31:24		ICFS	PCFS			EVTSEL[2:0]		
		23:16							SLEWRATE[1:0]	
		15:8	SCHMITT	OPD	IFSCEN	IFEN		PDEN	PUEN	DIR
		7:0						FUNC[2:0]		
0x48	PIO_PDSR1	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x4C	PIO_LOCKSR1	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x50	PIO_SODR1	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x54	PIO_CODR1	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x58	PIO_ODSR1	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x5C ... 0x5F	Reserved									
0x60	PIO_IER1	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x64	PIO_IDR1	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x68	PIO_IMR1	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x6C	PIO_ISR1	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x70 ... 0x7B	Reserved									
0x7C	PIO_IOFR1	31:24	FRZKEY[23:16]							
		23:16	FRZKEY[15:8]							
		15:8	FRZKEY[7:0]							
		7:0							FINT	FPHY
0x80	PIO_MSKR2	31:24	MSK31	MSK30	MSK29	MSK28	MSK27	MSK26	MSK25	MSK24
		23:16	MSK23	MSK22	MSK21	MSK20	MSK19	MSK18	MSK17	MSK16
		15:8	MSK15	MSK14	MSK13	MSK12	MSK11	MSK10	MSK9	MSK8
		7:0	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0
0x84	PIO_CFGR2	31:24		ICFS	PCFS			EVTSEL[2:0]		
		23:16							SLEWRATE[1:0]	
		15:8	SCHMITT	OPD	IFSCEN	IFEN		PDEN	PUEN	DIR
		7:0						FUNC[2:0]		

© 2026 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x88	PIO_PDSR2	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x8C	PIO_LOCKSR2	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x90	PIO_SODR2	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x94	PIO_CODR2	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x98	PIO_ODSR2	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x9C ... 0x9F	Reserved									
0xA0	PIO_IER2	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0xA4	PIO_IDR2	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0xA8	PIO_IMR2	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0xAC	PIO_ISR2	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0xB0 ... 0xBB	Reserved									
0xBC	PIO_IOFR2	31:24	FRZKEY[23:16]							
		23:16	FRZKEY[15:8]							
		15:8	FRZKEY[7:0]							
		7:0							FINT	FPHY
0xC0 ... 0x05DF	Reserved									
0x05E0	PIO_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0							WPITEN	WPEN
0x05E4	PIO_WPSR	31:24								
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0								WPVS
0x05E8 ... 0x0FFF	Reserved									

© 2026 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x1000	PIO_P_MSKR0	31:24	MSK31	MSK30	MSK29	MSK28	MSK27	MSK26	MSK25	MSK24
		23:16	MSK23	MSK22	MSK21	MSK20	MSK19	MSK18	MSK17	MSK16
		15:8	MSK15	MSK14	MSK13	MSK12	MSK11	MSK10	MSK9	MSK8
		7:0	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0
0x1004	PIO_P_CFGR0	31:24		ICFS	PCFS			EVTSEL[2:0]		
		23:16							SLEWRATE[1:0]	
		15:8	SCHMITT	OPD	IFSCEN	IFEN		PDEN	PUEN	DIR
		7:0						FUNC[2:0]		
0x1008	PIO_P_PDSR0	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x100C	PIO_P_LOCKSR0	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x1010	PIO_P_SODR0	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x1014	PIO_P_CODR0	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x1018	PIO_P_ODSR0	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x101C ... 0x101F	Reserved									
0x1020	PIO_P_IER0	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x1024	PIO_P_IDR0	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x1028	PIO_P_IMR0	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x102C	PIO_P_ISR0	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x1030	PIO_P_SIO_UAR0	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x1034	PIO_P_SIO_PAR0	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x1038	PIO_P_IOSSR0	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0

© 2026 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x103C	PIO_P_IOFR0	31:24	FRZKEY[23:16]							
		23:16	FRZKEY[15:8]							
		15:8	FRZKEY[7:0]							
		7:0							FINT	FPHY
0x1040	PIO_P_MSKR1	31:24	MSK31	MSK30	MSK29	MSK28	MSK27	MSK26	MSK25	MSK24
		23:16	MSK23	MSK22	MSK21	MSK20	MSK19	MSK18	MSK17	MSK16
		15:8	MSK15	MSK14	MSK13	MSK12	MSK11	MSK10	MSK9	MSK8
		7:0	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0
0x1044	PIO_P_CFGR1	31:24		ICFS	PCFS			EVTSEL[2:0]		
		23:16							SLEWRATE[1:0]	
		15:8	SCHMITT	OPD	IFSCEN	IFEN		PDEN	PUEN	DIR
		7:0						FUNC[2:0]		
0x1048	PIO_P_PDSR1	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x104C	PIO_P_LOCKSR1	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x1050	PIO_P_SODR1	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x1054	PIO_P_CODR1	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x1058	PIO_P_ODSR1	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x105C ... 0x105F	Reserved									
0x1060	PIO_P_IER1	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x1064	PIO_P_IDR1	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x1068	PIO_P_IMR1	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x106C	PIO_P_ISR1	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x1070	PIO_P_SIO_UAR1	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x1074	PIO_P_SIO_PAR1	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0

© 2026 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x1078	PIO_P_IOSSR1	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x107C	PIO_P_IOFR1	31:24	FRZKEY[23:16]							
		23:16	FRZKEY[15:8]							
		15:8	FRZKEY[7:0]							
		7:0							FINT	FPHY
0x1080	PIO_P_MSKR2	31:24	MSK31	MSK30	MSK29	MSK28	MSK27	MSK26	MSK25	MSK24
		23:16	MSK23	MSK22	MSK21	MSK20	MSK19	MSK18	MSK17	MSK16
		15:8	MSK15	MSK14	MSK13	MSK12	MSK11	MSK10	MSK9	MSK8
		7:0	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0
0x1084	PIO_P_CFGR2	31:24		ICFS	PCFS			EVTSEL[2:0]		
		23:16							SLEWRATE[1:0]	
		15:8	SCHMITT	OPD	IFSCEN	IFEN		PDEN	PUEN	DIR
		7:0						FUNC[2:0]		
0x1088	PIO_P_PDSR2	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x108C	PIO_P_LOCKSR2	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x1090	PIO_P_SODR2	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x1094	PIO_P_CODR2	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x1098	PIO_P_ODSR2	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x109C ... 0x109F	Reserved									
0x10A0	PIO_P_IER2	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x10A4	PIO_P_IDR2	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x10A8	PIO_P_IMR2	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x10AC	PIO_P_ISR2	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x10B0	PIO_P_SIO_UAR2	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0

© 2026 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x10B4	PIO_P_SIO_PAR2	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x10B8	PIO_P_IOSSR2	31:24	P31	P30	P29	P28	P27	P26	P25	P24
		23:16	P23	P22	P21	P20	P19	P18	P17	P16
		15:8	P15	P14	P13	P12	P11	P10	P9	P8
		7:0	P7	P6	P5	P4	P3	P2	P1	P0
0x10BC	PIO_P_IOFR2	31:24	FRZKEY[23:16]							
		23:16	FRZKEY[15:8]							
		15:8	FRZKEY[7:0]							
		7:0							FINT	FPHY
0x10C0 ... 0x14FF	Reserved									
0x1500	PIO_P_SCDR	31:24								
		23:16								
		15:8			DIV[13:8]					
		7:0	DIV[7:0]							
0x1504 ... 0x15DF	Reserved									
0x15E0	PIO_P_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0							WPITEN	WPEN
0x15E4	PIO_P_WPSR	31:24								
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0								WPVS

© 2026 Microchip Technology Inc. and its subsidiaries

21.7.1. PIO Mask Register

Name: PIO_MSKRx
Offset: 0x00 + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the PIO Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	MSK31	MSK30	MSK29	MSK28	MSK27	MSK26	MSK25	MSK24
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	MSK23	MSK22	MSK21	MSK20	MSK19	MSK18	MSK17	MSK16
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	MSK15	MSK14	MSK13	MSK12	MSK11	MSK10	MSK9	MSK8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – MSKy PIO Line y Mask

These bits define the I/O lines to be configured when writing the PIO Configuration Register.
0 (DISABLED): Writing the PIO_CFGRx, PIO_ODSRx or PIO_IOFRx does not affect the corresponding I/O line configuration.
1 (ENABLED): Writing the PIO_CFGRx, PIO_ODSRx or PIO_IOFRx updates the corresponding I/O line configuration.

© 2026 Microchip Technology Inc. and its subsidiaries

21.7.2. PIO Configuration Register

Name: PIO_CFGRx
Offset: 0x04 + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the PIO Write Protection Mode Register.

Writing this register will only affect I/O lines enabled in the PIO_MSKRx.

Bit	31	30	29	28	27	26	25	24
		ICFS	PCFS			EVTSEL[2:0]		
Access		R	R			R/W	R/W	R/W
Reset		0	0			0	0	0

Bit	23	22	21	20	19	18	17	16
							SLEWRATE[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	15	14	13	12	11	10	9	8
	SCHMITT	OPD	IFSCEN	IFEN		PDEN	PUEN	DIR
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bit	7	6	5	4	3	2	1	0
						FUNC[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 30 – ICFS Interrupt Configuration Freeze Status (read-only)
Gives information about the freeze state of the following fields of the read I/O line configuration:
• IFEN: Input Filter Enable
• IFSCEN: Input Filter Slow Clock Enable
• EVTSEL: Event Selection
0 (NOT_FROZEN): The fields are not frozen and can be written for this I/O line.
1 (FROZEN): The fields are frozen and cannot be written for this I/O line. Only a hardware reset can release these fields.

Bit 29 – PCFS Physical Configuration Freeze Status (read-only)
Gives information about the freeze state of the following fields of the read I/O line configuration:
• FUNC: I/O Line Function
• DIR: Direction
• PUEN: Pull-Up Enable
• PDEN: Pull-Down Enable
• OPD: Open-Drain
• SCHMITT: Schmitt Trigger
• SLEWRATE: Slew Rate
0 (NOT_FROZEN): The fields are not frozen and can be written for this I/O line.
1 (FROZEN): The fields are frozen and cannot be written for this I/O line. Only a hardware reset can release these fields.

© 2026 Microchip Technology Inc. and its subsidiaries

Bits 26:24 – EVTSEL[2:0] Event Selection

Defines the type of event to detect on the I/O lines of the I/O group x according to the PIO Mask Register.

Value	Name	Description
0	FALLING	Event detection on input falling edge
1	RISING	Event detection on input rising edge
2	BOTH	Event detection on input both edge
3	LOW	Event detection on low level input
4	HIGH	Event detection on high level input
5	–	Reserved
6	–	Reserved
7	–	Reserved

Bits 17:16 – SLEWRATE[1:0] Slew Rate

Defines the slew rate of the I/O lines of the I/O group x according to the PIO Mask Register.

Value	Name	Description
0	FAST	Fast slew rate
1	MEDIUMFAST	Medium-fast slew rate
2	MEDIUM	Medium slew rate
3	SLOW	High slew rate

Bit 15 – SCHMITT Schmitt Trigger

Defines the Schmitt trigger configuration of the I/O lines of the I/O group x according to the PIO Mask Register.
0 (ENABLED): Schmitt trigger is enabled for the selected I/O lines.
1 (DISABLED): Schmitt trigger is disabled for the selected I/O lines.

Bit 14 – OPD Open Drain

Defines the open drain configuration of the I/O lines of the I/O group x according to the PIO Mask Register.
0 (DISABLED): The open-drain is disabled for the selected I/O lines. I/O lines are driven at high- and low-level.
1 (ENABLED): The open-drain is enabled for the selected I/O lines. I/O lines are driven at low-level only.

Bit 13 – IFSCEN Input Filter Slow Clock Enable

Defines the clock source of the glitch filtering for the I/O lines of the I/O group x according to the PIO Mask Register.
0 (DISABLED): The glitch filter is able to filter glitches with a duration less than 1 peripheral clock cycle for the selected I/O lines.
1 (ENABLED): The debouncing filter is able to filter pulses with a duration less than 1 divided slow clock cycle for the selected I/O lines.

Bit 12 – IFEN Input Filter Enable

Defines if the glitch filtering is used for the I/O lines of the I/O group x according to the PIO Mask Register.
0 (DISABLED): The input filter is disabled for the selected I/O lines.
1 (ENABLED): The input filter is enabled for the selected I/O lines.

Bit 10 – PDEN Pull-Down Enable

Defines the pull-down configuration of the I/O lines of the I/O group x according to the PIO Mask Register.
PDEN can be written to 1 only if PUEN is written to 0.
0 (DISABLED): Pull-down is disabled for the selected I/O lines.

© 2026 Microchip Technology Inc. and its subsidiaries

1 (ENABLED): Pull-down is enabled for the selected I/O lines only if PUEN is 0.

Bit 9 – PUEN Pull-Up Enable

Defines the pull-up configuration of the I/O lines of the I/O group x according to the PIO Mask Register.
0 (DISABLED): Pull-up is disabled for the selected I/O lines.
1 (ENABLED): Pull-up is enabled for the selected I/O lines.

Bit 8 – DIR Direction

Defines the direction of the I/O lines of the I/O group x according to the PIO Mask Register.
0 (INPUT): The selected I/O lines are pure inputs.
1 (OUTPUT): The selected I/O lines are enabled in output.

Bits 2:0 – FUNC[2:0] I/O Line Function

Defines the function for I/O lines of the I/O group x according to the PIO Mask Register.

Value	Name	Description
0	GPIO	Selects the PIO mode for the selected I/O lines.
1	PERIPH_A	Selects peripheral A for the selected I/O lines.
2	PERIPH_B	Selects peripheral B for the selected I/O lines.
3	PERIPH_C	Selects peripheral C for the selected I/O lines.
4	PERIPH_D	Selects peripheral D for the selected I/O lines.

© 2026 Microchip Technology Inc. and its subsidiaries

21.7.3. PIO Pin Data Status Register

Name: PIO_PDSRx
Offset: 0x08 + x*0x40 [x=0..2]
Reset: –
Property: Read-only

Reset value of PIO_PDSR depends on the level of the I/O lines. Reading the I/O line levels requires the clock of the PIO Controller to be enabled, otherwise PIO_PDSR reads the levels present on the I/O line at the time the clock was disabled.

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	R	R	R	R	R	R	R	R
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	R	R	R	R	R	R	R	R
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	R	R	R	R	R	R	R	R
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	R	R	R	R	R	R	R	R
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – Px Input Data Status

Value	Description
0	The I/O line of the I/O group x is at level 0.
1	The I/O line of the I/O group x is at level 1.

© 2026 Microchip Technology Inc. and its subsidiaries

21.7.4. PIO Lock Status Register

Name: PIO_LOCKSRx
Offset: 0x0C + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – Px Lock Status

Value	Description
0	The I/O line of the I/O group x is not locked.
1	The I/O line of the I/O group x is locked.

© 2026 Microchip Technology Inc. and its subsidiaries

21.7.5. PIO Set Output Data Register

Name: PIO_SODRx
Offset: 0x10 + x*0x40 [x=0..2]
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – Px Set Output Data

Value	Description
0	No effect.
1	Sets the data to be driven on the I/O line of I/O group x.

© 2026 Microchip Technology Inc. and its subsidiaries

21.7.6. PIO Clear Output Data Register

Name: PIO_CODRx
Offset: 0x14 + x*0x40 [x=0..2]
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – Px Clear Output Data

Value	Description
0	No effect.
1	Clears the data to be driven on the I/O line of the I/O group x.

© 2026 Microchip Technology Inc. and its subsidiaries

21.7.7. PIO Output Data Status Register

Name: PIO_ODSRx
Offset: 0x18 + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read/Write

Writing this register will only affect I/O lines enabled in the PIO_MSKRx.

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – Px Output Data Status

Value	Description
0	The data to be driven on the I/O line of the I/O group x is 0.
1	The data to be driven on the I/O line of the I/O group x is 1.

© 2026 Microchip Technology Inc. and its subsidiaries

21.7.8. PIO Interrupt Enable Register

Name: PIO_IERx
Offset: 0x20 + x*0x40 [x=0..2]
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the PIO Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – Px Input Change Interrupt Enable

Value	Description
0	No effect.
1	Enables the Input Change interrupt on the I/O line of the I/O group x.

© 2026 Microchip Technology Inc. and its subsidiaries

21.7.9. PIO Interrupt Disable Register

Name: PIO_IDRx
Offset: 0x24 + x*0x40 [x=0..2]
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the PIO Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – Px Input Change Interrupt Disable

Value	Description
0	No effect.
1	Disables the Input Change interrupt on the I/O line of the I/O group x.

© 2026 Microchip Technology Inc. and its subsidiaries

21.7.10. PIO Interrupt Mask Register

Name: PIO_IMRx
Offset: 0x28 + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – Px Input Change Interrupt Mask

Value	Description
0	Input Change interrupt is disabled on the I/O line of the I/O group x.
1	Input Change interrupt is enabled on the I/O line of the I/O group x.

© 2026 Microchip Technology Inc. and its subsidiaries

21.7.11. PIO Interrupt Status Register

Name: PIO_ISRx
Offset: 0x2C + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read-only

PIO_ISR is reset at 0x00000000. However, the first read of the register may read a different value as input changes may have occurred.

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – Px Input Change Interrupt Status

Value	Description
0	No Input Change has been detected on the I/O line of the I/O group x since PIO_ISRx was last read or since reset.
1	At least one Input Change has been detected on the I/O line of the I/O group since PIO_ISRx was last read or since reset.

© 2026 Microchip Technology Inc. and its subsidiaries

21.7.12. PIO I/O Freeze Configuration Register

Name: PIO_IOFRx
Offset: 0x3C + x*0x40 [x=0..2]
Reset: –
Property: Write-only

Writing this register will only affect I/O lines enabled in the PIO_MSKRx.

Bit	31	30	29	28	27	26	25	24
	FRZKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	FRZKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	FRZKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
							FINT	FPHY
Access							W	W
Reset							–	–

Bits 31:8 – FRZKEY[23:0] Freeze Key

Value	Name	Description
0x494F46	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit.

Bit 1 – FINT Freeze Interrupt Configuration
Only a hardware reset can reset the FINT bit.

Value	Description
0	No effect.
1	Freezes the following configuration fields if FRZKEY corresponds to 0x494F46 (“IOF” in ASCII): • IFEN: Input Filter Enable • IFSCEN: Input Filter Slow Clock Enable • EVTSEL: Event Selection

Bit 0 – FPHY Freeze Physical Configuration
Only a hardware reset can reset the FPHY bit.

Value	Description
0	No effect.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
1	Freezes the following configuration fields if FRZKEY corresponds to 0x494F46 (“IOF” in ASCII): • FUNC: I/O Line Function • DIR: Direction • PUEN: Pull-Up Enable • PDEN: Pull-Down Enable • OPD: Open-Drain • SCHMITT: Schmitt Trigger • SLEWRATE: Slew Rate

© 2026 Microchip Technology Inc. and its subsidiaries

21.7.13. PIO Write Protection Mode Register

Name: PIO_WPMR
Offset: 0x5E0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
							WPITEN	WPEN
Access							R/W	R/W
Reset							0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x50494F	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0.

Bit 1 – WPITEN Write Protection Interrupt Enable
See Register Write Protection for the list of registers that can be protected.

Value	Description
0	Disables the write protection on interrupt registers if WPKEY corresponds to 0x50494F (“PIO” in ASCII).
1	Enables the write protection on interrupt registers if WPKEY corresponds to 0x50494F (“PIO” in ASCII).

Bit 0 – WPEN Write Protection Enable
See Register Write Protection for the list of registers that can be protected.

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x50494F (“PIO” in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x50494F (“PIO” in ASCII).

© 2026 Microchip Technology Inc. and its subsidiaries

21.7.14. PIO Write Protection Status Register

Name: PIO_WPSR
Offset: 0x5E4
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	WPVSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPVS
Access								R
Reset								0

Bits 23:8 – WPVSRC[15:0] Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protection Violation Status

Value	Description
0	No write protection violation has occurred since the last read of the PIO_WPSR.
1	A write protection violation has occurred since the last read of the PIO_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

© 2026 Microchip Technology Inc. and its subsidiaries

21.7.15. PIO Privilege Mask Register

Name: PIO_P_MSKRx
Offset: 0x1000 + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the PIO Privilege Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	MSK31	MSK30	MSK29	MSK28	MSK27	MSK26	MSK25	MSK24
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	MSK23	MSK22	MSK21	MSK20	MSK19	MSK18	MSK17	MSK16
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	MSK15	MSK14	MSK13	MSK12	MSK11	MSK10	MSK9	MSK8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – MSKy PIO Line y Mask

Define the I/O lines to be configured when writing the PIO Privilege Configuration Register.
0 (DISABLED): Writing the PIO_P_CFGRx, PIO_P_ODSRx or PIO_P_IOFRx does not affect the corresponding I/O line configuration.
1 (ENABLED): Writing the PIO_P_CFGRx, PIO_P_ODSRx or PIO_P_IOFRx updates the corresponding I/O line configuration.

© 2026 Microchip Technology Inc. and its subsidiaries

21.7.16. PIO Privilege Configuration Register

Name: PIO_P_CFGRx
Offset: 0x1004 + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the PIO Privilege Write Protection Mode Register.

Writing this register will only affect I/O lines enabled in the PIO Privilege Mask Register.

Bit	31	30	29	28	27	26	25	24
		ICFS	PCFS			EVTSEL[2:0]		
Access		R	R			R/W	R/W	R/W
Reset		0	0			0	0	0

Bit	23	22	21	20	19	18	17	16
							SLEWRATE[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	15	14	13	12	11	10	9	8
	SCHMITT	OPD	IFSCEN	IFEN		PDEN	PUEN	DIR
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bit	7	6	5	4	3	2	1	0
						FUNC[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 30 – ICFS Interrupt Configuration Freeze Status
Gives information about the freeze state of the following fields of the read I/O line configuration:
• IFEN: Input Filter Enable
• IFSCEN: Input Filter Slow Clock Enable
• EVTSEL: Event Selection
0 (NOT_FROZEN): The fields are not frozen and can be written for this I/O line.
1 (FROZEN): The fields are frozen and cannot be written for this I/O line. Only a hardware reset can release these fields.

Bit 29 – PCFS Physical Configuration Freeze Status
Gives information about the freeze state of the following fields of the read I/O line configuration:
• FUNC: I/O Line Function
• DIR: Direction
• PUEN: Pull-Up Enable
• PDEN: Pull-Down Enable
• OPD: Open-Drain
• SCHMITT: Schmitt Trigger
• SLEWRATE: Slew Rate
0 (NOT_FROZEN): The fields are not frozen and can be written for this I/O line.
1 (FROZEN): The fields are frozen and cannot be written for this I/O line. Only a hardware reset can release these fields.

© 2026 Microchip Technology Inc. and its subsidiaries

Bits 26:24 – EVTSEL[2:0] Event Selection

Defines the type of event to detect on the I/O lines of the I/O group x according to the PIO Privilege Mask Register.

Value	Name	Description
0	FALLING	Event detection on input falling edge
1	RISING	Event detection on input rising edge
2	BOTH	Event detection on input both edge
3	LOW	Event detection on low level input
4	HIGH	Event detection on high level input
5	–	Reserved
6	–	Reserved
7	–	Reserved

Bits 17:16 – SLEWRATE[1:0] Slew Rate

Defines the slew rate of the I/O lines of the I/O group x according to the PIO Privilege Mask Register.

Value	Name	Description
0	FAST	Fast slew rate
1	MEDIUMFAST	Medium-fast slew rate
2	MEDIUM	Medium slew rate
3	SLOW	High slew rate

Bit 15 – SCHMITT Schmitt Trigger

Defines the Schmitt trigger configuration of the I/O lines of the I/O group x according to the PIO Privilege Mask Register.
0 (ENABLED): Schmitt trigger is enabled for the selected I/O lines.
1 (DISABLED): Schmitt trigger is disabled for the selected I/O lines.

Bit 14 – OPD Open Drain

Defines the open drain configuration of the I/O lines of the I/O group x according to the PIO Privilege Mask Register.
0 (DISABLED): The open drain is disabled for the selected I/O lines. I/O lines are driven at high- and low-level.
1 (ENABLED): The open drain is enabled for the selected I/O lines. I/O lines are driven at low-level only.

Bit 13 – IFSCEN Input Filter Slow Clock Enable

Defines the clock source of the glitch filtering for the I/O lines of the I/O group x according to the PIO Privilege Mask Register.

Value	Description
0	The glitch filter is able to filter glitches with a duration less than 1 peripheral clock cycle for the selected I/O lines.
1	The debouncing filter is able to filter pulses with a duration less than 1 divided slow clock cycle for the selected I/O lines.

Bit 12 – IFEN Input Filter Enable

Defines if the glitch filtering is used for the I/O lines of the I/O group x according to the PIO Privilege Mask Register.
0 (DISABLED): The input filter is disabled for the selected I/O lines.
1 (ENABLED): The input filter is enabled for the selected I/O lines.

Bit 10 – PDEN Pull-Down Enable

Defines the pull-down configuration of the I/O lines of the I/O group x according to the PIO Privilege Mask Register.

© 2026 Microchip Technology Inc. and its subsidiaries

PDEN can be written to 1 only if PUEN is written to 0.
0 (DISABLED): Pull-down is disabled for the selected I/O lines.
1 (ENABLED): Pull-down is enabled for the selected I/O lines only if PUEN is 0.

Bit 9 – PUEN Pull-Up Enable
Defines the pull-up configuration of the I/O lines of the I/O group x according to the PIO Privilege Mask Register.
0 (DISABLED): Pull-up is disabled for the selected I/O lines.
1 (ENABLED): Pull-up is enabled for the selected I/O lines.

Bit 8 – DIR Direction
Defines the direction of the I/O lines of the I/O group x according to the PIO Privilege Mask Register.
0 (INPUT): The selected I/O lines are pure inputs.
1 (OUTPUT): The selected I/O lines are enabled in output.

Bits 2:0 – FUNC[2:0] I/O Line Function
Defines the function for I/O lines of the I/O group x according to the PIO Privilege Mask Register.

Value	Name	Description
0	GPIO	Selects the PIO mode for the selected I/O lines.
1	PERIPH_A	Selects peripheral A for the selected I/O lines.
2	PERIPH_B	Selects peripheral B for the selected I/O lines.
3	PERIPH_C	Selects peripheral C for the selected I/O lines.
4	PERIPH_D	Selects peripheral D for the selected I/O lines.

© 2026 Microchip Technology Inc. and its subsidiaries

21.7.17. PIO Privilege Pin Data Status Register

Name: PIO_P_PDSRx
Offset: 0x1008 + x*0x40 [x=0..2]
Reset: –
Property: Read-only

Reset values of PIO_PDSR and PIO_P_PDSR depend on the level of the I/O lines. Reading the I/O line levels requires the clock of the PIO Controller to be enabled, otherwise PIO_PDSR reads the levels present on the I/O line at the time the clock was disabled.

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	R	R	R	R	R	R	R	R
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	R	R	R	R	R	R	R	R
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	R	R	R	R	R	R	R	R
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	R	R	R	R	R	R	R	R
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – Px Input Data Status

Value	Description
0	The I/O line of the I/O group x is at level 0.
1	The I/O line of the I/O group x is at level 1.

© 2026 Microchip Technology Inc. and its subsidiaries

21.7.18. PIO Privilege Lock Status Register

Name: PIO_P_LOCKSRx
Offset: 0x100C + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – Px Lock Status

Value	Description
0	The I/O line of the I/O group x is not locked.
1	The I/O line of the I/O group x is locked.

© 2026 Microchip Technology Inc. and its subsidiaries

21.7.19. PIO Privilege Set Output Data Register

Name: PIO_P_SODRx
Offset: 0x1010 + x*0x40 [x=0..2]
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – Px Set Output Data

Value	Description
0	No effect.
1	Sets the data to be driven on the I/O line of I/O group x.

© 2026 Microchip Technology Inc. and its subsidiaries

21.7.20. PIO Privilege Clear Output Data Register

Name: PIO_P_CODRx
Offset: 0x1014 + x*0x40 [x=0..2]
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – Px Clear Output Data

Value	Description
0	No effect.
1	Clears the data to be driven on the I/O line of the I/O group x.

© 2026 Microchip Technology Inc. and its subsidiaries

21.7.21. PIO Privilege Output Data Status Register

Name: PIO_P_ODSRx
Offset: 0x1018 + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read/Write

Writing this register will only affect I/O lines enabled in the PIO_P_MSKRx.

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – Px Output Data Status

Value	Description
0	The data to be driven on the I/O line of the I/O group x is 0.
1	The data to be driven on the I/O line of the I/O group x is 1.

© 2026 Microchip Technology Inc. and its subsidiaries

21.7.22. PIO Privilege Interrupt Enable Register

Name: PIO_P_IERx
Offset: 0x1020 + x*0x40 [x=0..2]
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the PIO Privilege Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – Px Input Change Interrupt Enable

Value	Description
0	No effect.
1	Enables the Input Change interrupt on the I/O line of the I/O group x.

© 2026 Microchip Technology Inc. and its subsidiaries

21.7.23. PIO Privilege Interrupt Disable Register

Name: PIO_P_IDRx
Offset: 0x1024 + x*0x40 [x=0..2]
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the PIO Privilege Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – Px Input Change Interrupt Disable

Value	Description
0	No effect.
1	Disables the Input Change interrupt on the I/O line of the I/O group x.

© 2026 Microchip Technology Inc. and its subsidiaries

21.7.24. PIO Privilege Interrupt Mask Register

Name: PIO_P_IMRx
Offset: 0x1028 + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – Px Input Change Interrupt Mask

Value	Description
0	Input Change interrupt is disabled on the I/O line of the I/O group x.
1	Input Change interrupt is enabled on the I/O line of the I/O group x.

© 2026 Microchip Technology Inc. and its subsidiaries

21.7.25. PIO Privilege Interrupt Status Register

Name: PIO_P_ISRx
Offset: 0x102C + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read-only

PIO_ISR and PIO_P_ISR are reset at 0x000000000. However, the first read of the register may read a different value as input changes may have occurred.

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – Px Input Change Interrupt Status

Value	Description
0	No Input Change has been detected on the I/O line of the I/O group x since PIO_P_ISRx was last read or since reset.
1	At least one Input Change has been detected on the I/O line of the I/O group since PIO_P_ISRx was last read or since reset.

© 2026 Microchip Technology Inc. and its subsidiaries

21.7.26. PIO Privilege Set I/O User Access Register

Name: PIO_P_SIO_UARx
Offset: 0x1030 + x*0x40 [x=0..2]
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – Px Set I/O in User-Access Mode

Value	Description
0	No effect.
1	Sets the I/O line of the I/O group x in User-Access mode.

© 2026 Microchip Technology Inc. and its subsidiaries

21.7.27. PIO Privilege Set I/O Privilege Access Register

Name: PIO_P_SIO_PARx
Offset: 0x1034 + x*0x40 [x=0..2]
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – Px Set I/O in Privileged-Access Mode

Value	Description
0	No effect.
1	Sets the I/O line of the I/O group x in Privileged-Access mode.

© 2026 Microchip Technology Inc. and its subsidiaries

21.7.28. PIO Privilege I/O Security Status Register

Name: PIO_P_IOSSRx
Offset: 0x1038 + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – Px I/O Security Status
0 (PRIVILEGED_ACCESS): The I/O line of the I/O group x is in Privileged-Access mode.
1 (USER_ACCESS): The I/O line of the I/O group x is in User-Access mode.

© 2026 Microchip Technology Inc. and its subsidiaries

21.7.29. PIO Privilege I/O Freeze Configuration Register

Name: PIO_P_IOFRx
Offset: 0x103C + x*0x40 [x=0..2]
Reset: –
Property: Write-only

Writing this register will only affect I/O lines enabled in the PIO_P_MSKRx.

Bit	31	30	29	28	27	26	25	24
	FRZKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	FRZKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	FRZKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
							FINT	FPHY
Access							W	W
Reset							–	–

Bits 31:8 – FRZKEY[23:0] Freeze Key

Value	Name	Description
0x494F46	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit.

Bit 1 – FINT Freeze Interrupt Configuration
Only a hardware reset can reset the FINT bit.

Value	Description
0	No effect.
1	Freezes the following configuration fields of Privileged-Access I/O lines if FRZKEY corresponds to 0x494F46 (“IOF” in ASCII): • IFEN: Input Filter Enable • IFSCEN: Input Filter Slow Clock Enable • EVTSEL: Event Selection

Bit 0 – FPHY Freeze Physical Configuration
Only a hardware reset can reset the FPHY bit.

Value	Description
0	No effect.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
1	Freezes the following configuration fields of Privileged-Access mode I/O lines if FRZKEY corresponds to 0x494F46 (“IOF” in ASCII): • FUNC: I/O Line Function • DIR: Direction • PUEN: Pull-Up Enable • PDEN: Pull-Down Enable • OPD: Open-Drain • SCHMITT: Schmitt Trigger • SLEWRATE: Slew Rate

© 2026 Microchip Technology Inc. and its subsidiaries

21.7.30. PIO Privilege Slow Clock Divider Debouncing Register

Name: PIO_P_SCDR
Offset: 0x1500
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the PIO Privilege Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			DIV[13:8]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DIV[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 13:0 – DIV[13:0] Slow Clock Divider Selection for Debouncing

$t_{div\_slck} = ((DIV + 1) \times 2) \times t_{slck}$

© 2026 Microchip Technology Inc. and its subsidiaries

21.7.31. PIO Privilege Write Protection Mode Register

Name: PIO_P_WPMR
Offset: 0x15E0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
							WPITEN	WPEN
Access							R/W	R/W
Reset							0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x50494F	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0.

Bit 1 – WPITEN Write Protection Interrupt Enable
See Register Write Protection for the list of registers that can be protected.

Value	Description
0	Disables the write protection on secure interrupt registers if WPKEY corresponds to 0x50494F (“PIO” in ASCII).
1	Enables the write protection on secure interrupt registers if WPKEY corresponds to 0x50494F (“PIO” in ASCII).

Bit 0 – WPEN Write Protection Enable
See Register Write Protection for the list of registers that can be protected.

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x50494F (“PIO” in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x50494F (“PIO” in ASCII).

© 2026 Microchip Technology Inc. and its subsidiaries

21.7.32. PIO Privilege Write Protection Status Register

Name: PIO_P_WPSR
Offset: 0x15E4
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	WPVSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPVS
Access								R
Reset								0

Bits 23:8 – WPVSRC[15:0] Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protection Violation Status

Value	Description
0	No write protection violation has occurred since the last read of the PIO_P_WPSR.
1	A write protection violation has occurred since the last read of the PIO_P_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

© 2026 Microchip Technology Inc. and its subsidiaries

22. Real-Time Clock (RTC)

22.1. Description

The Real-time Clock (RTC) is designed for very low power consumption. The RTC requires an accurate external 32.768 kHz clock, which can be provided by a crystal oscillator.

The RTC combines a complete time-of-day clock with alarm and a Gregorian or Persian calendar or UTC mode, complemented by a programmable periodic interrupt. The alarm and calendar registers are accessed by a 32-bit data bus.

In Gregorian mode, the time and calendar values are coded in binary-coded decimal (BCD) format. The time format can be 24-hour mode or 12-hour mode with an AM/PM indicator.

Updating time and calendar fields and configuring the alarm fields are performed by a parallel capture on the 32-bit data bus. An entry control is performed to avoid loading registers with incompatible BCD format data or with an incompatible date according to the current month/year/century.

A clock divider calibration circuitry can compensate for crystal oscillator frequency variations.

Two RTC outputs can be programmed to generate several waveforms, including a prescaled clock derived from 32.768 kHz.

Timestamping capability reports the first and last occurrences for each tamper event detected on TMPx pins.

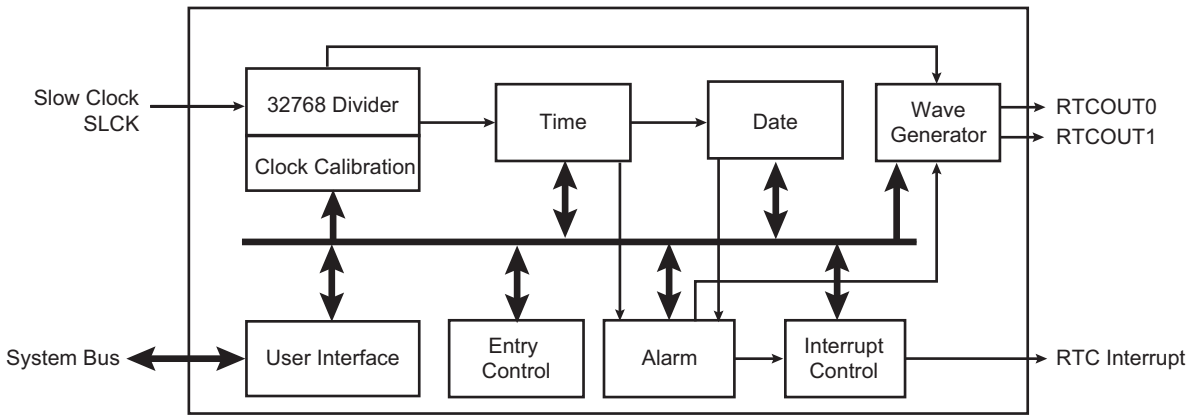
22.2. Embedded Characteristics

- Full Asynchronous Design for Ultra Low-Power Consumption
- Gregorian, Persian and UTC Modes Supported
- Programmable Periodic Interrupt
- Safety/Security Features:
  - Valid Time and Date Programming Check
  - On-The-Fly Time and Date Validity Check
- Counters Calibration Circuitry to Compensate for Crystal Oscillator Variations
- Waveform Generation
- Tamper Timestamping Registers
- Register Write Protection

© 2026 Microchip Technology Inc. and its subsidiaries

22.3. Block Diagram

Figure 22-1. RTC Block Diagram



22.4. Product Dependencies

22.4.1. Power Management

The RTC is continuously clocked at 32.768 kHz. The Power Management Controller has no effect on RTC behavior.

22.4.2. Interrupt

Within the System Controller, the RTC interrupt is OR-wired with all the other module interrupts.

The System Controller interrupt line is connected on one of the internal sources of the interrupt controller.

The RTC interrupt requires the interrupt controller to be programmed first.

When a System Controller interrupt occurs, the service routine must first determine the cause of the interrupt. This is done by reading each status register of the System Controller peripherals successively.

22.5. Functional Description

The RTC provides a full binary-coded decimal (BCD) clock that includes century (20), year (with leap years), month, date, day, hours, minutes and seconds reported in the Time register (RTC_TIMR) and the Calendar register (RTC_CALR).

The RTC can operate in UTC mode, giving the number of seconds elapsed since a reference time defined by the user (the UTC standard—ISO 8601—reference time is the 30th of June 1972). In this mode, the time is reported on 32-bit and coded in hexadecimal format.

The valid year range is up to 2099 in Gregorian mode (or 1300 to 1499 in Persian mode).

The RTC can operate in 24-hour mode or in 12-hour mode with an AM/PM indicator.

Corrections for leap years are included (all years divisible by 4 being leap years). This is correct up to the year 2099.

The RTC can generate configurable waveforms on RTCOUT0/1 outputs.

22.5.1. Reference Clock

The reference clock is the Slow Clock (TD_SLCK). It can be driven internally or by an external 32.768 kHz crystal.

© 2026 Microchip Technology Inc. and its subsidiaries

During Low-Power modes of the processor, the oscillator runs and power consumption is critical. The crystal selection has to take into account the current consumption for power saving and the frequency drift due to temperature effect on the circuit for time accuracy.

22.5.2. Timing

In Gregorian and Persian modes, the RTC is updated in real time at one-second intervals in Normal mode for the counters of seconds, at one-minute intervals for the counter of minutes and so on.

In UTC mode, the RTC is updated in real time at one-second intervals (32-bit UTC counter default configuration).

Due to the asynchronous operation of the RTC with respect to the rest of the chip, to be certain that the value read in the RTC registers (century, year, month, date, day, hours, minutes, seconds) are valid and stable, it is necessary to read these registers twice. If the data is the same both times, then it is valid. Therefore, a minimum of two and a maximum of three accesses are required.

22.5.3. Alarm

In Gregorian and Persian modes, the RTC has five programmable fields for month, date, hours, minutes and seconds.

Each of these fields can be enabled or disabled to match the alarm condition:

- If all the fields are enabled, an alarm flag is generated (the corresponding flag is asserted and an interrupt generated if enabled) at a given month, date, hour/minute/second.
- If only the “seconds” field is enabled, then an alarm is generated every minute.

Depending on the fields that are enabled in Calendar Alarm register (RTC_CALALR) and Time Alarm register (RTC_TIMALR), a large number of possibilities are available to the user ranging from minutes to 365/366 days.

Note: To change one of the RTC_TIMALR.SEC, MIN, HOUR, RTC_CALALR.DATE, MONTH fields, it is recommended to disable the field before changing the value and then re-enable it after the change has been made. This requires up to three accesses to the RTC_TIMALR or RTC_CALALR. The first access clears the enable corresponding to the field to change (SECEN, MINEN, HOUREN, DATEEN, MTHEN). If the field is already cleared, this access is not required. The second access performs the change of the value (SEC, MIN, HOUR, DATE, MONTH). The third access is required to re-enable the field by writing 1 in SECEN, MINEN, HOUREn, DATEEN, MTHEN fields.

In UTC mode, RTC_TIMALR must be configured to set the UTC alarm value and bit 0 in RTC_CALALR must be used to enable or disable the UTC alarm. If the UTC alarm is enabled, the alarm is generated once the UTC time matches the programmed UTC_TIME alarm field.

To change the RTC_TIMALR.UTC_TIME alarm field, proceed as follows:

1. Disable the UTC alarm by clearing the RTC_CALALR.UTCEN bit if it is not already cleared.
2. Change the RTC_TIMALR.UTC_TIME value.
3. Re-enable the UTC alarm by setting the RTC_CALALR.UTCEN bit.

22.5.4. Error Checking when Programming

Verification on user interface data is performed when accessing the century, year, month, date, day, hours, minutes, seconds and alarms. A check is performed on illegal BCD entries such as illegal date of the month with regard to the year and century configured.

If one of the time fields is not correct, the data is not loaded into the register/counter and a flag is set in the Valid Entry register (RTC_VER). The user cannot reset this flag. It is reset as soon as an acceptable value is programmed. This avoids any abnormal behavior in the system (non-BCD format read by software). The same procedure is followed for the alarm.

The following checks are performed:

© 2026 Microchip Technology Inc. and its subsidiaries

1. Century (check if it is 20 in Gregorian mode or 13-14 in Persian mode)
2. Year (BCD entry check)
3. Date (check range 01–31)
4. Month (check if it is in BCD range 01–12, check validity regarding “date”)
5. Day (check range 1–7)
6. Hour (BCD checks: in 24-hour mode, check range 00–23 and check that AM/PM flag is not set if RTC is set in 24-hour mode; in 12-hour mode check range 01–12)
7. Minute (check BCD and range 00–59)
8. Second (check BCD and range 00–59)

Note: If the 12-hour mode is selected by means of the Mode register (RTC_MR), a 12-hour value can be programmed and the returned value on RTC_TIMR will be the corresponding 24-hour value. The entry control checks the value of the AM/PM indicator (bit 22 of RTC_TIMR) to determine the range to be checked.

Note: In UTC mode, no check is performed on the entries. The RTC does not report any failure.

22.5.5. RTC Internal Free Running Counter Error Checking

To improve the reliability and security of the RTC, a permanent check is performed on the internal free running counters to report non-BCD or invalid date/time values.

An error is reported by RTC_SR.TDERR bit in the status register if an incorrect value has been detected. The flag can be cleared by setting RTC_SCCR.TDERRCLR.

In all cases, the RTC_SR.TDERR error flag will be set again if the source of the error has not been cleared before clearing the RTC_SR.TDERR flag. The clearing of the source of such an error can be done by reprogramming a correct value in RTC_CALR and/or RTC_TIMR.

The RTC internal free-running counters may automatically clear the source of RTC_SR.TDERR due to their roll-over (i.e., every 10 seconds for RTC_TIMR.SECONDS[3:0] field). In this case the RTC_SR.TDERR is held high until a clear command is asserted by RTC_SCCR.TDERRCLR.

22.5.6. Updating Time/Calendar

22.5.6.1. Gregorian and Persian Modes

To update time and date, the RTC must be stopped by setting the corresponding field in the Control register (RTC_CR). The RTC_CR.UPDTIM bit must be set to update time fields (hour, minute, second) and RTC_CR.UPDCAL must be set to update calendar fields (century, year, month, date, day). When both time and calendar fields require an update, RTC_CR.UPDTIM and RTC_CR.UPDCAL must be written to 1 in the same access.

RTC_SR.ACKUPD must then be read to 1 by either polling the RTC_SR or by enabling the acknowledge update interrupt by writing a 1 in RTC_IER.ACKEN. Once RTC_SR.ACKUPD is read to 1, it is mandatory to clear this flag by writing a 1 in RTC_SCCR.ACKCLR, after which the user can write to RTC_TIMR, RTC_CALR, or both.

Once the update is finished, the user must write a 0 in the bits RTC_CR.UPDTIM and RTC_CR.UPDCAL.

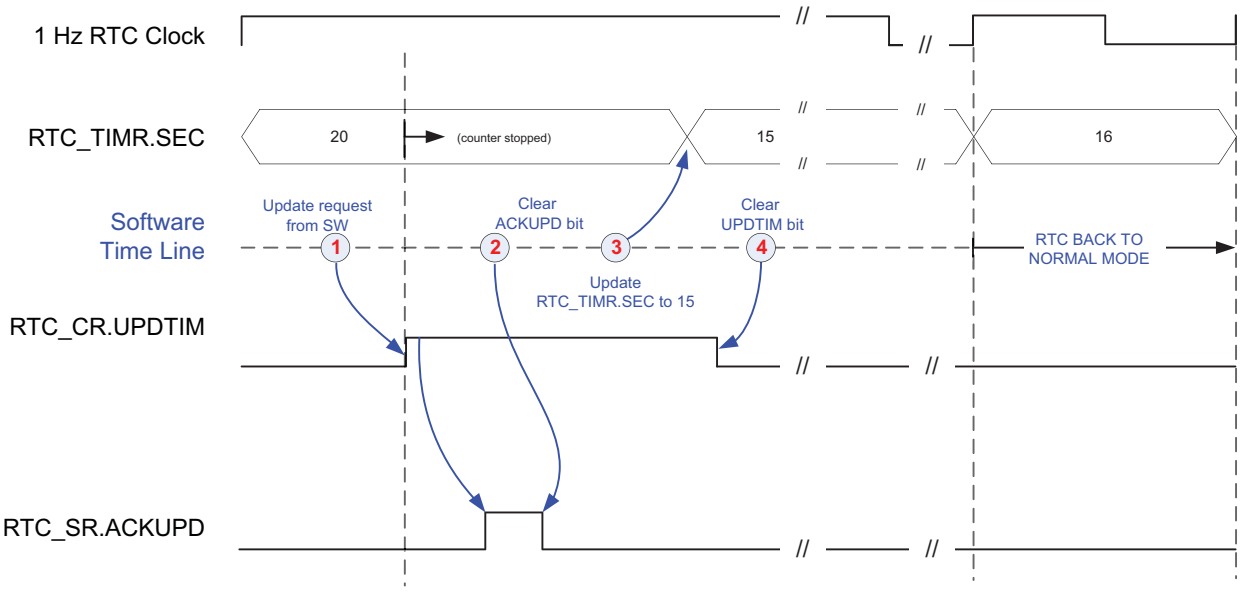
The timing sequence of the time/calendar update is described in Figure 22-2.

When entering the Programming mode of the calendar fields, the time fields remain enabled. When entering the Programming mode of the time fields, both the time and the calendar fields are stopped. This is due to the location of the calendar logical circuity (downstream for low-power considerations).

In successive update operations, the user must first check that RTC_CR.UPDTIM and RTC_CR.UPDCAL read 0 before writing a 1 in these bits.

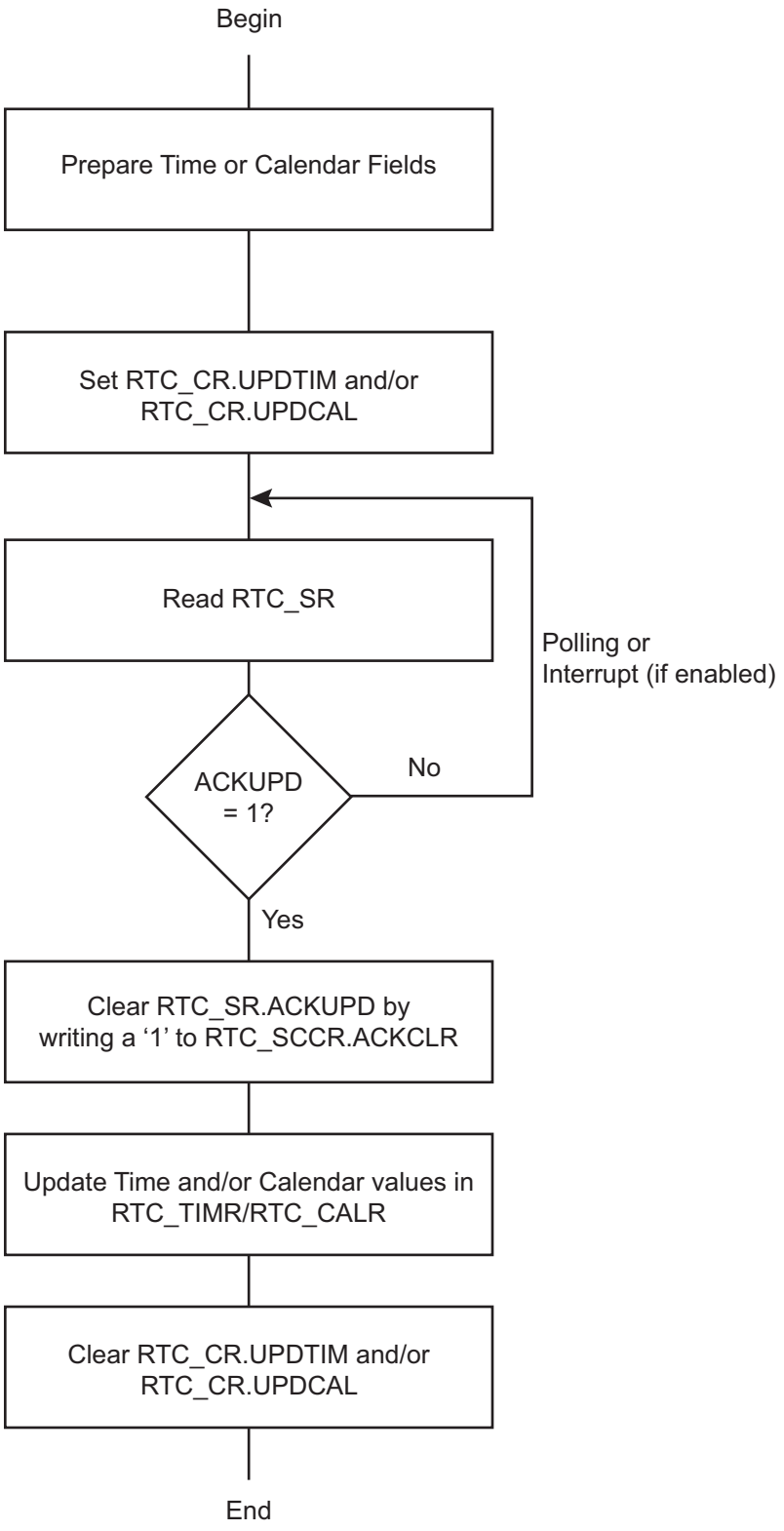
© 2026 Microchip Technology Inc. and its subsidiaries

Figure 22-2. Time/Calendar Update Timing Diagram



© 2026 Microchip Technology Inc. and its subsidiaries

Figure 22-3. Gregorian and Persian Modes Update Sequence



22.5.6.2. UTC Mode

To update the UTC time, the RTC must be stopped by writing a 1 in RTC_CR.UPDTIM and RTC_CR.UPDCAL.

© 2026 Microchip Technology Inc. and its subsidiaries

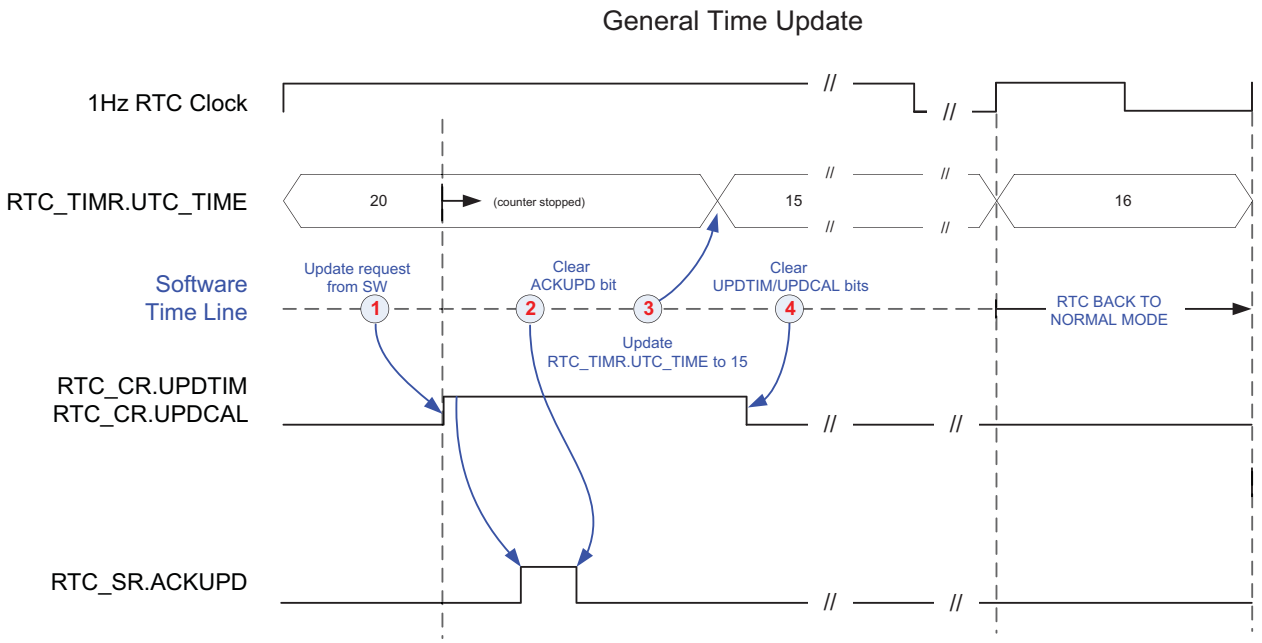
RTC_SR.ACKUPD must then be read to 1 by either polling the RTC_SR or by enabling the acknowledge update interrupt by writing a 1 in RTC_IER.ACKEN. Once RTC_SR.ACKUPD is read to 1, it is mandatory to clear this flag by writing a 1 in RTC_SCCR.ACKCLR, after which the user can write to RTC_TIMR.

Once the update is finished, the user must write a 0 in RTC_CR.UPDTIM and RTC_CR.UPDCAL.

In successive update operations, the user must first check that RTC_CR.UPDTIM and RTC_CR.UPDCAL read 0 before writing a 1 in these bits.

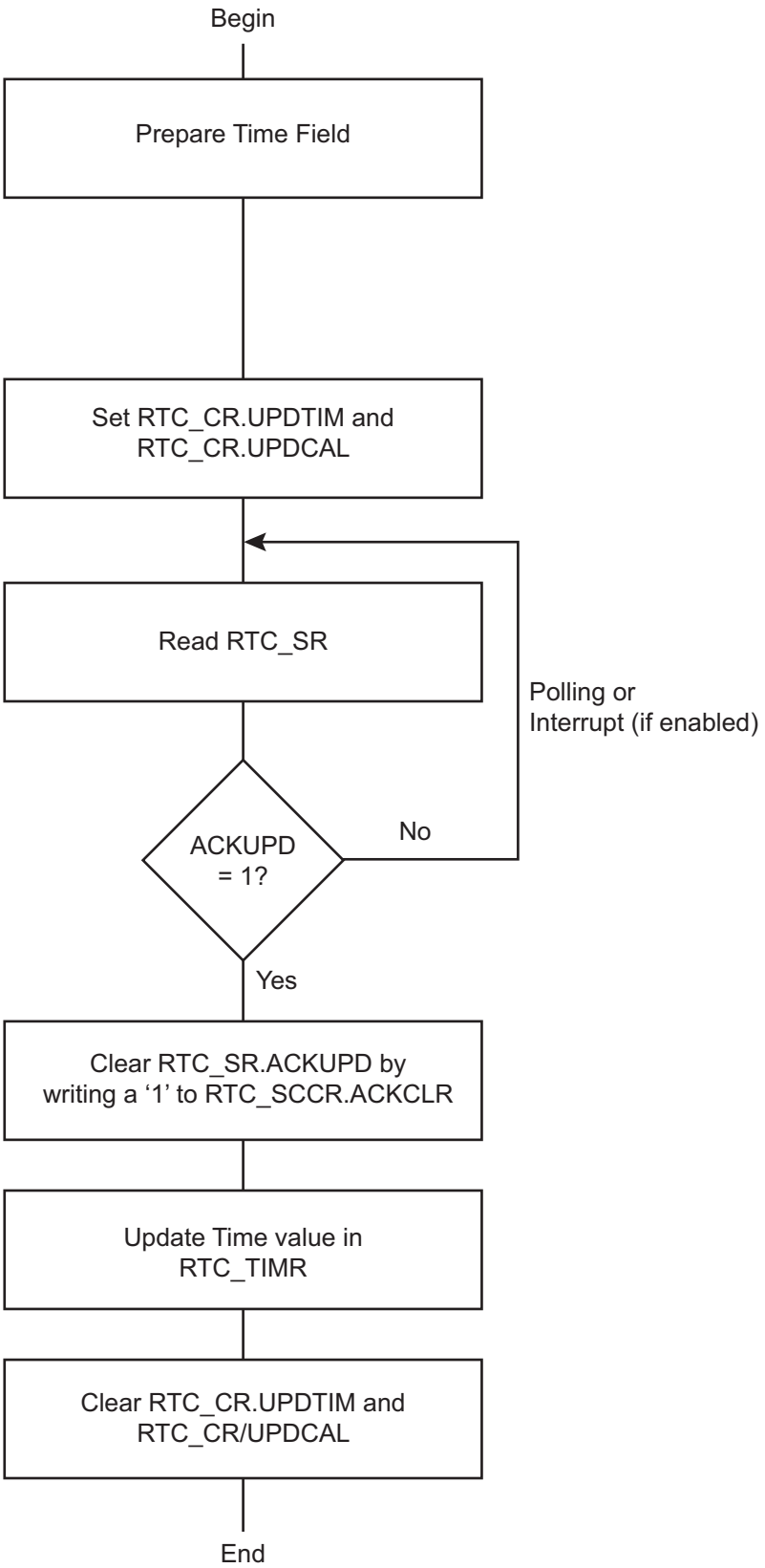
The timing sequence of the UTC time update is described in Figure 22-4.

Figure 22-4. UTC Time Update Timing Diagram



© 2026 Microchip Technology Inc. and its subsidiaries

Figure 22-5. UTC Mode Update Sequence



© 2026 Microchip Technology Inc. and its subsidiaries

22.5.7. RTC Accurate Clock Calibration

The crystal oscillator that drives the RTC may not be as accurate as expected mainly due to temperature variation. The RTC is equipped with circuitry able to compensate the slow clock crystal drift.

To compensate for possible temperature variations over time, this accurate clock calibration circuitry can be programmed on-the-fly and also programmed during application manufacturing, in order to correct the crystal frequency accuracy at room temperature (20–25°C). The typical clock drift range at room temperature is ±20 ppm.

In the device operating temperature range, the 32.768 kHz crystal oscillator clock inaccuracy can be up to -200 ppm.

The RTC clock calibration circuitry allows positive or negative correction in a range of 1.5 ppm to 1950 ppm.

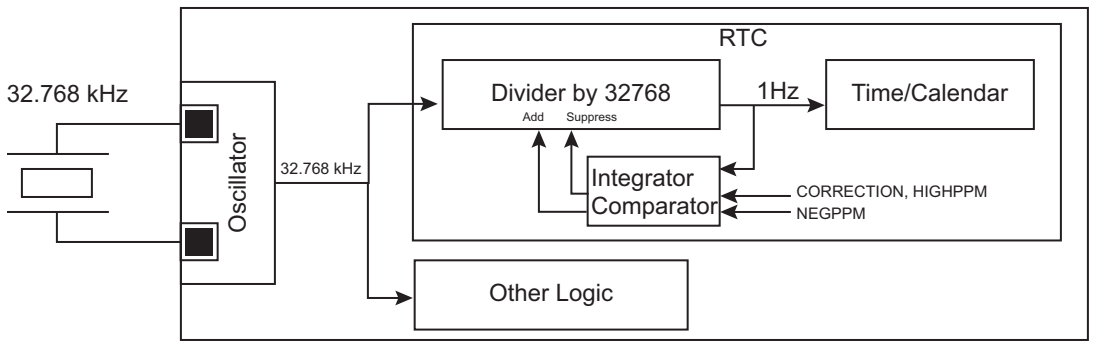
The calibration circuitry is fully digital. Thus, the configured correction is independent of temperature, voltage, process, etc., and no additional measurement is required to check that the correction is effective.

If the correction value configured in the calibration circuitry results from an accurate crystal frequency measure, the remaining accuracy is bounded by the values listed below:

- Below 1 ppm, for an initial crystal drift between 1.5 ppm up to 20 ppm, and from 30 ppm to 90 ppm
- Below 2 ppm, for an initial crystal drift between 20 ppm up to 30 ppm, and from 90 ppm to 130 ppm
- Below 5 ppm, for an initial crystal drift between 130 ppm up to 200 ppm

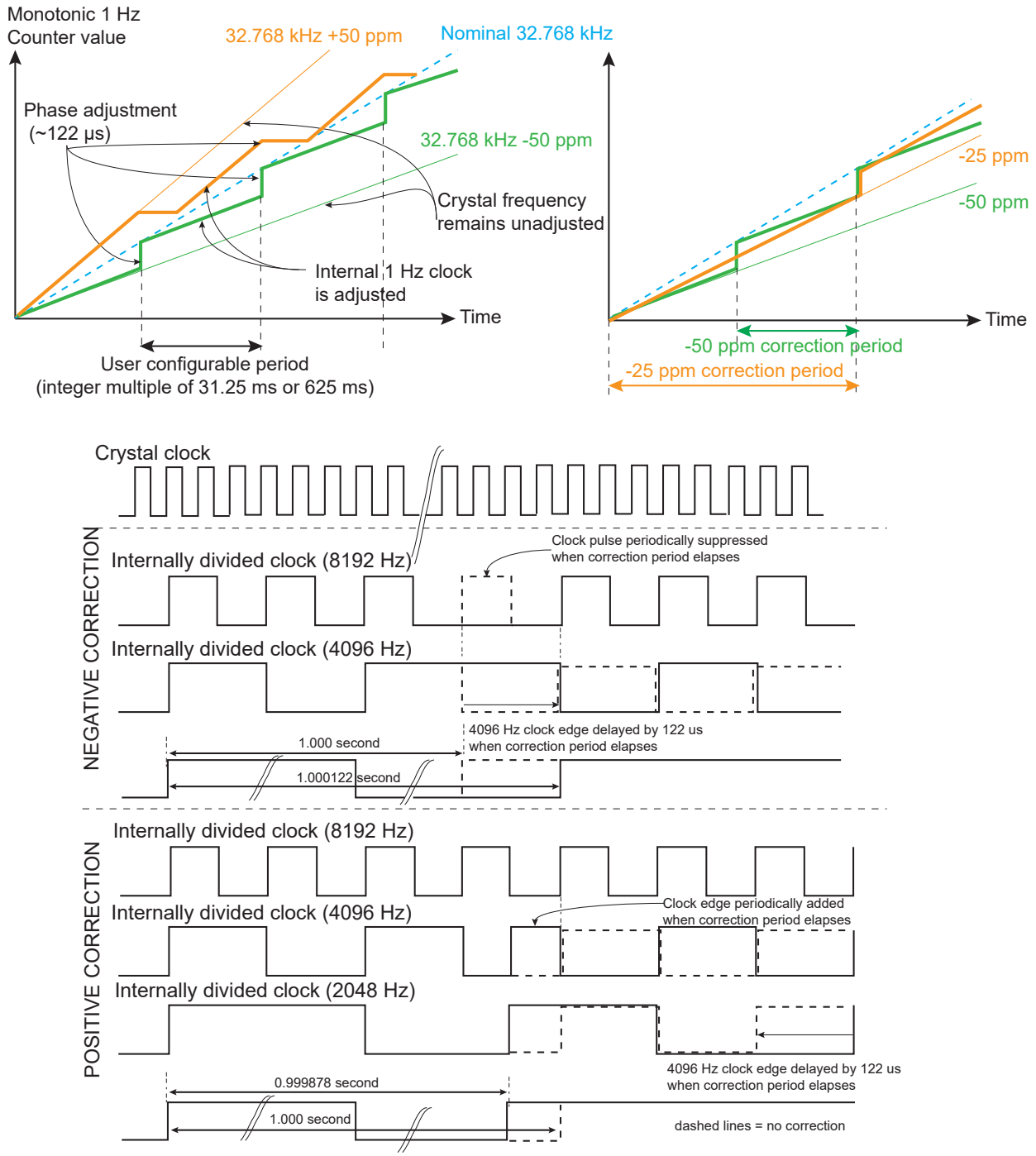
The calibration circuitry does not modify the 32.768 kHz crystal oscillator clock frequency but it acts by slightly modifying the 1 Hz clock period and RTCOUTx outputs from time to time. The correction event occurs every 1 + [(20 - (19 x HIGHPPM)) x CORRECTION]/32 seconds. When the period is modified, depending on the sign of the correction, the 1 Hz clock period increases or reduces by around 122 µs. Depending on the values of RTC_MR.CORRECTION, RTC_MR.NEGPPM and RTC_MR.HIGHPPM, the period interval between two correction events differs.

Figure 22-6. Calibration Circuitry



© 2026 Microchip Technology Inc. and its subsidiaries

Figure 22-7. Calibration Circuitry Waveforms



The inaccuracy of a crystal oscillator at typical room temperature (±20 ppm at 20–25 °C) can be compensated if a reference clock/signal is used to measure such inaccuracy. This kind of calibration operation can be set up during the final product manufacturing by means of measurement equipment embedding such a reference clock. The correction of value must be programmed into

© 2026 Microchip Technology Inc. and its subsidiaries

the RTC_MR, and this value is kept as long as the circuitry is powered (backup area). Removing the backup power supply cancels this calibration.

To ease the comparison of the inherent crystal accuracy with the reference clock/signal during manufacturing, an internal prescaled 32.768 kHz clock derivative signal can be assigned to drive an RTC output (RTCOUT0 or RTCOUT1). To accommodate the measure, several clock frequencies can be selected among 1 Hz, 32 Hz, 64 Hz, 512 Hz.

The clock calibration correction drives the internal RTC counters but can also be observed in the RTC outputs when one of the following three frequencies 1 Hz, 32 Hz or 64 Hz is configured. The correction is not visible in the RTC output if 512 Hz frequency is configured.

The frequency drift (up to -200 ppm) due to temperature variation can be compensated using a reference time if the application can access such a reference. If a reference time cannot be used, a temperature sensor can be placed close to the crystal oscillator in order to get the operating temperature of the crystal oscillator. Once obtained, the temperature may be converted using a lookup table (describing the accuracy/temperature curve of the crystal oscillator used) and RTC_MR configured accordingly. The calibration can be performed on-the-fly. This adjustment method is not based on a measurement of the crystal frequency/drift and therefore can be improved by means of the networking capability of the target application.

If no crystal frequency adjustment has been done during manufacturing, it is still possible to do it. In the case where a reference time of the day can be obtained through LAN/WAN network, it is possible to calculate the drift of the application crystal oscillator by comparing the values read on RTC_TIMR and programming RTC_MR.HIGHPPM and RTC_MR.CORRECTION according to the difference measured between the reference time and the time read on RTC_TIMR and RTC_CALR.

22.5.8. Waveform Generation

Waveforms can be generated by the RTC in order to take advantage of the RTC inherent prescalers while the RTC is the only powered circuitry (Low-power mode of operation, Backup mode) or in any active mode. Going into Backup or Low-power operating modes does not affect the waveform generation outputs.

The RTC outputs (RTCOUT0 and RTCOUT1) have a source driver selected among seven possibilities.

The first selection choice sticks the associated output at 0. This is the reset value and it can be used at any time to disable the waveform generation.

Selection choices 1 to 4 select 1 Hz, 32 Hz, 64 Hz and 512 Hz, respectively.

64 Hz can drive, for example, a twisted nematic (TN) LCD backplane signal while 1 Hz can be used to drive a blinking character like ":" for basic time display (hour, minute) on TN LCDs.

Selection choice 5 provides a toggling signal when the RTC alarm is reached.

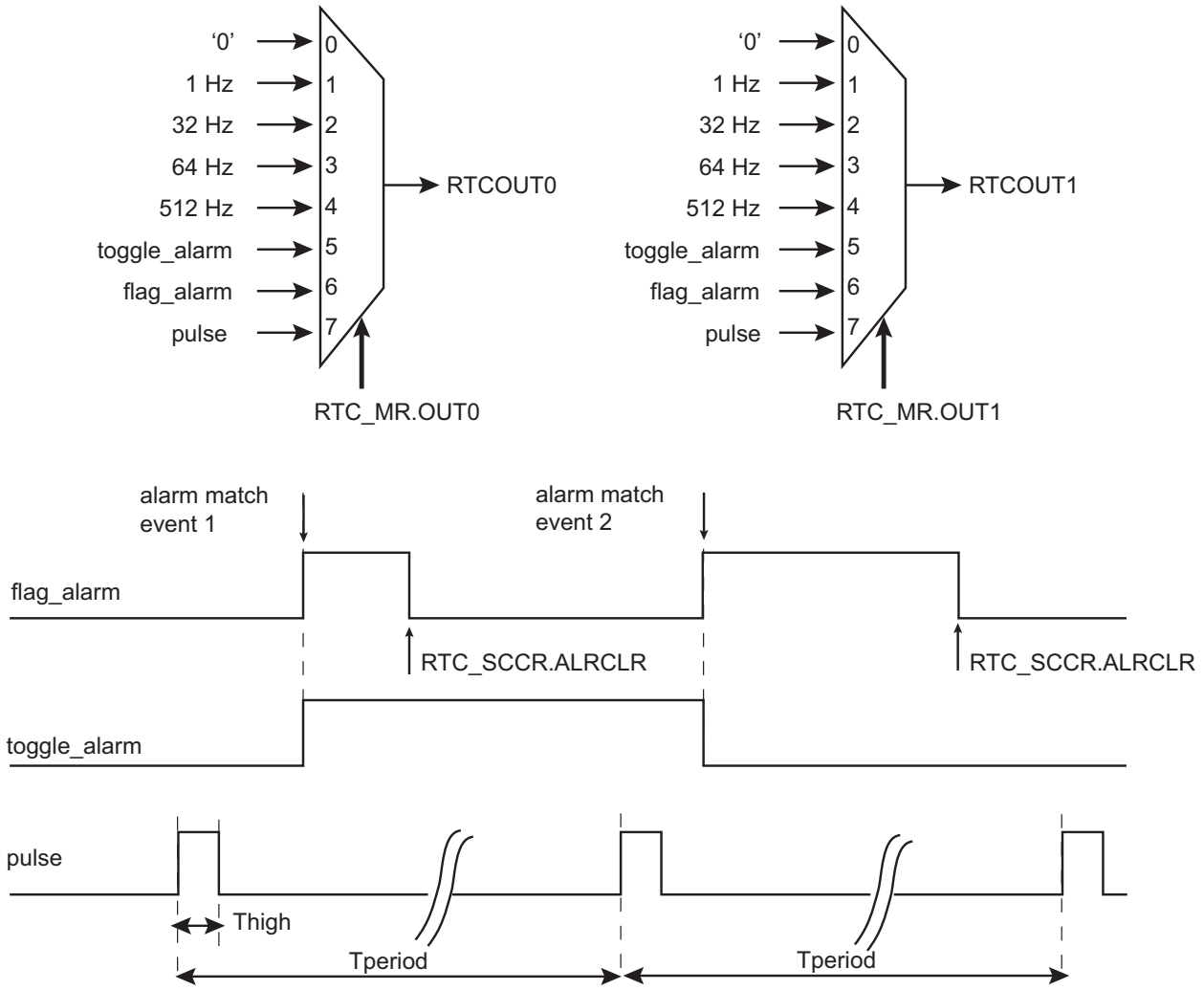
Selection choice 6 provides a copy of the alarm flag, so the associated output is set high (logical 1) when an alarm occurs and immediately cleared when software clears the alarm interrupt source.

Selection choice 7 provides a duty cycle programmable pulse that can be used to drive external devices for power consumption reduction or any other purpose.

PIO lines associated to RTC outputs automatically select these waveforms as soon as RTC_MR corresponding fields RTCOUT0 and RTCOUT1 differ from 0 .

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 22-8. Waveform Generation



22.5.9. Tamper Timestamping

The RTC can be used to stamp up to 5 tamper events. Tamper inputs are located on pins TMP0 to TMP4

As soon as a tamper is detected, the tamper event counter of the corresponding tamper input is incremented and the RTC stores the time of the day and the date of the tamper event in registers located in the backup area. The RTC stores the time, the date and the number of event of the first and the last tamper event on each TMP0 to TMP4 inputs.

In UTC mode, only the UTC time is stored. The date information is not relevant

The tamper counter saturates at 15. Once this limit is reached, the exact number of tamper occurrences since the last read of stamping registers cannot be known.

The time stamping on the TMPx input is enabled by setting the corresponding RTC_TAMPER.TAMPENx bit. The associated Low-power debouncer must be configured in the SUPC by writing a '1' to the corresponding SUPC_WUMR.LPDBCENx and configuring the debouncing value in the corresponding SUPC_WUMR.LPDBCx field.

The first set of timestamping registers (RTC_FSTRx, RTC_FSDRx) cannot be overwritten, so once they have been written all data are stored until the registers are reset. Therefore these registers are storing the first tamper occurrence after a clear.

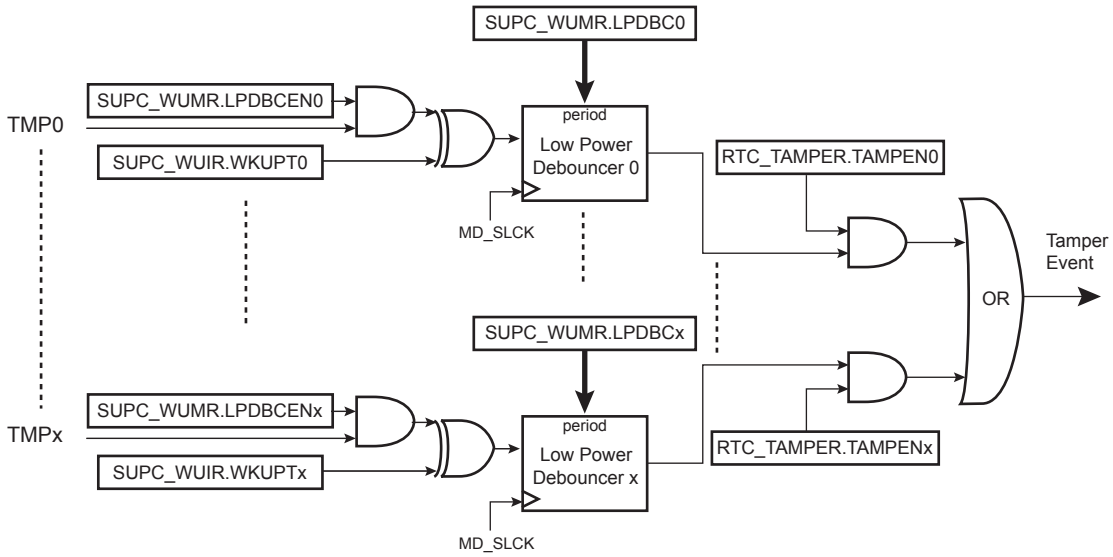
© 2026 Microchip Technology Inc. and its subsidiaries

The second set of timestamping registers (RTC_LSTRx, RTC_LSDRx) is overwritten each time a tamper event is detected. Thus the date and the time data of the first and the second stamping registers may be equal. This occurs when the tamper counter value carried on field TEVCNT in RTC_LSTRx equals 1. Thus this second set of registers stores the last occurrence of tamper after a clear.

Reading a set of timestamping registers associated to a tamper input TMPx requires four accesses, one for the time of the day, one for the date and one for the tamper source.

Reading the RTC_LSDRx of a tamper input clears all content of the registers associated to this tamper input (RTC_FSTRx, RTC_FSDRX, RTC_LSTRx, RTC_LSDRx) and makes the timestamping registers available to store a new event.

Figure 22-9. Tamper Management Block Diagram



22.5.10. GPBR Clear

On a tamper event, the GPBR registers content can be fully or partially erased.

By setting RTC_TAMPER.TAMPCLR to 1, the GPBR registers content are fully or partially cleared depending on the value of RTC_TCR.FGPBRCLR:

- if FGPBRCLR is set to ‘0’, a tamper event clears GPBR0 to GPBR11
- if FGPBRCLR is set to ‘1’, a tamper event clears all GPBRs

© 2026 Microchip Technology Inc. and its subsidiaries

22.6. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	RTC_CR	31:24								
		23:16							CALEVSEL[1:0]	
		15:8							TIMEVSEL[1:0]	
		7:0							UPDCAL	UPDTIM
0x04	RTC_MR	31:24			TPERIOD[1:0]			THIGH[2:0]		
		23:16		OUT1[2:0]				OUT0[2:0]		
		15:8	HIGHPPM	CORRECTION[6:0]						
		7:0				NEGPPM		UTC	PERSIAN	HRMOD
0x08	RTC_TIMR	31:24								
		23:16		AMPM	HOUR[5:0]					
		15:8		MIN[6:0]						
		7:0		SEC[6:0]						
0x08	RTC_TIMR (UTC_MODE)	31:24	UTC_TIME[31:24]							
		23:16	UTC_TIME[23:16]							
		15:8	UTC_TIME[15:8]							
		7:0	UTC_TIME[7:0]							
0x0C	RTC_CALR	31:24			DATE[5:0]					
		23:16	DAY[2:0]			MONTH[4:0]				
		15:8	YEAR[7:0]							
		7:0		CENT[6:0]						
0x10	RTC_TIMALR	31:24								
		23:16	HOUREN	AMPM	HOUR[5:0]					
		15:8	MINEN	MIN[6:0]						
		7:0	SECEN	SEC[6:0]						
0x10	RTC_TIMALR (UTC_MODE)	31:24	UTC_TIME[31:24]							
		23:16	UTC_TIME[23:16]							
		15:8	UTC_TIME[15:8]							
		7:0	UTC_TIME[7:0]							
0x14	RTC_CALALR	31:24	DATEEN		DATE[5:0]					
		23:16	MTHEN			MONTH[4:0]				
		15:8								
		7:0								
0x14	RTC_CALALR (UTC_MODE)	31:24								
		23:16								
		15:8								
		7:0								UTCEN
0x18	RTC_SR	31:24								
		23:16								
		15:8								
		7:0			TDERR	CALEV	TIMEV	SEC	ALARM	ACKUPD
0x1C	RTC_SCCR	31:24								
		23:16								
		15:8								
		7:0			TDERRCLR	CALCLR	TIMCLR	SECCLR	ALRCLR	ACKCLR
0x20	RTC_IER	31:24								
		23:16								
		15:8								
		7:0			TDERREN	CALEN	TIMEN	SECEN	ALREN	ACKEN
0x24	RTC_IDR	31:24								
		23:16								
		15:8								
		7:0			TDERRDIS	CALDIS	TIMDIS	SECDIS	ALRDIS	ACKDIS
0x28	RTC_IMR	31:24								
		23:16								
		15:8								
		7:0			TDERR	CAL	TIM	SEC	ALR	ACK

© 2026 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x2C	RTC_VER	31:24								
		23:16								
		15:8								
		7:0					NVCALALR	NVTIMALR	NVCAL	NVTIM
0x30 ... 0x33	Reserved									
0x34	RTC_TCR	31:24								
		23:16							FGPBRCLR	TAMPCLR
		15:8								
		7:0				TAMPEN4	TAMPEN3	TAMPEN2	TAMPEN1	TAMPEN0
0x38	RTC_TISR	31:24								
		23:16								
		15:8								
		7:0				TISR4	TISR3	TISR2	TISR1	TISR0
0x3C ... 0x3F	Reserved									
0x40	RTC_FSTR0	31:24	BACKUP				TEVCNT[3:0]			
		23:16		AMPM	HOUR[5:0]					
		15:8		MIN[6:0]						
		7:0		SEC[6:0]						
0x40	RTC_FSTR0 (UTC_MODE)	31:24	BACKUP				TEVCNT[3:0]			
		23:16								
		15:8								
		7:0								
0x44	RTC_FSDR0	31:24			DATE[5:0]					
		23:16		DAY[2:0]		MONTH[4:0]				
		15:8		YEAR[6:0]						
		7:0		CENT[6:0]						
0x44	RTC_FSDR0 (UTC_MODE)	31:24	UTC_TIME[31:24]							
		23:16	UTC_TIME[23:16]							
		15:8	UTC_TIME[15:8]							
		7:0	UTC_TIME[7:0]							
0x48	RTC_LSTR0	31:24	BACKUP							
		23:16		AMPM	HOUR[5:0]					
		15:8		MIN[6:0]						
		7:0		SEC[6:0]						
0x48	RTC_LSTR0 (UTC_MODE)	31:24	BACKUP							
		23:16								
		15:8								
		7:0								
0x4C	RTC_LSDR0	31:24			DATE[5:0]					
		23:16		DAY[2:0]		MONTH[4:0]				
		15:8		YEAR[6:0]						
		7:0		CENT[6:0]						
0x4C	RTC_LSDR0 (UTC_MODE)	31:24	UTC_TIME[31:24]							
		23:16	UTC_TIME[23:16]							
		15:8	UTC_TIME[15:8]							
		7:0	UTC_TIME[7:0]							
0x50	RTC_FSTR1	31:24	BACKUP				TEVCNT[3:0]			
		23:16		AMPM	HOUR[5:0]					
		15:8		MIN[6:0]						
		7:0		SEC[6:0]						
0x50	RTC_FSTR1 (UTC_MODE)	31:24	BACKUP				TEVCNT[3:0]			
		23:16								
		15:8								
		7:0								

© 2026 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x54	RTC_FSDR1	31:24			DATE[5:0]					
		23:16	DAY[2:0]			MONTH[4:0]				
		15:8		YEAR[6:0]						
		7:0		CENT[6:0]						
0x54	RTC_FSDR1 (UTC_MODE)	31:24	UTC_TIME[31:24]							
		23:16	UTC_TIME[23:16]							
		15:8	UTC_TIME[15:8]							
		7:0	UTC_TIME[7:0]							
0x58	RTC_LSTR1	31:24	BACKUP							
		23:16		AMPM	HOUR[5:0]					
		15:8		MIN[6:0]						
		7:0		SEC[6:0]						
0x58	RTC_LSTR1 (UTC_MODE)	31:24	BACKUP							
		23:16								
		15:8								
		7:0								
0x5C	RTC_LSDR1	31:24			DATE[5:0]					
		23:16	DAY[2:0]			MONTH[4:0]				
		15:8		YEAR[6:0]						
		7:0		CENT[6:0]						
0x5C	RTC_LSDR1 (UTC_MODE)	31:24	UTC_TIME[31:24]							
		23:16	UTC_TIME[23:16]							
		15:8	UTC_TIME[15:8]							
		7:0	UTC_TIME[7:0]							
0x60	RTC_FSTR2	31:24	BACKUP				TEVCNT[3:0]			
		23:16		AMPM	HOUR[5:0]					
		15:8		MIN[6:0]						
		7:0		SEC[6:0]						
0x60	RTC_FSTR2 (UTC_MODE)	31:24	BACKUP				TEVCNT[3:0]			
		23:16								
		15:8								
		7:0								
0x64	RTC_FSDR2	31:24			DATE[5:0]					
		23:16	DAY[2:0]			MONTH[4:0]				
		15:8		YEAR[6:0]						
		7:0		CENT[6:0]						
0x64	RTC_FSDR2 (UTC_MODE)	31:24	UTC_TIME[31:24]							
		23:16	UTC_TIME[23:16]							
		15:8	UTC_TIME[15:8]							
		7:0	UTC_TIME[7:0]							
0x68	RTC_LSTR2	31:24	BACKUP							
		23:16		AMPM	HOUR[5:0]					
		15:8		MIN[6:0]						
		7:0		SEC[6:0]						
0x68	RTC_LSTR2 (UTC_MODE)	31:24	BACKUP							
		23:16								
		15:8								
		7:0								
0x6C	RTC_LSDR2	31:24			DATE[5:0]					
		23:16	DAY[2:0]			MONTH[4:0]				
		15:8		YEAR[6:0]						
		7:0		CENT[6:0]						
0x6C	RTC_LSDR2 (UTC_MODE)	31:24	UTC_TIME[31:24]							
		23:16	UTC_TIME[23:16]							
		15:8	UTC_TIME[15:8]							
		7:0	UTC_TIME[7:0]							
0x70	RTC_FSTR3	31:24	BACKUP				TEVCNT[3:0]			
		23:16		AMPM	HOUR[5:0]					
		15:8		MIN[6:0]						
		7:0		SEC[6:0]						

© 2026 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x70	RTC_FSTR3 (UTC_MODE)	31:24	BACKUP				TEVCNT[3:0]			
		23:16								
		15:8								
		7:0								
0x74	RTC_FSDR3	31:24			DATE[5:0]					
		23:16		DAY[2:0]		MONTH[4:0]				
		15:8		YEAR[6:0]						
		7:0		CENT[6:0]						
0x74	RTC_FSDR3 (UTC_MODE)	31:24	UTC_TIME[31:24]							
		23:16	UTC_TIME[23:16]							
		15:8	UTC_TIME[15:8]							
		7:0	UTC_TIME[7:0]							
0x78	RTC_LSTR3	31:24	BACKUP							
		23:16		AMPM	HOUR[5:0]					
		15:8		MIN[6:0]						
		7:0		SEC[6:0]						
0x78	RTC_LSTR3 (UTC_MODE)	31:24	BACKUP							
		23:16								
		15:8								
		7:0								
0x7C	RTC_LSDR3	31:24			DATE[5:0]					
		23:16		DAY[2:0]		MONTH[4:0]				
		15:8		YEAR[6:0]						
		7:0		CENT[6:0]						
0x7C	RTC_LSDR3 (UTC_MODE)	31:24	UTC_TIME[31:24]							
		23:16	UTC_TIME[23:16]							
		15:8	UTC_TIME[15:8]							
		7:0	UTC_TIME[7:0]							
0x80	RTC_FSTR4	31:24	BACKUP				TEVCNT[3:0]			
		23:16		AMPM	HOUR[5:0]					
		15:8		MIN[6:0]						
		7:0		SEC[6:0]						
0x80	RTC_FSTR4 (UTC_MODE)	31:24	BACKUP				TEVCNT[3:0]			
		23:16								
		15:8								
		7:0								
0x84	RTC_FSDR4	31:24			DATE[5:0]					
		23:16		DAY[2:0]		MONTH[4:0]				
		15:8		YEAR[6:0]						
		7:0		CENT[6:0]						
0x84	RTC_FSDR4 (UTC_MODE)	31:24	UTC_TIME[31:24]							
		23:16	UTC_TIME[23:16]							
		15:8	UTC_TIME[15:8]							
		7:0	UTC_TIME[7:0]							
0x88	RTC_LSTR4	31:24	BACKUP							
		23:16		AMPM	HOUR[5:0]					
		15:8		MIN[6:0]						
		7:0		SEC[6:0]						
0x88	RTC_LSTR4 (UTC_MODE)	31:24	BACKUP							
		23:16								
		15:8								
		7:0								
0x8C	RTC_LSDR4	31:24			DATE[5:0]					
		23:16		DAY[2:0]		MONTH[4:0]				
		15:8		YEAR[6:0]						
		7:0		CENT[6:0]						
0x8C	RTC_LSDR4 (UTC_MODE)	31:24	UTC_TIME[31:24]							
		23:16	UTC_TIME[23:16]							
		15:8	UTC_TIME[15:8]							
		7:0	UTC_TIME[7:0]							

© 2026 Microchip Technology Inc. and its subsidiaries

22.6.1. RTC Control Register

Name: RTC_CR
Offset: 0x00
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode Register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							CALEVSEL[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	15	14	13	12	11	10	9	8
							TIMEVSEL[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
							UPDCAL	UPDTIM
Access							R/W	R/W
Reset							0	0

Bits 17:16 – CALEVSEL[1:0] Calendar Event Selection
The event that generates the flag CALEV in RTC_SR depends on the value of CALEVSEL
In UTC mode, this field has no effect on the RTC_SR.

Value	Name	Description
0	WEEK	Week change (every Monday at time 00:00:00)
1	MONTH	Month change (every 01 of each month at time 00:00:00)
2	YEAR	Year change (every January 1 at time 00:00:00)
3	–	Reserved

Bits 9:8 – TIMEVSEL[1:0] Time Event Selection
The event that generates the flag TIMEV in RTC_SR depends on the value of TIMEVSEL.
In UTC mode, this field has no effect on the RTC_SR.

Value	Name	Description
0	MINUTE	Minute change
1	HOUR	Hour change
2	MIDNIGHT	Every day at midnight
3	NOON	Every day at noon

Bit 1 – UPDCAL Update Request Calendar Register
Calendar counting consists of day, date, month, year and century counters. Calendar counters can be programmed once this bit is set and acknowledged by RTC_SR.ACKUPD.
In UTC mode, this field has no effect on the RTC_SR.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Name	Description
0	STOP_UPDATE	No effect or, if UPDCAL has been previously written to 1, stops the update procedure.
1	START_UPDATE	Stops the RTC calendar counting.

Bit 0 – UPDTIM Update Request Time Register

Value	Name	Description
0	STOP_UPDATE	No effect or, if UPDTIM has been previously written to 1, stops the update procedure.
1	START_UPDATE	Stops the RTC time counting. Second, minute and hour counters can be programmed once this bit is set and acknowledged by RTC_SR.ACKUPD.

© 2026 Microchip Technology Inc. and its subsidiaries

22.6.2. RTC Mode Register

Name: RTC_MR
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode Register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
			TPERIOD[1:0]			THIGH[2:0]		
Access			R/W	R/W		R/W	R/W	R/W
Reset			0	0		0	0	0

Bit	23	22	21	20	19	18	17	16
		OUT1[2:0]				OUT0[2:0]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bit	15	14	13	12	11	10	9	8
	HIGHPPM	CORRECTION[6:0]						
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
				NEGPPM		UTC	PERSIAN	HRMOD
Access				R/W		R/W	R/W	R/W
Reset				0		0	0	0

Bits 29:28 – TPERIOD[1:0] Period of the Output Pulse

Value	Name	Description
0	P_1S	1 second
1	P_500MS	500 ms
2	P_250MS	250 ms
3	P_125MS	125 ms

Bits 26:24 – THIGH[2:0] High Duration of the Output Pulse

Value	Name	Description
0	H_31MS	31.2 ms
1	H_16MS	15.6 ms
2	H_4MS	3.91 ms
3	H_976US	976 µs
4	H_488US	488 µs
5	H_122US	122 µs
6	H_30US	30.5 µs
7	H_15US	15.2 µs

Bits 22:20 – OUT1[2:0] RTCOUT1 Output Source Selection

Value	Name	Description
0	NO_WAVE	No waveform, stuck at ‘0’
1	FREQ1HZ	1 Hz square wave
2	FREQ32HZ	32 Hz square wave

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Name	Description
3	FREQ64HZ	64 Hz square wave
4	FREQ512HZ	512 Hz square wave
5	ALARM_TOGGLE	Output toggles when alarm flag rises
6	ALARM_FLAG	Output is a copy of the alarm flag
7	PROG_PULSE	Duty cycle programmable pulse

Bits 18:16 – OUT0[2:0] RTCOUT0 OutputSource Selection

Value	Name	Description
0	NO_WAVE	No waveform, stuck at '0'
1	FREQ1HZ	1 Hz square wave
2	FREQ32HZ	32 Hz square wave
3	FREQ64HZ	64 Hz square wave
4	FREQ512HZ	512 Hz square wave
5	ALARM_TOGGLE	Output toggles when alarm flag rises
6	ALARM_FLAG	Output is a copy of the alarm flag
7	PROG_PULSE	Duty cycle programmable pulse

Bit 15 – HIGHPPM HIGH PPM Correction

If the absolute value of the correction to be applied is lower than 30 ppm, it is recommended to clear HIGHPPM. HIGHPPM set to 1 is recommended for 30 ppm correction and above.

Formula:

If HIGHPPM = 0, then the clock frequency correction range is from 1.5 ppm up to 98 ppm. The RTC accuracy is less than 1 ppm for a range correction from 1.5 ppm up to 30 ppm.

The correction field must be programmed according to the required correction in ppm; the formula is as follows:

$$\text{CORRECTION} = \frac{3906}{20 \times \text{ppm}} - 1$$

The value obtained must be rounded to the nearest integer prior to being programmed into CORRECTION field.

If HIGHPPM = 1, then the clock frequency correction range is from 30.5 ppm up to 1950 ppm. The RTC accuracy is less than 1 ppm for a range correction from 30.5 ppm up to 90 ppm.

The correction field must be programmed according to the required correction in ppm; the formula is as follows:

$$\text{CORRECTION} = \frac{3906}{\text{ppm}} - 1$$

The value obtained must be rounded to the nearest integer prior to be programmed into CORRECTION field.

If NEGPPM is set to 1, the ppm correction is negative (used to correct crystals that are faster than the nominal 32.768 kHz).

Value	Name	Description
0	DISABLED	Lower range ppm correction with accurate correction (below 30 ppm correction).
1	ENABLED	Higher range ppm correction with accurate correction (above 30 ppm correction).

Bits 14:8 – CORRECTION[6:0] Slow Clock Correction

Value	Name	Description
0	DISABLED	No correction
1–127	ENABLED	The slow clock will be corrected according to the formula given in HIGHPPM description.

Bit 4 – NEGPPM Negative PPM Correction

See CORRECTION and HIGHPPM field descriptions.

NEGPPM must be cleared to correct a crystal oscillator frequency slower than 32.768 kHz.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Name	Description
0	DISABLED	Positive correction (the divider will be slightly higher than 32768).
1	ENABLED	Negative correction (the divider will be slightly lower than 32768).

Bit 2 – UTC UTC Time Format

It is forbidden to write a one to the UTC and PERSIAN bits at the same time.

Value	Name	Description
0	DISABLED	Gregorian or Persian calendar.
1	ENABLED	UTC format.

Bit 1 – PERSIAN PERSIAN Calendar

It is forbidden to write a one to the UTC and PERSIAN bits at the same time.

Value	Name	Description
0	DISABLED	Gregorian calendar.
1	ENABLED	Persian calendar.

Bit 0 – HRMOD 12-/24-hour Mode

Value	Name	Description
0	24HOURS	24-hour mode is selected.
1	AMPM	12-hour mode is selected.

© 2026 Microchip Technology Inc. and its subsidiaries

22.6.3. RTC Time Register

Name: RTC_TIMR
Offset: 0x08
Reset: 0x00000000
Property: Read/Write

In UTC mode, this register view is not relevant, see RTC_Time Alarm register (UTC_MODE).

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode Register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
		AMPM	HOUR[5:0]					
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
		MIN[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		SEC[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit 22 – AMPM Ante Meridiem Post Meridiem Indicator
This bit is the AM/PM indicator in 12-hour mode.

Value	Name	Description
0	AM	AM.
1	PM	PM.

Bits 21:16 – HOUR[5:0] Current Hour
The range that can be set is 1–12 (BCD) in 12-hour mode or 0–23 (BCD) in 24-hour mode.

Bits 14:8 – MIN[6:0] Current Minute
The range that can be set is 0–59 (BCD).
The lowest four bits encode the units. The higher bits encode the tens.

Bits 6:0 – SEC[6:0] Current Second
The range that can be set is 0–59 (BCD).
The lowest four bits encode the units. The higher bits encode the tens.

© 2026 Microchip Technology Inc. and its subsidiaries

22.6.4. RTC Time Register (UTC_MODE)

Name: RTC_TIMR (UTC_MODE)
Offset: 0x08
Reset: 0x00000000
Property: Read/Write

This configuration is relevant only if UTC = 1 in RTC_MR.

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
	UTC_TIME[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	UTC_TIME[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	UTC_TIME[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	UTC_TIME[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – UTC_TIME[31:0] Current UTC Time
Any value can be set.

© 2026 Microchip Technology Inc. and its subsidiaries

22.6.5. RTC Calendar Register

Name: RTC_CALR
Offset: 0x0C
Reset: 0x01411920
Property: Read/Write

In UTC mode, values read in this register are not relevant.

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode Register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
			DATE[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	1

Bit	23	22	21	20	19	18	17	16
	DAY[2:0]			MONTH[4:0]				
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8
	YEAR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	1	1	0	0	1

Bit	7	6	5	4	3	2	1	0
		CENT[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	1	0	0	0	0	0

Bits 29:24 – DATE[5:0] Current Day in Current Month
The range that can be set is 01–31 (BCD).
The lowest four bits encode the units. The higher bits encode the tens.

Bits 23:21 – DAY[2:0] Current Day in Current Week
The range that can be set is 1–7 (BCD).
The coding of the number (which number represents which day) is user-defined as it has no effect on the date counter.

Bits 20:16 – MONTH[4:0] Current Month
The range that can be set is 01–12 (BCD).
The lowest four bits encode the units. The higher bits encode the tens.

Bits 15:8 – YEAR[7:0] Current Year
The range that can be set is 00–99 (BCD).
The lowest four bits encode the units. The higher bits encode the tens.

Bits 6:0 – CENT[6:0] Current Century
Note: Value 20 (BCD) is always written in CENT whatever the value entered, thus there is no trigger event on RTC_VER.NVCAL regarding CENT.

The range that can be set is 20 (Gregorian) (BCD) or 13-14 (Persian) (BCD).
The lowest four bits encode the units. The higher bits encode the tens.

© 2026 Microchip Technology Inc. and its subsidiaries

22.6.6. RTC Time Alarm Register

Name: RTC_TIMALR
Offset: 0x10
Reset: 0x00000000
Property: Read/Write

In UTC mode, this register view is not relevant, see RTC_TIMALR (UTC_MODE).

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode Register (SYSC_WPMR).

To change one of the SEC, MIN, HOUR fields, it is recommended to disable the field before changing the value and then re-enable it after the change has been made. This requires up to three accesses to the RTC_TIMALR. The first access clears the enable corresponding to the field to change (SECEN, MINEN, HOUREN). If the field is already cleared, this access is not required. The second access performs the change of the value (SEC, MIN, HOUR). The third access is required to re-enable the field by writing 1 in SECEN, MINEN, HOUREN fields.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	HOUREN	AMPM	HOUR[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	MINEN	MIN[6:0]						
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SECEN	SEC[6:0]						
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 23 – HOUREN Hour Alarm Enable

Value	Name	Description
0	DISABLED	The hour-matching alarm is disabled.
1	ENABLED	The hour-matching alarm is enabled.

Bit 22 – AMPM AM/PM Indicator
Corresponds to the BCD-coded hour counter.

Bits 21:16 – HOUR[5:0] Hour Alarm
Corresponds to the BCD-coded hour counter.

Bit 15 – MINEN Minute Alarm Enable

Value	Name	Description
0	DISABLED	The minute-matching alarm is disabled.
1	ENABLED	The minute-matching alarm is enabled.

© 2026 Microchip Technology Inc. and its subsidiaries

Bits 14:8 – MIN[6:0] Minute Alarm
Corresponds to the BCD-coded minute counter.

Bit 7 – SECEN Second Alarm Enable

Value	Name	Description
0	DISABLED	The second-matching alarm is disabled.
1	ENABLED	The second-matching alarm is enabled.

Bits 6:0 – SEC[6:0] Second Alarm
Corresponds to the BCD-coded second counter.

© 2026 Microchip Technology Inc. and its subsidiaries

22.6.7. RTC Time Alarm Register (UTC_MODE)

Name: RTC_TIMALR (UTC_MODE)
Offset: 0x10
Reset: 0x00000000
Property: Read/Write

This register view is relevant only if UTC = 1 in RTC_MR.

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
	UTC_TIME[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	UTC_TIME[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	UTC_TIME[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	UTC_TIME[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – UTC_TIME[31:0] UTC_TIME Alarm
Corresponds to the UTC time counter. To change it, proceed as follows:

1. Disable the UTC alarm by clearing the UTCEN bit in RTC_CALALR if it is not already cleared.
2. Change the UTC_TIME alarm value.
3. Enable the UTC alarm by setting the UTCEN bit in RTC_CALALR.

© 2026 Microchip Technology Inc. and its subsidiaries

22.6.8. RTC Calendar Alarm Register

Name: RTC_CALALR
Offset: 0x14
Reset: 0x01010000
Property: Read/Write

In UTC mode, this register view is not relevant, see RTC_CALALR (UTC_MODE).

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode Register (SYSC_WPMR).

To change one of the DATE, MONTH fields, it is recommended to disable the field before changing the value and then re-enable it after the change has been made. This requires up to three accesses to the RTC_CALALR. The first access clears the enable corresponding to the field to change (DATEEN, MTHEN). If the field is already cleared, this access is not required. The second access performs the change of the value (DATE, MONTH). The third access is required to re-enable the field by writing 1 in DATEEN, MTHEN fields.

Bit	31	30	29	28	27	26	25	24
	DATEEN		DATE[5:0]					
Access	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset	0		0	0	0	0	0	1

Bit	23	22	21	20	19	18	17	16
	MTHEN			MONTH[4:0]				
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			0	0	0	0	1

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 31 – DATEEN Date Alarm Enable

Value	Name	Description
0	DISABLED	The date-matching alarm is disabled.
1	ENABLED	The date-matching alarm is enabled.

Bits 29:24 – DATE[5:0] Date Alarm
Corresponds to the BCD-coded date counter.

Bit 23 – MTHEN Month Alarm Enable

Value	Name	Description
0	DISABLED	The month-matching alarm is disabled.
1	ENABLED	The month-matching alarm is enabled.

Bits 20:16 – MONTH[4:0] Month Alarm
Corresponds to the BCD-coded month counter.

© 2026 Microchip Technology Inc. and its subsidiaries

22.6.9. RTC Calendar Alarm Register (UTC_MODE)

Name: RTC_CALALR (UTC_MODE)
Offset: 0x14
Reset: 0x01010000
Property: Read/Write

This register view is relevant only if UTC = 1 in RTC_MR.

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								UTCEN
Access								R/W
Reset								0

Bit 0 – UTCEN UTC Alarm Enable
Corresponds to the BCD-coded month counter.

Value	Name	Description
0	DISABLED	The UTC-matching alarm is disabled.
1	ENABLED	The UTC-matching alarm is enabled.

© 2026 Microchip Technology Inc. and its subsidiaries

22.6.10. RTC Status Register

Name: RTC_SR
Offset: 0x18
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			TDERR	CALEV	TIMEV	SEC	ALARM	ACKUPD
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit 5 – TDERR Time and/or Date Free Running Error
If the RTC is configured in UTC mode, the value returned by this field is not relevant.

Value	Name	Description
0	CORRECT	The internal free running counters are carrying valid values since the last read of the Status Register (RTC_SR).
1	ERR_TIMEDATE	The internal free running counters have been corrupted (invalid date or time, non-BCD values) since the last read and/or they are still invalid.

Bit 4 – CALEV Calendar Event
The calendar event is selected in the CALEVSEL field in the Control Register (RTC_CR) and can be any one of the following events: week change, month change and year change. If the RTC is configured in UTC mode, the value returned by this field is not relevant.

Value	Name	Description
0	NO_CALEVENT	No calendar event has occurred since the last clear.
1	CALEVENT	At least one calendar event has occurred since the last clear.

Bit 3 – TIMEV Time Event
The time event is selected in the TIMEVSEL field in the Control Register (RTC_CR) and can be any one of the following events: minute change, hour change, noon, midnight (day change). If the RTC is configured in UTC mode, the value returned by this field is not relevant.

Value	Name	Description
0	NO_TIMEVENT	No time event has occurred since the last clear.
1	TIMEVENT	At least one time event has occurred since the last clear.

Bit 2 – SEC Second Event

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Name	Description
0	NO_SECEVENT	No second event has occurred since the last clear.
1	SECEVENT	At least one second event has occurred since the last clear.

Bit 1 – ALARM Alarm Flag

Value	Name	Description
0	NO_ALARMEVENT	No alarm matching condition occurred.
1	ALARMEVENT	An alarm matching condition has occurred.

Bit 0 – ACKUPD Acknowledge for Update

Value	Name	Description
0	FREERUN	Time and calendar registers cannot be updated.
1	UPDATE	Time and calendar registers can be updated.

© 2026 Microchip Technology Inc. and its subsidiaries

22.6.11. RTC Status Clear Command Register

Name: RTC_SCCR
Offset: 0x1C
Reset: –
Property: Write-only

Due to the asynchronous operation of the RTC with respect to the system bus, three slow clock cycles must be completed between two accesses to this register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Clears the corresponding status flag in the Status register (RTC_SR).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			TDERRCLR	CALCLR	TIMCLR	SECCLR	ALRCLR	ACKCLR
Access			W	W	W	W	W	W
Reset			–	–	–	–	–	–

Bit 5 – TDERRCLR Time and/or Date Free Running Error Clear
If the RTC is configured in UTC mode, this bit has no effect.

Bit 4 – CALCLR Calendar Clear
If the RTC is configured in UTC mode, this bit has no effect.

Bit 3 – TIMCLR Time Clear
If the RTC is configured in UTC mode, this bit has no effect.

Bit 2 – SECCLR Second Clear

Bit 1 – ALRCLR Alarm Clear

Bit 0 – ACKCLR Acknowledge Clear

© 2026 Microchip Technology Inc. and its subsidiaries

22.6.12. RTC Interrupt Enable Register

Name: RTC_IER
Offset: 0x20
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the System Controller Write Protection Mode Register (SYSC_WPMR).

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			TDERREN	CALEN	TIMEN	SECEN	ALREN	ACKEN
Access			W	W	W	W	W	W
Reset			–	–	–	–	–	–

Bit 5 – TDERREN Time and/or Date Error Interrupt Enable
If the RTC is configured in UTC mode, this bit has no effect.

Bit 4 – CALEN Calendar Event Interrupt Enable
If the RTC is configured in UTC mode, this bit has no effect.

Bit 3 – TIMEN Time Event Interrupt Enable
If the RTC is configured in UTC mode, this bit has no effect.

Bit 2 – SECEN Second Event Interrupt Enable

Bit 1 – ALREN Alarm Interrupt Enable

Bit 0 – ACKEN Acknowledge Update Interrupt Enable

© 2026 Microchip Technology Inc. and its subsidiaries

22.6.13. RTC Interrupt Disable Register

Name: RTC_IDR
Offset: 0x24
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the System Controller Write Protection Mode Register (SYSC_WPMR).

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			TDERRDIS	CALDIS	TIMDIS	SECDIS	ALRDIS	ACKDIS
Access			W	W	W	W	W	W
Reset			–	–	–	–	–	–

Bit 5 – TDERRDIS Time and/or Date Error Interrupt Disable
If the RTC is configured in UTC mode, this bit has no effect.

Bit 4 – CALDIS Calendar Event Interrupt Disable
If the RTC is configured in UTC mode, this bit has no effect.

Bit 3 – TIMDIS Time Event Interrupt Disable
If the RTC is configured in UTC mode, this bit has no effect.

Bit 2 – SECDIS Second Event Interrupt Disable

Bit 1 – ALRDIS Alarm Interrupt Disable

Bit 0 – ACKDIS Acknowledge Update Interrupt Disable

© 2026 Microchip Technology Inc. and its subsidiaries

22.6.14. RTC Interrupt Mask Register

Name: RTC_IMR
Offset: 0x28
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is disabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			TDERR	CAL	TIM	SEC	ALR	ACK
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit 5 – TDERR Time and/or Date Error Mask
If the RTC is configured in UTC mode, this bit has no effect.

Bit 4 – CAL Calendar Event Interrupt Mask
If the RTC is configured in UTC mode, this bit has no effect.

Bit 3 – TIM Time Event Interrupt Mask
If the RTC is configured in UTC mode, this bit has no effect.

Bit 2 – SEC Second Event Interrupt Mask

Bit 1 – ALR Alarm Interrupt Mask

Bit 0 – ACK Acknowledge Update Interrupt Mask

© 2026 Microchip Technology Inc. and its subsidiaries

22.6.15. RTC Valid Entry Register

Name: RTC_VER
Offset: 0x2C
Reset: 0x00000000
Property: Read-only

If the RTC is configured in UTC mode, the values returned by this register are not relevant.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					NVCALALR	NVTIMALR	NVCAL	NVTIM
Access					R	R	R	R
Reset					0	0	0	0

Bit 3 – NVCALALR Non-valid Calendar Alarm

Value	Description
0	No invalid data has been detected in RTC_CALALR or in RTC_MR.UTC=1.
1	RTC_CALALR has contained invalid data since it was last programmed.

Bit 2 – NVTIMALR Non-valid Time Alarm

Value	Description
0	No invalid data has been detected in RTC_TIMALR or in RTC_MR.UTC=1.
1	RTC_TIMALR has contained invalid data since it was last programmed.

Bit 1 – NVCAL Non-valid Calendar

Value	Description
0	No invalid data has been detected in RTC_CALR or in RTC_MR.UTC=1.
1	RTC_CALR has contained invalid data since it was last programmed.

Bit 0 – NVTIM Non-valid Time

Value	Description
0	No invalid data has been detected in RTC_TIMR or in RTC_MR.UTC=1.
1	RTC_TIMR has contained invalid data since it was last programmed.

© 2026 Microchip Technology Inc. and its subsidiaries

22.6.16. RTC Tamper Control Register

Name: RTC_TCR
Offset: 0x34
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode Register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							FGPBRCLR	TAMPCLR
Access							R/W	R/W
Reset							0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				TAMPEN4	TAMPEN3	TAMPEN2	TAMPEN1	TAMPEN0
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit 17 – FGPBRCLR Full GPBR Clear

Value	Name	Description
0	HALF	If TAMPCLR is set to '1', a tamper event immediately clears GPBR0 to GPBR11.
1	FULL	If TAMPCLR is set to '1', a tamper event immediately clears all GPBRs.

Bit 16 – TAMPCLR Tamper Clear

Value	Name	Description
0	NOT_ENABLE	A Tamper event does not create an immediate clear on GPBR registers.
1	ENABLE	Tamper event on TMP0 or TMP4 generates an immediate clear on GPBR registers. The number of cleared GPBR registers depends on the value of FGPBRCLR.

Bits 0, 1, 2, 3, 4 – TAMPENx Tampering of TMPx input enabled

Value	Name	Description
0	TIMESTAMP_OFF	Tamper event on TMPx input is not timestamped.
1	TIMESTAMP_ON	Tamper event on TMPx input is timestamped.

© 2026 Microchip Technology Inc. and its subsidiaries

22.6.17. RTC Tamper Input Status Register

Name: RTC_TISR
Offset: 0x38
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				TISR4	TISR3	TISR2	TISR1	TISR0
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bits 0, 1, 2, 3, 4 – TISRx TMPx Input Status Register

Value	Description
0	TMPx input level is low
1	TMPx input level is high

© 2026 Microchip Technology Inc. and its subsidiaries

22.6.18. RTC First Stamping Time Register x

Name: RTC_FSTRx
Offset: 0x40 + x*0x04 [x=0..4]
Reset: 0x00000000
Property: Read-only

This configuration is relevant only if RTC_MR.UTC = 0.

RTC_FSTRx reports the timestamp time of the first tamper event that occurred on TMPx after the last read of RTC_LSDRx.

Bit	31	30	29	28	27	26	25	24
	BACKUP				TEVCNT[3:0]			
Access	R				R	R	R	R
Reset	0				0	0	0	0

Bit	23	22	21	20	19	18	17	16
		AMPM	HOUR[5:0]					
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
		MIN[6:0]						
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		SEC[6:0]						
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit 31 – BACKUP System Mode of the Tamper (cleared by reading RTC_LSDRx)

Value	Description
0	The system is not in Backup mode when the tamper event occurs.
1	The system is in Backup mode when the tamper event occurs.

Bits 27:24 – TEVCNT[3:0] Tamper Events Counter (cleared by reading RTC_LSDRx)
Each time a tamper event occurs, this counter is incremented. This counter saturates at 15. Once this value is reached, it is no longer possible to know the exact number of tamper events.
If this field is not null, at least one tamper event has occurred since the last register reset and the values stored in RTC_FSTRx, RTC_FSDRx, RTC_LSTRx, RTC_LSDRx registers are valid.

Bit 22 – AMPM AM/PM Indicator of the Tamper (cleared by reading RTC_LSDRx)

Bits 21:16 – HOUR[5:0] Hours of the Tamper (cleared by reading RTC_LSDRx)

Bits 14:8 – MIN[6:0] Minutes of the Tamper (cleared by reading RTC_LSDRx)

Bits 6:0 – SEC[6:0] Seconds of the Tamper (cleared by reading RTC_LSDRx)

© 2026 Microchip Technology Inc. and its subsidiaries

22.6.19. RTC First Stamping Time Register x (UTC_MODE)

Name: RTC_FSTRx (UTC_MODE)
Offset: 0x40 + x*0x04 [x=0..4]
Reset: 0x00000000
Property: Read-only

This configuration is relevant only if RTC_MR.UTC = 1.

RTC_FSTRx reports the timestamp time of the first tamper event that occurred on TMPx after the last read of RTC_LSDRx.

Bit	31	30	29	28	27	26	25	24
	BACKUP				TEVCNT[3:0]			
Access	R				R	R	R	R
Reset	0				0	0	0	0

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 31 – BACKUP System Mode of the Tamper (cleared by reading RTC_LSDRx)

Value	Description
0	The system is not in Backup mode when the tamper event occurs.
1	The system is in Backup mode when the tamper event occurs.

Bits 27:24 – TEVCNT[3:0] Tamper Events Counter (cleared by reading RTC_LSDRx)
Each time a tamper event occurs, this counter is incremented. This counter saturates at 15. Once this value is reached, it is no longer possible to know the exact number of tamper events.
If this field is not null, at least one tamper event has occurred since the last register reset and the values stored in RTC_FSTRx, RTC_FSDRx, RTC_LSTRx, RTC_LSDRx registers are valid.

© 2026 Microchip Technology Inc. and its subsidiaries

22.6.20. RTC First Stamping Date Register x

Name: RTC_FSDRx
Offset: 0x44 + x*0x04 [x=0..4]
Reset: 0x00000000
Property: Read-only

This configuration is relevant only if RTC_MR.UTC = 0.

RTC_FSDRx reports the timestamp date of the first tamper event that occurred on TMPx pins after the last read of RTC_LSDRx.

Bit	31	30	29	28	27	26	25	24
			DATE[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DAY[2:0]			MONTH[4:0]				
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
		YEAR[6:0]						
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		CENT[6:0]						
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bits 29:24 – DATE[5:0] Date of the Tamper (cleared by reading RTC_LSDRx)

Bits 23:21 – DAY[2:0] Day of the Tamper (cleared by reading RTC_LSDRx)

Bits 20:16 – MONTH[4:0] Month of the Tamper (cleared by reading RTC_LSDRx)

Bits 14:8 – YEAR[6:0] Year of the Tamper (cleared by reading RTC_LSDRx)

Bits 6:0 – CENT[6:0] Century of the Tamper (cleared by reading RTC_LSDRx)

© 2026 Microchip Technology Inc. and its subsidiaries

22.6.21. RTC First Stamping Date Register x (UTC_MODE)

Name: RTC_FSDRx (UTC_MODE)
Offset: 0x44 + x*0x04 [x=0..4]
Reset: 0x00000000
Property: Read-only

This configuration is relevant only if RTC_MR.UTC = 1.

RTC_FSDRx reports the timestamp date of the first tamper event that occurred on TMPx pins after the last read of RTC_LSDRx.

Bit	31	30	29	28	27	26	25	24
	UTC_TIME[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	UTC_TIME[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	UTC_TIME[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	UTC_TIME[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – UTC_TIME[31:0] Time of the Tamper (cleared by reading RTC_LSDRx)

© 2026 Microchip Technology Inc. and its subsidiaries

22.6.22. RTC Last Stamping Time Register x

Name: RTC_LSTRx
Offset: 0x48 + x*0x04 [x=0..4]
Reset: 0x00000000
Property: Read-only

This configuration is relevant only if RTC_MR.UTC = 0.

RTC_LSTRx reports the timestamp time of the last tamper event that occured on TMPx pins after the last read of RTC_LSDRx.

Bit	31	30	29	28	27	26	25	24
	BACKUP							
Access	R							
Reset	0							

Bit	23	22	21	20	19	18	17	16
		AMPM	HOUR[5:0]					
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
		MIN[6:0]						
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		SEC[6:0]						
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit 31 – BACKUP System Mode of the Tamper (cleared by reading RTC_LSDRx)

Value	Description
0	The system is not in Backup mode when the tamper event occurs.
1	The system is in Backup mode when the tamper event occurs.

Bit 22 – AMPM AM/PM Indicator of the Tamper (cleared by reading RTC_LSDRx)

Bits 21:16 – HOUR[5:0] Hours of the Tamper (cleared by reading RTC_LSDRx)

Bits 14:8 – MIN[6:0] Minutes of the Tamper (cleared by reading RTC_LSDRx)

Bits 6:0 – SEC[6:0] Seconds of the Tamper (cleared by reading RTC_LSDRx)

© 2026 Microchip Technology Inc. and its subsidiaries

22.6.23. RTC Last Stamping Time Register x (UTC_MODE)

Name: RTC_LSTRx (UTC_MODE)
Offset: 0x48 + x*0x04 [x=0..4]
Reset: 0x00000000
Property: Read-only

This configuration is relevant only if RTC_MR.UTC = 1.

RTC_LSTRx reports the timestamp time of the last tamper event that occured on TMPx pins after the last read of RTC_LSDRx.

Bit	31	30	29	28	27	26	25	24
	BACKUP							
Access	R							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 31 – BACKUP System Mode of the Tamper (cleared by reading RTC_LSDRx)

Value	Description
0	The system is not in Backup mode when the tamper event occurs.
1	The system is in Backup mode when the tamper event occurs.

© 2026 Microchip Technology Inc. and its subsidiaries

22.6.24. RTC Last Stamping Date Register x

Name: RTC_LSDRx
Offset: 0x4C + x*0x04 [x=0..4]
Reset: 0x00000000
Property: Read-only

This configuration is relevant only if RTC_MR.UTC = 0.

RTC_LSDRx reports the timestamp date of the last tamper event that occurred on TMPx pins after the last read of RTC_LSDRx.

Bit	31	30	29	28	27	26	25	24
			DATE[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DAY[2:0]			MONTH[4:0]				
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
		YEAR[6:0]						
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		CENT[6:0]						
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bits 29:24 – DATE[5:0] Date of the Tamper (cleared on read)

Bits 23:21 – DAY[2:0] Day of the Tamper (cleared on read)

Bits 20:16 – MONTH[4:0] Month of the Tamper (cleared on read)

Bits 14:8 – YEAR[6:0] Year of the Tamper (cleared on read)

Bits 6:0 – CENT[6:0] Century of the Tamper (cleared on read)

© 2026 Microchip Technology Inc. and its subsidiaries

22.6.25. RTC Last Stamping Date Register x (UTC_MODE)

Name: RTC_LSDRx (UTC_MODE)
Offset: 0x4C + x*0x04 [x=0..4]
Reset: 0x00000000
Property: Read-only

This configuration is relevant only if RTC_MR.UTC = 1.

RTC_LSDRx reports the timestamp date of the last tamper event that occurred on TMPx pins after the last read of RTC_LSDRx.

Bit	31	30	29	28	27	26	25	24
	UTC_TIME[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	UTC_TIME[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	UTC_TIME[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	UTC_TIME[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – UTC_TIME[31:0] Time of the Tamper (cleared on read)

© 2026 Microchip Technology Inc. and its subsidiaries

23. Real-time Timer (RTT)

23.1. Description

The Real-time Timer (RTT) is built around a 32-bit counter used to count roll-over events of the programmable 16-bit prescaler driven from the 32-kHz slow clock source. It generates a periodic interrupt and/or triggers an alarm on a programmed value.

The RTT can also be configured to be driven by the 1Hz RTC signal, thus taking advantage of a calibrated 1Hz clock.

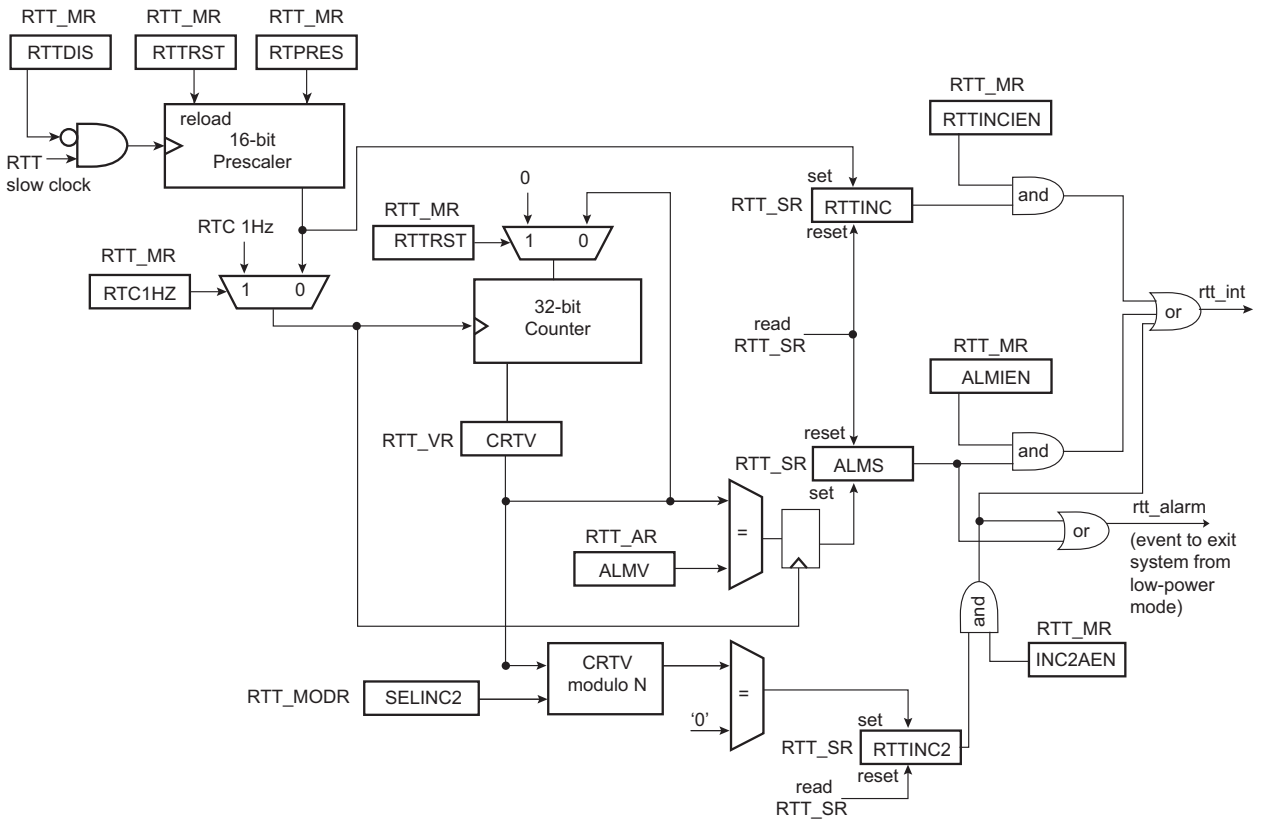
The slow clock source can be fully disabled to reduce power consumption when only an elapsed seconds count is required.

23.2. Embedded Characteristics

- 32-bit Free-running Counter on Prescaled Slow Clock or RTC Calibrated 1Hz Clock
- 16-bit Configurable Prescaler
- Interrupt on Alarm or Counter Increment
- Programmable Event

23.3. Block Diagram

Figure 23-1. RTT Block Diagram



23.4. Functional Description

The programmable 16-bit prescaler value can be configured through the RTPRES field in the RTT Mode register (RTT_MR).

© 2026 Microchip Technology Inc. and its subsidiaries

Configuring RTPRES to 0x8000 (default value) corresponds to feeding the real-time counter with a 1Hz signal (if the slow clock is 32.768 kHz). The 32-bit counter can count up to 2^{32} seconds, corresponding to more than 136 years, then roll over to 0. Bit RTTINC in the RTT Status register (RTT_SR) is set each time there is a prescaler roll-over (see the figure below).

The real-time 32-bit counter can also be supplied by the 1Hz RTC clock. This mode is applicable when the RTC 1Hz is calibrated (RTC_MR.CORRECTION ≠ 0) in order to ensure the synchronism between RTC and RTT counters.

Setting RTT_MR.RTC1HZ drives the 32-bit RTT counter from the 1Hz RTC clock. In this mode, RTPRES has no effect on the 32-bit counter.

The prescaler roll-over generates an increment of the RTT counter if RTC1HZ = 0. Otherwise, if RTC1HZ = 1, the RTT counter is incremented every second. RTTINC is set independently from the 32-bit counter increment.

The RTT can also be used as a free-running timer with a lower time-base. The best accuracy is achieved by writing RTT_MR.RTPRES to 3.

Programming RTPRES to 1 or 2 is forbidden.

The CRTV field can be read at any time in the RTT Value register (RTT_VR). As this value can be updated asynchronously with the peripheral bus clock, CRTV must be read twice at the same value to read a correct value.

The current value of the counter is compared with the value written in the RTT Alarm register (RTT_AR). If the counter value matches the alarm, the ALMS bit in the RTT_SR is set. The RTT_AR is set to its maximum value (0xFFFFFFFF) after a reset.

ALMS is always a source of the RTT alarm signal that may be used to exit the system from low power modes (see the RTT Block Diagram above).

The alarm interrupt must be disabled (RTT_MR.ALMIEN must be cleared) when writing a new value to RTT_AR.ALMV.

RTT_SR.RTTINC can be used to start a periodic interrupt. The period is one second when RTT_MR.RTPRES = 0x8000 and the slow clock = 32.768 kHz.

RTT_MR.RTTINCIEN must be cleared prior to writing a new value in RTT_MR. RTPRES.

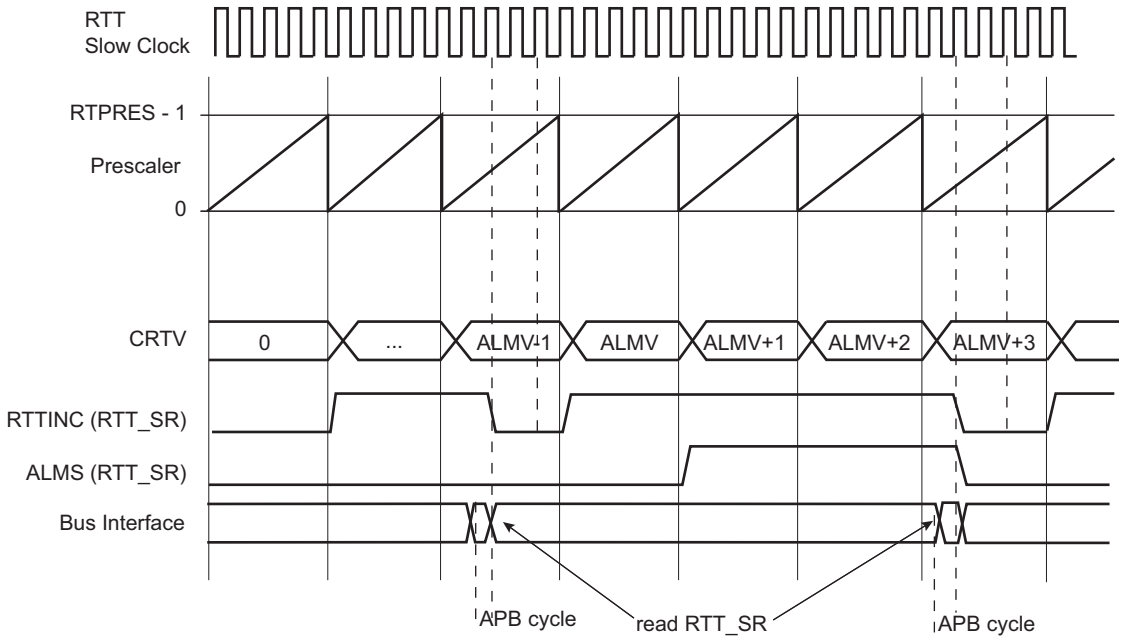
Reading RTT_SR automatically clears RTT_SR.RTTINC and RTT_SR.ALMS.

Writing RTT_MR.RTTRST immediately reloads and restarts the clock divider with the new programmed value. This also resets the 32-bit counter.

When not used, the RTT can be disabled in order to suppress dynamic power consumption in this module. This can be achieved by setting RTT_MR.RTTDIS.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 23-2. RTT Counting



The RTTINC2 flag is set when the number of prescaler roll-overs programmed through the SELINC2 field in the RTT Modulo Selection register (RTT_MODR) has been reached since the last read of RTT_SR.

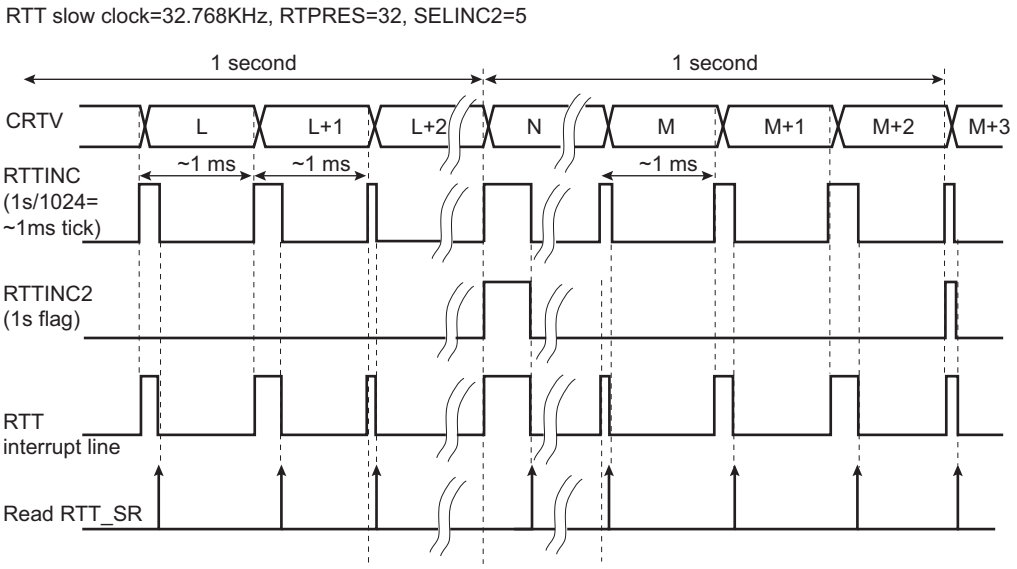
For example, it is possible to generate 2 sources of interrupt of different periods with flags RTTINC and RTTINC2. If the RTT slow clock frequency is 32.768 kHz and RTPRES=32, the RTTINC flag rises 1024 times per second (less than 1 ms period). If the field SELINC2=5, the RTTINC2 flag rises once per second.

If RTTINC is defined as the unique source of interrupt (RTTINCEN=1, ALMIEN=0 and INC2AEN=0 in RTT_MR), the value read in RTT_SR by the interrupt handler determines if the current interrupt event corresponds to a 1-second event (RTT_SR[2:1]=3) or to a 1-millisecond event (RTT_SR[2:1]=1). See the figure below.

If the bit INC2AEN=1, RTTINC2 flag is also a source for the RTT alarm signal. See RTT Block Diagram.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 23-3. RTTINC2 Behavior



© 2026 Microchip Technology Inc. and its subsidiaries

23.5. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	RTT_MR	31:24								RTC1HZ
		23:16			INC2AEN	RTTDIS		RTTRST	RTTINCIEN	ALMIEN
		15:8	RTPRES[15:8]							
		7:0	RTPRES[7:0]							
0x04	RTT_AR	31:24	ALMV[31:24]							
		23:16	ALMV[23:16]							
		15:8	ALMV[15:8]							
		7:0	ALMV[7:0]							
0x08	RTT_VR	31:24	CRTV[31:24]							
		23:16	CRTV[23:16]							
		15:8	CRTV[15:8]							
		7:0	CRTV[7:0]							
0x0C	RTT_SR	31:24								
		23:16								
		15:8								
		7:0						RTTINC2	RTTINC	ALMS
0x10	RTT_MODR	31:24								
		23:16								
		15:8								
		7:0						SELINC2[2:0]		

© 2026 Microchip Technology Inc. and its subsidiaries

23.5.1. Real-time Timer Mode Register

Name: RTT_MR
Offset: 0x00
Reset: 0x00008000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode Register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
								RTC1HZ
Access								R/W
Reset								0

Bit	23	22	21	20	19	18	17	16
			INC2AEN	RTTDIS		RTTRST	RTTINCIEN	ALMIEN
Access			R/W	R/W		R/W	R/W	R/W
Reset			0	0		0	0	0

Bit	15	14	13	12	11	10	9	8
	RTPRES[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RTPRES[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 24 – RTC1HZ Real-Time Clock 1Hz Clock Selection

Value	Description
0	The RTT 32-bit counter is driven by the 16-bit prescaler roll-over events.
1	The RTT 32-bit counter is driven by the 1Hz RTC clock.

Bit 21 – INC2AEN RTTINC2 Alarm and Interrupt Enable

Value	Description
0	The RTTINC2 flag is not a source of the RTT alarm signal nor a source of interrupt.
1	The RTTINC2 flag is a source of the RTT alarm signal and a source of interrupt.

Bit 20 – RTTDIS Real-time Timer Disable

Value	Description
0	The RTT is enabled.
1	The RTT is disabled (no dynamic power consumption).

Bit 18 – RTTRST Real-time Timer Restart

Value	Description
0	No effect.
1	Reloads and restarts the clock divider with the new programmed value. This also resets the 32-bit counter.

Bit 17 – RTTINCIEN Real-time Timer Increment Interrupt Enable

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	The bit RTTINC in RTT_SR has no effect on interrupt.
1	The bit RTTINC in RTT_SR asserts interrupt.

Bit 16 – ALMIEN Alarm Interrupt Enable

Value	Description
0	The bit ALMS in RTT_SR has no effect on interrupt.
1	The bit ALMS in RTT_SR asserts interrupt.

Bits 15:0 – RTPRES[15:0] Real-time Timer Prescaler Value

Defines the number of RTT slow clock periods required to increment the Real-time timer. The RTTINCIEN bit must be cleared prior to writing a new RTPRES value.
RTPRES is defined as follows:

- RTPRES = 0: The prescaler period is equal to 2^{16} * slow clock periods.
- RTPRES = 1 or 2: forbidden.
- RTPRES ≠ 0, 1 or 2: The prescaler period is equal to RTPRES * slow clock periods.

© 2026 Microchip Technology Inc. and its subsidiaries

23.5.2. Real-time Timer Alarm Register

Name: RTT_AR
Offset: 0x04
Reset: 0xFFFFFFFF
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode Register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
	ALMV[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	23	22	21	20	19	18	17	16
	ALMV[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8
	ALMV[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
	ALMV[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 31:0 – ALMV[31:0] Alarm Value
When the CRTV value in RTT_VR equals the ALMV field, the ALMS flag is set in RTT_SR.
The alarm interrupt must be disabled (ALMIEN must be cleared in RTT_MR) when writing a new ALMV value.

© 2026 Microchip Technology Inc. and its subsidiaries

23.5.3. Real-time Timer Value Register

Name: RTT_VR
Offset: 0x08
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	CRTV[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CRTV[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CRTV[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CRTV[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CRTV[31:0] Current Real-time Value
Returns the current value of the RTT.
As CRTV can be updated asynchronously, it must be read twice at the same value.

© 2026 Microchip Technology Inc. and its subsidiaries

23.5.4. Real-time Timer Status Register

Name: RTT_SR
Offset: 0x0C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						RTTINC2	RTTINC	ALMS
Access						R	R	R
Reset						0	0	0

Bit 2 – RTTINC2 Predefined Number of Prescaler Roll-overs Status (cleared on read)

Value	Description
0	SELINC2 = 0 or the number of prescaler roll-overs programmed through the SELINC2 field in RTT_MODR has not been reached since the last read of the RTT_SR.
1	The number of prescaler roll-overs programmed through the SELINC2 field has been reached since the last read of the RTT_SR.

Bit 1 – RTTINC Prescaler Roll-over Status (cleared on read)

Value	Description
0	No prescaler roll-over occurred since the last read of the RTT_SR.
1	Prescaler roll-over occurred since the last read of the RTT_SR.

Bit 0 – ALMS Real-time Alarm Status (cleared on read)

Value	Description
0	The Real-time Alarm has not occurred since the last read of RTT_SR.
1	The Real-time Alarm occurred since the last read of RTT_SR.

© 2026 Microchip Technology Inc. and its subsidiaries

23.5.5. Real-time Timer Modulo Selection Register

Name: RTT_MODR
Offset: 0x10
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode Register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						SELINC2[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 2:0 – SELINC2[2:0] Selection of the 32-bit Counter Modulo to generate RTTINC2 Flag

Value	Name	Description
0	NO_RTTINC2	The RTTINC2 flag never rises.
1	MOD64	The RTTINC2 flag is set when CRTV modulo 64 equals 0.
2	MOD128	The RTTINC2 flag is set when CRTV modulo 128 equals 0.
3	MOD256	The RTTINC2 flag is set when CRTV modulo 256 equals 0.
4	MOD512	The RTTINC2 flag is set when CRTV modulo 512 equals 0.
5	MOD1024	The RTTINC2 flag is set when CRTV modulo 1024 equals 0. Example: If RTPRES=32 then RTTINC2 flag rises once per second if the slow clock is 32.768 kHz.
6	MOD2048	The RTTINC2 flag is set when CRTV modulo 2048 equals 0.
7	MOD4096	The RTTINC2 flag is set when CRTV modulo 4096 equals 0.

© 2026 Microchip Technology Inc. and its subsidiaries

24. General Purpose Backup Registers (GPBR)

24.1. Description

The System Controller embeds 768 bits of General Purpose Backup registers organized as 24 32-bit registers.

It is possible to generate an immediate clear of the content of General Purpose Backup registers 0 to 11 (first half), if a tamper event is detected on one of the tamper pins, TMP0 to TMP4 by writing RTC_TCR.TAMPCLR to ‘1’. The content of the other General Purpose Backup registers (second half) remains unchanged.

The immediate clear on tamper detection can be extended to all General Purpose Backup registers by writing RTC_TCR.FGPBRCLR to ‘1’.

It is also possible to generate an immediate clear of all General Purpose Backup registers if the embedded Flash is erased by writing SUPC_EMR.FLRSGPBR to ‘1’ in the Supply Controller.

SYS_GPBR0 to SYS_GPBR15 can be individually (each 32-bit part-select) read- and write-protected by configuring GPBR_MR. This register is write-once, which means that once it has been configured, the read or write protection is available until the loss of VDDBU.

24.2. Embedded Characteristics

- 768 bits of General Purpose Backup Registers
- Immediate Clear on Tamper Event
- Read and Write Protection for SYS_GPBR0 to SYS_GPBR15

© 2026 Microchip Technology Inc. and its subsidiaries

24.3. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	GPBR_MR	31:24	GPBRRP15	GPBRRP14	GPBRRP13	GPBRRP12	GPBRRP11	GPBRRP10	GPBRRP9	GPBRRP8
		23:16	GPBRRP7	GPBRRP6	GPBRRP5	GPBRRP4	GPBRRP3	GPBRRP2	GPBRRP1	GPBRRP0
		15:8	GPBRWP15	GPBRWP14	GPBRWP13	GPBRWP12	GPBRWP11	GPBRWP10	GPBRWP9	GPBRWP8
		7:0	GPBRWP7	GPBRWP6	GPBRWP5	GPBRWP4	GPBRWP3	GPBRWP2	GPBRWP1	GPBRWP0
0x04 ... 0x07	Reserved									
0x08	SYS_GPBR0	31:24	GPBR_VALUE[31:24]							
		23:16	GPBR_VALUE[23:16]							
		15:8	GPBR_VALUE[15:8]							
		7:0	GPBR_VALUE[7:0]							
...										
0x64	SYS_GPBR23	31:24	GPBR_VALUE[31:24]							
		23:16	GPBR_VALUE[23:16]							
		15:8	GPBR_VALUE[15:8]							
		7:0	GPBR_VALUE[7:0]							

© 2026 Microchip Technology Inc. and its subsidiaries

24.3.1. GPBR Mode Register

Name: GPBR_MR
Offset: 0x0
Reset: 0x00000000
Property: Read/Write-Once

This register is write-once. All bits are cleared at first power-up and on each loss of VDDBU.

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode register (SYSC_WPMR).

Bit	31	30	29	28	27	26	25	24
	GPBRRP15	GPBRRP14	GPBRRP13	GPBRRP12	GPBRRP11	GPBRRP10	GPBRRP9	GPBRRP8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	GPBRRP7	GPBRRP6	GPBRRP5	GPBRRP4	GPBRRP3	GPBRRP2	GPBRRP1	GPBRRP0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	GPBRWP15	GPBRWP14	GPBRWP13	GPBRWP12	GPBRWP11	GPBRWP10	GPBRWP9	GPBRWP8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	GPBRWP7	GPBRWP6	GPBRWP5	GPBRWP4	GPBRWP3	GPBRWP2	GPBRWP1	GPBRWP0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – GPBRRPx GPBRx Read Protection

Value	Description
0	The content of the corresponding GPBR register (32-bit part-select) can be read.
1	The corresponding GPBR register (32-bit part-select) always returns zero when read.

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – GPBRWPx GPBRx Write Protection

Value	Description
0	The corresponding GPBR register (32-bit part-select) can be written.
1	The corresponding GPBR register (32-bit part-select) is write-protected.

© 2026 Microchip Technology Inc. and its subsidiaries

24.3.2. General Purpose Backup Register x [x=0..23]

Name: SYS_GPBRx
Offset: 0x08 + x*0x04 [x=0..23]
Reset: 0x00000000
Property: R/W

This register can only be written if the WPEN bit is cleared in the System Controller Write Protection Mode register (SYSC_WPMR).

These registers are reset at first power-up and on each loss of VDDBU.

Bit	31	30	29	28	27	26	25	24
	GPBR_VALUE[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	GPBR_VALUE[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	GPBR_VALUE[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	GPBR_VALUE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – GPBR_VALUE[31:0] Value of SYS_GPBRx

Note: If a Tamper event has been detected, it is not possible to write GPBR_VALUE as long as the LPDBCS0 or LPDBCS4 flag has not been cleared in the Supply Controller Status register (SUPC_SR).

© 2026 Microchip Technology Inc. and its subsidiaries

25. Special Function Registers (SFR)

25.1. Description

Special Function Registers (SFR) manage specific aspects of the integrated memory, bridge implementations, processor and other functionality not controlled elsewhere.

25.2. Embedded Characteristics

- 32-bit Special Function Registers Control Specific Behavior of the Product

25.3. Functional Description

25.3.1. Register Write Protection

To prevent any single software error from corrupting SFR behavior, certain registers in the address space can be write-protected by setting the WPEN bit in the SFR Write Protection Mode Register (SFR_WPMR).

If a write access to a write-protected register is detected, the WPVS flag in the SFR Write Protection Status Register (SFR_WPSR) is set and the field WPSRC indicates the register in which the write access has been attempted.

The following registers can be write-protected by setting SFR_WPMR.WPEN:

- Optical Link Register
- ERASE Flash Register
- PWM Debug Register

© 2026 Microchip Technology Inc. and its subsidiaries

25.4. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00 ... 0x6F	Reserved									
0x70	SFR_FLASH	31:24								
		23:16								
		15:8								
		7:0								PATCH_BYPASS
0x74 ... 0x7F	Reserved									
0x80	SFR_OPT_LINK	31:24								
		23:16								
		15:8								
		7:0								CLK_SELECT
0x84 ... 0xAF	Reserved									
0xB0	SFR_ERASE_FLASH_SRAM	31:24								
		23:16								
		15:8								
		7:0							SRAM0	HW_ERASE
0xB4	SFR_PWM_DEBUG	31:24								
		23:16								
		15:8								
		7:0								CORE
0xB8 ... 0xE3	Reserved									
0xE4	SFR_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0								WPEN
0xE8	SFR_WPSR	31:24								
		23:16	WPSRC[15:8]							
		15:8	WPSRC[7:0]							
		7:0								WPVS

© 2026 Microchip Technology Inc. and its subsidiaries

25.4.1. Flash Memory Configuration Register

Name: SFR_FLASH
Offset: 0x70
Reset: 0x00000001
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								PATCH_BYPASS
Access								R/W
Reset								1

Bit 0 – PATCH_BYPASS Patch Bypass

Value	Description
0	Enables Flash high-speed patch.
1	Disables Flash high-speed patch.

© 2026 Microchip Technology Inc. and its subsidiaries

25.4.2. Optical Link Register

Name: SFR_OPT_LINK
Offset: 0x80
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								CLK_SELECT
Access								R/W
Reset								0

Bit 0 – CLK_SELECT Clock Selection

Value	Description
0	PLLA Clock1 output is selected.
1	PMC PCK3 is selected.

© 2026 Microchip Technology Inc. and its subsidiaries

25.4.3. Erase Flash/SRAM Register

Name: SFR_ERASE_FLASH_SRAM
Offset: 0xB0
Reset: 0x00000001
Property: Read/Write-Once

This is a write-once register and is only unlocked by a reset.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							SRAM0	HW_ERASE
Access							R/WO	R/WO
Reset							0	1

Bit 1 – SRAM0 Erase SRAM0 Content

Value	Name	Description
0	NOT_DELETED	If HW_ERASE = 1, and HW erase signal assertion occurs, SRAM0 content is not deleted.
1	DELETED	If HW_ERASE = 1, and HW erase signal assertion occurs, SRAM0 content is deleted.

Bit 0 – HW_ERASE PB2/Peripherals or Hardware Erase Signal Assignment

Value	Name	Description
0	DISABLE	Hardware erase signal disabled. PB2 pin can be used in GPIO or Peripheral IO mode.
1	ENABLE	Hardware erase signal enabled. PB2 pin is assigned to Flash erase function.

© 2026 Microchip Technology Inc. and its subsidiaries

25.4.4. PWM Debug Register

Name: SFR_PWM_DEBUG
Offset: 0xB4
Reset: 0x00000003
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								CORE
Access								R/W
Reset								1

Bit 0 – CORE Debug Information Propagation Mode

Value	Name	Description
0	NOT_SENT	Core does not send the debug signal to the PWM.
1	SENT	Core sends the debug signal to PWM. Refer to “Debug Mode” in the section “Pulse Width Modulation Controller (PWM)” for details on configuring an action when the debug signal is active.

© 2026 Microchip Technology Inc. and its subsidiaries

25.4.5. SFR Write Protection Mode Register

Name: SFR_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
								WPEN
Access								R/W
Reset								0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x534652	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0.

Bit 0 – WPEN Write Protection Enable

Value	Description
0	Disables the write protection if WPKEY value corresponds to 0x534652 (“SFR” in ASCII).
1	Enables the write protection if WPKEY value corresponds to 0x534652 (“SFR” in ASCII).

© 2026 Microchip Technology Inc. and its subsidiaries

25.4.6. SFR Write Protection Status Register

Name: SFR_WPSR
Offset: 0xE8
Reset: 0x00000000
Property: Read/Write

Refer to Register Write Protection for the list of write-protected registers.

This register is fully cleared when a write is done.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	WPSRC[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPSRC[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPVS
Access								R/W
Reset								0

Bits 23:8 – WPSRC[15:0] Write Protection Source
When WPVS = 1, WPSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protection Violation Status

Value	Description
0	No write protection violation has occurred since the last read of SFR_WPSR.
1	A write protection violation has occurred since the last read of SFR_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPSRC.

© 2026 Microchip Technology Inc. and its subsidiaries

26. Special Function Registers Backup (SFRBU)

26.1. Description

Special Function Register Backup (SFRBU) manage specific aspects of the integrated memory, bridge implementations, processor and other functionality not controlled elsewhere.

26.2. Embedded Characteristics

- 32-bit Special Function Registers Control Specific Behavior of the Product

26.3. Functional Description

26.3.1. Register Write Protection

To prevent any single software error from corrupting SFRBU behavior, certain registers in the address space can be write-protected by setting the WPEN bit in the SFRBU Write Protection Mode Register (SFRBU_WPMR).

If a write access to a write-protected register is detected, the WPVS flag in the SFRBU Write Protection Status Register (SFRBU_WPSR) is set and the field WPSRC indicates the register in which the write access has been attempted.

The following registers can be write-protected by setting SFRBU_WPMR.WPEN:

- IO Retention Register

© 2026 Microchip Technology Inc. and its subsidiaries

26.4. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00 ... 0x2F	Reserved									
0x30	SFRBU_IO_RETENTION	31:24								
		23:16								NRST
		15:8	PD_30_24	PD_23_16	PD_15_8	PD_7_0		PC_22_16	PC_15_8	PC_7_0
		7:0	PB_26_24	PB_23_16	PB_15_8	PB_7_0	PA_31_24	PA_23_16	PA_15_8	PA_7_0
0x34 ... 0x3F	Reserved									
0x40	SFRBU_BODCORE	31:24								
		23:16								
		15:8								
		7:0								STATUS
0x44 ... 0xE3	Reserved									
0xE4	SFRBU_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0								WPEN
0xE8	SFRBU_WPSR	31:24								
		23:16	WPSRC[15:8]							
		15:8	WPSRC[7:0]							
		7:0								WPVS

© 2026 Microchip Technology Inc. and its subsidiaries

26.4.1. SFRBU IO Retention Register

Name: SFRBU_IO_RETENTION
Offset: 0x30
Reset: 0x0001F7FF
Property: Read/Write

The following configuration values are valid for all listed bit names of this register:
0: Keep function (IO_retention) activated in pads.

1: No Keep function (IO_retention) activated in pads.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								NRST
Access								R/W
Reset								1

Bit	15	14	13	12	11	10	9	8
	PD_30_24	PD_23_16	PD_15_8	PD_7_0		PC_22_16	PC_15_8	PC_7_0
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	1	1	1	1		1	1	1

Bit	7	6	5	4	3	2	1	0
	PB_26_24	PB_23_16	PB_15_8	PB_7_0	PA_31_24	PA_23_16	PA_15_8	PA_7_0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit 16 – NRST Keep Function on NRST Pad

Bit 15 – PD_30_24 Keep Function on PD24 to PD30

Bit 14 – PD_23_16 Keep Function on PD16 to PD23

Bit 13 – PD_15_8 Keep Function on PD8 to PD15

Bit 12 – PD_7_0 Keep Function on PD0 to PD7

Bit 10 – PC_22_16 Keep Function on PC16 to PC22

Bit 9 – PC_15_8 Keep Function on PC8 to PC15

Bit 8 – PC_7_0 Keep Function on PC0 to PC7

Bit 7 – PB_26_24 Keep Function on PB24 to PB26

Bit 6 – PB_23_16 Keep Function on PB16 to PB23

Bit 5 – PB_15_8 Keep Function on PB8 to PB15

Bit 4 – PB_7_0 Keep Function on PB0 to PB7

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 3 – PA_31_24 Keep Function on PA24 to PA31

Bit 2 – PA_23_16 Keep Function on PA16 to PA23

Bit 1 – PA_15_8 Keep Function on PA8 to PA15

Bit 0 – PA_7_0 Keep Function on PA0 to PA7

© 2026 Microchip Technology Inc. and its subsidiaries

26.4.2. SFRBU BOD Core Register

Name: SFRBU_BODCORE
Offset: 0x40
Reset: 0x00000000
Property: Read/Write

Write in this register to clear this flag.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								STATUS
Access								R/W
Reset								0

Bit 0 – STATUS Core Brownout Detector

Value	Description
0	Indicates there is no BODCORE event from the Power-on Reset or from the last time this register has been cleared.
1	Indicates there is at least one BODCORE event from the Power-on Reset or from the last time this register has been cleared.

© 2026 Microchip Technology Inc. and its subsidiaries

26.4.3. SFRBU Write Protection Mode Register

Name: SFRBU_WPMR
Offset: 0xE4
Reset: 0x00000001
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
								WPEN
Access								R/W
Reset								1

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x534652	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0.

Bit 0 – WPEN Write Protection Enable

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x534652 (“SFR” in ASCII).
1	Enables the write protection if WPKEY value corresponds to 0x534652 (“SFR” in ASCII).

© 2026 Microchip Technology Inc. and its subsidiaries

26.4.4. SFRBU Write Protection Status Register

Name: SFRBU_WPSR
Offset: 0xE8
Reset: 0x00000000
Property: Read/Write

Refer to Register Write Protection for the list of write-protected registers.

This register is fully cleared when a write is done.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	WPSRC[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPSRC[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPVS
Access								R/W
Reset								0

Bits 23:8 – WPSRC[15:0] Write Protection Source
When WPVS = 1, WPSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protection Violation Status

Value	Description
0	No write protection violation has occurred since the last read of SFRBU_WPSR.
1	A write protection violation has occurred since the last read of SFRBU_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPSRC.

© 2026 Microchip Technology Inc. and its subsidiaries

27. Bus Matrix (MATRIX)

27.1. Description

The Bus Matrix (MATRIX) implements a multi-layer AHB, based on the AHB-Lite protocol, that enables parallel access paths between multiple hosts and clients in a system, thus increasing the overall bandwidth.

The MATRIX interconnects hosts to clients. The normal latency to connect a host to a client is one cycle, except for the default host of the accessed client which is connected directly (zero cycle latency).

The MATRIX user interface is compliant with the Arm Advanced Peripheral Bus.

27.1.1. Matrix 0 and 1 Hosts and Clients

Table 27-1. MATRIX0 Hosts

Host No.	Host Name	Description
0	CM4-C0 ICode Bus	–
1	CM4-C0 DCode Bus	–
2	CM4-C0 S-Bus	–
3	CMCC0 (ICache/ITCM)	–
4	CMCC1 (DCache/DTCM)	–
5	MATRIX2	PDC1
6	MATRIX1	PDC0, PDC2 and ICM

Table 27-2. MATRIX0 Clients

Client No.	Client Name	Description
0	SRAM0, Port 0	–
1	SRAM0, Port 1	–
2	SRAM0, Port 2	–
3	ROM	–
4	MATRIX1	Provides access to: • QSPI • QSPI through AESB • APB peripherals accessible through Bridges 0 and 2
5	CPKCC	–
6	APB Bridge 3	–
7	MATRIX2	Provides access to: • APB peripherals accessible through Bridges 1 and 4
8	CMCC0 (ICache/ITCM)	–
9	CMCC1 (DCache/DTCM)	–
10	Flash	–
11	MATRIX1	Provides access to: • QSPI • QSPI through AESB

Note: For consistency, each SRAM port shall have the same Privilege Protection access management configuration.

© 2026 Microchip Technology Inc. and its subsidiaries

Table 27-3. MATRIX0 Host to Client Connections

		MATRIX0 Hosts						
		0	1	2	3	4	5	6
	MATRIX0 Clients	CM4-C0 ICode Bus	CM4-C0 DCode Bus	CM4-C0 S-Bus	CMCC0 (ICache/ITCM)	CMCC1 (DCache/DTCM	MATRIX2	MATRIX1
0	SRAM0, Port 0	–	–	X	–	–	–	–
1	SRAM0, Port 1	–	–	–	–	–	–	X
2	SRAM0, Port 2	–	–	–	–	–	X	–
3	ROM	X	X	–	–	–	–	–
4	MATRIX1	X	X	X	–	–	X	–
5	CPKCC	X	X	–	–	–	–	–
6	APB Bridge 3	–	–	X	–	–	–	–
7	MATRIX2	–	–	X	–	–	–	–
8	CMCC0 (ICache/ITCM)	X	X(3)	–	–	–	–	–
9	CMCC1 (DCache/DTCM)	–	X	–	–	–	–	–
10	Flash(4)	X	X	X(6)	X	X	–	X
11	MATRIX1	–	–	–	X	X	–	–

Notes:

1. For consistency, each SRAM port shall have the same Privilege Protection access management configuration.
2. Connection needed to enable write of ITCM for CM4-C0.
3. Client Flash must be configured in NO_DEFAULT_MASTER.
4. Used to access the Flash Controller Page Buffer (Write Operation) in a strongly ordered manner.

Table 27-4. MATRIX1 Hosts

Host No.	Host Name	Description
0	MATRIX0	• CM4-C0-ICode and CM4-C0-DCode to QSPI and AESB • CM4-C0-S-Bus to APB Bridges (0 and 2)
1	MATRIX0	CMCC0 (I), CMCC1 (D) to QSPI and AESB
2	AESB	QSPI Cached or Non cached through AESB
3	ICM	Flash Non cached QSPI Non cached SRAM0 port1
4	PDC0	QSPI Non cached AESB Non cached - through AESB SRAM0 port1
5	PDC2	Flash Non cached QSPI Non cached SRAM0 port1

Table 27-5. MATRIX1 Clients

Client No.	Client Name	Description
0	QSPI	–
1	AESB	–

© 2026 Microchip Technology Inc. and its subsidiaries

Table 27-5. MATRIX1 Clients (continued)

Client No.	Client Name	Description
2	MATRIX0	Provides access to: • SRAM0 port 1 • Flash for ICM
3	APB Bridge 0	–
4	APB Bridge 2	–

Table 27-6. MATRIX1 Host to Client Connections

MATRIX1 Clients		MATRIX1 Hosts					
		0	1	2	3	4	5
		MATRIX0	MATRIX0	AESB	ICM	PDC0	PDC2
0	QSPI	X	X	X	X	X	X
1	AESB	X	X	–	–	X	–
2	MATRIX0	–	–	–	X	X	X
3	APB Bridge 0	X	–	–	–	X	–
4	APB Bridge 2	X	–	–	–	–	X

27.1.2. Matrix 2 and 3 Hosts and Clients

Table 27-7. MATRIX2 Hosts

Host No.	Host Name	Description
0	MATRIX0	CM4-C0-S-Bus, PDC0, PDC2, ICM
1	MATRIX3	PDC1

Table 27-8. MATRIX2 Clients

Client No.	Client Name	Description
0	MATRIX0	Provides access through MATRIX0 to: • SRAM0 (for PDC1 only)
1	MATRIX3	Provides access through MATRIX3 to APB Bridge 1
2	APB Bridge 4	–

Note: For consistency, each SRAM port shall have the same Privilege Protection access management configuration.

Table 27-9. MATRIX2 Host to Client Connections

MATRIX2 Clients		MATRIX2 Hosts	
		0	1
		MATRIX0	MATRIX3
0	MATRIX2	–	X
1	MATRIX3	X	–
2	APB Bridge 4	X	–

Table 27-10. MATRIX3 Hosts

Host No.	Host Name	Description
0	MATRIX2	• CM4-C0 S-Bus to APB Bridge 1
1	PDC1	–

© 2026 Microchip Technology Inc. and its subsidiaries

Table 27-11. MATRIX3 Clients

Client No.	Client Name	Description
0	MATRIX2	–
1	APB Bridge 1	–

Table 27-12. MATRIX3 Host to Client Connections

MATRIX3 Clients		MATRIX3 Hosts	
		0	1
		MATRIX2	PDC1
0	MATRIX2 (1)	–	X
1	APB Bridge 1	X	X

Note:

1. Client port provides access to SRAM0 port 2 for PDC1.

27.2. Embedded Characteristics

- 32- or 64-bit Data Bus
- Configurable Number of Hosts (Up to Sixteen)
- Configurable Number of Clients (Up to Sixteen)
- One Decoder for Each Host
- Several Possible Boot Memories for Each Host before Remap
- One Remap Function for Each Host
- Support for Long Bursts of Length 32, 64, 128 and Up to the Limit of 256-bit Burst Beats of Words
- Enhanced Programmable Mixed Arbitration for Each Client
  - Round-robin
  - Fixed priority
  - Latency Quality of Service (QoS)
- Programmable Default Host for Each Client
  - No default host
  - Last accessed default host
  - Fixed default host
- Deterministic Maximum Access Latency for Hosts
- Zero or One Cycle Arbitration Latency for the First Access of a Burst
- Bus Lock Forwarding to Clients
- Host Number Forwarding to Clients
- Register Write Protection of User Interface Registers
- User/Privilege Protection

27.3. Memory Mapping

The MATRIX provides one decoder for every host interface. The decoder offers each host several memory mappings. Each memory area can be assigned to several clients. Booting at the same address while using different clients (i.e., external RAM, internal ROM or internal Flash, etc.) becomes possible.

27.4. Special Bus Granting Techniques

The MATRIX provides some speculative bus granting techniques in order to anticipate access requests from hosts. Hence, latency is reduced at first access of a burst or, for a single transfer,

© 2026 Microchip Technology Inc. and its subsidiaries

as long as the client is free from any other host access. It does not provide any benefit if the client is continuously accessed by more than one host, since arbitration is pipelined and has no negative effect on the client bandwidth or access latency.

This bus granting technique sets a different default host for every client.

At the end of the current access, if no other request is pending, the client remains connected to its associated default host. A client can be associated with three kinds of default hosts:

- no default host
- last access host
- fixed default host

To change from one type of default host to another, the user interface provides Client Configuration registers (MATRIX_SCFGx), one for every client, which set a default host for each client. MATRIX_SCFGx contain two fields to manage host selection: DEFMSTR_TYPE and FIXED_DEFMSTR. The 2-bit DEFMSTR_TYPE field selects the default host type (no default, last access host, fixed default host), whereas the 4-bit FIXED_DEFMSTR field selects a fixed default host provided that DEFMSTR_TYPE is set to fixed default host. See MATRIX_SCFGx.

27.5. No Default Host

After the end of the current access, if no other request is pending, the client is disconnected from all hosts.

This configuration incurs one latency clock cycle for the first access of a burst after bus Idle. Arbitration without default host may be used for hosts that perform significant bursts or several transfers with no Idle cycle in-between, or if the client bus bandwidth is widely used by one or more hosts.

This configuration provides no benefit on access latency or bandwidth when reaching maximum client bus throughput whatever the number of requesting hosts.

27.6. Last Access Host

After the end of the current access, if no other request is pending, the client remains connected to the last host that performed an access request.

This enables the MATRIX to remove the one latency cycle for the last host that accessed the client. Other non-privileged hosts still get one latency clock cycle if they need to access the same client. This technique is useful for hosts that mainly perform single accesses or short bursts with some Idle cycles in-between.

This configuration provides no benefit on access latency or bandwidth when reaching maximum client bus throughput whatever is the number of requesting hosts.

27.7. Fixed Default Host

After the end of the current access, if no other request is pending, the client connects to its fixed default host. Unlike the last access host, the fixed default host does not change unless the user modifies it by software (FIXED_DEFMSTR field of the related MATRIX_SCFG).

This allows the MATRIX arbiters to remove the one latency clock cycle for the fixed default host of the client. All requests attempted by the fixed default host do not cause any arbitration latency, whereas other non-privileged hosts get one latency cycle. This technique is useful for a host that mainly performs single accesses or short bursts with Idle cycles in-between.

This configuration provides no benefit on access latency or bandwidth when reaching maximum client bus throughput, regardless of the number of requesting hosts.

© 2026 Microchip Technology Inc. and its subsidiaries

27.8. Arbitration

The MATRIX provides an arbitration technique that reduces latency when conflicts occur, i.e., when two or more hosts try to access the same client at the same time. One arbiter per client is provided, thus arbitrating each client specifically.

The user can either choose one of the following arbitration types, or mix them for each client:

1. Round-robin Arbitration (default)
2. Fixed Priority Arbitration

The resulting algorithm may be complemented by selecting a default host configuration for each client.

When re-arbitration must be done, specific conditions apply. See Arbitration Scheduling.

27.8.1. Arbitration Scheduling

Each arbiter has the ability to arbitrate between two or more different host requests. In order to avoid burst breaking as well as to provide the maximum throughput for client interfaces, arbitration may only take place during the following cycles:

- Idle Cycles: When a client is not connected to any host or is connected to a host which is not currently accessing it.
- Single Cycles: When a client is currently doing a single access.
- End of Burst Cycles: When the current cycle is the last cycle of a burst transfer. For defined length burst, predicted end of burst matches the size of the transfer but is managed differently for undefined length burst. See Undefined Length Burst Arbitration.
- Slot Cycle Limit: When the slot cycle counter has reached the limit value indicating that the current host access is too long and must be broken. See Slot Cycle Limit Arbitration.

27.8.1.1. Undefined Length Burst Arbitration

In order to prevent long burst lengths that can lock the access to the client for an excessive period of time, the user can trigger the re-arbitration before the end of the incremental bursts. The re-arbitration period can be selected from the following Undefined Length Burst Type (ULBT) possibilities:

- Unlimited: no predetermined end of burst is generated. This value enables 1 Kbyte burst lengths.
- 1-beat bursts: predetermined end of burst is generated at each single transfer during the INCR transfer.
- 4-beat bursts: predetermined end of burst is generated at the end of each 4-beat boundary during INCR transfer.
- 8-beat bursts: predetermined end of burst is generated at the end of each 8-beat boundary during INCR transfer.
- 16-beat bursts: predetermined end of burst is generated at the end of each 16-beat boundary during INCR transfer.
- 32-beat bursts: predetermined end of burst is generated at the end of each 32-beat boundary during INCR transfer.
- 64-beat bursts: predetermined end of burst is generated at the end of each 64-beat boundary during INCR transfer.
- 128-beat bursts: predetermined end of burst is generated at the end of each 128-beat boundary during INCR transfer.

Undefined-length bursts lower than 8 beats should not be used since this may decrease the overall bus bandwidth due to arbitration and client latencies at each first access of a burst.

© 2026 Microchip Technology Inc. and its subsidiaries

However, if the length of undefined-length bursts is known for a host, it is recommended to configure MATRIX_MCFG.ULBT accordingly.

27.8.1.2. Slot Cycle Limit Arbitration

The MATRIX contains specific logic to break long accesses, such as very long bursts on a very slow client (e.g., an external low speed memory). At each arbitration time, a counter is loaded with the value previously written in MATRIX_SCFGx.SLOT_CYCLE and decreased at each clock cycle. When the counter elapses, the arbiter has the ability to re-arbitrate at the end of the current system bus access cycle.

Unless a host has a very tight access latency constraint, which could lead to data overflow or underflow due to a badly undersized internal FIFO with respect to its throughput, the Slot Cycle Limit should be disabled (SLOT_CYCLE = 0) or set to its default maximum value in order not to inefficiently break long bursts performed by some hosts.

In most cases, this feature is not needed and should be disabled for power saving.

WARNING This feature cannot prevent any client from locking its access indefinitely.

27.8.2. Arbitration Priority Scheme

The MATRIX arbitration scheme is organized in priority pools, each corresponding to an access criticality class as shown in the “Latency Quality of Service” column in the table below. When the Latency Quality of Service is enabled for a host-client pair through the MATRIX, the priority pool number to use for arbitration at the client port is determined from the host. When the Latency Quality of Service is disabled, it is determined through the MATRIX user interface. See MATRIX_PRASx.

After reset, the Latency Quality of Service is enabled by default on all of the host ports that are connected to a host driving the Latency Quality of Service signals, as shown in the bit LQOSEN of MATRIX_PRASx and MATRIX_PRBSx.

Table 27-13. Arbitration Priority Pools

Priority Pool	Latency Quality of Service
3	Latency Critical
2	Latency Sensitive
1	Bandwidth Sensitive
0	Background Transfers

Round-robin priority is used in the highest and lowest priority pools 3 and 0, whereas fixed level priority is used between priority pools and in the intermediate priority pools 2 and 1.

For each client, each host is assigned to one of the client priority pools based on the Latency Quality of Service inputs or to the priority registers for clients (MxPR fields of MATRIX_PRAS and MATRIX_PRBS). When evaluating host requests, this priority pool level always takes precedence.

After reset, most of the hosts belong to the lowest priority pool (MxPR = 0, Background Transfer) and are therefore granted bus access in a true round-robin order.

The highest priority pool must be specifically reserved for hosts requiring very low access latency. If more than one host belongs to this pool, those hosts are granted bus access in a biased round-robin manner which enables tight and deterministic maximum access latency from system bus requests. In the worst case, any currently occurring high-priority host request is granted after the current bus host access has ended and any other high priority pool host requests have been granted once each.

The lowest priority pool shares the remaining bus bandwidth between hosts.

© 2026 Microchip Technology Inc. and its subsidiaries

Intermediate priority pools enable fine priority tuning. Typically, a latency-sensitive host or a bandwidth-sensitive host use such a priority level. The higher the priority level (MxPR value), the higher the host priority.

For good CPU performance, it is recommended to let the CPU priority configured with the default reset value 2 (Latency Sensitive).

All combinations of MxPR values are allowed for all hosts and clients. For example, some hosts might be assigned the highest priority pool (round-robin), and remaining hosts the lowest priority pool (round-robin), with no host for intermediate fixed priority levels.

27.8.2.1. Fixed Priority Arbitration

The fixed priority arbitration algorithm is the first and only arbitration algorithm applied between hosts from distinct priority pools. It is also used in priority pools other than the highest and lowest priority pools (intermediate priority pools).

Fixed priority arbitration enables the MATRIX arbiters to dispatch the requests from different hosts to the same client by using the fixed priority defined by the user in the MxPR field for each host in the registers, MATRIX_PRAS and MATRIX_PRBS. If two or more host requests are active at the same time, the host with the highest priority MxPR number is serviced first.

In intermediate priority pools, if two or more host requests with the same priority are active at the same time, the host with the highest number is serviced first.

27.8.2.2. Round-Robin Arbitration

This algorithm is only used in the highest and lowest priority pools. It allows the MATRIX arbiters to properly dispatch requests from different hosts to the same client. If two or more host requests are active at the same time in the priority pool, they are serviced in a round-robin increasing host number order.

27.9. Register Write Protection

To prevent any single software error from corrupting MATRIX behavior, certain registers in the address space can be write-protected by setting the WPEN bit in the Write Protection Mode register (MATRIX_WPMR).

If a write access to a write-protected register is detected, the WPVS flag in the Write Protection Status register (MATRIX_WPSR) is set and the field WPVSRC indicates the register in which the write access has been attempted.

The WPVS flag is reset by writing the MATRIX_WPMR with the appropriate access key WPKEY.

The registers listed below can be write-protected.

© 2026 Microchip Technology Inc. and its subsidiaries

Related Links

MATRIX_MCFGx
MATRIX Host Configuration Register x

MATRIX_SCFGx
MATRIX Client Configuration Register x

MATRIX_PRASx
MATRIX Priority Register A For Clients x

MATRIX_PRBSx
MATRIX Priority Register B For Clients x

MATRIX_MRCR
MATRIX Host Remap Control Register

MATRIX_MEIER
MATRIX Host Error Interrupt Enable Register

MATRIX_MEIDR
MATRIX Host Error Interrupt Disable Register

MATRIX_PSRx
MATRIX Protection Client Register x

MATRIX_PASSRx
MATRIX Protected Areas Split Client Register x

MATRIX_PRTSRx
MATRIX Protected Region Top Client Register x

MATRIX_PPSELRx
MATRIX Protected Peripheral Select x Registers [x=1..3]

27.10. Privilege Protection of AHB and APB

User/Privilege protection is supported through the filtering of each client access with the host User/Privilege protection bit.

The Protection Unit adds the ability to manage the access rights for Privilege and User accesses. The access rights are defined through the hardware and software configuration of the device. The operating mode is the following:

- The Bus Hosts transmit requests with the Privilege or User access right.
- The MATRIX, according to its configuration and the request, grants or denies the access.

The client address space is divided into one or more client regions. The client regions are generally contiguous parts of the client address space. The client region is potentially split into an access denied area (upper part) and a protected region which can be split (lower part), unless the client-protected region occupies the whole client region. The protected region itself may or may not be split into one Privilege-eligible area and one User area. The Privilege-eligible area may be independently Privileged for read access and for write access.

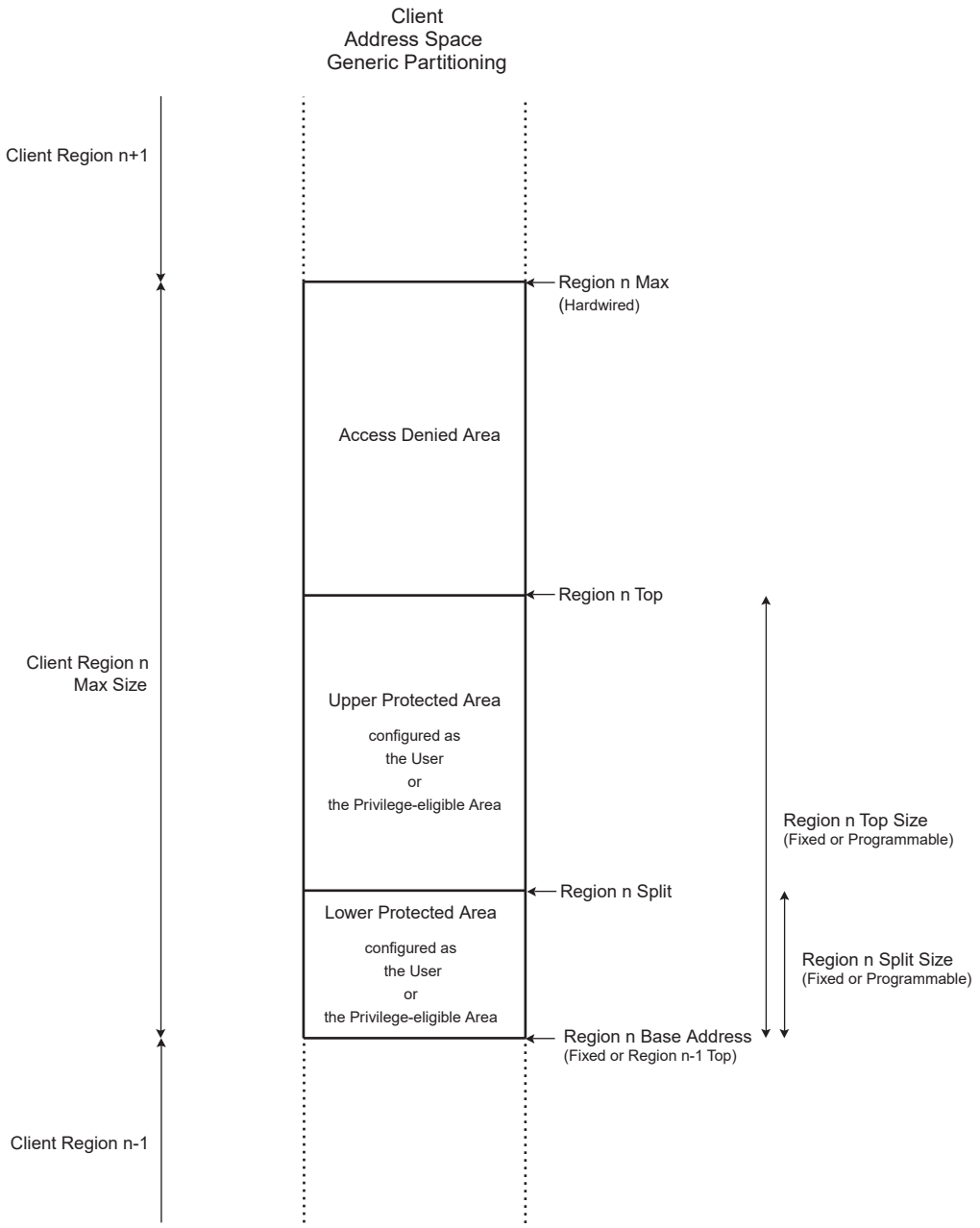
For one client region, the following characteristics are configured by hardware or software:

- Base Address of the client region
- The Max Size of the client region—a maximum size for the region's physical content
- The Top Size of the client-protected region—the actually programmed or fixed size for the region's physical content
- The Split Size of the client-protected region—the size of one of the two protected areas of the region
- Whether Split Size is defined downward from the top of the protected region, or upward from its base address

© 2026 Microchip Technology Inc. and its subsidiaries

The following figure shows how the terms defined here are implemented in a Client address space.

Figure 27-1. Generic Partitioning of the Client Address Space



A set of MATRIX protection registers are used to specify, for each client, client-protected region or client-protected area, the protection mode required for accessing this client, client-protected region or client-protected area. See MATRIX_PSRx, MATRIX_PASSRx and MATRIX_PRTSRx.

Additional MATRIX protection registers are used to specify, for each peripheral bus client, the protection mode required for accessing this client (see MATRIX_PPSELRx).

The MATRIX registers can only be accessed in Privileged mode.

© 2026 Microchip Technology Inc. and its subsidiaries

The MATRIX propagates the protection bit down to the clients to let them perform additional protection checks, and the MATRIX itself allows or not the access to the clients via its embedded protection unit.

Access violations may be reported either by a client through the bus error response (example from the system-to-peripheral bus (AHB/APB Bridge)), or by the MATRIX embedded protection unit. In both cases, a bus error response is sent to the offending host and the error is flagged in the Host Error Status Register. An interrupt can be sent if it has been enabled for that host by writing into the Host Error Interrupt Enable Register. Thus, the offending host is identified. The offending address is registered in the Host Error Address Registers, so that the client and the targeted protected region are also known.

Depending on the hardware parameters and software configuration, the address space of each client-protected region may or may not be split into two parts: one User and one Privileged.

Five different protection types of clients are supported. The number of protected regions is set by design for each client, independently, from 1 to 8, totaling from 1 up to 16 protected areas for each protection-configurable client.

27.10.1. Protection Types of System Bus Clients

27.10.1.1. Principles

The MATRIX supports five different protection types of clients: two fixed types and three configurable types. The protection type of a client is set at hardware design among the following:

- Always User
- Always Privileged
- Internal Protected
- External Protected
- Scalable Protected

The protection type is set at hardware design on a per-host and a per-client basis. Always User and Always Privilege protection types are not software-configurable.

The different protection types have the following characteristics:

- Always User clients have no access right restriction. Their address space is precisely set by design. Any out-of-address range access is denied and reported.
- Always Privilege clients can only be accessed by a Privileged host request. Their address space is precisely set by design. Any User access or out-of-address range access is denied and reported.
- Internal Protected type is intended for internal memories such as RAM, ROM or embedded Flash. The Internal Protected client has one or several client regions, and each has a hardware-fixed base address and Protected Region Top. Each client region may be split through software configuration into one User area plus one Privilege-eligible area. Inside each client-protected region, the split boundary is programmable in powers of 2 from 4 Kbytes up to the full client-protected region address space. The protected area located below the split boundary may be configured as the User or the Privilege-eligible one. The Privilege-eligible area may be independently configured as Read User and/or Write Privileged. Any access with access right or address range violation is denied and reported.
- External Protected type is intended for external memories on the EBI, such as DDR, SDRAM, external ROM or NAND Flash. The External Protected client has identical features as the Internal Protected client, plus the ability to configure each of its client-protected region address space sizes according to the external memory parts used. This avoids mirroring Privileged areas into User areas, and further restricts the overall accessible address range. Any access with access right violation or configured address range violation is denied and reported.

© 2026 Microchip Technology Inc. and its subsidiaries

- Scalable Protected type is intended for external memories with a dedicated client, such as DDR. The Scalable Protected client is divided into a fixed number of scalable, equally sized, and contiguous protected regions. Each of them can be split in the same way as for Internal or External Protected clients. The protected region size must be configured by software, so that the equally-sized regions fill the actual available memory. This avoids mirroring Privileged areas into User areas, and further restricts the overall accessible address range. Any access with access right violation or configured address range violation is denied and reported.

As the protection type is set at hardware design on a per-host and per-client basis, it is possible to set some client access protection as configurable from one or some particular hosts, and to set the access as Always Privileged from all the other hosts.

As the protection type is set by design at the client region level, different protected region types can be mixed inside a single client.

Likewise, the mapping base address and the accessible address range of each client or client region may have been hardware-restricted on a per-host basis from no access to full client address space.

27.10.1.1.1. Client Protection Types by Matrix

Table 27-14. MATRIX0 Client Protection Types

Client	Peripheral	#hsel (8)	Mode	Max Size
0	SRAM0, port 0	0 @ 0x20000000 1 @ 0x20020000 2 @ 0x20040000 3 @ 0x20060000	Internal Protected	128 Kbytes 128 Kbytes 128 Kbytes 128 Kbytes
1	SRAM0, port 1	0 @ 0x20000000 1 @ 0x20020000 2 @ 0x20040000 3 @ 0x20060000	Internal Protected	128 Kbytes 128 Kbytes 128 Kbytes 128 Kbytes
2	SRAM0, port 2	0 @ 0x20000000 1 @ 0x20020000 2 @ 0x20040000 3 @ 0x20060000	Internal Protected	128 Kbytes 128 Kbytes 128 Kbytes 128 Kbytes
3	ROM	0 @ 0x00000000	Always User	32 Kbytes
4	HS2LSASB (bridge to MATRIX1 host port 0)	0 @ 0x04000000 0 @ 0x06000000 0 @ 0x40000000 0 @ 0x44000000	Always User (1)	32 Mbytes (QSPI Non-Cached) 32 Mbytes (AESB Non-Cached) 16 Mbytes (APB Bridge 0) 16 Mbytes (APB Bridge 2)
5	CPKCC	0 @ 0x02020000 0 @ 0x02031000	Internal Protected	64 Kbytes (CPKCC ROM) 4 Kbytes (CPKCC RAM)
6	APB Bridge 3	0 @ 0x46000000	Always User(2)	16 Mbytes
7	HSAS2HSMSB (bridge to MATRIX2 host port 0)	0 @ 0x48000000 0 @ 0x4A000000 0 @ 0x20080000 0 @ 0x20088000	Always User(3)	16 Mbytes (APB Bridge 1) 16 Mbytes (APB Bridge 4)
8	CMCCI/ITCM	0 @ 0x11000000 0 @ 0x14000000 0 @ 0x16000000 1 @ 0x1FFFC000	Always User(4)	16 Mbytes (Flash Cached) 32 Mbytes (QSPI Cached) 32 Mbytes (AESB Cached) 16 Kbytes (ITCM)

© 2026 Microchip Technology Inc. and its subsidiaries

Table 27-14. MATRIX0 Client Protection Types (continued)

Client	Peripheral	#hsel (8)	Mode	Max Size
9	CMCCD/DTCM	0 @ 0x11000000 0 @ 0x14000000 0 @ 0x16000000 1 @ 0x1FFFA000	Always User[4]	16 Mbytes (Flash Cached) 32 Mbytes (QSPI Cached) 32 Mbytes (AESB Cached) 8 Kbytes (DTCM)
10	Flash	0 @ 0x01000000 1 @ 0x01040000 2 @ 0x01080000 3 @ 0x010C0000 4 @ 0x01100000 5 @ 0x01140000 6 @ 0x01180000 7 @ 0x011C0000 0 @ 0x11000000 1 @ 0x11040000 2 @ 0x11080000 3 @ 0x110C0000 4 @ 0x11100000 5 @ 0x11140000 6 @ 0x11180000 7 @ 0x111C0000 0 @ 0xA1000000 1 @ 0xA1040000 2 @ 0xA1080000 3 @ 0xA10C0000 4 @ 0xA1100000 5 @ 0xA1140000 6 @ 0xA1180000 7 @ 0xA11C0000	Internal Protected	256 Kbytes (Flash Non-Cached) 256 Kbytes (Flash Non-Cached) 256 Kbytes (Flash Non-Cached) 256 Kbytes (Flash Non-Cached) 256 Kbytes (Flash Non-Cached) 256 Kbytes (Flash Non-Cached) 256 Kbytes (Flash Non-Cached) 256 Kbytes (Flash Non-Cached) 256 Kbytes (Flash Cached) 256 Kbytes (Flash Cached) 256 Kbytes (Flash Cached) 256 Kbytes (Flash Cached) 256 Kbytes (Flash Cached) 256 Kbytes (Flash Cached) 256 Kbytes (Flash Cached) 256 Kbytes (Flash Cached) 256 Kbytes (Internal Flash Write) 256 Kbytes (Internal Flash Write) 256 Kbytes (Internal Flash Write) 256 Kbytes (Internal Flash Write) 256 Kbytes (Internal Flash Write) 256 Kbytes (Internal Flash Write) 256 Kbytes (Internal Flash Write) 256 Kbytes (Internal Flash Write)
11	HS2LSASB2 (bridge to MATRIX1 host port 1)	0 @ 0x04000000 0 @ 0x06000000	Always User[1]	32 Mbytes (QSPI Cached) 32 Mbytes (AESB Cached)

Notes:

1. Protection managed in MATRIX1.
2. Protection managed in APB Bridge 3 (through the use of PPSELR registers, see MATRIX PPSELRx Register Descriptions).
3. Protection managed in MATRIX2.
4. Protection managed on final client (in MATRIX0 for Flash, in MATRIX1 for QSPI).

© 2026 Microchip Technology Inc. and its subsidiaries

Table 27-15. MATRIX1 Client Protection Types

Client	Peripheral	#hsel (4)	Mode	Max Size
0	QSPI	0 @ 0x04000000	Scalable Protected	32 Mbytes (QSPI Cached)
		1 @ 0x04000000		32 Mbytes (QSPI Cached)
		2 @ 0x04000000		32 Mbytes (QSPI Cached)
		3 @ 0x04000000		32 Mbytes (QSPI Cached)
		0 @ 0x14000000		32 Mbytes (QSPI Non-Cached)
		1 @ 0x14000000		32 Mbytes (QSPI Non-Cached)
		2 @ 0x14000000		32 Mbytes (QSPI Non-Cached)
		3 @ 0x14000000		32 Mbytes (QSPI Non-Cached)
		0 @ 0x06000000		32 Mbytes (QSPI Cached or Non-Cached through AESB)
		1 @ 0x06000000		32 Mbytes (QSPI Cached or Non-Cached through AESB)
		2 @ 0x06000000		32 Mbytes (QSPI Cached or Non-Cached through AESB)
		3 @ 0x06000000		32 Mbytes (QSPI Cached or Non-Cached through AESB)
1	AESB	0 @ 0x16000000	Always User (1)	32 Mbytes (QSPI Cached through AESB)
		0 @ 0x06000000		32 Mbytes (QSPI Non-Cached through AESB)
2	LS2HSASB (bridge to MATRIX0 host port 6)	0 @ 0x01000000	Always User(2)	16 Mbytes (Flash Non-Cached)
		0 @ 0x20000000		512 Kbytes (SRAM0 port 1)
		0 @ 0x20080000		
		0 @ 0x20088000		
3	APB Bridge 0	0 @ 0x40000000	Always User(3)	16 Mbytes
4	APB Bridge 2	0 @ 0x44000000	Always User(4)	16 Mbytes

Notes:

1. Protection managed on final client QSPI in MATRIX1.
2. Protection managed in MATRIX0.
3. Protection managed in APB Bridge 0 (through the use of PPSELR registers, see MATRIX PPSELRx Register Descriptions).
4. Protection managed in APB Bridge 2 (through the use of PPSELR registers, see MATRIX PPSELRx Register Descriptions).

Table 27-16. MATRIX2 Client Protection Types

Client	Peripheral	#hsel (1)	Mode	Max Size
0	HSMS2HSASB (bridge to MATRIX0 host port 5)	0 @0x20000000	Always User (1)	512 Kbytes (SRAM0, port 2)
		0 @ 0x40000000		16 Mbytes (APB Bridge 0)
		0 @ 0x46800000		8 Mbytes (APB Bridge 3, PMC)
1	HS2LSMSB (bridge to MATRIX3 host port 0)	0 @ 0x48000000	Always User(2)	16 Mbytes (APB Bridge 1)
2	APB Bridge 4	0 @ 0x4A000000	Always User(3)	16 Mbytes

Notes:

1. Protection managed in MATRIX0 for final client SRAM0 and in APB bridge 0 and APB bridge 3 (through the use of PPSELR registers, see MATRIX PPSELRx Register Descriptions).
2. Protection managed in APB bridge 1 (through the use of PPSELR registers, see MATRIX PPSELRx Register Descriptions).
3. Protection managed in APB Bridge 4 (through the use of PPSELR registers, see MATRIX PPSELRx Register Descriptions).

© 2026 Microchip Technology Inc. and its subsidiaries

Table 27-17. MATRIX3 Client Protection Types

Client	Peripheral	#hsel (1)	Mode	Max Size
0	LS2HSMSB (bridge to MATRIX2 host port 1)	0 @ 0x20000000 0 @ 0x20080000 0 @ 0x20088000	Always User	512 Kbytes (SRAM0, port 2)
1	APB Bridge 1	0 @ 0x48000000	Always User	16 Mbytes

Notes:

1. Protection managed in MATRIX0 for final client SRAM0.
2. Protection managed in APB bridge 1 (through the use of PPSELR registers, see MATRIX PPSELRx Register Descriptions).

27.10.2. Protection of APB Clients

To be able to configure the protection mode required for accessing a particular peripheral bus client connected behind the system-to-peripheral bus (AHB/APB Bridge), the MATRIX features three 32-bit Protected Peripheral Select x Registers. Some of these bits may have been set by design to a Privileged or a User value, when others are programmed by software (see Protected Peripheral Select x Registers).

The system-to-peripheral bus bridge compares the incoming host request protection bit with the required protection for the selected peripheral, and accepts or denies access. In the last case, its bus error response is internally flagged in the MATRIX Host Error Status Register; the offending address is registered in the Host Error Address Registers.

27.10.2.1. MATRIXx PPSELRx Register Descriptions

Table 27-18. MATRIX0 PPSELR2 Description

Bit	Peripheral	Mode	Reset Value
0	SEFC0	Programmable	Privilege only
1	SEFC1	Programmable	Privilege only
2	PMC	Programmable	Privilege only

Table 27-19. MATRIX0 PPSELR3 Description

Bit	Peripheral	Mode	Reset Value
0	CPKCC	Programmable	Privilege only
1	MATRIX0	Fixed	Privilege only
2	CMCCI	Programmable	Privilege only
3	CMCCD	Programmable	Privilege only

Table 27-20. MATRIX1 PPSELR1 Description

Bit	Peripheral	Mode	Reset Value
0	SYSC	Programmable	Privilege only

Table 27-21. MATRIX1 PPSELR2 Description

Bit	Peripheral	Mode	Reset Value
0	Reserved	–	–
1	CHIPID	Programmable	Privilege only
2	SFR	Programmable	Privilege only
3	SFRBU	Programmable	Privilege only

© 2026 Microchip Technology Inc. and its subsidiaries

Table 27-22. MATRIX1 PPSELR3 Description

Bit	Peripheral	Mode	Reset Value
0	FLEXCOM0	Programmable	Privilege only
1	FLEXCOM1	Programmable	Privilege only
2	FLEXCOM2	Programmable	Privilege only
3	FLEXCOM3	Programmable	Privilege only
4	FLEXCOM4	Programmable	Privilege only
5	FLEXCOM5	Programmable	Privilege only
6	FLEXCOM6	Programmable	Privilege only
7	FLEXCOM7	Programmable	Privilege only
8	QSPI	Programmable	Privilege only
9	ADCC	Programmable	Privilege only
10	ACC	Programmable	Privilege only
11	Reserved	–	–
12	SLCDC	Programmable	Privilege only
13	MEM2MEM0	Programmable	Privilege only
14	TC0	Programmable	Privilege only
15	TC1	Programmable	Privilege only
16	TC2	Programmable	Privilege only
17	MATRIX1	Fixed	Privilege only
18	PIOA	Programmable	Privilege only
19	AES	Programmable	Privilege only
20	AESB	Programmable	Privilege only
21	SHA	Programmable	Privilege only
22	TRNG	Programmable	Privilege only
23	ICM	Programmable	Privilege only

Table 27-23. MATRIX2 PPSELR3 Description

Bit	Peripheral	Mode	Reset Value
0	MATRIX2	Fixed	Privilege only

Table 27-24. MATRIX3 PPSELR3 Description

Bit	Peripheral	Mode	Reset Value
0	Reserved	–	–
1	MEM2MEM1	Programmable	Privilege only
2	TC3	Programmable	Privilege only
3	PIOD	Programmable	Privilege only
4	UART	Programmable	Privilege only
5	Reserved	–	–
6	MCSPI	Programmable	Privilege only
7	PWM	Programmable	Privilege only
8	MATRIX3	Fixed	Privilege only

© 2026 Microchip Technology Inc. and its subsidiaries

27.11. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	MATRIX_MCFG0	31:24								
		23:16								
		15:8								
		7:0						ULBT[2:0]		
...										
0x3C	MATRIX_MCFG15	31:24								
		23:16								
		15:8								
		7:0						ULBT[2:0]		
0x40	MATRIX_SCFG0	31:24								
		23:16			FIXED_DEFMSTR[3:0]				DEFMSTR_TYPE[1:0]	
		15:8								SLOT_CYCLE[8]
		7:0	SLOT_CYCLE[7:0]							
...										
0x7C	MATRIX_SCFG15	31:24								
		23:16			FIXED_DEFMSTR[3:0]				DEFMSTR_TYPE[1:0]	
		15:8								SLOT_CYCLE[8]
		7:0	SLOT_CYCLE[7:0]							
0x80	MATRIX_PRAS0	31:24		LQOSEN7	M7PR[1:0]			LQOSEN6	M6PR[1:0]	
		23:16		LQOSEN5	M5PR[1:0]			LQOSEN4	M4PR[1:0]	
		15:8		LQOSEN3	M3PR[1:0]			LQOSEN2	M2PR[1:0]	
		7:0		LQOSEN1	M1PR[1:0]			LQOSEN0	M0PR[1:0]	
0x84	MATRIX_PRBS0	31:24		LQOSEN15	M15PR[1:0]			LQOSEN14	M14PR[1:0]	
		23:16		LQOSEN13	M13PR[1:0]			LQOSEN12	M12PR[1:0]	
		15:8		LQOSEN11	M11PR[1:0]			LQOSEN10	M10PR[1:0]	
		7:0		LQOSEN9	M9PR[1:0]			LQOSEN8	M8PR[1:0]	
0x88	MATRIX_PRAS1	31:24		LQOSEN7	M7PR[1:0]			LQOSEN6	M6PR[1:0]	
		23:16		LQOSEN5	M5PR[1:0]			LQOSEN4	M4PR[1:0]	
		15:8		LQOSEN3	M3PR[1:0]			LQOSEN2	M2PR[1:0]	
		7:0		LQOSEN1	M1PR[1:0]			LQOSEN0	M0PR[1:0]	
0x8C	MATRIX_PRBS1	31:24		LQOSEN15	M15PR[1:0]			LQOSEN14	M14PR[1:0]	
		23:16		LQOSEN13	M13PR[1:0]			LQOSEN12	M12PR[1:0]	
		15:8		LQOSEN11	M11PR[1:0]			LQOSEN10	M10PR[1:0]	
		7:0		LQOSEN9	M9PR[1:0]			LQOSEN8	M8PR[1:0]	
0x90	MATRIX_PRAS2	31:24		LQOSEN7	M7PR[1:0]			LQOSEN6	M6PR[1:0]	
		23:16		LQOSEN5	M5PR[1:0]			LQOSEN4	M4PR[1:0]	
		15:8		LQOSEN3	M3PR[1:0]			LQOSEN2	M2PR[1:0]	
		7:0		LQOSEN1	M1PR[1:0]			LQOSEN0	M0PR[1:0]	
0x94	MATRIX_PRBS2	31:24		LQOSEN15	M15PR[1:0]			LQOSEN14	M14PR[1:0]	
		23:16		LQOSEN13	M13PR[1:0]			LQOSEN12	M12PR[1:0]	
		15:8		LQOSEN11	M11PR[1:0]			LQOSEN10	M10PR[1:0]	
		7:0		LQOSEN9	M9PR[1:0]			LQOSEN8	M8PR[1:0]	
0x98	MATRIX_PRAS3	31:24		LQOSEN7	M7PR[1:0]			LQOSEN6	M6PR[1:0]	
		23:16		LQOSEN5	M5PR[1:0]			LQOSEN4	M4PR[1:0]	
		15:8		LQOSEN3	M3PR[1:0]			LQOSEN2	M2PR[1:0]	
		7:0		LQOSEN1	M1PR[1:0]			LQOSEN0	M0PR[1:0]	
0x9C	MATRIX_PRBS3	31:24		LQOSEN15	M15PR[1:0]			LQOSEN14	M14PR[1:0]	
		23:16		LQOSEN13	M13PR[1:0]			LQOSEN12	M12PR[1:0]	
		15:8		LQOSEN11	M11PR[1:0]			LQOSEN10	M10PR[1:0]	
		7:0		LQOSEN9	M9PR[1:0]			LQOSEN8	M8PR[1:0]	
0xA0	MATRIX_PRAS4	31:24		LQOSEN7	M7PR[1:0]			LQOSEN6	M6PR[1:0]	
		23:16		LQOSEN5	M5PR[1:0]			LQOSEN4	M4PR[1:0]	
		15:8		LQOSEN3	M3PR[1:0]			LQOSEN2	M2PR[1:0]	
		7:0		LQOSEN1	M1PR[1:0]			LQOSEN0	M0PR[1:0]	

© 2026 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xA4	MATRIX_PRBS4	31:24		LQOSEN15	M15PR[1:0]			LQOSEN14	M14PR[1:0]	
		23:16		LQOSEN13	M13PR[1:0]			LQOSEN12	M12PR[1:0]	
		15:8		LQOSEN11	M11PR[1:0]			LQOSEN10	M10PR[1:0]	
		7:0		LQOSEN9	M9PR[1:0]			LQOSEN8	M8PR[1:0]	
0xA8	MATRIX_PRAS5	31:24		LQOSEN7	M7PR[1:0]			LQOSEN6	M6PR[1:0]	
		23:16		LQOSEN5	M5PR[1:0]			LQOSEN4	M4PR[1:0]	
		15:8		LQOSEN3	M3PR[1:0]			LQOSEN2	M2PR[1:0]	
		7:0		LQOSEN1	M1PR[1:0]			LQOSEN0	M0PR[1:0]	
0xAC	MATRIX_PRBS5	31:24		LQOSEN15	M15PR[1:0]			LQOSEN14	M14PR[1:0]	
		23:16		LQOSEN13	M13PR[1:0]			LQOSEN12	M12PR[1:0]	
		15:8		LQOSEN11	M11PR[1:0]			LQOSEN10	M10PR[1:0]	
		7:0		LQOSEN9	M9PR[1:0]			LQOSEN8	M8PR[1:0]	
0xB0	MATRIX_PRAS6	31:24		LQOSEN7	M7PR[1:0]			LQOSEN6	M6PR[1:0]	
		23:16		LQOSEN5	M5PR[1:0]			LQOSEN4	M4PR[1:0]	
		15:8		LQOSEN3	M3PR[1:0]			LQOSEN2	M2PR[1:0]	
		7:0		LQOSEN1	M1PR[1:0]			LQOSEN0	M0PR[1:0]	
0xB4	MATRIX_PRBS6	31:24		LQOSEN15	M15PR[1:0]			LQOSEN14	M14PR[1:0]	
		23:16		LQOSEN13	M13PR[1:0]			LQOSEN12	M12PR[1:0]	
		15:8		LQOSEN11	M11PR[1:0]			LQOSEN10	M10PR[1:0]	
		7:0		LQOSEN9	M9PR[1:0]			LQOSEN8	M8PR[1:0]	
0xB8	MATRIX_PRAS7	31:24		LQOSEN7	M7PR[1:0]			LQOSEN6	M6PR[1:0]	
		23:16		LQOSEN5	M5PR[1:0]			LQOSEN4	M4PR[1:0]	
		15:8		LQOSEN3	M3PR[1:0]			LQOSEN2	M2PR[1:0]	
		7:0		LQOSEN1	M1PR[1:0]			LQOSEN0	M0PR[1:0]	
0xBC	MATRIX_PRBS7	31:24		LQOSEN15	M15PR[1:0]			LQOSEN14	M14PR[1:0]	
		23:16		LQOSEN13	M13PR[1:0]			LQOSEN12	M12PR[1:0]	
		15:8		LQOSEN11	M11PR[1:0]			LQOSEN10	M10PR[1:0]	
		7:0		LQOSEN9	M9PR[1:0]			LQOSEN8	M8PR[1:0]	
0xC0	MATRIX_PRAS8	31:24		LQOSEN7	M7PR[1:0]			LQOSEN6	M6PR[1:0]	
		23:16		LQOSEN5	M5PR[1:0]			LQOSEN4	M4PR[1:0]	
		15:8		LQOSEN3	M3PR[1:0]			LQOSEN2	M2PR[1:0]	
		7:0		LQOSEN1	M1PR[1:0]			LQOSEN0	M0PR[1:0]	
0xC4	MATRIX_PRBS8	31:24		LQOSEN15	M15PR[1:0]			LQOSEN14	M14PR[1:0]	
		23:16		LQOSEN13	M13PR[1:0]			LQOSEN12	M12PR[1:0]	
		15:8		LQOSEN11	M11PR[1:0]			LQOSEN10	M10PR[1:0]	
		7:0		LQOSEN9	M9PR[1:0]			LQOSEN8	M8PR[1:0]	
0xC8	MATRIX_PRAS9	31:24		LQOSEN7	M7PR[1:0]			LQOSEN6	M6PR[1:0]	
		23:16		LQOSEN5	M5PR[1:0]			LQOSEN4	M4PR[1:0]	
		15:8		LQOSEN3	M3PR[1:0]			LQOSEN2	M2PR[1:0]	
		7:0		LQOSEN1	M1PR[1:0]			LQOSEN0	M0PR[1:0]	
0xCC	MATRIX_PRBS9	31:24		LQOSEN15	M15PR[1:0]			LQOSEN14	M14PR[1:0]	
		23:16		LQOSEN13	M13PR[1:0]			LQOSEN12	M12PR[1:0]	
		15:8		LQOSEN11	M11PR[1:0]			LQOSEN10	M10PR[1:0]	
		7:0		LQOSEN9	M9PR[1:0]			LQOSEN8	M8PR[1:0]	
0xD0	MATRIX_PRAS10	31:24		LQOSEN7	M7PR[1:0]			LQOSEN6	M6PR[1:0]	
		23:16		LQOSEN5	M5PR[1:0]			LQOSEN4	M4PR[1:0]	
		15:8		LQOSEN3	M3PR[1:0]			LQOSEN2	M2PR[1:0]	
		7:0		LQOSEN1	M1PR[1:0]			LQOSEN0	M0PR[1:0]	
0xD4	MATRIX_PRBS10	31:24		LQOSEN15	M15PR[1:0]			LQOSEN14	M14PR[1:0]	
		23:16		LQOSEN13	M13PR[1:0]			LQOSEN12	M12PR[1:0]	
		15:8		LQOSEN11	M11PR[1:0]			LQOSEN10	M10PR[1:0]	
		7:0		LQOSEN9	M9PR[1:0]			LQOSEN8	M8PR[1:0]	
0xD8	MATRIX_PRAS11	31:24		LQOSEN7	M7PR[1:0]			LQOSEN6	M6PR[1:0]	
		23:16		LQOSEN5	M5PR[1:0]			LQOSEN4	M4PR[1:0]	
		15:8		LQOSEN3	M3PR[1:0]			LQOSEN2	M2PR[1:0]	
		7:0		LQOSEN1	M1PR[1:0]			LQOSEN0	M0PR[1:0]	
0xDC	MATRIX_PRBS11	31:24		LQOSEN15	M15PR[1:0]			LQOSEN14	M14PR[1:0]	
		23:16		LQOSEN13	M13PR[1:0]			LQOSEN12	M12PR[1:0]	
		15:8		LQOSEN11	M11PR[1:0]			LQOSEN10	M10PR[1:0]	
		7:0		LQOSEN9	M9PR[1:0]			LQOSEN8	M8PR[1:0]	

© 2026 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xE0	MATRIX_PRAS12	31:24		LQOSEN7	M7PR[1:0]			LQOSEN6	M6PR[1:0]	
		23:16		LQOSEN5	M5PR[1:0]			LQOSEN4	M4PR[1:0]	
		15:8		LQOSEN3	M3PR[1:0]			LQOSEN2	M2PR[1:0]	
		7:0		LQOSEN1	M1PR[1:0]			LQOSEN0	M0PR[1:0]	
0xE4	MATRIX_PRBS12	31:24		LQOSEN15	M15PR[1:0]			LQOSEN14	M14PR[1:0]	
		23:16		LQOSEN13	M13PR[1:0]			LQOSEN12	M12PR[1:0]	
		15:8		LQOSEN11	M11PR[1:0]			LQOSEN10	M10PR[1:0]	
		7:0		LQOSEN9	M9PR[1:0]			LQOSEN8	M8PR[1:0]	
0xE8	MATRIX_PRAS13	31:24		LQOSEN7	M7PR[1:0]			LQOSEN6	M6PR[1:0]	
		23:16		LQOSEN5	M5PR[1:0]			LQOSEN4	M4PR[1:0]	
		15:8		LQOSEN3	M3PR[1:0]			LQOSEN2	M2PR[1:0]	
		7:0		LQOSEN1	M1PR[1:0]			LQOSEN0	M0PR[1:0]	
0xEC	MATRIX_PRBS13	31:24		LQOSEN15	M15PR[1:0]			LQOSEN14	M14PR[1:0]	
		23:16		LQOSEN13	M13PR[1:0]			LQOSEN12	M12PR[1:0]	
		15:8		LQOSEN11	M11PR[1:0]			LQOSEN10	M10PR[1:0]	
		7:0		LQOSEN9	M9PR[1:0]			LQOSEN8	M8PR[1:0]	
0xF0	MATRIX_PRAS14	31:24		LQOSEN7	M7PR[1:0]			LQOSEN6	M6PR[1:0]	
		23:16		LQOSEN5	M5PR[1:0]			LQOSEN4	M4PR[1:0]	
		15:8		LQOSEN3	M3PR[1:0]			LQOSEN2	M2PR[1:0]	
		7:0		LQOSEN1	M1PR[1:0]			LQOSEN0	M0PR[1:0]	
0xF4	MATRIX_PRBS14	31:24		LQOSEN15	M15PR[1:0]			LQOSEN14	M14PR[1:0]	
		23:16		LQOSEN13	M13PR[1:0]			LQOSEN12	M12PR[1:0]	
		15:8		LQOSEN11	M11PR[1:0]			LQOSEN10	M10PR[1:0]	
		7:0		LQOSEN9	M9PR[1:0]			LQOSEN8	M8PR[1:0]	
0xF8	MATRIX_PRAS15	31:24		LQOSEN7	M7PR[1:0]			LQOSEN6	M6PR[1:0]	
		23:16		LQOSEN5	M5PR[1:0]			LQOSEN4	M4PR[1:0]	
		15:8		LQOSEN3	M3PR[1:0]			LQOSEN2	M2PR[1:0]	
		7:0		LQOSEN1	M1PR[1:0]			LQOSEN0	M0PR[1:0]	
0xFC	MATRIX_PRBS15	31:24		LQOSEN15	M15PR[1:0]			LQOSEN14	M14PR[1:0]	
		23:16		LQOSEN13	M13PR[1:0]			LQOSEN12	M12PR[1:0]	
		15:8		LQOSEN11	M11PR[1:0]			LQOSEN10	M10PR[1:0]	
		7:0		LQOSEN9	M9PR[1:0]			LQOSEN8	M8PR[1:0]	
0x0100	MATRIX_MRCR	31:24								
		23:16								
		15:8	RCB15	RCB14	RCB13	RCB12	RCB11	RCB10	RCB9	RCB8
		7:0	RCB7	RCB6	RCB5	RCB4	RCB3	RCB2	RCB1	RCB0
0x0104 ... 0x014F	Reserved									
0x0150	MATRIX_MEIER	31:24								
		23:16								
		15:8	MERR15	MERR14	MERR13	MERR12	MERR11	MERR10	MERR9	MERR8
		7:0	MERR7	MERR6	MERR5	MERR4	MERR3	MERR2	MERR1	MERR0
0x0154	MATRIX_MEIDR	31:24								
		23:16								
		15:8	MERR15	MERR14	MERR13	MERR12	MERR11	MERR10	MERR9	MERR8
		7:0	MERR7	MERR6	MERR5	MERR4	MERR3	MERR2	MERR1	MERR0
0x0158	MATRIX_MEIMR	31:24								
		23:16								
		15:8	MERR15	MERR14	MERR13	MERR12	MERR11	MERR10	MERR9	MERR8
		7:0	MERR7	MERR6	MERR5	MERR4	MERR3	MERR2	MERR1	MERR0
0x015C	MATRIX_MESR	31:24								
		23:16								
		15:8	MERR15	MERR14	MERR13	MERR12	MERR11	MERR10	MERR9	MERR8
		7:0	MERR7	MERR6	MERR5	MERR4	MERR3	MERR2	MERR1	MERR0
0x0160	MATRIX_MEAR0	31:24	ERRADD[31:24]							
		23:16	ERRADD[23:16]							
		15:8	ERRADD[15:8]							
		7:0	ERRADD[7:0]							
...										

© 2026 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x019C	MATRIX_MEAR15	31:24	ERRADD[31:24]							
		23:16	ERRADD[23:16]							
		15:8	ERRADD[15:8]							
		7:0	ERRADD[7:0]							
0x01A0 ... 0x01E3	Reserved									
0x01E4	MATRIX_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0	CFGFRZ							WPEN
0x01E8	MATRIX_WPSR	31:24								
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0								WPVS
0x01EC ... 0x01FF	Reserved									
0x0200	MATRIX_PSR0	31:24	DPSOA7	DPSOA6	DPSOA5	DPSOA4	DPSOA3	DPSOA2	DPSOA1	DPSOA0
		23:16	WRUSERH7	WRUSERH6	WRUSERH5	WRUSERH4	WRUSERH3	WRUSERH2	WRUSERH1	WRUSERH0
		15:8	RDUSERH7	RDUSERH6	RDUSERH5	RDUSERH4	RDUSERH3	RDUSERH2	RDUSERH1	RDUSERH0
		7:0	LAUSERH7	LAUSERH6	LAUSERH5	LAUSERH4	LAUSERH3	LAUSERH2	LAUSERH1	LAUSERH0
...										
0x023C	MATRIX_PSR15	31:24	DPSOA17	DPSOA16	DPSOA15	DPSOA14	DPSOA13	DPSOA12	DPSOA11	DPSOA10
		23:16	WRUSERH17	WRUSERH16	WRUSERH15	WRUSERH14	WRUSERH13	WRUSERH12	WRUSERH11	WRUSERH10
		15:8	RDUSERH17	RDUSERH16	RDUSERH15	RDUSERH14	RDUSERH13	RDUSERH12	RDUSERH11	RDUSERH10
		7:0	LAUSERH17	LAUSERH16	LAUSERH15	LAUSERH14	LAUSERH13	LAUSERH12	LAUSERH11	LAUSERH10
0x0240	MATRIX_PASSR0	31:24	PASPLIT7[3:0]				PASPLIT6[3:0]			
		23:16	PASPLIT5[3:0]				PASPLIT4[3:0]			
		15:8	PASPLIT3[3:0]				PASPLIT2[3:0]			
		7:0	PASPLIT1[3:0]				PASPLIT0[3:0]			
...										
0x027C	MATRIX_PASSR15	31:24	PASPLIT17[3:0]				PASPLIT16[3:0]			
		23:16	PASPLIT15[3:0]				PASPLIT14[3:0]			
		15:8	PASPLIT13[3:0]				PASPLIT12[3:0]			
		7:0	PASPLIT11[3:0]				PASPLIT10[3:0]			
0x0280	MATRIX_PRTSR0	31:24	PRTOP7[3:0]				PRTOP6[3:0]			
		23:16	PRTOP5[3:0]				PRTOP4[3:0]			
		15:8	PRTOP3[3:0]				PRTOP2[3:0]			
		7:0	PRTOP1[3:0]				PRTOP0[3:0]			
...										
0x02BC	MATRIX_PRTSR15	31:24	PRTOP17[3:0]				PRTOP16[3:0]			
		23:16	PRTOP15[3:0]				PRTOP14[3:0]			
		15:8	PRTOP13[3:0]				PRTOP12[3:0]			
		7:0	PRTOP11[3:0]				PRTOP10[3:0]			
0x02C0	MATRIX_PPSELR1	31:24	USERP31	USERP30	USERP29	USERP28	USERP27	USERP26	USERP25	USERP24
		23:16	USERP23	USERP22	USERP21	USERP20	USERP19	USERP18	USERP17	USERP16
		15:8	USERP15	USERP14	USERP13	USERP12	USERP11	USERP10	USERP9	USERP8
		7:0	USERP7	USERP6	USERP5	USERP4	USERP3	USERP2	USERP1	USERP0
0x02C4	MATRIX_PPSELR2	31:24	USERP31	USERP30	USERP29	USERP28	USERP27	USERP26	USERP25	USERP24
		23:16	USERP23	USERP22	USERP21	USERP20	USERP19	USERP18	USERP17	USERP16
		15:8	USERP15	USERP14	USERP13	USERP12	USERP11	USERP10	USERP9	USERP8
		7:0	USERP7	USERP6	USERP5	USERP4	USERP3	USERP2	USERP1	USERP0
0x02C8	MATRIX_PPSELR3	31:24	USERP31	USERP30	USERP29	USERP28	USERP27	USERP26	USERP25	USERP24
		23:16	USERP23	USERP22	USERP21	USERP20	USERP19	USERP18	USERP17	USERP16
		15:8	USERP15	USERP14	USERP13	USERP12	USERP11	USERP10	USERP9	USERP8
		7:0	USERP7	USERP6	USERP5	USERP4	USERP3	USERP2	USERP1	USERP0

© 2026 Microchip Technology Inc. and its subsidiaries

27.11.1. MATRIX Host Configuration Register x

Name: MATRIX_MCFGx
Offset: 0x00 + x*0x04 [x=0..15]
Reset: 0x00000004
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						ULBT[2:0]		
Access						R/W	R/W	R/W
Reset						1	0	0

Bits 2:0 – ULBT[2:0] Undefined Length Burst Type

Value	Name	Description
0	UNLIMITED	Unlimited Length Burst—No predicted end of burst is generated, therefore INCR bursts coming from this host can only be broken if the Client Slot Cycle Limit is reached. If the Slot Cycle Limit is not reached, the burst is normally completed by the host, at the latest, on the next system bus 1 Kbyte address boundary, allowing up to 256-beat word bursts or 128-beat double-word bursts. This value should not be used in the very particular case of a host capable of performing back-to-back undefined length bursts on a single client, since this could indefinitely freeze the client arbitration and thus prevent another host from accessing this client.
1	SINGLE	Single Access—The undefined length burst is treated as a succession of single accesses, allowing re-arbitration at each beat of the INCR burst or bursts sequence.
2	4_BEAT	4-beat Burst—The undefined length burst or bursts sequence is split into 4-beat bursts or less, allowing re-arbitration every 4 beats.
3	8_BEAT	8-beat Burst—The undefined length burst or bursts sequence is split into 8-beat bursts or less, allowing re-arbitration every 8 beats.
4	16_BEAT	16-beat Burst—The undefined length burst or bursts sequence is split into 16-beat bursts or less, allowing re-arbitration every 16 beats.
5	32_BEAT	32-beat Burst—The undefined length burst or bursts sequence is split into 32-beat bursts or less, allowing re-arbitration every 32 beats.
6	64_BEAT	64-beat Burst—The undefined length burst or bursts sequence is split into 64-beat bursts or less, allowing re-arbitration every 64 beats.
7	128_BEAT	128-beat Burst—The undefined length burst or bursts sequence is split into 128-beat bursts or less, allowing re-arbitration every 128 beats. Unless duly needed, the ULBT should be left at its default 0 value for power saving.

© 2026 Microchip Technology Inc. and its subsidiaries

27.11.2. MATRIX Client Configuration Register x

Name: MATRIX_SCFGx
Offset: 0x40 + x*0x04 [x=0..15]
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the Write Protection Mode Register.

Table 27-25. MATRIX_SCFG Reset Values

Register MATRIX_	Matrix 0	Matrix 1	Matrix 2	Matrix 3
SCFG0	0x000001FF	0x000001FF	0x001201FF	0x000101FF
SCFG1	0x000001FF	0x000001FF	0x001201FF	0x000101FF
SCFG2	0x000001FF	0x000001FF	0x001201FF	0x000001FF
SCFG3	0x000001FF	0x000001FF	0x000A01FF	0x000001FF
SCFG4	0x000001FF	0x000001FF	0x000D01FF	0x000001FF
SCFG5	0x000001FF	0x000001FF	0x001201FF	0x000001FF
SCFG6	0x000001FF	0x000001FF	0x000101FF	0x000001FF
SCFG7	0x000001FF	0x000001FF	0x000A01FF	0x000001FF
SCFG8	0x000001FF	0x000001FF	0x000D01FF	0x000001FF
SCFG9	0x000001FF	0x000001FF	0x001201FF	0x000001FF
SCFG10	0x000001FF	0x000001FF	0x000101FF	0x00001FF
SCFG11	0x000001FF	0x000001FF	0x000001FF	0x000001FF
SCFG12	0x000001FF	0x000001FF	0x000001FF	0x000001FF
SCFG13	0x000001FF	0x000001FF	0x000001FF	0x000001FF
SCFG14	0x000001FF	0x000001FF	0x000001FF	0x000001FF
SCFG15	0x000001FF	0x000001FF	0x000001FF	0x000001FF

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			FIXED_DEFMSTR[3:0]				DEFMSTR_TYPE[1:0]	
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset								

Bit	15	14	13	12	11	10	9	8
								SLOT_CYCLE[8]
Access								R/W
Reset								

Bit	7	6	5	4	3	2	1	0
	SLOT_CYCLE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								

Bits 21:18 – FIXED_DEFMSTR[3:0] Fixed Default Host
This is the number of the Default Host for this client. Only used if DEFMSTR_TYPE is 2. Specifying the number of a host which is not connected to the selected client is equivalent to setting DEFMSTR_TYPE to 0.

Bits 17:16 – DEFMSTR_TYPE[1:0] Default Host Type

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Name	Description
0	NONE	No Default Host—At the end of the current client access, if no other host request is pending, the client is disconnected from all hosts. This results in a one clock cycle latency for the first access of a burst transfer or for a single access.
1	LAST	Last Default Host—At the end of the current client access, if no other host request is pending, the client stays connected to the last host having accessed it. This results in not having one clock cycle latency when the last host tries to access the client again.
2	FIXED	Fixed Default Host—At the end of the current client access, if no other host request is pending, the client connects to the fixed host the number that has been written in the FIXED_DEFMSTR field. This results in not having one clock cycle latency when the fixed host tries to access the client again.

Bits 8:0 – SLOT_CYCLE[8:0] Maximum Bus Grant Duration for Hosts

When SLOT_CYCLE system bus clock cycles have elapsed since the last arbitration, a new arbitration takes place to let another host access this client. If another host is requesting the client bus, then the current host burst is broken.

If SLOT_CYCLE = 0, the Slot Cycle Limit feature is disabled and bursts always complete unless broken according to the ULBT.

This limit has been placed in order to enforce arbitration so as to meet potential latency constraints of hosts waiting for client access.

This limit must not be too small. Unreasonably small values break every burst and the MATRIX arbitrates without performing any data transfer. The default maximum value is usually an optimal conservative choice.

In most cases, this feature is not needed and should be disabled for power saving.

See Slot Cycle Limit Arbitration for details.

© 2026 Microchip Technology Inc. and its subsidiaries

27.11.3. MATRIX Priority Register A For Clients x

Name: MATRIX_PRASx
Offset: 0x80 + x*0x08 [x=0..15]
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the Write Protection Mode Register.

Table 27-26. MATRIX_PRAS Reset Values

Register MATRIX_	Matrix 0	Matrix 1	Matrix 2	Matrix 3
PRAS0	0x00000200	0x00000022	0x00000000	0x00000000
PRAS1	0x00000000	0x00000000	0x00000000	0x00000000
PRAS2	0x00200000	0x00022000	0x00000000	0x00000000
PRAS3	0x00000000	0x00000000	0x00000000	0x00000000
PRAS4	0x00200000	0x00000000	0x00000000	0x00000000
PRAS5	0x00000022	0x00000000	0x00000000	0x00000000
PRAS6	0x00000000	0x00000000	0x00000000	0x00000000
PRAS7	0x00000200	0x00000000	0x00000000	0x00000000
PRAS8	0x00000000	0x00000000	0x00000000	0x00000000
PRAS9	0x00000020	0x00000000	0x00000000	0x00000000
PRAS10	0x00000000	0x00000000	0x00000022	0x00000000
PRAS11	0x00022000	0x00000000	0x00000000	0x00000000
PRAS12	0x00000000	0x00000000	0x00000000	0x00000000
PRAS13	0x00000000	0x00000000	0x00000000	0x00000000
PRAS14	0x00000000	0x00000000	0x00000000	0x00000000
PRAS15	0x00000000	0x00000000	0x00000000	0x00000000

Bit	31	30	29	28	27	26	25	24
		LQOSEN7	M7PR[1:0]			LQOSEN6	M6PR[1:0]	
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset								

Bit	23	22	21	20	19	18	17	16
		LQOSEN5	M5PR[1:0]			LQOSEN4	M4PR[1:0]	
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset								

Bit	15	14	13	12	11	10	9	8
		LQOSEN3	M3PR[1:0]			LQOSEN2	M2PR[1:0]	
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset								

Bit	7	6	5	4	3	2	1	0
		LQOSEN1	M1PR[1:0]			LQOSEN0	M0PR[1:0]	
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset								

Bits 2, 6, 10, 14, 18, 22, 26, 30 – LQOSENx Latency Quality of Service Enable for Host x

Value	Description
0	Disables propagation of Latency Quality of Service from the Host x to the Client and apply MxPR priority for all access from Host x to the Client.
1	Enables the propagation of Latency Quality of Service from the Host x to the Client if supported by the Host x.

© 2026 Microchip Technology Inc. and its subsidiaries

Bits 0:1, 4:5, 8:9, 12:13, 16:17, 20:21, 24:25, 28:29 – MxPR Host x Priority

Fixed priority of Host x for accessing the selected client. The higher the number, the higher the priority.

All the hosts programmed with the same MxPR value for the client make up a priority pool.

Round-robin arbitration is used in the lowest (MxPR = 0) and highest (MxPR = 3) priority pools.

Fixed priority is used in intermediate priority pools (MxPR = 1) and (MxPR = 2).

See Arbitration Priority Scheme for details.

If LQOSENx bit is cleared, then this priority value is used as it for arbitration and downward propagation to the client. If LQOSENx bit is set, then this priority acts as the upper limit for the Latency Quality of Service from Host x.

For hosts other than the CPU, the usual value of this field should be 0x0 if LQOSENx bit is cleared, and 0x1 if LQOSENx bit is set. For the CPU host, the usual value of this field should be 0x2.

© 2026 Microchip Technology Inc. and its subsidiaries

27.11.4. MATRIX Priority Register B For Clients x

Name: MATRIX_PRBSx
Offset: 0x84 + x*0x08 [x=0..15]
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
		LQOSEN15	M15PR[1:0]			LQOSEN14	M14PR[1:0]	
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bit	23	22	21	20	19	18	17	16
		LQOSEN13	M13PR[1:0]			LQOSEN12	M12PR[1:0]	
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bit	15	14	13	12	11	10	9	8
		LQOSEN11	M11PR[1:0]			LQOSEN10	M10PR[1:0]	
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bit	7	6	5	4	3	2	1	0
		LQOSEN9	M9PR[1:0]			LQOSEN8	M8PR[1:0]	
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bits 2, 6, 10, 14, 18, 22, 26, 30 – LQOSENx Latency Quality of Service Enable for Host x

Value	Description
0	Disables propagation of Latency Quality of Service from the Host x to the Client and apply MxPR priority for all access from Host x to the Client.
1	Enables the propagation of Latency Quality of Service from the Host x to the Client if supported by the Host x.

Bits 0:1, 4:5, 8:9, 12:13, 16:17, 20:21, 24:25, 28:29 – MxPR Host x Priority

Fixed priority of Host x for accessing the selected client. The higher the number, the higher the priority.
All the hosts programmed with the same MxPR value for the client make up a priority pool.
Round-robin arbitration is used in the lowest (MxPR = 0) and highest (MxPR = 3) priority pools.
Fixed priority is used in intermediate priority pools (MxPR = 1) and (MxPR = 2).
See Arbitration Priority Scheme for details.
If LQOSENx bit is cleared, then this priority value is used as it for arbitration and downward propagation to the client. If LQOSENx bit is set, then this priority acts as the upper limit for the Latency Quality of Service from Host x.
For hosts other than the CPU, the usual value of this field should be 0x0 if LQOSENx bit is cleared, and 0x1 if LQOSENx bit is set. For the CPU host, the usual value of this field should be 0x2.

© 2026 Microchip Technology Inc. and its subsidiaries

27.11.5. MATRIX Host Remap Control Register

Name: MATRIX_MRCR
Offset: 0x0100
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	RCB15	RCB14	RCB13	RCB12	RCB11	RCB10	RCB9	RCB8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RCB7	RCB6	RCB5	RCB4	RCB3	RCB2	RCB1	RCB0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – RCBx Remap Command Bit for Host x

Value	Description
0	Disables remapped address decoding for the selected host.
1	Enables remapped address decoding for the selected host.

© 2026 Microchip Technology Inc. and its subsidiaries

27.11.6. MATRIX Host Error Interrupt Enable Register

Name: MATRIX_MEIER
Offset: 0x0150
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	MERR15	MERR14	MERR13	MERR12	MERR11	MERR10	MERR9	MERR8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	MERR7	MERR6	MERR5	MERR4	MERR3	MERR2	MERR1	MERR0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – MERRx Host x Access Error

Value	Description
0	No effect.
1	Enables Host x Access Error interrupt source.

© 2026 Microchip Technology Inc. and its subsidiaries

27.11.7. MATRIX Host Error Interrupt Disable Register

Name: MATRIX_MEIDR
Offset: 0x0154
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	MERR15	MERR14	MERR13	MERR12	MERR11	MERR10	MERR9	MERR8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	MERR7	MERR6	MERR5	MERR4	MERR3	MERR2	MERR1	MERR0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – MERRx Host x Access Error

Value	Description
0	No effect.
1	Disables Host x Access Error interrupt source.

© 2026 Microchip Technology Inc. and its subsidiaries

27.11.8. MATRIX Host Error Interrupt Mask Register

Name: MATRIX_MEIMR
Offset: 0x0158
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	MERR15	MERR14	MERR13	MERR12	MERR11	MERR10	MERR9	MERR8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MERR7	MERR6	MERR5	MERR4	MERR3	MERR2	MERR1	MERR0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – MERRx Host x Access Error

Value	Description
0	Host x Access Error does not trigger any interrupt.
1	Host x Access Error triggers the MATRIX interrupt line.

© 2026 Microchip Technology Inc. and its subsidiaries

27.11.9. MATRIX Host Error Status Register

Name: MATRIX_MESR
Offset: 0x015C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	MERR15	MERR14	MERR13	MERR12	MERR11	MERR10	MERR9	MERR8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MERR7	MERR6	MERR5	MERR4	MERR3	MERR2	MERR1	MERR0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – MERRx Host x Access Error

Value	Description
0	No Host Access Error has occurred since the last read of the MATRIX_MESR.
1	At least one Host Access Error has occurred since the last read of the MATRIX_MESR.

© 2026 Microchip Technology Inc. and its subsidiaries

27.11.10. MATRIX Host Error Address Register x

Name: MATRIX_MEARx
Offset: 0x0160 + x*0x04 [x=0..15]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	ERRADD[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ERRADD[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ERRADD[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ERRADD[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ERRADD[31:0] Host Error Address
32 most significant bits of the last access error address

© 2026 Microchip Technology Inc. and its subsidiaries

27.11.11. MATRIX Write Protection Mode Register

Name: MATRIX_WPMR
Offset: 0x01E4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	CFGFRZ							WPEN
Access	R/W							R/W
Reset	0							0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x4D4154	PASSWD	Writing any other value in this field aborts the write operation of the WPEN and CFGFRZ bits. Always reads as 0.

Bit 7 – CFGFRZ Configuration Freeze

Value	Description
0	The MATRIX configuration is not frozen.
1	Freezes the MATRIX configuration until hardware reset. The registers that can be protected by the WPEN bit and the Write Protection Mode Register are no longer modifiable.

Bit 0 – WPEN Write Protection Enable

See Register Write Protection for the list of registers that can be write-protected.

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x4D4154 (“MAT” in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x4D4154 (“MAT” in ASCII).

© 2026 Microchip Technology Inc. and its subsidiaries

27.11.12. MATRIX Write Protection Status Register

Name: MATRIX_WPSR
Offset: 0x01E8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	WPVSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPVS
Access								R
Reset								0

Bits 23:8 – WPVSRC[15:0] Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protection Violation Status

Value	Description
0	No write protection violation has occurred since the last write of the MATRIX_WPMR.
1	A write protection violation has occurred since the last write of the MATRIX_WPMR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

© 2026 Microchip Technology Inc. and its subsidiaries

27.11.13. MATRIX Protection Client Register x

Name: MATRIX_PSRx
Offset: 0x0200 + x*0x04 [x=0..15]
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the Write Protection Mode Register.

Table 27-27. MATRIX_PSR Reset Values

Register MATRIX_	Matrix 0	Matrix 1	Matrix 2	Matrix 3
PSR0	0x00000000	0x00000000	0x00010101	0x00010101
PSR1	0x00000000	0x00010101	0x00010101	0x00010101
PSR2	0x00000000	0x00010101	0x00010101	0x00000000
PSR3	0x00010101	0x00010101	0x00000000	0x00000000
PSR4	0x00010101	0x00010101	0x00000000	0x00000000
PSR5	0x00000000	0x00000000	0x00000000	0x00000000
PSR6	0x00010101	0x00000000	0x00000000	0x00000000
PSR7	0x00010101	0x00000000	0x00000000	0x00000000
PSR8	0x00030303	0x00000000	0x00000000	0x00000000
PSR9	0x00030303	0x00000000	0x00000000	0x00000000
PSR10	0x00000000	0x00000000	0x00000000	0x00000000
PSR11	0x00010101	0x00000000	0x00000000	0x00000000
PSR12	0x00000000	0x00000000	0x00000000	0x00000000
PSR13	0x00000000	0x00000000	0x00000000	0x00000000
PSR14	0x00000000	0x00000000	0x00000000	0x00000000
PSR15	0x00000000	0x00000000	0x00000000	0x00000000

Bit	31	30	29	28	27	26	25	24
	DPSOA7	DPSOA6	DPSOA5	DPSOA4	DPSOA3	DPSOA2	DPSOA1	DPSOA0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								

Bit	23	22	21	20	19	18	17	16
	WRUSERH7	WRUSERH6	WRUSERH5	WRUSERH4	WRUSERH3	WRUSERH2	WRUSERH1	WRUSERH0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								

Bit	15	14	13	12	11	10	9	8
	RDUSERH7	RDUSERH6	RDUSERH5	RDUSERH4	RDUSERH3	RDUSERH2	RDUSERH1	RDUSERH0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								

Bit	7	6	5	4	3	2	1	0
	LAUSERH7	LAUSERH6	LAUSERH5	LAUSERH4	LAUSERH3	LAUSERH2	LAUSERH1	LAUSERH0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								

Bits 24, 25, 26, 27, 28, 29, 30, 31 – DPSOAx Downward Protection Split Address for HSELx Protected Region

Value	Description
0	For the HSELx client-protected region, the protected area split MATRIX_PASSR / PASPLITx defines the size of the client-protected area starting from the base address of the client protected region and up.
1	For the HSELx client-protected region, the protected area split MATRIX_PASSR / PASPLITx defines the size of the client-protected area starting from the end address of the client-protected region and down.

© 2026 Microchip Technology Inc. and its subsidiaries

Bits 16, 17, 18, 19, 20, 21, 22, 23 – WRUSERHx Write USER for HSELx Protected Region
Privilege-eligible Area access rights:

WRUSERHx / RDUSERHx	User Access	Privileged Access
00	Denied	Write - Read
01	Read	Write - Read
10	Write	Write - Read
11	Write - Read	Write - Read

Value	Description
0	The HSELx client-protected region is split into one Write Privileged and one Write User area, according to LAUSERHx and MATRIX_PASSR / PASPLITx. That is, the so defined privilege-eligible high or low area is Privileged for Write access.
1	The HSELx client-protected region is User for Write access.

Bits 8, 9, 10, 11, 12, 13, 14, 15 – RDUSERHx Read USER for HSELx Protected Region

Value	Description
0	The HSELx client-protected region is split into one Read Privileged and one Read User area, according to LAUSERHx and MATRIX_PASSR / PASPLITx. That is, the so defined privilege-eligible high or low area is Privileged for Read access.
1	The HSELx client-protected region is User for Read access.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – LAUSERHx Low Area USER in HSELx Protected Region

Value	Description
0	The protection of the HSELx client area laying below the corresponding MATRIX_PASSR / PASPLITx boundary is configured according to RDUSERHx and WRUSERHx. The whole remaining HSELx upper address space is configured as User access.
1	The HSELx client address area laying below the corresponding MATRIX_PASSR / PASPLITx boundary is configured as User access, and the whole remaining upper address space according to RDUSERHx and WRUSERHx.

© 2026 Microchip Technology Inc. and its subsidiaries

27.11.14. MATRIX Protected Areas Split Client Register x

Name: MATRIX_PASSRx
Offset: 0x0240 + x*0x04 [x=0..15]
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the Write Protection Mode Register.

Table 27-28. MATRIX_PASSR Reset Values

Register MATRIX_	Matrix 0	Matrix 1	Matrix 2	Matrix 3
PASSR0	0x0000FFFF	0x0000FFFF	0x0000000F	0x0000000F
PASSR1	0x0000FFFF	0x0000000F	0x0000000F	0x0000000F
PASSR2	0x0000FFFF	0x0000000F	0x0000000F	0x00000000
PASSR3	0x0000000F	0x0000000F	0x0000000F	0x00000000
PASSR4	0x0000000F	0x0000000F	0x0000000F	0x00000000
PASSR5	0x0000000F	0x00000000	0x0000000F	0x00000000
PASSR6	0x0000000F	0x00000000	0x0000000F	0x00000000
PASSR7	0x0000000F	0x00000000	0x0000000F	0x00000000
PASSR8	0x000000FF	0x00000000	0x0000000F	0x00000000
PASSR9	0x000000FF	0x00000000	0x0000000F	0x00000000
PASSR10	0xFFFFFFFF	0x00000000	0x0000000F	0x00000000
PASSR11	0x0000000F	0x00000000	0x00000000	0x00000000
PASSR12	0x00000000	0x00000000	0x00000000	0x00000000
PASSR13	0x00000000	0x00000000	0x00000000	0x00000000
PASSR14	0x00000000	0x00000000	0x00000000	0x00000000
PASSR15	0x00000000	0x00000000	0x00000000	0x00000000

Bit	31	30	29	28	27	26	25	24
	PASPLIT7[3:0]				PASPLIT6[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								

Bit	23	22	21	20	19	18	17	16
	PASPLIT5[3:0]				PASPLIT4[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								

Bit	15	14	13	12	11	10	9	8
	PASPLIT3[3:0]				PASPLIT2[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								

Bit	7	6	5	4	3	2	1	0
	PASPLIT1[3:0]				PASPLIT0[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								

Bits 0:3, 4:7, 8:11, 12:15, 16:19, 20:23, 24:27, 28:31 – PASPLITx Protected Areas Split for HSELx Protected Region

This field defines the boundary address offset where the HSELx client-protected region splits into two Protected Areas whose access is controlled according to the corresponding MATRIX_PSR.
If this size is set at or above the HSELx Region Size, then the MATRIX_PSR register settings apply to the unique protected area then covering the whole HSELx Protected Region.

PASPLITx	Split Offset	Protected Low Area Size
0000	0x00001000	4 Kbytes

© 2026 Microchip Technology Inc. and its subsidiaries

PASPLITx (continued)

PASPLITx	Split Offset	Protected Low Area Size
0001	0x00002000	8 Kbytes
0010	0x00004000	16 Kbytes
0011	0x00008000	32 Kbytes
0100	0x00010000	64 Kbytes
0101	0x00020000	128 Kbytes
0110	0x00040000	256 Kbytes
0111	0x00080000	512 Kbytes
1000	0x00100000	1 Mbyte
1001	0x00200000	2 Mbytes
1010	0x00400000	4 Mbytes
1011	0x00800000	8 Mbytes
1100	0x01000000	16 Mbytes
1101	0x02000000	32 Mbytes
1110	0x04000000	64 Mbytes
1111	0x08000000	128 Mbytes

© 2026 Microchip Technology Inc. and its subsidiaries

27.11.15. MATRIX Protected Region Top Client Register x

Name: MATRIX_PRTSRx
Offset: 0x0280 + x*0x04 [x=0..15]
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	PRTOP7[3:0]				PRTOP6[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	PRTOP5[3:0]				PRTOP4[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	PRTOP3[3:0]				PRTOP2[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PRTOP1[3:0]				PRTOP0[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0:3, 4:7, 8:11, 12:15, 16:19, 20:23, 24:27, 28:31 – PRTOPx HSELx Protected Region Top
This field defines the size of the HSELx protected region address space. Invalid sizes for the client region must never be programmed. Valid sizes and number of protected regions are product, client and client configuration dependent.
Note: The clients featuring multiple scalable contiguous protected regions have a single PRTOP0 field for all the protected regions.

If this HSELx protected region size is set at or below the MATRIX_PASSR / PASPLITx Protected Areas Split size, then the MATRIX_PSR register settings apply to the unique protected area then covering the whole HSELx protected region

PRTOPx	Top Offset	Protected Region Size
0000	0x00001000	4 Kbytes
0001	0x00002000	8 Kbytes
0010	0x00004000	16 Kbytes
0011	0x00008000	32 Kbytes
0100	0x00010000	64 Kbytes
0101	0x00020000	128 Kbytes
0110	0x00040000	256 Kbytes
0111	0x00080000	512 Kbytes
1000	0x00100000	1 Mbyte
1001	0x00200000	2 Mbytes
1010	0x00400000	4 Mbytes
1011	0x00800000	8 Mbytes
1100	0x01000000	16 Mbytes

© 2026 Microchip Technology Inc. and its subsidiaries

PRTOPx (continued)

PRTOPx	Top Offset	Protected Region Size
1101	0x02000000	32 Mbytes
1110	0x04000000	64 Mbytes
1111	0x08000000	128 Mbytes

© 2026 Microchip Technology Inc. and its subsidiaries

27.11.16. MATRIX Protected Peripheral Select x Registers [x=1..3]

Name: MATRIX_PPSELRx
Offset: 0x02C0 + (x-1)*0x04 [x=1..3]
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	USERP31	USERP30	USERP29	USERP28	USERP27	USERP26	USERP25	USERP24
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	USERP23	USERP22	USERP21	USERP20	USERP19	USERP18	USERP17	USERP16
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	USERP15	USERP14	USERP13	USERP12	USERP11	USERP10	USERP9	USERP8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	USERP7	USERP6	USERP5	USERP4	USERP3	USERP2	USERP1	USERP0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – USERPy User PSELy Peripheral

Value	Description
0	The PSELx[y] APB Peripheral address space is configured as Privileged access.
1	The PSELx[y] APB Peripheral address space is configured as User access.

© 2026 Microchip Technology Inc. and its subsidiaries

28. Chip Identifier (CHIPID)

28.1. Description

Chip Identifier (CHIPID) registers are used to recognize the device and its revision. These registers provide the sizes and types of the on-chip memories, as well as the set of embedded peripherals.

Two CHIPID registers are embedded: Chip ID Register (CHIPID_CIDR) and Chip ID Extension Register (CHIPID_EXID). Both registers contain a hard-wired value that is read-only.

The CHIPID_CIDR register contains the following fields:

- VERSION: Identifies the revision of the silicon
- EPROC: Indicates the embedded ARM processor
- NVPTYP and NVPSIZ: Identify the type of embedded non-volatile memory and the size
- SRAMSIZ: Indicates the size of the embedded SRAM
- ARCH: Identifies the set of embedded peripherals
- EXT: Shows the use of the extension identifier register

The CHIPID_EXID register is device-dependent and reads 0 if CHIPID_CIDR.EXT = 0.

28.2. Embedded Characteristics

- Chip ID Registers
  - Identification of the Device Revision, Sizes of the Embedded Memories, Set of Peripherals, Embedded Processor

Table 28-1. PIC32CXMTG Chip ID Registers

Chip Name	CHIPID_CIDR	CHIPID_EXID
PIC32CX2051MTG128	0x2C8F0EE0	0x00000000
PIC32CX2051MTG64	0x2C6F0EE0	0x00000000
PIC32CX1025MTG128	0x2C8D0CE0	0x00000000
PIC32CX1025MTG64	0x2C6D0CE0	0x00000000
PIC32CX5112MTG128	0x2C8C0AE0	0x00000000

© 2026 Microchip Technology Inc. and its subsidiaries

28.3. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CHIPID_CIDR	31:24	EXT	NVPTYP[2:0]			ARCH[7:4]			
		23:16	ARCH[3:0]				SRAMSIZ[3:0]			
		15:8	NVPSIZ2[3:0]				NVPSIZ[3:0]			
		7:0	EPROC[2:0]			VERSION[4:0]				
0x04	CHIPID_EXID	31:24	EXID[31:24]							
		23:16	EXID[23:16]							
		15:8	EXID[15:8]							
		7:0	EXID[7:0]							

© 2026 Microchip Technology Inc. and its subsidiaries

28.3.1. Chip ID Register

Name: CHIPID_CIDR
Offset: 0x0
Reset: –
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	EXT	NVPTYP[2:0]			ARCH[7:4]			
Access	R	R	R	R	R	R	R	R
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	ARCH[3:0]				SRAMSIZ[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	NVPSIZ2[3:0]				NVPSIZ[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	EPROC[2:0]			VERSION[4:0]				
Access	R	R	R	R	R	R	R	R
Reset	–	–	–	–	–	–	–	–

Bit 31 – EXT Extension Flag

Value	Description
0	Chip ID has a single register definition without extension.
1	An extended Chip ID exists.

Bits 30:28 – NVPTYP[2:0] Nonvolatile Program Memory Type
Must be '2'.

Bits 27:20 – ARCH[7:0] Architecture Identifier

Value	Name	Description
0xC6	PIC32CXMTG64	Single Core (64-lead version)
0xC8	PIC32CXMTG128	Single Core (128-lead version)

Bits 19:16 – SRAMSIZ[3:0] Internal SRAM Size

Value	Name	Description
0–11	–	Reserved
12	128K	128 Kbytes
13	256K	256 Kbytes
14	–	Reserved
15	512K	512 Kbytes

Bits 15:12 – NVPSIZ2[3:0] Second Nonvolatile Program Memory Size
Must be '0'.

Bits 11:8 – NVPSIZ[3:0] Nonvolatile Program Memory Size

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Name	Description
0-9	–	Reserved
10	512K	512 Kbytes
11	–	Reserved
12	1024K	1024 Kbytes
13	–	Reserved
14	2048K	2048 Kbytes
15	–	Reserved

Bits 7:5 – EPROC[2:0] Embedded Processor
Must be '7'.

Bits 4:0 – VERSION[4:0] Version of the Device
Current version of the device.

© 2026 Microchip Technology Inc. and its subsidiaries

28.3.2. Chip ID Extension Register

Name: CHIPID_EXID
Offset: 0x4
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	EXID[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	EXID[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	EXID[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	EXID[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – EXID[31:0] Chip ID Extension
This field is cleared if CHIPID_CIDR.EXT = 0.

© 2026 Microchip Technology Inc. and its subsidiaries

29. Secure Embedded Flash Controller (SEFC)

29.1. Description

The Secure Embedded Flash Controller (SEFC) manages the programming, erasing, locking and unlocking sequences of the Flash using a full set of commands.

Security in the SEFC is based on access rights, secure key storage and a Private Key bus. The SEFC manages safety features, including Error Correction Code and a self-check mechanism reported by the Flash block.

29.2. Embedded Characteristics

- Memory Organization
  - 512-byte pages granularity
  - 32-Kbyte user signature area
  - Unique identifier area
  - Flash descriptor of Flash configuration
  - 9 general-purpose GPNVM bits for device Booting option
- Safety and Security
  - 128 lock bits, each protecting a lock region
  - One non-volatile bit for code and device protection
  - User signature blocks of 4 Kbytes secured with locks or access rights
  - Cryptographic key transfer from user signature area to AES engine over private key bus
  - One Time Programmable (OTP) block for software monotonic counter
  - Erase Function Lock (EFL) to disable the chip hardware Erase signal
  - Register write and command protection
  - Program Verify and Erase Verify Fail self-checking mechanism
  - ECC single and multiple error flags report
- Erase and Write Operations
  - Write by page
  - Software erase by block (4 Kbytes), multiple blocks or sector
  - Full chip erase by hardware Erase signal
  - Suspend and resume of write and erase operations
- Read Operations
  - High performance in Thumb-2 mode with 128-bit-wide memory interface
  - Code loop and code read optimization
  - Automatic Clock-off mode for power reduction
  - Supports read of the calibration bits

29.3. Product Dependencies

29.3.1. Power Management

The Secure Embedded Flash Controller is continuously clocked. The Power Management Controller has no effect on its behavior.

© 2026 Microchip Technology Inc. and its subsidiaries

29.3.2. Interrupt Sources

The SEFC interrupt line is connected to the interrupt controller. Using the SEFC interrupt requires the interrupt controller to be programmed first. The SEFC interrupt is generated only if the value of the bit FRDY in Flash Mode register (EEFC_FMR) is ‘1’.

29.4. Functional Description

29.4.1. Embedded Flash Organization

The SEFC interfaces between an embedded Flash memory plane and the internal bus.

A memory subsystem is composed of:

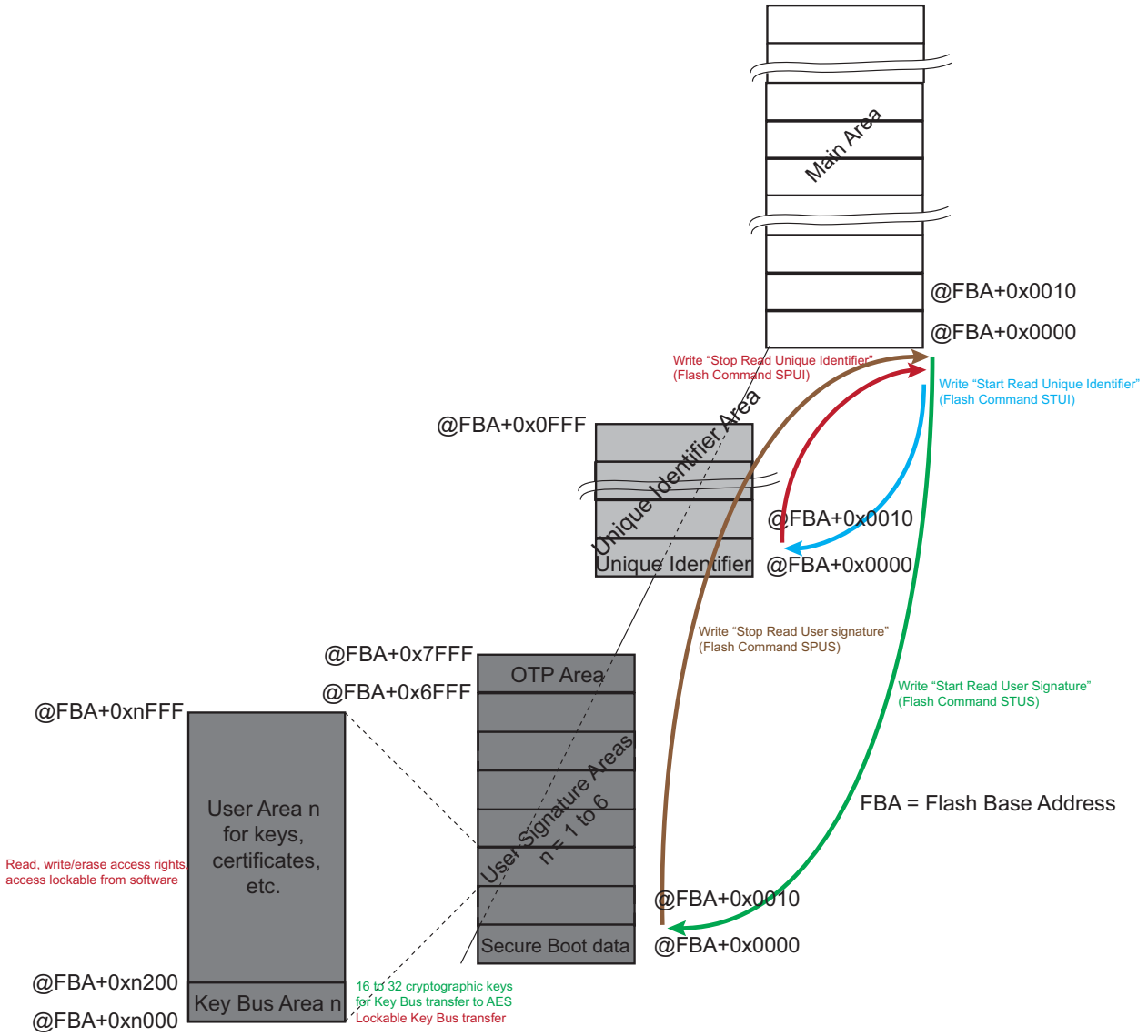
- The 256-KByte, 512-KByte or 1-MByte main area memory plane organized in several pages of 512 bytes for the code and data
- A separate 4-Kbyte memory area which includes the unique chip identifier
- A separate 32-Kbyte (8 x 4K) memory area for the user signature, cryptographic keys and One Time Programmable block
- Two 128-bit read buffers for code read optimization
- Two 128-bit read buffers for code loop optimization
- One 128-bit read buffer for data read optimization
- One write-only buffer equal to the page size that manages page programming along the full flash memory address space
- Lock bits used to protect write/erase operation on several pages (lock region); a lock bit is associated with a lock region composed of several pages in the memory plane
- General-purpose non-volatile memory bits (GPNVM) that may be set and cleared through the SEFC interface (only Plane 0 GPNVM bits are active)

A device may feature more than one memory subsystem. Refer to the table “Configuration Summary” for the device memory configuration.

The embedded Flash size, the page size, the organization of lock regions and the definition of GPNVM bits are specific to the device. The SEFC returns a descriptor of the Flash controller after a ‘Get Flash Descriptor’ command was issued by the application (see Get Flash Descriptor Command).

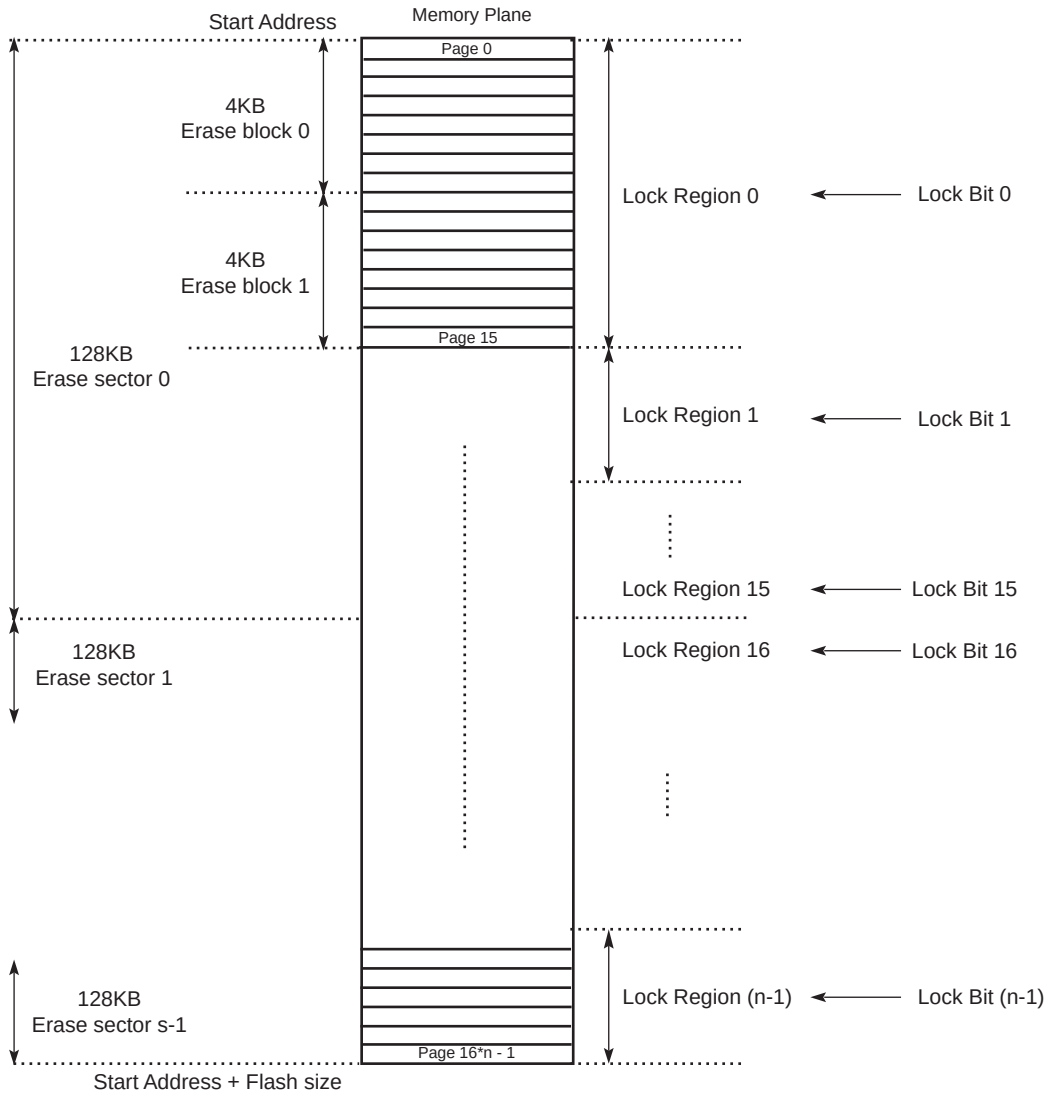
© 2026 Microchip Technology Inc. and its subsidiaries

Figure 29-1. Flash Memory Areas



© 2026 Microchip Technology Inc. and its subsidiaries

Figure 29-2. Embedded Flash Organization



Note: The above figure shows the organization of the memory plane of one embedded Flash. Memory plane organization is identical for devices with two memory planes and Flash controllers.

29.4.2. Read Operations

The SEFC manages embedded Flash reads. Performance is increased when the processor is running in Thumb-2 mode by means of the 128-bit-wide memory interface.

The Flash memory is accessible through 8-, 16- and 32-bit reads.

The read operations can be performed with or without wait states. Wait states must be programmed in the EEFC_FMR.FWS field. Defining FWS as 0 enables the single-cycle access of the embedded Flash. For more details, refer to the section “Electrical Characteristics” of this datasheet.

Reading in the memory plane stalls the bus when it is being programmed or erased except when the suspend feature is used.

Reading in one Flash memory plane through one SEFC does not stall the bus when a second Flash memory plane is being programmed or erased through the second SEFC.

29.4.2.1. Code Read Optimization

Code read optimization is enabled if the bit EEFC_FMR.SCOD is cleared.

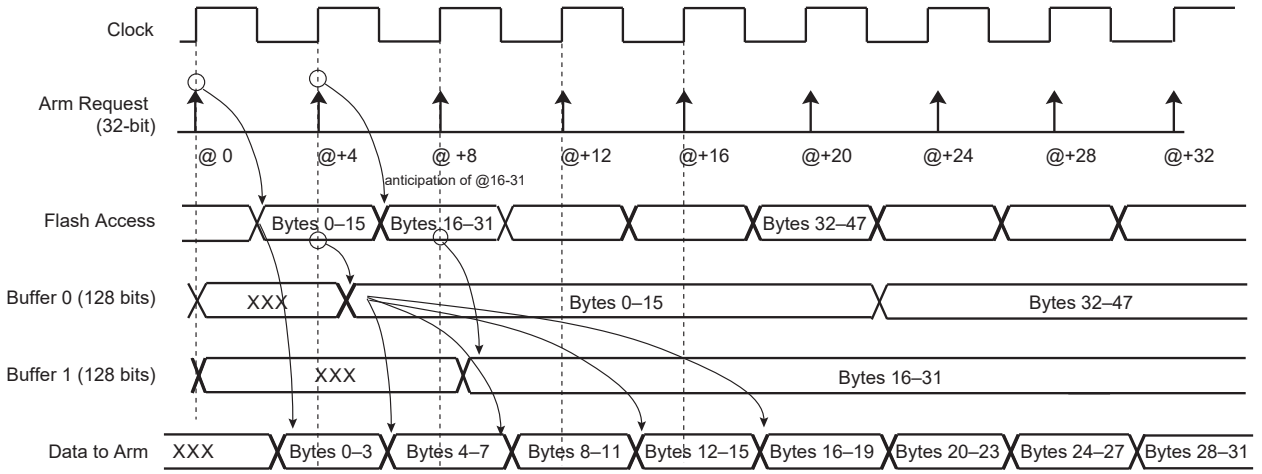
© 2026 Microchip Technology Inc. and its subsidiaries

A system of 2 x 128-bit buffers is added in order to optimize sequential code fetch.
Note: Immediate consecutive code read accesses are not mandatory to benefit from this optimization.

The sequential code read optimization is enabled by default. If the bit EEFC_FMR.SCOD is set, these buffers are disabled and the sequential code read is no longer optimized.

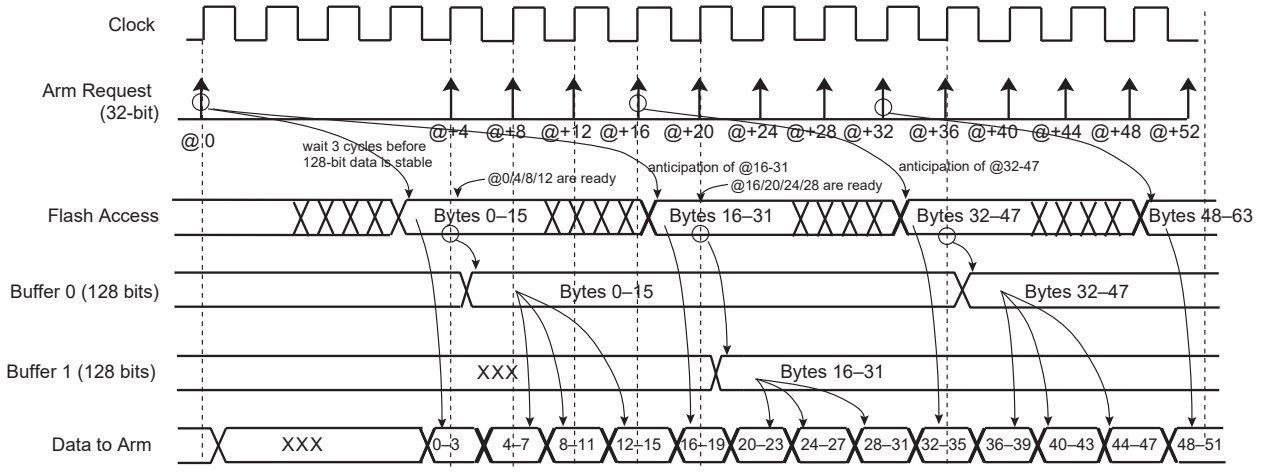
Another system of 2 x 128-bit buffers is added in order to optimize loop code fetch. See Code Loop Optimization for more details.

Figure 29-3. Code Read Optimization for FWS = 0



Note: When FWS is equal to 0, all the accesses are performed in a single-cycle access.

Figure 29-4. Code Read Optimization for FWS = 3



Note: When FWS is between 1 and 3, in case of sequential reads, the first access takes (FWS + 1) cycles. The following accesses take only one cycle.

29.4.2.2. Code Loop Optimization

Code loop optimization is enabled when the bit EEFC_FMR.CLOE is set.

When a backward jump is inserted in the code, the pipeline of the sequential optimization is broken and becomes inefficient. In this case, the loop code read optimization takes over from the sequential code read optimization to prevent the insertion of wait states. The loop code read optimization is enabled by default. In EEFC_FMR, if the bit CLOE is reset to 0 or the bit SCOD is set, these buffers are disabled and the loop code read is not optimized.

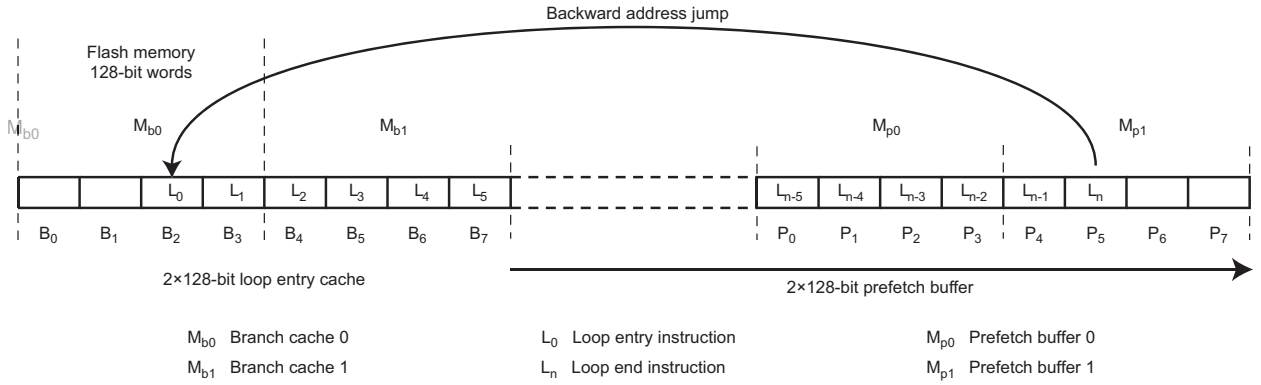
© 2026 Microchip Technology Inc. and its subsidiaries

When code loop optimization is enabled, if inner loop body instructions L_0 to L_n are positioned from the 128-bit Flash memory cell M_{b0} to the memory cell M_{p1}, after recognition of a first backward branch, the first two Flash memory cells M_{b0} and M_{b1} targeted by this branch are cached for fast access from the processor at the next loop iteration.

Then by combining the sequential prefetch (described in Code Read Optimization) through the loop body with the fast read access to the loop entry cache, the entire loop can be iterated with no wait state.

The following figure illustrates code loop optimization.

Figure 29-5. Code Loop Optimization

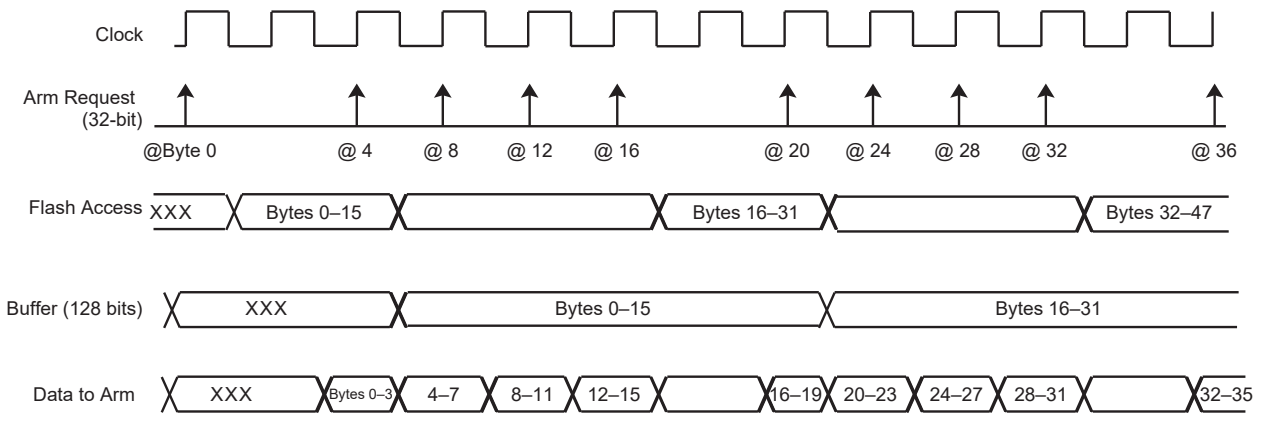


29.4.2.3. Data Read Optimization

The organization of the Flash in 128 bits is associated with two 128-bit prefetch buffers and one 128-bit data read buffer, thus providing maximum system performance. This buffer is added in order to store the requested data plus all the data contained in the 128-bit aligned data. This speeds up sequential data reads if, for example, FWS is equal to 1 (see the figure below). The data read optimization is enabled by default. If the bit EEFC_FMR.SCOD is set, this buffer is disabled and the data read is no longer optimized.

Note: Consecutive data read accesses need neither being consecutive in time nor strictly sequential in address to benefit from this optimization.

Figure 29-6. Data Read Optimization for FWS = 1



29.4.3. Flash Commands

The SEFC offers a set of commands to manage programming the Flash memory, locking and unlocking lock regions, consecutive programming, locking and full Flash erasing, etc.

The commands are listed in the following table.

© 2026 Microchip Technology Inc. and its subsidiaries

Table 29-1. Set of Commands

Command	Value	Mnemonic
Get Flash descriptor	0x00	GETD
Write page	0x01	WP
Write page and lock	0x02	WPL
Erase pages	0x07	EPA
Set lock bit	0x08	SLB
Clear lock bit	0x09	CLB
Get lock bit	0x0A	GLB
Set GPNVM bit	0x0B	SGPB
Clear GPNVM bit	0x0C	CGPB
Get GPNVM bit	0x0D	GGPB
Start read unique identifier	0x0E	STUI
Stop read unique identifier	0x0F	SPUI
Get CALIB bit	0x10	GCALB
Erase sector	0x11	ES
Write user signature	0x12	WUS
Erase user signature	0x13	EUS
Start read user signature	0x14	STUS
Stop read user signature	0x15	SPUS
Suspend	0x17	SUSP
Resume	0x18	RES
Send cryptographic key	0x19	SCK

In order to execute one of these commands, select the required command using the FCMD field in the Flash Command register (EEFC_FCR). As soon as EEFC_FCR is written, the FRDY flag is automatically cleared, except for the GETD command. The FVALUE field in the Flash Result register (EEFC_FRR) is also automatically cleared. Once the current command has completed, the FRDY flag is automatically set, except for the STUI and STUS commands. If an interrupt was enabled by setting the bit EEFC_FMR.FRDY, the corresponding interrupt line of the interrupt controller is activated.

Write and Erase commands can be suspended by writing the Suspend (SUSP) command in EEFC_FCR. Only read access to the Flash is permitted during suspended commands. The suspended command can be later resumed by writing a Resume command (RES) in EEFC_FCR.

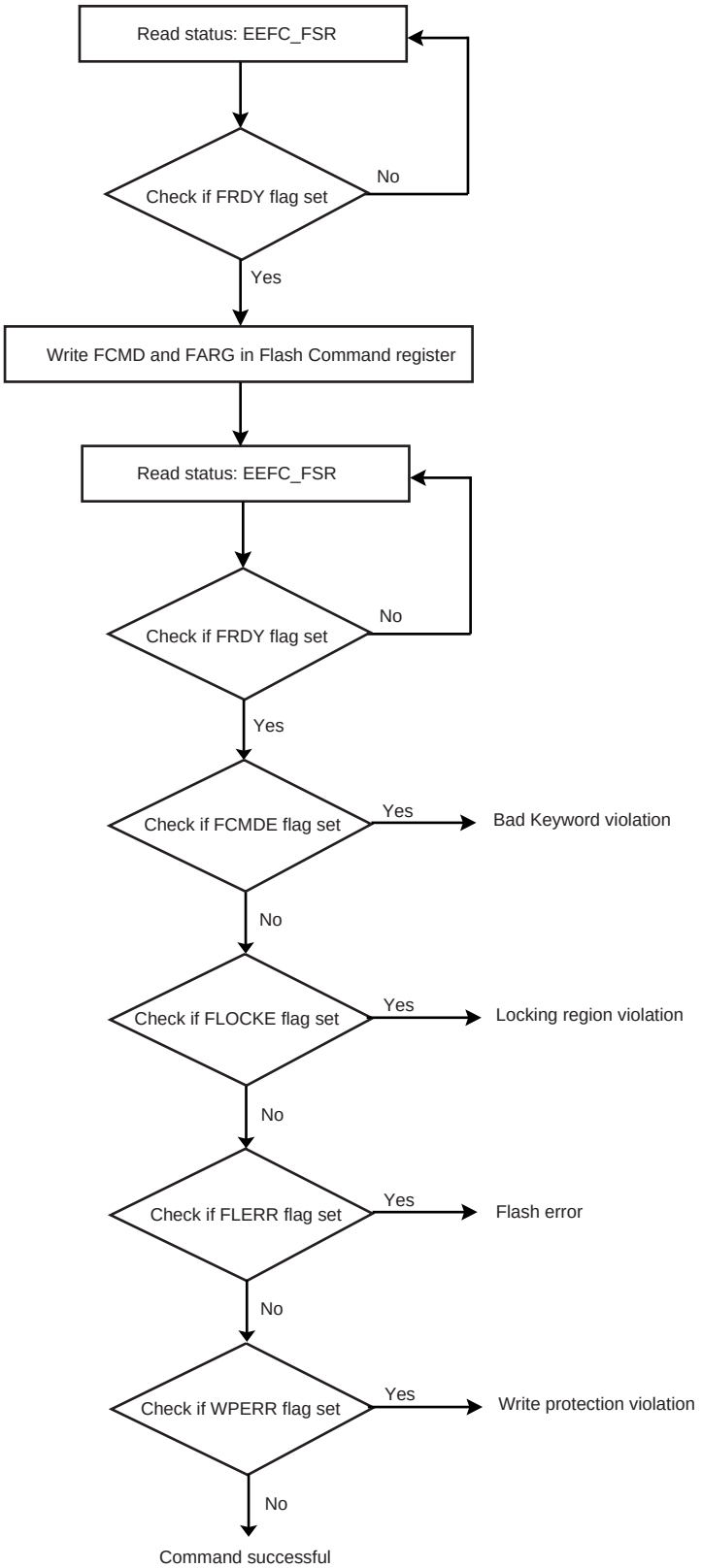
All the commands are protected by the same keyword, which must be written in the eight highest bits of EEFC_FCR.

Writing EEFC_FCR with data that does not contain the correct key and/or with an invalid command has no effect on the memory plane, but does set the FCMDE flag in the Flash Status register (EEFC_FSR). This flag is automatically cleared by a read access to EEFC_FSR.

When the current command writes page(s) or erases block(s) in a locked region, it has no effect on the memory plane but does set the FLOCKE flag in EEFC_FSR. This flag is automatically cleared by a read access to EEFC_FSR.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 29-7. Command State Chart



© 2026 Microchip Technology Inc. and its subsidiaries

29.4.3.1. Get Flash Descriptor Command

This command provides the system with information on the Flash organization. The system can take full advantage of this information. For instance, a device could be replaced by one with more Flash capacity, and so the software is able to adapt to the new configuration.

To get the embedded Flash descriptor, the application writes the GETD command in EEFC_FCR. The first word of the descriptor can be read by the software application in EEFC_FRR as soon as the FRDY flag in EEFC_FSR rises. The next reads of EEFC_FRR provide the following word of the descriptor. If extra read operations to EEFC_FRR are done after the last word of the descriptor was returned, the EEFC_FRR value is 0 until the next valid command.

Table 29-2. Flash Descriptor Definition

Symbol	Word Index	Description
FL_ID	0	Flash interface description
FL_SIZE	1	Flash size in bytes
FL_PAGE_SIZE	2	Page size in bytes
FL_NB_PLANE	3	Number of planes
FL_PLANE[0]	4	Number of bytes in the plane
FL_NB_LOCK	5	Number of lock bits. A bit is associated with a lock region. A lock bit is used to prevent write or erase operations in the lock region.
FL_LOCK[0]	6	Number of bytes in the first lock region

29.4.3.2. Write Commands

'Write Page' (WP) and 'Write Page and Lock' (WPL) commands are used to program the Flash main area.

Only 0 values can be programmed using the Flash technology; 1 is the erased value. In order to program words in a page, the corresponding 4K block that the page belongs to must be erased. Other commands are also available to erase by block, by multiple blocks or by sector.

After programming one or several 512-byte pages, the entire lock region containing these pages can be locked to prevent miscellaneous write or erase sequences. The lock bit can be automatically set after page programming using the WPL command.

Data to be programmed in the Flash must be written in an internal 512-byte latch buffer before writing the programming command in EEFC_FCR. Data can be written at their final destination address, as the latch buffer is mapped into the Flash memory address space and wraps around within this Flash address space.

Byte and half-word accesses to the latch buffer are not allowed. Only 32-bit word accesses are supported. If a single byte is to be written in a 32-bit word, the rest of the word must be written with ones.

32-bit words must be written continuously, in either ascending or descending order. Writing the latch buffer in a random order is not permitted. This prevents mapping a C-code structure to the latch buffer and accessing the data of the structure in any order. It is instead recommended to fill in a C-code structure in SRAM and copy it in the latch buffer in a continuous order.

Write operations in the latch buffer are performed with the number of wait states programmed for reading the Flash.

The latch buffer is automatically re-initialized, i.e., written with logical '1', after execution of each programming command. However, after power-up, the latch buffer is not initialized. If only part of the page is to be written with user data, the remaining part must be erased (written with '1').

The programming sequence is the following:

1. Write the data to be programmed in the latch buffer, up to 512 bytes.

© 2026 Microchip Technology Inc. and its subsidiaries

2. Write the programming command in EEFC_FCR. This automatically clears the bit EEFC_FSR.FRDY.
3. When Flash programming is completed, the bit EEFC_FSR.FRDY rises. If an interrupt was enabled by setting the bit EEFC_FMR.FRDY, the interrupt line of the SEFC is activated.

Three errors can be detected in EEFC_FSR after a programming sequence:

- Command error: a wrong keyword was written in EEFC_FCR.
- Lock error: the page to be programmed belongs to a locked region. A command must be run previously to unlock the corresponding region.
- Flash error: when programming is completed, the WriteVerify test of the Flash memory fails.

Only one page can be programmed at a time. It is possible to program all the bits of a page (full page programming) or only some of the bits of the page (partial page programming).

Depending on the number of bits to be programmed within the page, the SEFC adapts the write operations required to program the Flash.

When a 'Write Page' (WP) command is issued, the SEFC starts the programming sequence and all the bits written at 0 in the latch buffer are cleared in the Flash memory array.

During programming, i.e., until EEFC_FSR.FDRY rises, any access to the Flash is stalled until the programming completes. To avoid stalling the processor, the code can be run out of the internal SRAM.

If the Flash has to be read while a programming is active, the Write command can be suspended by writing the Suspend (SUSP) command in EEFC_FCR.

Consequently, EEFC_FSR.FDRY rises together with EEFC_FSR.FLSUSP, indicating the programming command is suspended.

Only Read accesses from the Flash are allowed during a suspended command.

The suspended programming command can be later resumed by writing a Resume command (RES) in EEFC_FCR.

If the SUSP command is issued late in the programming, the programming command may simply finish but not be suspended as indicated by EEFC_FSR.FDRY rising but EEFC_FSR.FLSUSP staying low.

If the same Write command is suspended then resumed multiple times, it is recommended that after the RES command, the next SUSP command should be given not earlier than 5 µs.

29.4.3.2.1. Full Page Programming

To program a full page, all the bits of the page must be erased before writing the latch buffer and issuing the WP command. The latch buffer must be written only in ascending or descending order. See Figure 29-8.

29.4.3.2.2. Partial Page Programming

To program only part of a page using the WP command, the following constraints apply:

- Data to be programmed must be contained in integer multiples of 64-bit address aligned words.
- 64-bit words can be programmed only if all the corresponding bits in the Flash array are erased (at logical value '1').

Attempting to write a new value in an already written 64-bit word will result in a corrupted word as seen by the ECC status flags after reading its content.

See Figure 29-9.

29.4.3.2.3. Optimized Partial Page Programming

The SEFC automatically detects the number of 128-bit words to be programmed. If only one 128-bit aligned word is to be programmed in the Flash array, the process is optimized to reduce the time needed for programming.

© 2026 Microchip Technology Inc. and its subsidiaries

If several 128-bit words are to be programmed, a standard page programming operation is performed.

See Figure 29-10.

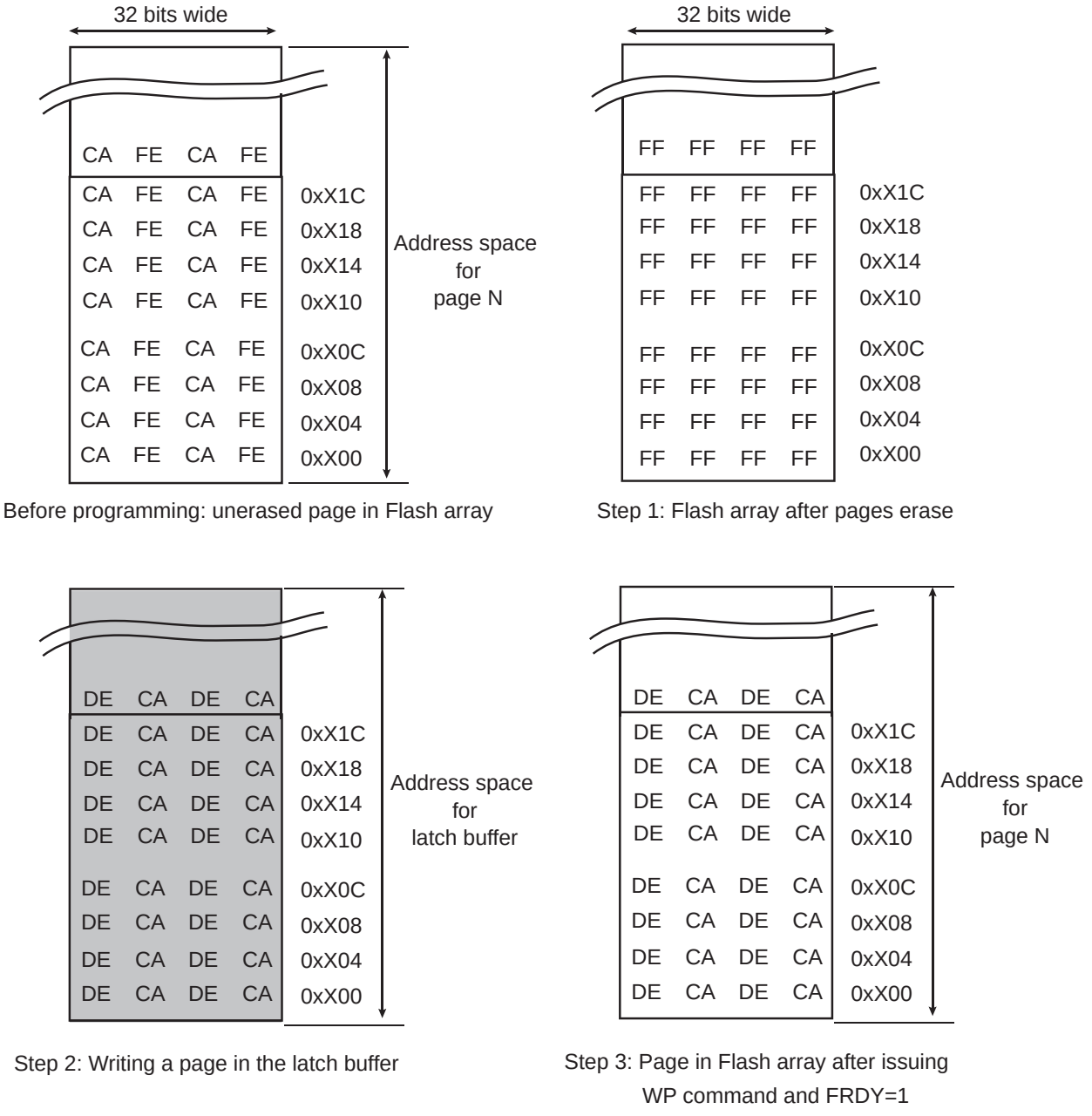
29.4.3.2.4. Programming Bytes

Individual bytes can be programmed using the Partial page programming mode.

In this case, an area of 64 bits must be reserved for each byte.

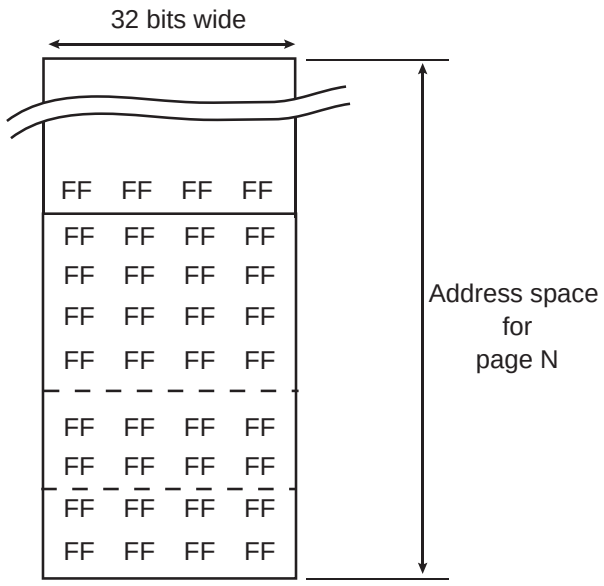
See Figure 29-11.

Figure 29-8. Full Page Programming

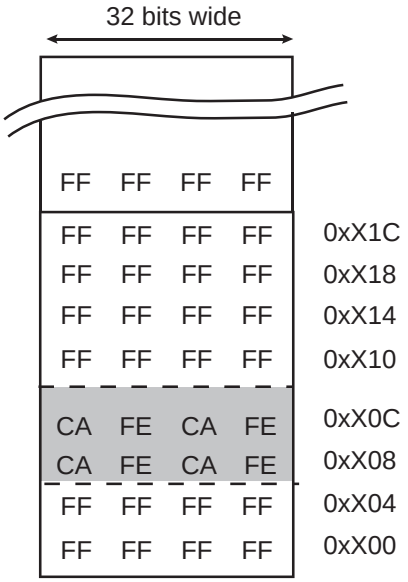


© 2026 Microchip Technology Inc. and its subsidiaries

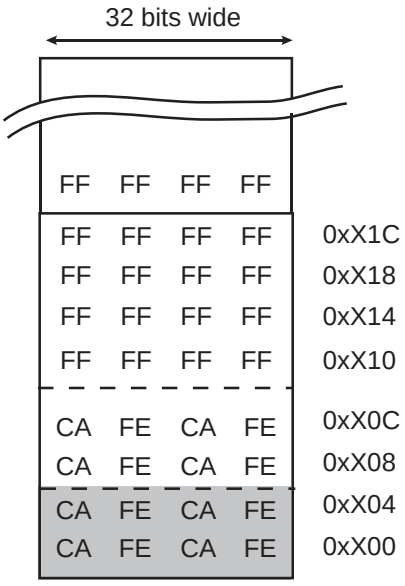
Figure 29-9. Partial Page Programming



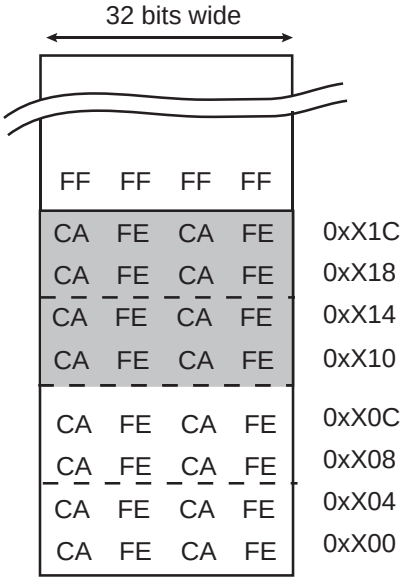
Step 1: Flash array after pages erase



Step 2: Flash array after programming
64-bit at address 0xX08 (write latch buffer + WP)



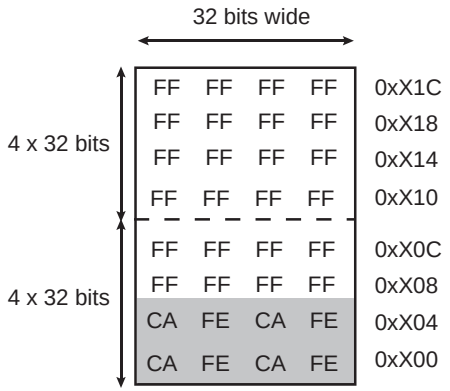
Step 3: Flash array after programming
a second 64-bit data at address 0xX00
(write latch buffer + WP)



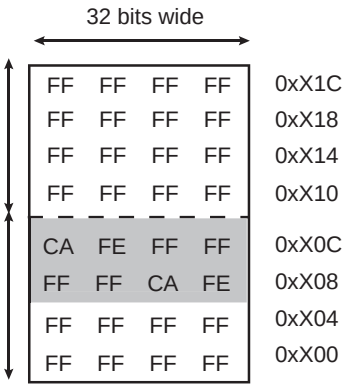
Step 4: Flash array after programming
a 128-bit data word at address 0xX10
(write latch buffer + WP)

© 2026 Microchip Technology Inc. and its subsidiaries

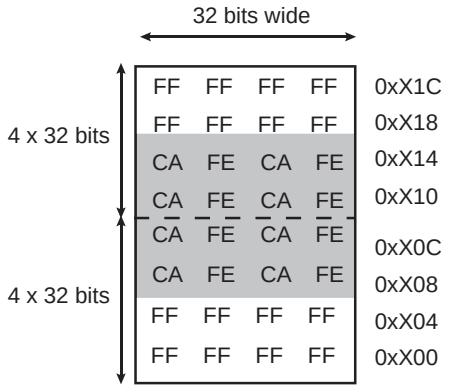
Figure 29-10. Optimized Partial Page Programming



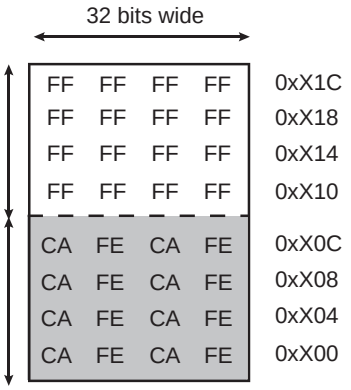
Case 1: 2 x 32 bits modified, not crossing 128-bit boundary
User programs WP, Flash Controller sends Write Word
=> Only 1 word programmed => programming period reduced



Case 2: 2 x 32 bits modified, not crossing 128-bit boundary
User programs WP, Flash Controller sends Write Word
=> Only 1 word programmed => programming period reduced



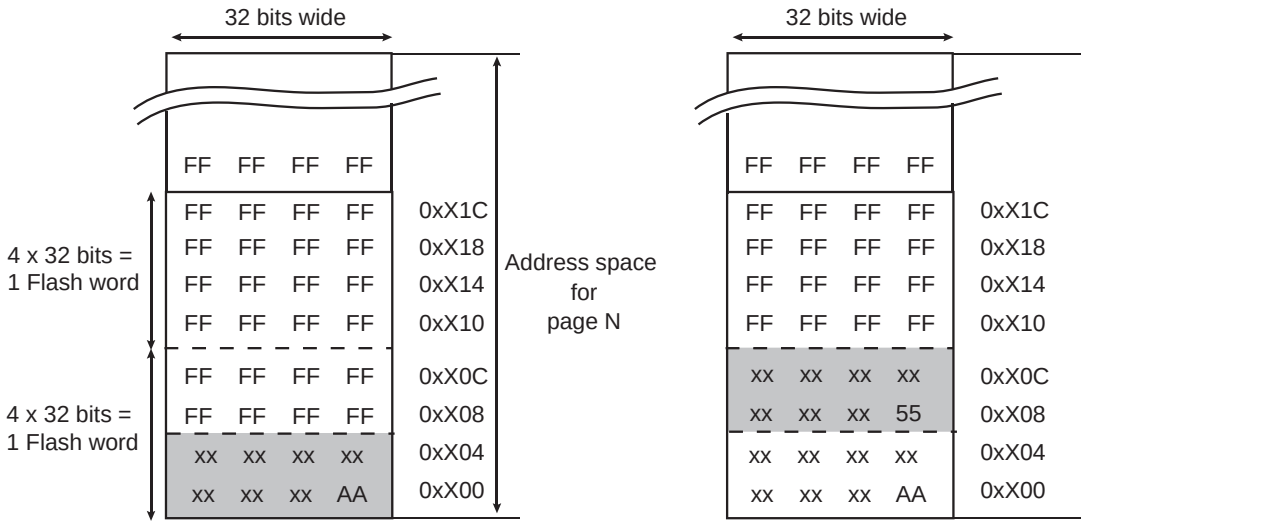
Case 3: 4 x 32 bits modified across 128-bit boundary
User programs WP, Flash Controller sends WP
=> Whole page programmed



Case 4: 4 x 32 bits modified, not crossing 128-bit boundary
User programs WP, Flash Controller sends Write Word
=> Only 1 word programmed => programming period reduced

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 29-11. Programming Bytes in the Flash



Step 1: Flash array after programming first byte (0xAA)
64-bit used at address 0xX00 (write latch buffer + WP)

Step 2: Flash array after programming second byte (0x55)
64-bit used at address 0xX08 (write latch buffer + WP)

Note: The byte location shown here is for example only, it can be any byte location within a 64-bit word.

29.4.3.3. Erase Commands

Erase commands are allowed only on unlocked regions of the Flash main area. Several commands can be used to erase the Flash:

- Erase Pages (EPA): 8, 16, 32 or 64 pages are erased in the Flash sector selected. The first page to be erased is specified in the EEFC_FCR.FARG[15:3] field. The first page number must be a multiple of 8, 16, 32 or 64, depending on the number of pages to erase at the same time.
- Erase Sector (ES): a full 128-Kbyte memory sector is erased. EEFC_FCR.FARG must be set with a page number that is in the sector to be erased.

If the processor is fetching code from the Flash memory while the EPA or ES command is being executed, the processor accesses are stalled until the EPA command is completed. To avoid stalling the processor, the code can be run out of the internal SRAM.

If the Flash has to be read while an EPA or ES command is performed, the Erase command can be suspended by writing the Suspend (SUSP) command in EEFC_FCR.

Consequently, EEFC_FSR.FDRY rises together with EEFC_FSR.FLSUSP, indicating the Erase command has been suspended.

Only Read accesses from the Flash are allowed during a suspended Erase command.

The suspended Erase command can be later resumed by writing a Resume command (RES) in EEFC_FCR.

If the SUSP command is issued late in the programming, the Erase command may simply finish but not be suspended as indicated by EEFC_FSR.FDRY rising but EEFC_FSR.FLSUSP staying low.

If the same Erase command is suspended then resumed multiple times, it is recommended that after RES command, the next SUSP command should be given not earlier than 1 ms.

The erase sequence is the following:

1. Erase starts as soon as one of the Erase commands and the FARG field are written in EEFC_FCR. For the EPA command, the two lowest bits of the FARG field define the number of pages to be erased (FARG[1:0]):

© 2026 Microchip Technology Inc. and its subsidiaries

Table 29-3. EEFC_FCR.FARG Field for EPA Command

FARG[2:0]	Number of pages to be erased with EPA command
1	8 pages
2	16 pages
3	32 pages
4	64 pages

2. When erasing is completed, the bit EEFC_FSR.FRDY rises. If an interrupt was enabled by setting the bit EEFC_FMR.FRDY, the interrupt line of the interrupt controller is activated.

Three errors can be detected in EEFC_FSR after an erasing sequence:

- Command error: a wrong keyword was written in EEFC_FCR.
- Lock error: at least one page to be erased belongs to a locked region. The Erase command has been refused, no page has been erased. A command must be run previously to unlock the corresponding region.
- Flash error: at the end of the erase period, the EraseVerify test of the Flash memory fails.

29.4.3.4. Lock Bit Protection

The Flash memory features 128 lock bits.

Lock bits are associated with several pages in the embedded Flash memory plane. This defines lock regions in the embedded Flash memory plane. They prevent writing/erasing protected pages.

The lock sequence is the following:

1. Execute the 'Set Lock Bit' command by writing EEFC_FCR.FCMD with the SLB command and EEFC_FCR.FARG with a page number to be protected.
2. When the locking completes, the bit EEFC_FSR.FRDY rises. If an interrupt was enabled by setting the bit EEFC_FMR.FRDY, the interrupt line of the interrupt controller is activated.
3. The result of the SLB command can be checked by running a 'Get Lock Bit' (GLB) command.
   Note: The value of the FARG argument passed together with the SLB command must not exceed the highest lock bit index available in the product.

Three errors can be detected in EEFC_FSR after a programming sequence:

- Command error: a wrong keyword was written in EEFC_FCR.
- Flash error: at the end of the programming, the EraseVerify or WriteVerify test of the Flash memory has failed.
- ECC error on lock bits: a single or multiple ECC error is present, as indicated by MECCEMSBL, SECCEMSBL, MECCELSBL or SECCELSBL.

It is possible to clear lock bits previously set. After the lock bits are cleared, the locked region can be erased or programmed. The unlock sequence is the following:

1. Execute the 'Clear Lock Bit' command by writing EEFC_FCR.FCMD with the CLB command and EEFC_FCR.FARG with a page number to be unprotected.
2. When the unlock completes, the bit EEFC_FSR.FRDY rises. If an interrupt was enabled by setting the bit EEFC_FMR.FRDY, the interrupt line of the interrupt controller is activated.
   Note: The value of the FARG argument passed together with the CLB command must not exceed the highest lock bit index available in the product.

Three errors can be detected in EEFC_FSR after a programming sequence:

- Command error: a wrong keyword was written in EEFC_FCR.
- Flash error: at the end of the programming, the EraseVerify or WriteVerify test of the Flash memory has failed.

© 2026 Microchip Technology Inc. and its subsidiaries

- ECC error on lock bits: a single or multiple ECC error is present, as indicated by MECCEMSBL, SECCEMSBL, MECCELSBL or SECCELSBL.

The status of lock bits can be returned by the SEFC. The 'Get Lock Bit' sequence is the following:

1. Execute the 'Get Lock Bit' command by writing EEFC_FCR.FCMD with the GLB command. Field EEFC_FCR.FARG is meaningless.
2. Lock bits can be read by the software application in EEFC_FRR. The first word read corresponds to the first 32 lock bits; the next reads provide the next 32 lock bits as long as it is meaningful. Extra reads to EEFC_FRR return 0.

For example, if the third bit of the first word read in EEFC_FRR is set, the third lock region is locked.

One error can be detected in EEFC_FSR after a programming sequence:

- Command error: a wrong keyword was written in EEFC_FCR.
  Note: Access to the Flash in read is permitted when a 'Set Lock Bit', 'Clear Lock Bit' or 'Get Lock Bit' command is executed, but the access is stalled until the command is completed.

29.4.4. GPNVM Bit

The Flash memory features 9 General Purpose Non Volatile Memory bits.

GPNVM bits do not interfere with the embedded Flash memory plane. For more details, refer to the section "Memories".

The 'Set GPNVM Bit' sequence is the following:

1. Execute the 'Set GPNVM Bit' command by writing EEFC_FCR.FCMD with the SGPB command and EEFC_FCR.FARG with the number of GPNVM bits to be set.
2. When the GPNVM bit is set, the bit EEFC_FSR.FRDY rises. If an interrupt was enabled by setting the bit EEFC_FMR.FRDY, the interrupt line of the interrupt controller is activated.
3. The result of the SGPB command can be checked by running a 'Get GPNVM Bit' (GGPB) command.
   Note: The value of the FARG argument passed together with SGPB command must not exceed the higher GPNVM index available in the product. Flash data content is not altered if FARG exceeds the limit. Command Error is detected only if FARG is greater than 8.

Two errors can be detected in EEFC_FSR after a programming sequence:

- Command error: a wrong keyword was written in EEFC_FCR.
- Flash error: at the end of the programming, the EraseVerify or WriteVerify test of the Flash memory has failed.

It is possible to clear GPNVM bits previously set. The 'Clear GPNVM Bit' sequence is the following:

1. Execute the 'Clear GPNVM Bit' command by writing EEFC_FCR.FCMD with the CGPB command and EEFC_FCR.FARG with the number of GPNVM bits to be cleared.
2. When the clear completes, the bit EEFC_FSR.FRDY rises. If an interrupt was enabled by setting the bit EEFC_FMR.FRDY, the interrupt line of the interrupt controller is activated.
   Note: The value of the FARG argument passed together with the CGPB command must not exceed the highest GPNVM index available in the product. Flash data content is not altered if FARG exceeds the limit. A command error is detected only if FARG is greater than 8.

© 2026 Microchip Technology Inc. and its subsidiaries

Important: GPNVM bits are set and cleared on a bit-by-bit basis and extreme caution must be used during the configuration sequence to avoid triggering undesired operations. As an example, if the bit-by-bit setting/clearing of GPNVM[4:2] results in the sequence '110' at any time during configuration, the EFL mode is set to "HW Erase function disabled - EFL bits are read-only". This mode permanently disables the Hardware Erase signal, the Flash can no longer be erased, and the operation is irreversible.
For a full description of GPNVM bit settings and functions, see the table "GPNVM Bits Overview" in the section "ROM Code and Boot Strategies".

Two errors can be detected in EEFC_FSR after a programming sequence:

- Command error: a wrong keyword was written in EEFC_FCR.
- Flash error: at the end of the programming, the EraseVerify or WriteVerify test of the Flash memory has failed.

The status of the GPNVM bits can be returned by the SEFC. The sequence is the following:

1. Execute the ‘Get GPNVM Bit’ command by writing EEFC_FCR.FCMD with the GGPB command. Field EEFC_FCR.FARG is meaningless.
2. GPNVM bits can be read by the software application in EEFC_FRR. The first word read corresponds to the first 32 GPNVM bits; the following reads provide the next 32 GPNVM bits as long as it is meaningful. Extra reads to EEFC_FRR return 0.

For example, if the third bit of the first word read in EEFC_FRR is set, the third GPNVM bit is active.

One error can be detected in EEFC_FSR after a programming sequence:

- Command error: a wrong keyword was written in EEFC_FCR.
  Note: Access to the Flash in read is permitted when a ‘Set GPNVM Bit’, ‘Clear GPNVM Bit’ or ‘Get GPNVM Bit’ command is executed, but the access is stalled until the command is completed.

29.4.5. Calibration Bits

Calibration bits do not interfere with the embedded Flash memory plane.

The calibration bits cannot be modified.

The status of calibration bits is returned by the SEFC. The sequence is the following:

1. Execute the ‘Get CALIB Bit’ command by writing EEFC_FCR.FCMD with the GCALB command. Field EEFC_FCR.FARG is meaningless.
2. Calibration bits can be read by the software application in EEFC_FRR. The first word read corresponds to the first 32 calibration bits. The following reads provide the next 32 calibration bits as long as it is meaningful. Extra reads to EEFC_FRR return 0.

The 12-MHz internal RC oscillator is calibrated in production. This calibration can be read through the GCALB command. The following table shows the bit implementation.

Table 29-4. Calibration Bit Indexes

Description	EEFC_FRR Bits
12-MHz RC calibration output	[38-32]

29.4.6. Security Bit Protection

When the security bit is enabled, any access to the Flash either through the Serial Wire Debug Port (SW-DP) or the Serial Wire JTAG Debug Port (SWJ-DP) or the Trace interface or the Fast Flash Programming Interface, is denied. This ensures the confidentiality of the code programmed in the Flash.

© 2026 Microchip Technology Inc. and its subsidiaries

The security bit is GPNVM0.

Disabling the security bit can only be achieved by asserting the ERASE pin at '1', and after a full Flash erase is performed. When the security bit is disabled, all accesses to the Flash are permitted.

29.4.7. Erase Function Lock

The Flash Erase Pin function can be locked, by programming the GPNVM[4:2], to prevent the device from being erased. The Erase Function Lock (EFL) is managed at application level.

By default the Erase pin is active. These GPNVM bits can be set to 0 or 1 by software, they apply only to the Erase Pin. These bits do not prevent software Erase commands.

Table 29-5. Erase Function Lock

Description	GPNVM[4:2] Bits
Not locked	000
Erase Function Locked modifiable by software	011
Erase Function Locked not modifiable	110
Erase Function Locked modifiable by software	Others

When the EFL is activated and the Security bit enabled, the device cannot be recovered. This function must be carefully used and should be validated at the very end of the production process.

29.4.8. Unique Identifier Area

Each device is programmed with a 4-Kbyte unique identifier area. See Figure 29-1.

The sequence to read the unique identifier area is the following:

1. Perform a dummy read at the Flash base address. Execute the 'Start Read Unique Identifier' command by writing EEFC_FCR.FCMD with the STUI command. EEFC_FCR.FARG is meaningless.
2. Wait until the bit EEFC_FSR.FRDY falls to read the unique identifier area. The unique identifier field is located in the first 128 bits of the Flash memory mapping. The 'Start Read Unique Identifier' command reuses some addresses of the main memory plane for code, but the unique identifier area is physically different from the main memory plane for code.
3. To stop reading the unique identifier area, execute the 'Stop Read Unique Identifier' command by writing EEFC_FCR.FCMD with the SPUI command. Field EEFC_FCR.FARG is meaningless.
4. When the SPUI command has been executed, EEFC_FSR.FRDY rises. If an interrupt was enabled by setting EEFC_FMR.FRDY, the interrupt line of the interrupt controller is activated.

Note that during the sequence, the software cannot be fetched from the Flash.

29.4.9. User Signature Area

Each product contains a user signature area of 32 Kbytes (8 x 4K) in one block. It can be used for storage. Read, write and erase of this area are allowed.

Only the first half of the signature area is erased by the erase pin assertion.

When reading from a user signature block, its Read and User/Privileged access rights must be honored as configured in the User Signature Rights register (EEFC_USR). Otherwise the read access returns a bus error which can be managed either as a synchronous abort exception for a CPU access or asynchronously as a security interrupt coming from the AHB matrix connected to the SEFC.

When writing the command to the Flash Command Register to program or erase a user signature block, its Write and User / Privileged access rights must be honored as configured in the User Signature Rights Register (EEFC_USR). Otherwise the command is ignored and a write protect violation error (WPERR) is flagged in EEFC_FSR.

© 2026 Microchip Technology Inc. and its subsidiaries

The Access Rights can be locked per each user signature block x until hardware reset by writing the corresponding LOCKUSRBx bit to 1.

See Figure 29-1.

The sequence to read the user signature area is the following:

1. Execute the 'Start Read User Signature' command by writing EEFC_FCR.FCMD with the STUS command. Field EEFC_FCR.FARG is meaningless.
2. Wait until the bit EEFC_FSR.FRDY falls to read the user signature area. The user signature area is located in the first 32 Kbytes of the Flash memory mapping. The 'Start Read User Signature' command reuses some addresses of the memory plane but the user signature area is physically different from the memory plane
3. To stop reading the user signature area, execute the 'Stop Read User Signature' command by writing EEFC_FCR.FCMD with the SPUS command. Field EEFC_FCR.FARG is meaningless.
4. When the SPUI command has been executed, the bit EEFC_FSR.FRDY rises. If an interrupt was enabled by setting the bit EEFC_FMR.FRDY, the interrupt line of the interrupt controller is activated.

Note that during the sequence, the software cannot be fetched from the Flash main memory area.

One error can be detected in EEFC_FSR after this sequence:

- Command error: a wrong keyword was written in EEFC_FCR.

The sequence to write the user signature area is the following:

1. Write the full page, at any page address, in the internal memory area address space.
2. Execute the 'Write User Signature' command by writing EEFC_FCR.FCMD with the WUS command. The page to be written is specified in the FARG field of the EEFC_FCR.
3. When programming is completed, the bit EEFC_FSR.FRDY rises. If an interrupt was enabled by setting the bit EEFC_FMR.FRDY, the corresponding interrupt line of the interrupt controller is activated.

Two errors can be detected in EEFC_FSR after this sequence:

- Command error: a wrong keyword was written in EEFC_FCR.
- Flash error: at the end of the programming, the WriteVerify test of the Flash memory failed.

The sequence to erase a user signature block is the following:

1. Execute the 'Erase User Signature' command by writing EEFC_FCR.FCMD with the EUS command. The block to be erased is specified in the EEFC_FCR.FARG[15:3] field.
2. When programming is completed, the bit EEFC_FSR.FRDY rises. If an interrupt was enabled by setting the bit EEFC_FMR.FRDY, the corresponding interrupt line of the interrupt controller is activated.

Two errors can be detected in EEFC_FSR after this sequence:

- Command error: a wrong keyword was written in EEFC_FCR.
- Flash error: at the end of the programming, the EraseVerify test of the Flash memory has failed.

The 4-Kbyte user signature block 7 is One Time Programmable only and cannot be erased by any mean. In particular, this means the 'Erase User Signature' command cannot be used with the FARG[15:3] field set to 7. Any attempt will have no effect but raise the Write Protection Error status flag (WPERR) in EEFC_FSR.

The One Time Programmable (OTP) block can be used for example to implement a monotonic counter in software, with 64 bits a least being written at a time as explained in Partial Page Programming.

© 2026 Microchip Technology Inc. and its subsidiaries

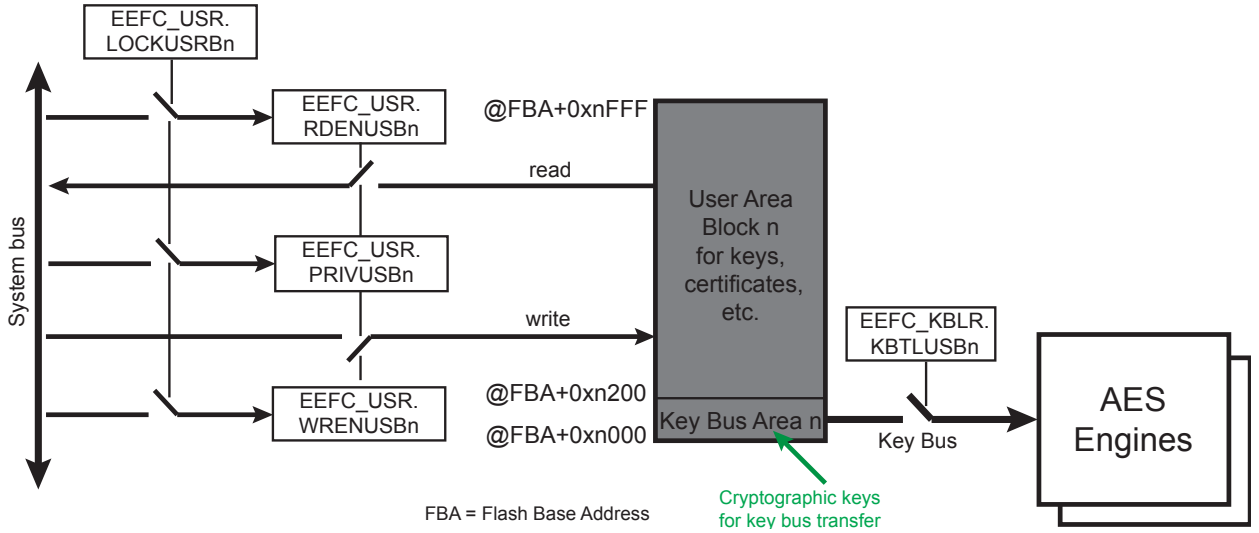
29.4.10. Cryptographic Key Bus

The first 512-byte page of each user signature block, including the OTP block, can be used to store cryptographic keys to be transferred through the key bus to the AES engine.

If the cryptographic key is stored in a user signature area to which read and write accesses have been disabled and locked until a hardware reset as programmed in EEFC_USR, then the CPU has no direct access to the cryptographic key. However, the CPU can still command the transfer of this key from the user signature area to the private key register of the AES or of the AESB engine through the key bus.

Transfer through the key bus can also be disabled until hardware reset and per each user signature block x by writing a 1 in the bit KBTLUSBx of the SEFC Key Bus Lock register (EEFC_KBLR).

Figure 29-12. User Areas and Key Bus Protections



The sequence to transfer a cryptographic key to the AES engine through the key bus is the following:

1. Execute the 'Send Cryptographic Key' command by writing EEFC_FCR.FCMD with the SCK value. At the same time, write the selected destination (AES or AESB), the source user signature block, the start address offset in the first page of this block and the length of the key in EEFC_FCR.FARG.
2. When transfer is completed, the bit EEFC_FSR.FRDY rises. If an interrupt was enabled by setting the bit EEFC_FMR.FRDY, the corresponding interrupt line of the interrupt controller is activated.

Two errors can be detected in EEFC_FSR after this sequence:

- Command error: a wrong keyword was written in EEFC_FCR.
- Write Protect error: the transfer from this user signature block is locked in the SEFC Key Bus Lock register.
  Note: The cryptographic key to be transferred must fit entirely into the first page of the selected user signature block, otherwise a Command error is flagged.
  Access to the Flash in read is permitted when a SCK command is executed.

Also check the AES engine Write Protection Status register for any key bus transfer error.

29.4.11. Programming Verify Functions

The Control Logic within the Embedded Flash Block features a self-check mechanism.

In functional programming mode, erase and write operations on either the GPNVM bits, lock bits or main memory array, end with verifications called respectively EraseVerify and WriteVerify.

© 2026 Microchip Technology Inc. and its subsidiaries

Verification processes are based on a “programming margin”. Programming here means both Erase or Write operations.

The FLERR status flag coming from the Flash cell is reported to the user through the Status register.

It is strongly advised to monitor the FLERR bit and all other error bits at application driver level.

29.4.12. ECC Errors and Corrections

The Flash embeds an ECC module able to correct one unique error and to detect two errors. The errors are detected while a read access is performed into memory array and stored in EEFC_FSR (see EEFC_FSR). The error report is kept until EEFC_FSR is read.

There is one flag for a unique error on lower half part of the Flash word (64 LSB) and one flag for the upper half part (MSB). Multiple errors are reported in the same way.

Due to the anticipation technique to improve bandwidth throughput on instruction fetch, a reported error can be located in the next sequential Flash word compared to the location of the instruction being executed, which is located in the previously fetched Flash word.

If a software routine processes the error detection independently from the main software routine, the entire Flash located software must be rewritten because there is no storage of the error location.

If only a software routine is running to program and check pages by reading EEFC_FSR, the situation differs from the previous case. Performing a check for ECC unique errors just after page programming completion involves a read of the newly programmed page. This read sequence is viewed as data accesses and is not optimized by the Flash controller. Thus, in case of unique error, only the current page must be reprogrammed.

A set of four similar ECC flags is available for the lock bits protection in EEFC_FSR.

Two Single Error Correction and two Multiple Error Detection flags protect respectively the 64 MSBs and 64 LSBs of the lock bits.

29.4.13. Register Write Protection

To prevent any single software error from corrupting SEFC behavior, certain registers in the address space can be write-protected.

The following registers can be write-protected by setting the WPEN bit in EEFC_WPMR:

- EEFC_FMR

The following registers can be write-protected by setting the USRWP bit in EEFC_WPMR:

- EEFC_USR
- EEFC_KBLR

29.4.14. Command Write Protection

To prevent any single software error from corrupting Flash content, lock bits or GPNVM bits, write and/or erase commands can be write-protected by setting the GPNVMWP, LOCKWP, ERASEWP and ERASEWL bits in EEFC_WPMR.

The GPNVM modification commands have no effect when EEFC_WPMR.GPNVMWP is set. Lock bit modifications are not possible when EEFC_WPMR.LOCKWP is set. Write and erase of pages, blocks or sectors are not possible when EEFC_WPMR.ERASEWP is set. Furthermore, these write and erase of pages, sectors or planes are not possible until hardware reset when EEFC_WPMR.ERASEWL is set.

© 2026 Microchip Technology Inc. and its subsidiaries

29.5. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	EEFC_FMR	31:24					ALWAYS1	CLOE		
		23:16								SCOD
		15:8					FWS[3:0]			
		7:0								FRDY
0x04	EEFC_FCR	31:24	FKEY[7:0]							
		23:16	FARG[15:8]							
		15:8	FARG[7:0]							
		7:0	FCMD[7:0]							
0x08	EEFC_FSR	31:24								
		23:16	MECCEMSBL	SECCEMSBL	MECCELSBL	SECCELSBL	MECCEMSBD	SECCEMSBD	MECCELSBD	SECCELSBD
		15:8								
		7:0			FLSUSP	WPERR	FLERR	FLOCKE	FCMDE	FRDY
0x0C	EEFC_FRR	31:24	FVALUE[31:24]							
		23:16	FVALUE[23:16]							
		15:8	FVALUE[15:8]							
		7:0	FVALUE[7:0]							
0x10	EEFC_USR	31:24	LOCKUSRB7	LOCKUSRB6	LOCKUSRB5	LOCKUSRB4	LOCKUSRB3	LOCKUSRB2	LOCKUSRB1	LOCKUSRB0
		23:16	PRIVUSB7	PRIVUSB6	PRIVUSB5	PRIVUSB4	PRIVUSB3	PRIVUSB2	PRIVUSB1	PRIVUSB0
		15:8	WRENUSB7	WRENUSB6	WRENUSB5	WRENUSB4	WRENUSB3	WRENUSB2	WRENUSB1	WRENUSB0
		7:0	RDENUSB7	RDENUSB6	RDENUSB5	RDENUSB4	RDENUSB3	RDENUSB2	RDENUSB1	RDENUSB0
0x14	EEFC_KBLR	31:24								
		23:16								
		15:8								
		7:0	KBTLUSB7	KBTLUSB6	KBTLUSB5	KBTLUSB4	KBTLUSB3	KBTLUSB2	KBTLUSB1	KBTLUSB0
0x18 ... 0xE3	Reserved									
0xE4	EEFC_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0	ERASEWL			USRWP	ERASEWP	LOCKWP	GPNVMWP	WPEN

© 2026 Microchip Technology Inc. and its subsidiaries

29.5.1. SEFC Flash Mode Register

Name: EEFC_FMR
Offset: 0x00
Reset: 0x0C010F00
Property: Read/Write

This register can only be written if the WPEN bit is cleared in EEFC_WPMR.

Bit	31	30	29	28	27	26	25	24
					ALWAYS1	CLOE		
Access					R/W	R/W		
Reset					1	1		

Bit	23	22	21	20	19	18	17	16
								SCOD
Access								R/W
Reset								1

Bit	15	14	13	12	11	10	9	8
					FWS[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					1	1	1	1

Bit	7	6	5	4	3	2	1	0
								FRDY
Access								R/W
Reset								0

Bit 27 – ALWAYS1 Always Written to One
This bit must be always written to 1.

Bit 26 – CLOE Code Loop Optimization Enable
No Flash read should be done during change of this field.

Value	Description
0	The opcode loop optimization is disabled.
1	The opcode loop optimization is enabled if the bit SCOD is reset to 0.

Bit 16 – SCOD Sequential Code Optimization Disable
No Flash read should be done during change of this field.

Value	Description
0	The sequential code optimization is enabled.
1	The sequential code optimization is disabled.

Bits 11:8 – FWS[3:0] Flash Wait State
This field defines the number of wait states for read and write operations:
FWS = Number of cycles for Read/Write operations - 1
Note: At hardware reset, this field is set to maximum value 15 to support any clock frequency in the available range.

Bit 0 – FRDY Flash Ready Interrupt Enable

Value	Description
0	Flash ready (to accept a new command) does not generate an interrupt.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
1	Flash ready (to accept a new command) generates an interrupt.

© 2026 Microchip Technology Inc. and its subsidiaries

29.5.2. SEFC Flash Command Register

Name: EEFC_FCR
Offset: 0x04
Reset: –
Property: Write-only

This register can only be written if the FKEY field is written to 0x5A.

Bit	31	30	29	28	27	26	25	24
	FKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	FARG[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	FARG[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	FCMD[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:24 – FKEY[7:0] Flash Writing Protection Key

Value	Name	Description
0x5A	PASSWD	The 0x5A value enables the command defined by the bits in the register. If the field is written with a different value, the write is not performed and no action is started.

Bits 23:8 – FARG[15:0] Flash Command Argument

GETD, GLB, GGPB, STUI, SPUI, GCALB, STUS, SPUS, SUSP, RES	Commands requiring no argument	FARG is meaningless, must be written with 0
ES	Erase sector command	FARG must be written with any page number within the sector to be erased

© 2026 Microchip Technology Inc. and its subsidiaries

EPA	Erase pages command	FARG[2:0] defines the number of pages to be erased The start page number divided by 8 must be written in FARG[15:3]. FARG[2:0] = 0: Reserved, FARG[2:0] = 1: Eight pages to be erased. FARG[15:3] = Page_Number/8, FARG[2:0] = 2: Sixteen pages to be erased. FARG[15:4] = Page_Number/16 FARG[3] = 0 FARG[2:0] = 3: Thirty-two pages to be erased. FARG[15:5] = Page_Number/32 FARG[4:3] = 0 FARG[2:0] = 4: Sixty-four pages to be erased. FARG[15:6] = Page_Number/64 FARG[5:3] = 0 FARG[2:0] > 4: Reserved. See Table 29-3.
EUS	Erase User Signature command	The start page number divided by 8 must be written in FARG[15:3]. FARG[2:0] = 0: Reserved
WP, WPL, WUS	Programming commands	FARG must be written with the page number to be programmed
SLB, CLB	Lock bit commands	FARG defines the page number to be locked or unlocked
SGPB, CGPB	GPNVM commands	FARG defines the GPNVM number to be programmed
SCK	Send cryptographic key to AES engine command	FARG[15]: Reserved FARG[14]: '0' if AES is selected, '1' if AESB is selected FARG[13]: Reserved FARG[12] defines the protection level attribute forwarded to the key bus during cryptographic key transfer. 0=User, 1=Privileged FARG[11:10]: Reserved FARG[9:8] defines the cryptographic key length to be transferred: 1 (128 bits), 2 (192 bits) or 3 (256 bits). FARG[7:5] defines the 4-Kbyte user signature block number where the cryptographic key is to be retrieved, from 0 to 7 FARG[4:0] defines the 128-bit flash word address offset in the first page of the user signature block where the cryptographic key is to be retrieved

Bits 7:0 – FCMD[7:0] Flash Command

Value	Name	Description
0x00	GETD	Get Flash descriptor
0x01	WP	Write page
0x02	WPL	Write page and lock
0x07	EPA	Erase pages (8, 16, 32, 64)
0x08	SLB	Set lock bit
0x09	CLB	Clear lock bit
0x0A	GLB	Get lock bit
0x0B	SGPB	Set GPNVM bit
0x0C	CGPB	Clear GPNVM bit
0x0D	GGPB	Get GPNVM bit
0x0E	STUI	Start read unique identifier
0x0F	SPUI	Stop read unique identifier
0x10	GCALB	Get CALIB bit
0x11	ES	Erase sector
0x12	WUS	Write user signature

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Name	Description
0x13	EUS	Erase user signature
0x14	STUS	Start read user signature
0x15	SPUS	Stop read user signature
0x17	SUSP	Suspend
0x18	RES	Resume
0x19	SCK	Send cryptographic key

© 2026 Microchip Technology Inc. and its subsidiaries

29.5.3. SEFC Flash Status Register

Name: EEFC_FSR
Offset: 0x08
Reset: 0x00000001
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	MECCEMSBL	SECCEMSBL	MECCELSBL	SECCELSBL	MECCEMSBD	SECCEMSBD	MECCELSBD	SECCELSBD
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			FLSUSP	WPERR	FLERR	FLOCKE	FCMDE	FRDY
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	1

Bit 23 – MECCEMSBL Multiple ECC Error on MSB Part of the Memory Lock Bits

Value	Description
0	No multiple error detected on 64 MSB of the Flash memory lock bits since the last read of EEFC_FSR
1	Multiple errors detected and NOT corrected on 64 MSB of the Flash memory lock bits since the last read of EEFC_FSR

Bit 22 – SECCEMSBL Single ECC Error on MSB Part of the Memory Lock Bits

Value	Description
0	No single error detected on 64 MSB lock bits of the Flash memory since the last read of EEFC_FSR
1	Single error detected but corrected on 64 MSB of the Flash memory lock bits since the last read of EEFC_FSR

Bit 21 – MECCELSBL Multiple ECC Error on LSB Part of the Memory Lock Bits

Value	Description
0	No multiple error detected on 64 LSB of the Flash memory Lock bits since the last read of EEFC_FSR.
1	Multiple errors detected and NOT corrected on 64 LSB of the Flash memory Lock bits since the last read of EEFC_FSR.

Bit 20 – SECCELSBL Single ECC Error on LSB Part of the Memory Lock Bits

Value	Description
0	No single error detected on 64 LSB of the Flash memory lock bits since the last read of EEFC_FSR
1	Single error detected but corrected on 64 LSB of the Flash memory lock bits since the last read of EEFC_FSR

Bit 19 – MECCEMSBD Multiple ECC Error on MSB Part of the Memory Flash Data Bus (cleared on read)

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	No multiple error detected on 64 MSB of the Flash memory data bus since the last read of EEFC_FSR
1	Multiple errors detected and NOT corrected on 64 MSB of the Flash memory data bus since the last read of EEFC_FSR

Bit 18 – SECCEMSBD Single ECC Error on MSB Part of the Memory Flash Data Bus (cleared on read)

Value	Description
0	No single error detected on 64 MSB of the Flash memory data bus since the last read of EEFC_FSR
1	Single error detected but corrected on 64 MSB of the Flash memory data bus since the last read of EEFC_FSR

Bit 17 – MECCELSBD Multiple ECC Error on LSB Part of the Memory Flash Data Bus (cleared on read)

Value	Description
0	No multiple error detected on 64 LSB of the Flash memory data bus since the last read of EEFC_FSR
1	Multiple errors detected and NOT corrected on 64 LSB of the Flash memory data bus since the last read of EEFC_FSR

Bit 16 – SECCELSBD Single ECC Error on LSB Part of the Memory Flash Data Bus (cleared on read)

Value	Description
0	No single error detected on 64 LSB of the Flash memory data bus since the last read of EEFC_FSR
1	Single error detected but corrected on 64 LSB of the Flash memory data bus since the last read of EEFC_FSR

Bit 5 – FLSUSP Flash Suspended Status (cleared when resuming the programming operation)

Value	Description
0	No Flash command is suspended.
1	The current Flash command is suspended.

Bit 4 – WPERR Write Protection Error Status (cleared on read)

Value	Description
0	No write protection violation occurred since the last read of EEFC_FSR.
1	A write protection violation occurred since the last read of EEFC_FSR.

Bit 3 – FLERR Flash Error Status (cleared when a programming operation starts)

Value	Description
0	No Flash memory error occurred at the end of programming (EraseVerify or WriteVerify test has passed).
1	A Flash memory error occurred at the end of programming (EraseVerify or WriteVerify test has failed).

Bit 2 – FLOCKE Flash Lock Error Status (cleared on read or by writing EEFC_FCR)

Value	Description
0	No programming/erase of at least one locked region has happened since the last read of EEFC_FSR.
1	Programming/erase of at least one locked region has happened since the last read of EEFC_FSR.

Bit 1 – FCMDE Flash Command Error Status (cleared on read)

Value	Description
0	No invalid commands and no wrong keywords were written in EEFC_FMR.
1	An invalid command and/or a wrong keyword was/were written in EEFC_FMR.

Bit 0 – FRDY Flash Ready Status (cleared when Flash is busy)
When set, this flag triggers an interrupt if the FRDY flag is set in EEFC_FMR. It is automatically cleared when the SEFC is busy.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	The SEFC is busy.
1	The SEFC is ready to start a new command.

© 2026 Microchip Technology Inc. and its subsidiaries

29.5.4. SEFC Flash Result Register

Name: EEFC_FRR
Offset: 0x0C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	FVALUE[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	FVALUE[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	FVALUE[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FVALUE[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – FVALUE[31:0] Flash Result Value
The result of a Flash command is returned in this register. If the size of the result is greater than 32 bits, the next resulting value is accessible at the next register read.

© 2026 Microchip Technology Inc. and its subsidiaries

29.5.5. SEFC User Signature Rights Register

Name: EEFC_USR
Offset: 0x10
Reset: 0x00000000
Property: Read/Write

This register can only be written if the USRWP bit is cleared in EEFC_WPMR.

This register can only be accessed in Privileged mode.

Bit	31	30	29	28	27	26	25	24
	LOCKUSRB7	LOCKUSRB6	LOCKUSRB5	LOCKUSRB4	LOCKUSRB3	LOCKUSRB2	LOCKUSRB1	LOCKUSRB0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	PRIVUSB7	PRIVUSB6	PRIVUSB5	PRIVUSB4	PRIVUSB3	PRIVUSB2	PRIVUSB1	PRIVUSB0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WRENUSB7	WRENUSB6	WRENUSB5	WRENUSB4	WRENUSB3	WRENUSB2	WRENUSB1	WRENUSB0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RDENUSB7	RDENUSB6	RDENUSB5	RDENUSB4	RDENUSB3	RDENUSB2	RDENUSB1	RDENUSB0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 24, 25, 26, 27, 28, 29, 30, 31 – LOCKUSRBx Lock User Signature Rights for User Signature Block x

Value	Description
0	User signature rights RDENUSBx, WRDENUSBx, PRIVUSBx, LOCKUSRBx can be modified.
1	User signature rights RDENUSBx, WRDENUSBx, PRIVUSBx, LOCKUSRBx cannot be modified.

Bits 16, 17, 18, 19, 20, 21, 22, 23 – PRIVUSBx Privileged Access for User Signature Block x

Value	Description
0	User signature area x can be accessed in both User and Privileged modes.
1	User signature area x can be accessed only in Privileged mode.

Bits 8, 9, 10, 11, 12, 13, 14, 15 – WRENUSBx Write Enable for User Signature Block x

Value	Description
0	User signature area x cannot be written.
1	User signature area x can be written.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – RDENUSBx Read Enable for User Signature Block x

Value	Description
0	User signature area x cannot be read.
1	User signature area x can be read.

© 2026 Microchip Technology Inc. and its subsidiaries

29.5.6. SEFC Key Bus Lock Register

Name: EEFC_KBLR
Offset: 0x14
Reset: 0x00000000
Property: Read/Write

This register can only be written if the USRWP bit is cleared in EEFC_WPMR.

This register can only be accessed in Privileged mode.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	KBTLUSB7	KBTLUSB6	KBTLUSB5	KBTLUSB4	KBTLUSB3	KBTLUSB2	KBTLUSB1	KBTLUSB0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7 – KBTLUSBx Key Bus Transfer Lock from User Signature Block x

Value	Description
0	Key bus transfer from the first page of User Signature Block x is enabled.
1	Key bus transfer from the first page of User Signature Block x is locked until hardware reset. When set to 1, this bit can no longer be modified until hardware reset.

© 2026 Microchip Technology Inc. and its subsidiaries

29.5.7. SEFC Write Protection Mode Register

Name: EEFC_WPMR
Offset: 0xE4
Reset: 0x00000010
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	ERASEWL			USRWP	ERASEWP	LOCKWP	GPNVMWP	WPEN
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			1	0	0	0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x454643	PASSWD	Writing any other value in this field aborts the write operation. Always reads as 0.

Bit 7 – ERASEWL Erase and Write Lock

Value	Description
0	All Write and Erase commands in the main memory plane as selected by EEFC_FCR.FCMD are enabled.
1	All Write and Erase commands in the main memory plane as selected by EEFC_FCR.FCMD are disabled until hardware reset. When set to 1, this bit can no longer be modified until hardware reset.

Bit 4 – USRWP User Signature Write Protection

Value	Description
0	Disables the write protection of EEFC_USR and EEFC_KBLR registers.
1	Enables the write protection of EEFC_USR and EEFC_KBLR registers.

Bit 3 – ERASEWP Erase and Write Protection

Value	Description
0	All Write and Erase commands in the main memory plane as selected by EEFC_FCR.FCMD are enabled.
1	All Write and Erase commands in the main memory plane as selected by EEFC_FCR.FCMD are disabled.

Bit 2 – LOCKWP Lock Bit Write Protection

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	The 'Set lock bit' and 'Clear lock bit' commands selected by EEFC_FCR.FCMD are enabled.
1	The 'Set lock bit' and 'Clear lock bit' commands selected by EEFC_FCR.FCMD are disabled.

Bit 1 – GPNVMWP GPNVM Bit Write Protection

Value	Description
0	The 'Set GPNVM bit' and 'Clear GPNVM bit' commands selected by EEFC_FCR.FCMD are enabled.
1	The 'Set GPNVM bit' and 'Clear GPNVM bit' commands selected by EEFC_FCR.FCMD are disabled.

Bit 0 – WPEN Write Protection Enable

See Register Write Protection for the list of registers that can be protected.

Value	Description
0	Disables write protection if WPKEY corresponds to 0x454643 ("EFC" in ASCII).
1	Enables write protection if WPKEY corresponds to 0x454643 ("EFC" in ASCII).

© 2026 Microchip Technology Inc. and its subsidiaries

30. Memory to Memory (MEM2MEM)

30.1. Description

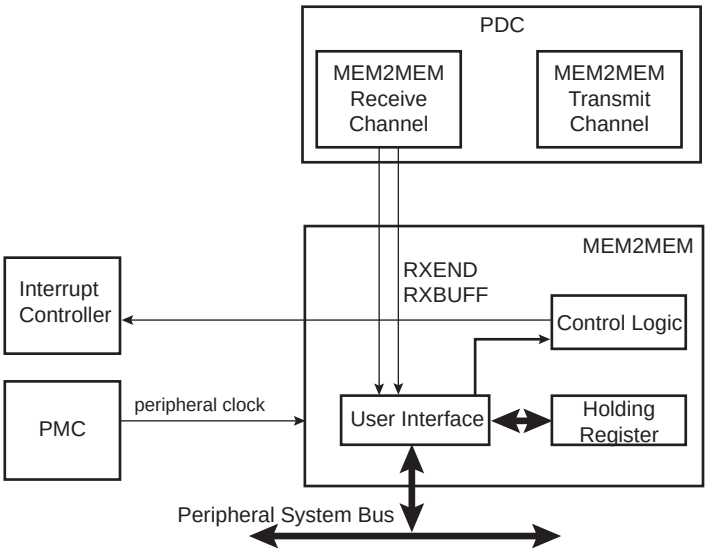
The Memory to Memory (MEM2MEM) module allows the Peripheral DMA Controller (PDC) to perform memory to memory transfer without CPU intervention. The transfer size can be configured in byte, half-word or word. Two PDC channels are required to perform the transfer; one channel defines the source of the transfer and the other defines the destination.

30.2. Embedded Characteristics

- Allows PDC to Perform Memory To Memory Transfer
- Supports Byte, Half-word and Word Transfer
- Interrupt for End of Transfer
- Register Write Protection

30.3. Block Diagram

Figure 30-1. Memory to Memory Block Diagram



30.4. Product Dependencies

30.4.1. Power Management

The MEM2MEM is not continuously clocked. The user must first configure the Power Management Controller (PMC) to enable the MEM2MEM clock.

30.4.2. Interrupt Sources

The MEM2MEM interface has an interrupt line connected to the Interrupt Controller.

Handling the MEM2MEM interrupt requires programming the Interrupt Controller before configuring the MEM2MEM.

30.5. Functional Description

The memory to memory transfer requires two operations.

The PDC receive channel associated to the MEM2MEM module must be configured with the transfer destination address and buffer size.

© 2026 Microchip Technology Inc. and its subsidiaries

The PDC transmit channel associated to the MEM2MEM module must be configured with the source address and buffer size. The transmit channel buffer size must be equal to the receive channel buffer size.

The two PDC channels exchange data through the Transfer Holding register (MEM2MEM_THR) which appears fully transparent from configuration. This register can be used as a general-purpose register in case the memory to memory transfer capability is not used.

The size of each element of the data buffer can be configured in byte, half-word or word by writing the TSIZE field in the Mode register (MEM2MEM_MR). Word transfer (32-bit) is the default size.

The transfer ends when either RXEND rises and/or RXBUFF rises in the Interrupt Status register (MEM2MEM_ISR).

An interrupt can be triggered at the end of transfer by configuring the Interrupt Enable register (MEM2MEM_IER). Refer to the section “Peripheral Data Controller (PDC)”.

30.6. Register Write Protection

To prevent any single software error from corrupting MEM2MEM behavior, certain registers in the address space can be write-protected by setting the WPEN and/or WPITEN bits in the MEM2MEM Write Protection Mode Register (MEM2MEM_WPMR).

The following register can be write-protected when MEM2MEM_WPMR.WPEN is set:

- MEM2MEM Mode Register

The following registers can be write-protected when MEM2MEM_WPMR.WPITEN is set:

- MEM2MEM Interrupt Enable Register
- MEM2MEM Interrupt Disable Register

© 2026 Microchip Technology Inc. and its subsidiaries

30.7. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	MEM2MEM_THR	31:24	THDATA[31:24]							
		23:16	THDATA[23:16]							
		15:8	THDATA[15:8]							
		7:0	THDATA[7:0]							
0x04	MEM2MEM_MR	31:24								
		23:16								
		15:8								
		7:0							TSIZE[1:0]	
0x08	MEM2MEM_IER	31:24								
		23:16								
		15:8								
		7:0							RXBUFF	RXEND
0x0C	MEM2MEM_IDR	31:24								
		23:16								
		15:8								
		7:0							RXBUFF	RXEND
0x10	MEM2MEM_IMR	31:24								
		23:16								
		15:8								
		7:0							RXBUFF	RXEND
0x14	MEM2MEM_ISR	31:24								
		23:16								
		15:8								
		7:0							RXBUFF	RXEND
0x18	MEM2MEM_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0							WPITEN	WPEN

© 2026 Microchip Technology Inc. and its subsidiaries

30.7.1. MEM2MEM Transfer Holding Register

Name: MEM2MEM_THR
Offset: 0x00
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	THDATA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	THDATA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	THDATA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	THDATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – THDATA[31:0] Transfer Holding Data
Must be written by the PDC transmit channel and read by the PDC receive channel.

© 2026 Microchip Technology Inc. and its subsidiaries

30.7.2. MEM2MEM Mode Register

Name: MEM2MEM_MR
Offset: 0x04
Reset: 0x00000002
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the MEM2MEM Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							TSIZE[1:0]	
Access							R/W	R/W
Reset							1	0

Bits 1:0 – TSIZE[1:0] Transfer Size

Value	Name	Description
0x0	T_8BIT	The buffer size is defined in bytes.
0x1	T_16BIT	The buffer size is defined in half-words (16-bit).
0x2	T_32BIT	The buffer size is defined in words (32-bit). Default value.

© 2026 Microchip Technology Inc. and its subsidiaries

30.7.3. MEM2MEM Interrupt Enable Register

Name: MEM2MEM_IER
Offset: 0x08
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the MEM2MEM Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							RXBUFF	RXEND
Access							W	W
Reset							–	–

Bit 1 – RXBUFF Buffer Full Interrupt Enable

Bit 0 – RXEND End of Transfer Interrupt Enable

© 2026 Microchip Technology Inc. and its subsidiaries

30.7.4. MEM2MEM Interrupt Disable Register

Name: MEM2MEM_IDR
Offset: 0x0C
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the MEM2MEM Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							RXBUFF	RXEND
Access							W	W
Reset							–	–

Bit 1 – RXBUFF Buffer Full Interrupt Disable

Bit 0 – RXEND End of Transfer Interrupt Disable

© 2026 Microchip Technology Inc. and its subsidiaries

30.7.5. MEM2MEM Interrupt Mask Register

Name: MEM2MEM_IMR
Offset: 0x10
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							RXBUFF	RXEND
Access							R	R
Reset							0	0

Bit 1 – RXBUFF Buffer Full Interrupt Mask

Bit 0 – RXEND End of Transfer Interrupt Mask

© 2026 Microchip Technology Inc. and its subsidiaries

30.7.6. MEM2MEM Interrupt Status Register

Name: MEM2MEM_ISR
Offset: 0x14
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							RXBUFF	RXEND
Access							R	R
Reset							0	0

Bit 1 – RXBUFF Buffer Full

Value	Description
0	The signal Buffer Full from the PDC receive channel is inactive.
1	The signal Buffer Full from the PDC receive channel is active.

Bit 0 – RXEND End of Transfer

Value	Description
0	The End of Transfer signal from the PDC receive channel is inactive.
1	The End of Transfer signal from the PDC receive channel is active.

© 2026 Microchip Technology Inc. and its subsidiaries

30.7.7. MEM2MEM Write Protection Mode Register

Name: MEM2MEM_WPMR
Offset: 0x18
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
							WPITEN	WPEN
Access							R/W	R/W
Reset							0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x4D454D	PASSWD	Writing any other value in this field aborts the write operation of the WPEN, WPITEN bits. Always reads as 0.

Bit 1 – WPITEN Write Protection Interruption Enable

Value	Description
0	Disables the write protection on interrupt registers if WPKEY corresponds to 0x4D454D (“MEM” in ASCII).
1	Enables the write protection on interrupt registers if WPKEY corresponds to 0x4D454D (“MEM” in ASCII).

Bit 0 – WPEN Write Protection Configuration Enable
See Section 7. “Register Write Protection” for the list of registers that can be write-protected.

Value	Description
0	Disables the write protection on configuration registers if WPKEY corresponds to 0x4D454D (“MEM” in ASCII).
1	Enables the write protection on configuration registers if WPKEY corresponds to 0x4D454D (“MEM” in ASCII).

© 2026 Microchip Technology Inc. and its subsidiaries

31. Peripheral DMA Controller (PDC)

31.1. Description

The Peripheral DMA Controller (PDC) transfers data between on-chip peripherals and the target memories. The PDC transfers data via the system bus matrix.

The user interface of each PDC channel is integrated into the user interface of the peripheral it serves. The user interface of mono-directional channels (receive-only or transmit-only) contains two 32-bit memory pointers and two 16-bit counters, one set (pointer, counter) for the current transfer and one set (pointer, counter) for the next transfer. The bidirectional channel user interface contains four 32-bit memory pointers and four 16-bit counters. Each set (pointer, counter) is used by the current transmit, next transmit, current receive and next receive.

Using the PDC decreases processor overhead by reducing its intervention during the transfer. This lowers significantly the number of clock cycles required for a data transfer, improving microcontroller performance.

To launch a transfer, the peripheral triggers its associated PDC channels by using transmit and receive signals. When the programmed data is transferred, an end of transfer interrupt is generated by the peripheral itself.

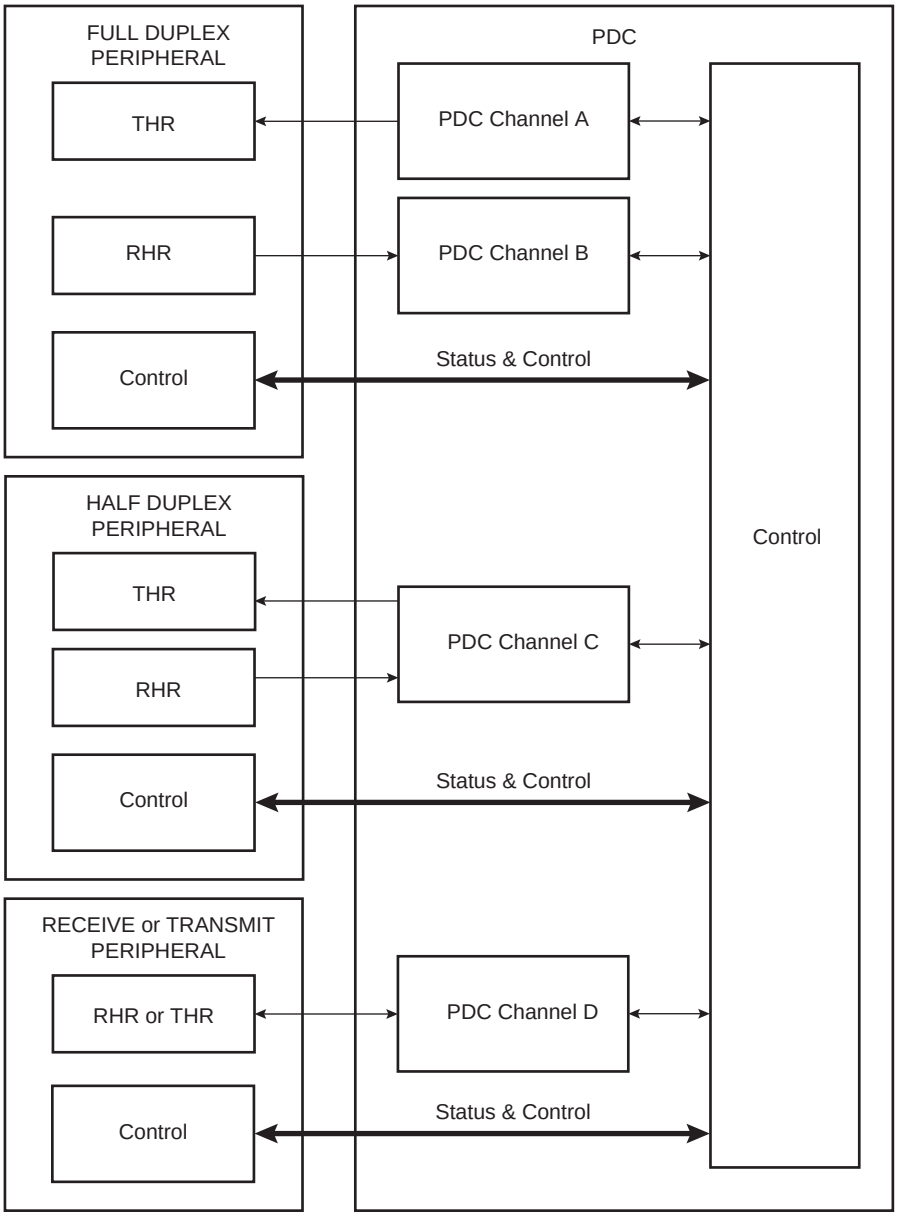
31.2. Embedded Characteristics

- Performs Transfers to/from Peripheral Bus On-Chip Peripherals
- Supports Half-duplex and Full-duplex Peripherals
- Automatic Circular Buffer Mode
- Transfer Bus Error Report

© 2026 Microchip Technology Inc. and its subsidiaries

31.3. Block Diagram

Figure 31-1. PDC Block Diagram



31.4. Functional Description

31.4.1. Configuration

The PDC channel user interface enables the user to configure and control data transfers for each channel. The user interface of each PDC channel is integrated into the associated peripheral user interface.

The user interface of a on-chip peripheral, whether it is full- or half-duplex, contains four 32-bit pointers (RPR, RNPR, TPR, TNPR) and four 16-bit counter registers (RCR, RNCR, TCR, TNCR). However, the transmit and receive parts of each type are programmed differently: the transmit and receive parts of a full-duplex peripheral can be programmed at the same time, whereas only one part (transmit or receive) of a half-duplex peripheral can be programmed at a time.

© 2026 Microchip Technology Inc. and its subsidiaries

32-bit pointers define the access location in memory for the current and next transfer, whether it is for read (transmit) or write (receive). 16-bit counters define the size of the current and next transfers. It is possible, at any moment, to read the number of transfers remaining for each channel.

The PDC has dedicated status registers which indicate if the transfer is enabled or disabled for each channel. The status for each channel is located in the associated peripheral status register. Transfers can be enabled and/or disabled by enabling the peripheral associated to a PDC channel (refer to the corresponding peripheral section).

At the end of a transfer, the PDC channel sends status flags to its associated peripheral. These flags are visible in the peripheral Status register (ENDRX, ENDTX, RXBUFF, and TXBUFE). See Section 5.5 and the associated peripheral user interface.

The peripheral where a PDC transfer is configured must have its peripheral clock enabled. The peripheral clock must be also enabled to access the PDC register set associated to this peripheral.

31.4.2. Memory Pointers

Each full-duplex peripheral is connected to the PDC by a receive channel and a transmit channel. Both channels have 32-bit memory pointers that point to a receive area and to a transmit area, respectively, in the target memory.

Each half-duplex peripheral is connected to the PDC by a bidirectional channel. This channel has two 32-bit memory pointers, one for current transfer and the other for next transfer. These pointers point to transmit or receive data depending on the operating mode of the peripheral.

Depending on the type of transfer (byte, half-word or word), the memory pointer is incremented respectively by 1, 2 or 4 bytes.

If a memory pointer address changes in the middle of a transfer, the PDC channel continues operating using the new address.

31.4.3. Transfer Counters

Each channel has two 16-bit counters, one for the current transfer and the one for the next transfer. These counters define the size of data to be transferred by the channel. The current transfer counter is decremented first as the data addressed by the current memory pointer starts to be transferred. When the current transfer counter reaches zero, the channel checks its next transfer counter. If the value of the next counter is zero, the channel stops transferring data and sets the appropriate flag. If the next counter value is greater than zero, the values of the next pointer/next counter are copied into the current pointer/current counter and the channel resumes the transfer, whereas next pointer/next counter get zero/zero as values. When the Circular buffer mode is activated, the register set {next counter, next pointer} is not reset when the next counter value is copied to the current counter, both next and current registers must be written to the same value. At the end of this transfer, the PDC channel sets the appropriate flags in the status register of the associated peripheral.

31.4.4. Data Transfers

The PDC channel first access is triggered when the peripheral is enabled.

When the peripheral receives external data, it sends a trigger event to its PDC receive channel which then requests access to the system bus matrix. When access is granted, the PDC receive channel reads the data in the peripheral. The read data are stored in an internal buffer of the PDC and then written to memory.

When the peripheral is ready to send data, it sends a trigger event to its PDC transmit channel which then requests access to the system bus matrix. When access is granted, the PDC transmit channel reads data from memory and transfers the data to the associated peripheral.

In case of invalid memory address resulting from a badly programmed PDC transmit or receive pointer, the system bus matrix does not perform the requested access and signals a bus error to

© 2026 Microchip Technology Inc. and its subsidiaries

the PDC. This transfer bus error drives the PERIPH_PTSR.ERR bit high to flag the error in the Transfer Status register.

31.4.5. PDC Flags and Peripheral Status Register

Each peripheral connected to the PDC sends out receive ready and transmit ready flags and the PDC returns flags to the peripheral. All these flags are only visible in the peripheral's Status register.

Depending on whether the peripheral is half- or full-duplex, the flags belong to either one single channel or two different channels.

These status flags are read in the status register of the peripheral associated to a PDC channel.

31.4.5.1. Receive Transfer End

The receive transfer end flag ENDRX is set when PERIPH_RCR reaches zero and the last data has been transferred from peripheral to memory.

This flag is cleared by writing a non-zero value to PERIPH_RCR or PERIPH_RNCR.

31.4.5.2. Transmit Transfer End

The transmit transfer end flag ENDTX is set when PERIPH_TCR reaches zero and the last data has been written to the peripheral.

This flag is cleared by writing a non-zero value to PERIPH_TCR or PERIPH_TNCR.

31.4.5.3. Receive Buffer Full

The receive buffer full flag RXBUFF is set when PERIPH_RCR reaches zero, with PERIPH_RNCR also set to zero and the last data transferred from peripheral to memory.

This flag is cleared by writing a non-zero value to PERIPH_TCR or PERIPH_TNCR.

31.4.5.4. Transmit Buffer Empty

The transmit buffer empty flag TXBUFE is set when PERIPH_TCR reaches zero, with PERIPH_TNCR also set to zero and the last data written to peripheral.

This flag is cleared by writing a non-zero value to PERIPH_TCR or PERIPH_TNCR.

31.4.6. Register Write Protection

To prevent any single software error from corrupting PDC behavior, certain registers in the address space can be write-protected by setting the WPPTREN,WPCTREN, WPCREN bits in the Write Protection Mode register (PERIPH_PWPMR). There is one write protection mode register per PDC channel,

The following registers can be write-protected when WPPTREN=1:

- Receive Pointer Register
- Transmit Pointer Register
- Receive Next Pointer Register
- Transmit Next Pointer Register

The following registers can be write-protected when WPCTREN=1:

- Receive Counter Register
- Transmit Counter Register
- Receive Next Counter Register
- Transmit Next Counter Register

The following registers can be write-protected when WPCREN=1:

- Transfer Control Register

© 2026 Microchip Technology Inc. and its subsidiaries

31.5. Register Summary

Note: PERIPH_XXX registers are mapped in the associated peripheral memory space at the same offset. These can be configured by the user depending on the function and the desired peripheral.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	PERIPH_RPR	31:24	RXPTR[31:24]							
		23:16	RXPTR[23:16]							
		15:8	RXPTR[15:8]							
		7:0	RXPTR[7:0]							
0x04	PERIPH_RCR	31:24								
		23:16								
		15:8	RXCTR[15:8]							
		7:0	RXCTR[7:0]							
0x08	PERIPH_TPR	31:24	TXPTR[31:24]							
		23:16	TXPTR[23:16]							
		15:8	TXPTR[15:8]							
		7:0	TXPTR[7:0]							
0x0C	PERIPH_TCR	31:24								
		23:16								
		15:8	TXCTR[15:8]							
		7:0	TXCTR[7:0]							
0x10	PERIPH_RNPR	31:24	RXNPTR[31:24]							
		23:16	RXNPTR[23:16]							
		15:8	RXNPTR[15:8]							
		7:0	RXNPTR[7:0]							
0x14	PERIPH_RNCR	31:24								
		23:16								
		15:8	RXNCTR[15:8]							
		7:0	RXNCTR[7:0]							
0x18	PERIPH_TNPR	31:24	TXNPTR[31:24]							
		23:16	TXNPTR[23:16]							
		15:8	TXNPTR[15:8]							
		7:0	TXNPTR[7:0]							
0x1C	PERIPH_TNCR	31:24								
		23:16								
		15:8	TXNCTR[15:8]							
		7:0	TXNCTR[7:0]							
0x20	PERIPH_PTCR	31:24								ERRCLR
		23:16					TXCBDIS	TXCBEN	RXCBDIS	RXCBEN
		15:8							TXTDIS	TXTEN
		7:0							RXTDIS	RXTEN
0x24	PERIPH_PTSR	31:24								ERR
		23:16						TXCBEN		RXCBEN
		15:8								TXTEN
		7:0								RXTEN
0x28	PERIPH_PWPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0						WPCREN	WPCTREN	WPPTREN

© 2026 Microchip Technology Inc. and its subsidiaries

31.5.1. Receive Pointer Register

Name: PERIPH_RPR
Offset: 0x00
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPPTREN bit is cleared in the Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	RXPTR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RXPTR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RXPTR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RXPTR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – RXPTR[31:0] Receive Pointer Register
RXPTR must be set to receive buffer address.
When a half-duplex peripheral is connected to the PDC, RXPTR = TXPTR.

© 2026 Microchip Technology Inc. and its subsidiaries

31.5.2. Receive Counter Register

Name: PERIPH_RCR
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPCTREN bit is cleared in the Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	RXCTR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RXCTR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – RXCTR[15:0] Receive Counter Register
RXCTR must be set to receive buffer size.
When a half-duplex peripheral is connected to the PDC, RXCTR = TXCTR.

Value	Description
0	Stops peripheral data transfer to the receiver.
1-65535	Starts peripheral data transfer if the corresponding channel is active.

© 2026 Microchip Technology Inc. and its subsidiaries

31.5.3. Transmit Pointer Register

Name: PERIPH_TPR
Offset: 0x08
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPPTREN bit is cleared in the Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	TXPTR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	TXPTR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TXPTR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TXPTR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – TXPTR[31:0] Transmit Counter Register
TXPTR must be set to transmit buffer address.
When a half-duplex peripheral is connected to the PDC, RXPTR = TXPTR.

© 2026 Microchip Technology Inc. and its subsidiaries

31.5.4. Transmit Counter Register

Name: PERIPH_TCR
Offset: 0x0C
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPCTREN bit is cleared in the Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	TXCTR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TXCTR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – TXCTR[15:0] Transmit Counter Register
TXCTR must be set to transmit buffer size.
When a half-duplex peripheral is connected to the PDC, RXCTR = TXCTR.

Value	Description
0	Stops peripheral data transfer to the transmitter.
1–65535	Starts peripheral data transfer if the corresponding channel is active.

© 2026 Microchip Technology Inc. and its subsidiaries

31.5.5. Receive Next Pointer Register

Name: PERIPH_RNPR
Offset: 0x10
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPPTREN bit is cleared in the Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	RXNPTR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RXNPTR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RXNPTR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RXNPTR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – RXNPTR[31:0] Receive Next Pointer
RXNPTR contains the next receive buffer address.
When a half-duplex peripheral is connected to the PDC, RXNPTR = TXNPTR.

© 2026 Microchip Technology Inc. and its subsidiaries

31.5.6. Receive Next Counter Register

Name: PERIPH_RNCR
Offset: 0x14
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPCTREN bit is cleared in the Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	RXNCTR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RXNCTR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – RXNCTR[15:0] Receive Next Counter
RXNCTR contains the next receive buffer size.
When a half-duplex peripheral is connected to the PDC, RXNCTR = TXNCTR.

© 2026 Microchip Technology Inc. and its subsidiaries

31.5.7. Transmit Next Pointer Register

Name: PERIPH_TNPR
Offset: 0x18
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPPTREN bit is cleared in the Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	TXNPTR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	TXNPTR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TXNPTR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TXNPTR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – TXNPTR[31:0] Transmit Next Pointer
TXNPTR contains the next transmit buffer address.
When a half-duplex peripheral is connected to the PDC, RXNPTR = TXNPTR.

© 2026 Microchip Technology Inc. and its subsidiaries

31.5.8. Transmit Next Counter Register

Name: PERIPH_TNCR
Offset: 0x1C
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPCTREN bit is cleared in the Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	TXNCTR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TXNCTR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – TXNCTR[15:0] Transmit Counter Next
TXNCTR contains the next transmit buffer size.
When a half-duplex peripheral is connected to the PDC, RXNCTR = TXNCTR.

© 2026 Microchip Technology Inc. and its subsidiaries

31.5.9. Transfer Control Register

Name: PERIPH_PTCR
Offset: 0x20
Reset: –
Property: Write-only

This register can only be written if the WPCREN bit is cleared in the Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
								ERRCLR
Access								W
Reset								–

Bit	23	22	21	20	19	18	17	16
					TXCBDIS	TXCBEN	RXCBDIS	RXCBEN
Access					W	W	W	W
Reset					–	–	–	–

Bit	15	14	13	12	11	10	9	8
							TXTDIS	TXTEN
Access							W	W
Reset							–	–

Bit	7	6	5	4	3	2	1	0
							RXTDIS	RXTEN
Access							W	W
Reset							–	–

Bit 24 – ERRCLR Transfer Bus Error Clear

Value	Description
0	No effect.
1	Clears the transfer bus error status bit.

Bit 19 – TXCBDIS Transmitter Circular Buffer Disable

Value	Description
0	No effect.
1	Enables the PDC circular buffer for the transmitter operation.

Bit 18 – TXCBEN Transmitter Circular Buffer Enable

Value	Description
0	No effect.
1	Enables the PDC circular buffer for the transmitter operation.

Bit 17 – RXCBDIS Receiver Circular Buffer Disable

Value	Description
0	No effect.
1	Disables the PDC circular buffer for the receiver operation.

Bit 16 – RXCBEN Receiver Circular Buffer Enable

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	No effect.
1	Enables the PDC circular buffer for the receiver operation.

Bit 9 – TXTDIS Transmitter Transfer Disable

When a half-duplex peripheral is connected to the PDC, disabling the transmitter channel requests disables the receiver channel requests.

Value	Description
0	No effect.
1	Disables the PDC transmitter channel requests.

Bit 8 – TXTEN Transmitter Transfer Enable

When a half-duplex peripheral is connected to the PDC, it enables the transmitter channel requests only if RXTEN is not set. It is forbidden to set both TXTEN and RXTEN for a half-duplex peripheral.

Value	Description
0	No effect.
1	Enables the PDC transmitter channel requests.

Bit 1 – RXTDIS Receiver Transfer Disable

When a half-duplex peripheral is connected to the PDC, disabling the receiver channel requests also disables the transmitter channel requests.

Value	Description
0	No effect.
1	Disables the PDC receiver channel requests.

Bit 0 – RXTEN Receiver Transfer Enable

When a half-duplex peripheral is connected to the PDC, enabling the receiver channel requests automatically disables the transmitter channel requests. It is forbidden to set both TXTEN and RXTEN for a half-duplex peripheral.

Value	Description
0	No effect.
1	Enables PDC receiver channel requests if RXTDIS is not set.

© 2026 Microchip Technology Inc. and its subsidiaries

31.5.10. Transfer Status Register

Name: PERIPH_PTSR
Offset: 0x24
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
								ERR
Access								R
Reset								0

Bit	23	22	21	20	19	18	17	16
						TXCBEN		RXCBEN
Access						R		R
Reset						0		0

Bit	15	14	13	12	11	10	9	8
								TXTEN
Access								R
Reset								0

Bit	7	6	5	4	3	2	1	0
								RXTEN
Access								R
Reset								0

Bit 24 – ERR Transfer Bus Error

Value	Description
0	PDC accesses are performed on valid memory address since the last write of ERRCLR bit in PERIPH_PTCR.
1	PDC transmit or receive pointer (or next pointer) is programmed with an invalid memory address since the last write of ERRCLR bit in PERIPH_PTCR.

Bit 18 – TXCBEN Transmitter Circular Buffer Enable

Value	Description
0	PDC Transmitter circular buffer mode is disabled.
1	PDC Transmitter circular buffer mode is enabled.

Bit 16 – RXCBEN Receiver Circular Buffer Enable

Value	Description
0	PDC Receiver circular buffer mode is disabled.
1	PDC Receiver circular buffer mode is enabled.

Bit 8 – TXTEN Transmitter Transfer Enable

Value	Description
0	PDC transmitter channel requests are disabled.
1	PDC transmitter channel requests are enabled.

Bit 0 – RXTEN Receiver Transfer Enable

Value	Description
0	PDC receiver channel requests are disabled.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
1	PDC receiver channel requests are enabled.

© 2026 Microchip Technology Inc. and its subsidiaries

31.5.11. Write Protection Mode Register

Name: PERIPH_PWPMR
Offset: 0x28
Reset: 0x00000000
Property: Read/Write

See Register Write Protection for the list of registers that can be write-protected.

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
						WPCREN	WPCTREN	WPPTREN
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x504443	PASSWD	Writing any other value in this field aborts the write operation. Always reads as 0.

Bit 2 – WPCREN Write Protection Control Register Enable

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x504443 (“PDC” in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x504443 (“PDC” in ASCII).

Bit 1 – WPCTREN Write Protection Counter Registers Enable

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x504443 (“PDC” in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x504443 (“PDC” in ASCII).

Bit 0 – WPPTREN Write Protection Pointer Registers Enable

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x504443 (“PDC” in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x504443 (“PDC” in ASCII).

© 2026 Microchip Technology Inc. and its subsidiaries

32. Cortex-M Cache Controller (CMCC)

32.1. Description

The Cortex-M Cache Controller (CMCC) is a 4-way set associative unified cache controller. It integrates a controller, a tag directory, data memory, metadata memory and a configuration interface.

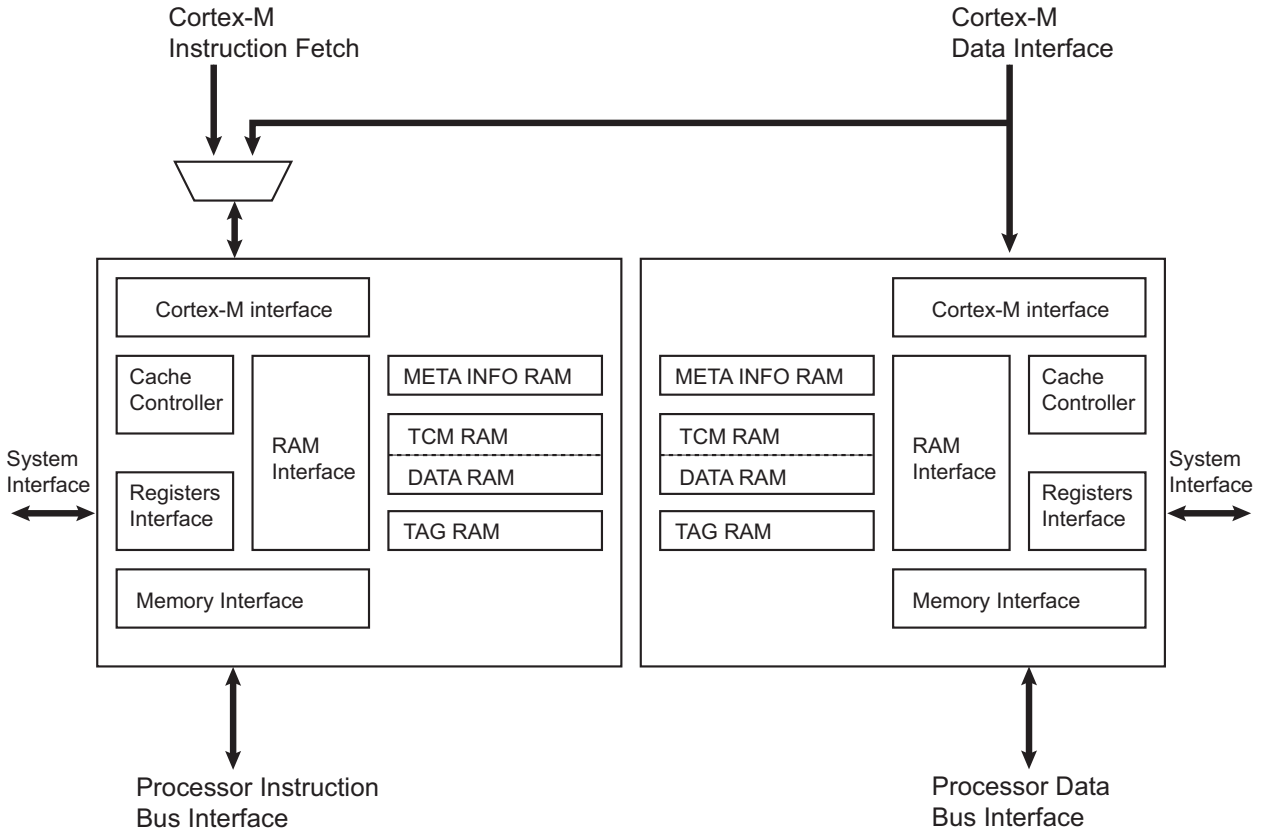
32.2. Embedded Characteristics

- Physically Addressed and Physically Tagged
- L1 Data Cache Set to 16/8 Kbytes
- L1 Cache Line Size Set to 16 bytes
- L1 Cache Integrates 32 System Bus Interface
- Unified Direct Mapped Cache Architecture
- Unified 4-way Set Associative Cache Architecture
- Write Accesses Forwarded, Cache State Not Modified. Allocate On Read.
- Round Robin Victim Selection Policy
- Event Monitoring, with One Programmable 32-bit Counter
- Configuration Registers Accessible through Cortex-M Private Peripheral Bus (PPB)
- Cache Interface Includes Cache Maintenance Operations Registers
- Register Write Protection

© 2026 Microchip Technology Inc. and its subsidiaries

32.3. Block Diagram

Figure 32-1. CMCC Block Diagram



32.4. Functional Description

32.4.1. Cache Operation

On reset, the cache controller data entries are all invalidated and the cache is disabled. The cache is transparent to processor operations. The cache controller is activated with its configuration registers. The configuration interface is memory-mapped in the private peripheral bus.

Use the following sequence to enable the cache controller:

1. Verify that the cache controller is disabled by reading the value of the CSTS (Cache Controller Status) bit of the Status register (CMCC_SR).
2. Enable the cache controller by writing a one to the CEN (Cache Enable) bit of the Control register (CMCC_CTRL).

Important: When used as TCM, SRAM memories embedded in both cache controllers (CMCC0 and CMCC1) are 32-bit accessible memories only. 8-bit or 16-bit accesses are not allowed. The code to load the ITCM during application initialization must use a "32-bit wide" memory copy (memcpy) function.

32.4.2. Cache Maintenance

If the contents seen by the cache have changed, the user must invalidate the cache entries. This can be done line-by-line or for all cache entries.

© 2026 Microchip Technology Inc. and its subsidiaries

32.4.2.1. Cache Invalidate-by-Line Operation

When an invalidate-by-line command is issued, the cache controller resets the valid bit information of the decoded cache line. As the line is no longer valid, the replacement counter points to that line.

Use the following sequence to invalidate one line of cache:

1. Disable the cache controller by clearing CMCC_CTRL.CEN.
2. Check the CSTS bit of CMCC_SR to verify that the cache is successfully disabled.
3. Perform an invalidate-by-line by configuring the bits INDEX and WAY in the Maintenance register 1 (CMCC_MAINT1).
4. Enable the cache controller by writing a one to CMCC_CTRL.CEN.

32.4.2.2. Cache Invalidate All Operation

To invalidate all cache entries, write a one to the INVALL bit of the Maintenance register 0 (CMCC_MAINT0).

32.4.3. Cache Performance Monitoring

The Cortex-M cache controller includes a programmable 32-bit monitor counter. The monitor can be configured to count the number of clock cycles, the number of data hits or the number of instruction hits.

Use the following sequence to activate the counter:

1. Configure the monitor counter by writing to the MODE field of the Monitor Configuration register (CMCC_MCFG).
2. Enable the counter by writing a one to the MENABLE bit of the Monitor Enable register (CMCC_MEN).
3. If required, clear the counter by writing a one to the SWRST bit of the Monitor Control register (CMCC_MCTRL).
4. Check the value of the monitor counter by reading the EVENT_CNT field of the Monitor Status register (CMCC_MSR).

32.4.4. Register Write Protection

To prevent any single software error from corrupting CMCC behavior, certain registers in the address space can be write-protected by setting the WPCFG and/or WPCR bits in the CMCC Write Protection Mode Register (CMCC_WPMR).

The following register can be write-protected when WPCFG is set in CMCC_WPMR:

- CMCC Configuration Register

The following register can be write-protected when WPCR is set in CMCC_WPMR:

- CMCC Control Register

© 2026 Microchip Technology Inc. and its subsidiaries

32.5. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CMCC_TYPE	31:24								
		23:16								
		15:8			CLSIZE[2:0]			CSIZE[2:0]		
		7:0	LCKDOWN	WAYNUM[1:0]		RRP	LRUP	RANDP	GCLK	AP
0x04	CMCC_CFG	31:24								
		23:16								
		15:8								LOCK
		7:0		PRGCSIZE[2:0]				DCDIS	ICDIS	GCLKDIS
0x08	CMCC_CTRL	31:24								
		23:16								
		15:8								LOCK
		7:0								CEN
0x0C	CMCC_SR	31:24								
		23:16								
		15:8								
		7:0								CSTS
0x10 ... 0x1F	Reserved									
0x20	CMCC_MAINT0	31:24								
		23:16								
		15:8								
		7:0								INVALL
0x24	CMCC_MAINT1	31:24	WAY[1:0]							
		23:16								
		15:8						INDEX[6:4]		
		7:0	INDEX[3:0]							
0x28	CMCC_MCFG	31:24								
		23:16								
		15:8								
		7:0							MODE[1:0]	
0x2C	CMCC_MEN	31:24								
		23:16								
		15:8								
		7:0								MENABLE
0x30	CMCC_MCTRL	31:24								
		23:16								
		15:8								
		7:0								SWRST
0x34	CMCC_MSR	31:24	EVENT_CNT[31:24]							
		23:16	EVENT_CNT[23:16]							
		15:8	EVENT_CNT[15:8]							
		7:0	EVENT_CNT[7:0]							
0x38 ... 0xE3	Reserved									
0xE4	CMCC_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0						WPCR		WPCFG

© 2026 Microchip Technology Inc. and its subsidiaries

32.5.1. CMCC Type Register

Name: CMCC_TYPE
Offset: 0x00
Reset: CMCC_TYPE_RESET
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			CLSIZE[2:0]			CSIZE[2:0]		
Access			R	R	R	R	R	R
Reset			e	s	e	r	_	e

Bit	7	6	5	4	3	2	1	0
	LCKDOWN	WAYNUM[1:0]		RRP	LRUP	RANDP	GCLK	AP
Access	R	R	R	R	R	R	R	R
Reset	p	y	t	_	c	c	m	c

Bits 13:11 – CLSIZE[2:0] Cache Line Size

Value	Name	Description
0	CLSIZE_1KB	Cache line size is 4 bytes
1	CLSIZE_2KB	Cache line size is 8 bytes
2	CLSIZE_4KB	Cache line size is 16 bytes
3	CLSIZE_8KB	Cache line size is 32 bytes

Bits 10:8 – CSIZE[2:0] Data Cache Size
Note: The Instruction cache size is 16KB, and the data cache size is 8KB

Value	Name	Description
0	CSIZE_1KB	Cache size is 1 Kbyte
1	CSIZE_2KB	Cache size is 2 Kbytes
2	CSIZE_4KB	Cache size is 4 Kbytes
3	CSIZE_8KB	Cache size is 8 Kbytes
4	CSIZE_16KB	Cache size is 16 Kbytes
5	CSIZE_32KB	Cache size is 32 Kbytes
6	CSIZE_64KB	Cache size is 64 Kbytes

Bit 7 – LCKDOWN Lockdown Supported

Value	Description
0	Lockdown is not supported.
1	Lockdown is supported.

Bits 6:5 – WAYNUM[1:0] Number of Ways

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Name	Description
0	DMAPPED	Direct Mapped Cache
1	ARCH2WAY	2-way set associative
2	ARCH4WAY	4-way set associative
3	ARCH8WAY	8-way set associative

Bit 4 – RRP Random Selection Policy Supported

Value	Description
0	Random Selection Policy is not supported.
1	Random Selection Policy is supported.

Bit 3 – LRUP Least Recently Used Policy Supported

Value	Description
0	Least Recently Used Policy is not supported.
1	Least Recently Used Policy is supported.

Bit 2 – RANDP Random Selection Policy Supported

Value	Description
0	Random victim selection is not supported.
1	Random victim selection is supported.

Bit 1 – GCLK Dynamic Clock Gating Supported

Value	Description
0	Cache controller does not support clock gating.
1	Cache controller uses dynamic clock gating.

Bit 0 – AP Access Port Access Allowed

Value	Description
0	Access Port Access is disabled.
1	Access Port Access is enabled.

© 2026 Microchip Technology Inc. and its subsidiaries

32.5.2. CMCC Configuration Register

Name: CMCC_CFG
Offset: 0x04
Reset: 0x0000000020
Property: Read/Write

This register can only be written if WPCFG is cleared in CMCC Write Protection Register .

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								LOCK
Reset								R/W
								0

Bit	7	6	5	4	3	2	1	0
Access		PRGCSIZE[2:0]				DCDIS	ICDIS	GCLKDIS
Reset		R/W	R/W	R/W		R/W	R/W	R/W
		0	1	0		0	0	0

Bit 8 – LOCK Configuration Lock Until Next System Reset (Write-once)

Value	Description
0	No effect
1	CMCC_CFG cannot be written until a system reset occurs.

Bits 6:4 – PRGCSIZE[2:0] Programmable Cache Size

Value	Name	Description
0	-	Reserved
1	PRGCSIZE_2KB	Programmable cache size is 2 Kbytes
2	PRGCSIZE_4KB	Programmable cache size is 4 Kbytes (default value)
3	PRGCSIZE_8KB	Programmable cache size is 8 Kbytes
4	PRGCSIZE_16KB	Programmable cache size is 16 Kbytes

Bit 2 – DCDIS Data Caching Disable
Note: DCDIS is only relevant for unified cache and data cache architecture.

Value	Description
0	Data caching enabled.
1	Data caching disabled.

Bit 1 – ICDIS Instruction Caching Disable
Note: ICDIS is only relevant for unified cache and Instruction cache architecture

Value	Description
0	Instruction caching enabled.
1	Instruction caching disabled.

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 0 – GCLKDIS Disable Clock Gating

Value	Description
0	Clock gating is activated.
1	Clock gating is disabled.

© 2026 Microchip Technology Inc. and its subsidiaries

32.5.3. CMCC Control Register

Name: CMCC_CTRL
Offset: 0x08
Reset: –
Property: Write-only

This register can only be written if WPCR is cleared in CMCC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								W
Reset								–

Bit	7	6	5	4	3	2	1	0
Access								W
Reset								–

Bit 8 – LOCK Control Lock Until Next System Reset (Write-once)

Value	Description
0	No effect
1	CMCC_CTRL cannot be written until a system reset occurs.

Bit 0 – CEN Cache Controller Enable

Value	Description
0	The cache controller is disabled.
1	The cache controller is enabled.

© 2026 Microchip Technology Inc. and its subsidiaries

32.5.4. CMCC Status Register

Name: CMCC_SR
Offset: 0x0C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								CSTS
Access								R
Reset								0

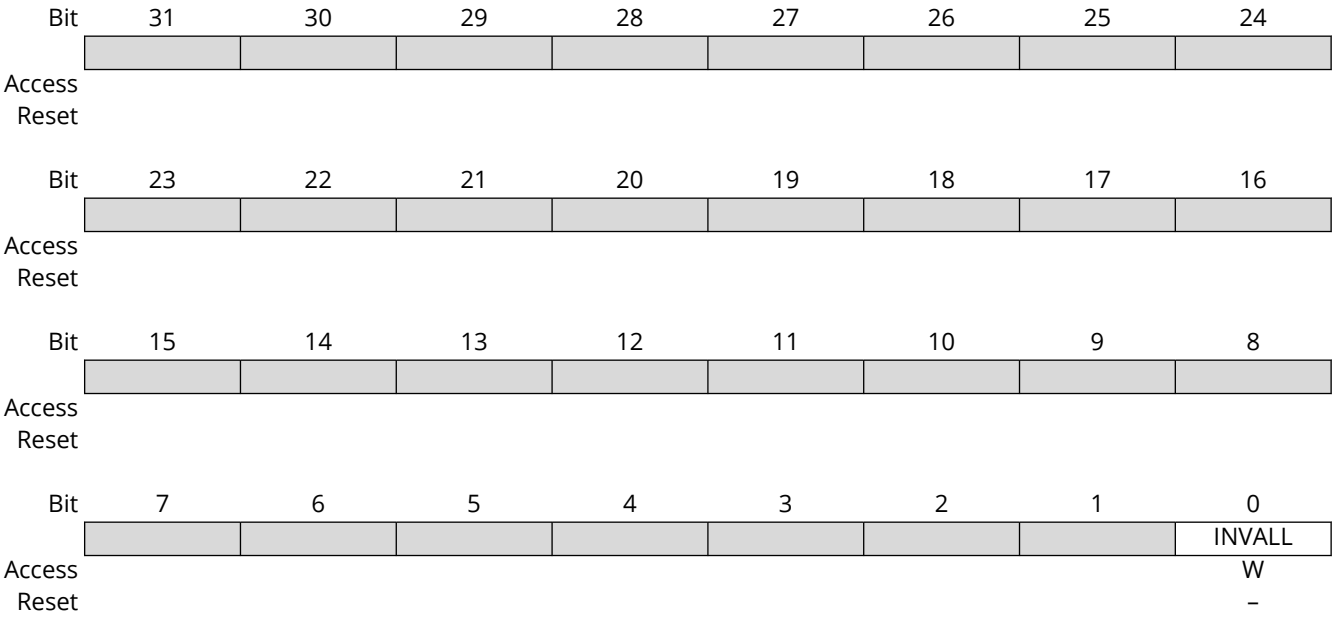
Bit 0 – CSTS Cache Controller Status

Value	Description
0	The cache controller is disabled.
1	The cache controller is enabled.

© 2026 Microchip Technology Inc. and its subsidiaries

32.5.5. CMCC Maintenance Register 0

Name: CMCC_MAINT0
Offset: 0x20
Reset: –
Property: Write-only



Bit 0 – INVALL Cache Controller Invalidate All

Value	Description
0	No effect.
1	All cache entries are invalidated.

© 2026 Microchip Technology Inc. and its subsidiaries

32.5.6. CMCC Maintenance Register 1

Name: CMCC_MAINT1
Offset: 0x24
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	WAY[1:0]							
Access	W	W						
Reset	–	–						

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						INDEX[6:4]		
Access						W	W	W
Reset						–	–	–

Bit	7	6	5	4	3	2	1	0
	INDEX[3:0]							
Access	W	W	W	W				
Reset	–	–	–	–				

Bits 31:30 – WAY[1:0] Invalidate Way

Value	Name	Description
0	WAY0	Way 0 is selection for index invalidation.
1	WAY1	Way 1 is selection for index invalidation.
2	WAY2	Way 2 is selection for index invalidation.
3	WAY3	Way 3 is selection for index invalidation.

Bits 10:4 – INDEX[6:0] Invalidate Index
Indicates the cache line to invalidate.

© 2026 Microchip Technology Inc. and its subsidiaries

32.5.7. CMCC Monitor Configuration Register

Name: CMCC_MCFG
Offset: 0x28
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							MODE[1:0]	
Access							R/W	R/W
Reset							0	0

Bits 1:0 – MODE[1:0] Cache Controller Monitor Counter Mode

Value	Name	Description
0	CYCLE_COUNT	Cycle counter
1	IHIT_COUNT	Instruction hit counter, Only relevant for unified cache and instruction cache architecture.
2	DHIT_COUNT	Data hit counter, Only relevant for unified cache and data cache architecture.

© 2026 Microchip Technology Inc. and its subsidiaries

32.5.8. CMCC Monitor Enable Register

Name: CMCC_MEN
Offset: 0x2C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								MENABLE
Access								R/W
Reset								0

Bit 0 – MENABLE Cache Controller Monitor Enable

Value	Description
0	The monitor counter is disabled.
1	The monitor counter is enabled.

© 2026 Microchip Technology Inc. and its subsidiaries

32.5.9. CMCC Monitor Control Register

Name: CMCC_MCTRL
Offset: 0x30
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								SWRST
Access								W
Reset								–

Bit 0 – SWRST Monitor

Value	Description
0	No effect.
1	Resets the event counter register.

© 2026 Microchip Technology Inc. and its subsidiaries

32.5.10. CMCC Monitor Status Register

Name: CMCC_MSR
Offset: 0x34
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	EVENT_CNT[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	EVENT_CNT[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	EVENT_CNT[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	EVENT_CNT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – EVENT_CNT[31:0] Monitor Event Counter

© 2026 Microchip Technology Inc. and its subsidiaries

32.5.11. CMCC Write Protection Mode Register

Name: CMCC_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
						WPCR		WPCFG
Access						R/W		R/W
Reset						0		0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x434D43	PASSWD	Writing any other value in this field aborts the write operation of the WPCFG and WPCR bits. Always reads as 0.

Bit 2 – WPCR Write Protection Control Enable

Value	Description
0	Disables the write protection on the control register if WPKEY corresponds to 0x0x434D43 (“CMC” in ASCII).
1	Enables the write protection on the control register if WPKEY corresponds to 0x0x434D43 (“CMC” in ASCII).

Bit 0 – WPCFG Write Protection Configuration Enable
See Register Write Protection for the list of registers that can be write-protected.

Value	Description
0	Disables the write protection on the configuration register if WPKEY corresponds to 0x0x434D43 (“CMC” in ASCII).
1	Enables the write protection on the configuration register if WPKEY corresponds to 0x0x434D43 (“CMC” in ASCII.

© 2026 Microchip Technology Inc. and its subsidiaries

33. Flexible Serial Communication Controller (FLEXCOM)

33.1. Description

The Flexible Serial Communication Controller (FLEXCOM) offers several serial communication protocols that are managed by the three submodules USART, SPI, and TWI (I2C).

The Universal Synchronous Asynchronous Receiver Transceiver (USART) provides one full-duplex universal synchronous asynchronous serial link. Data frame format is widely programmable (data length, parity, number of stop bits) to support a maximum of standards. The receiver implements parity error, framing error and overrun error detection. The receiver timeout enables handling variable-length frames and the transmitter timeguard facilitates communications with slow remote devices. Multidrop communications are also supported through address bit handling in reception and transmission.

The USART features three test modes: Remote Loopback, Local Loopback and Automatic Echo.

The USART supports specific operating modes providing interfaces on RS485, LIN, , with ISO7816 T = 0 or T = 1 smart card slots, and infrared transceivers. The hardware handshaking feature enables an out-of-band flow control by automatic management of the pins RTS and CTS.

The USART supports the connection to the Peripheral DMA Controller,which enables data transfers to the transmitter and from the receiver. The PDC provides chained buffer management without any intervention of the processor.

The Serial Peripheral Interface (SPI) circuit is a synchronous serial data link that provides communication with external devices in Host or Client mode. It also enables communication between processors if an external processor is connected to the system.

The Serial Peripheral Interface is essentially a shift register that serially transmits data bits to other SPIs. During a data transfer, one SPI system acts as the “host” which controls the data flow, while the other devices act as “clients” which have data shifted into and out by the host. Different CPUs can take turn being hosts (multiple host protocol, contrary to single host protocol where one CPU is always the host while all of the others are always clients). One host can simultaneously shift data into multiple clients. However, only one client can drive its output to write data back to the host at any given time.

A client device is selected when the host asserts its NSS signal. If multiple client devices exist, the host generates a separate client select signal for each client (NPCS).

The SPI system consists of two data lines and two control lines:

- Host Out Client In (MOSI)—This data line supplies the output data from the host shifted into the input(s) of the client(s).
- Host In Client Out (MISO)—This data line supplies the output data from a client to the input of the host. There may be no more than one client transmitting data during any particular transfer.
- Serial Clock (SPCK)—This control line is driven by the host and regulates the flow of the data bits. The host can transmit data at a variety of baud rates; there is one SPCK pulse for each bit that is transmitted.
- Client Select (NSS)—This control line allows clients to be turned on and off by hardware.

The Two-wire Interface (TWI) interconnects components on a unique two-wire bus, made up of one clock line and one data line based on a byte-oriented transfer format. It can be used with any Two-wire Interface bus Serial EEPROM and I2C-compatible devices, such as a Real-Time Clock (RTC), Dot Matrix/Graphic LCD Controller and temperature sensor. The TWI is programmable as a host or a client with sequential or single-byte access. Multiple host capability is supported.

Arbitration of the bus is performed internally and puts the TWI in Client mode automatically if the bus arbitration is lost.

© 2026 Microchip Technology Inc. and its subsidiaries

A configurable baud rate generator permits the output data rate to be adapted to a wide range of core clock frequencies.

The following table lists the compatibility level of any TWI in Host mode and a full I2C compatible device.

Table 33-1. TWI Compatibility with I2C Standard

I2C Standard	TWI
Standard mode speed (100 kHz)	Host, Multi-Host, Client supported
Fast mode speed (400 kHz)	Host, Multi-Host, Client supported
Fast mode Plus speed (1 MHz)	Host, Multi-Host, Client supported
High-speed mode (3.4 MHz)	Client supported
7- or 10-bit(1) Client addressing	Supported
Repeated Start (Sr) condition	Supported
ACK and NACK management	Supported
Input filtering	Supported
Slope control	Not supported
Clock stretching	Supported

Note:

1. 10-bit support in Host mode only.

33.2. Embedded Characteristics

33.2.1. USART/UART Characteristics

- 8-data Transmit and Receive FIFOs
- Programmable Baud Rate Generator
- Baud Rate can be Independent of the Processor/Peripheral Clock
- Supports Asynchronous Partial Wakeup on Receive Line Activity
- Comparison Function on Received Character
- 5-bit to 9-bit Full-duplex Synchronous or Asynchronous Serial Communications
  - 1, 1.5 or 2 stop bits in Asynchronous mode or 1 or 2 stop bits in Synchronous mode
  - Parity generation and error detection
  - Framing error detection, overrun error detection
  - Digital filter on receive line
  - MSB- or LSB-first
  - Optional break generation and detection
  - By 8 or by 16 oversampling receiver frequency
  - Optional hardware handshaking RTS-CTS
  - Receiver timeout and transmitter timeguard
  - Optional Multidrop mode with address generation and detection
- RS485 with Driver Control Signal
- ISO7816, T = 0 or T = 1 Protocols for Interfacing with Smart Cards
  - NACK handling, error counter with repetition and iteration limit
- IrDA Modulation and Demodulation
  - Communication at up to 115.2 kbit/s

© 2026 Microchip Technology Inc. and its subsidiaries

- OOK Modulation/Demodulation
- Up to 16-bit data, Manchester-encoded Frame Support
- LIN Mode
  - Compliant with LIN 1.3 and LIN 2.0 specifications
  - Host or client
  - Processing of frames with up to 256 data bytes
  - Response data length can be configurable or defined automatically by the identifier
  - Self-synchronization in client node configuration
  - Automatic processing and verification of the “synch break” and the “synch field”
  - “Synch break” detection even when partially superimposed with a data byte
  - Automatic identifier parity calculation/sending and verification
  - Parity sending and verification can be disabled
  - Automatic checksum calculation/sending and verification
  - Checksum sending and verification can be disabled
  - Support both “classic” and “enhanced” checksum types
  - Full LIN error checking and reporting
  - Frame Slot mode: host allocates slots to the scheduled frames automatically
  - Generation of the wakeup signal
- Test Modes
  - Remote Loopback, Local Loopback, Automatic Echo
- Supports Connection of:
  - Two Peripheral DMA Controller (PDC) channels
  - Offers buffer transfer without processor intervention
- Functional Safety: Protection, Monitors and Reports
  - Register Write protection
  - Reports any write-protected access

33.2.2. SPI Characteristics

- 8-data Transmit and Receive FIFOs
- Host or Client Serial Peripheral Bus Interface
  - 8-bit to 16-bit programmable data length per chip select
  - Programmable phase and polarity per chip select
  - Programmable transfer delay between consecutive transfers and delay before SPI clock per chip select
  - Programmable delay between chip selects
- Selectable Mode Fault Detection
- Host Mode Can Drive SPCK up to Peripheral Clock
- Host Mode Bit Rate Can Be Independent of the Processor/Peripheral Clock
- Client Mode Operates on SPCK, Asynchronously with Core and Bus Clock
- Two Chip Selects with External Decoder Support Allow Communication with up to 3 Peripherals
- Communication with Serial External Devices Supported
  - Serial memories, such as DataFlash and 3-wire EEPROMs

© 2026 Microchip Technology Inc. and its subsidiaries

  - Serial peripherals, such as ADCs, DACs, LCD controllers, CAN controllers and sensors
  - External coprocessors
- Connection to PDC Channels Optimizes Data Transfers
  - One channel for the receiver
  - One channel for the transmitter
- Functional Safety: Protection, Monitors and Reports
  - Register Write protection
  - Reports any write-protected access

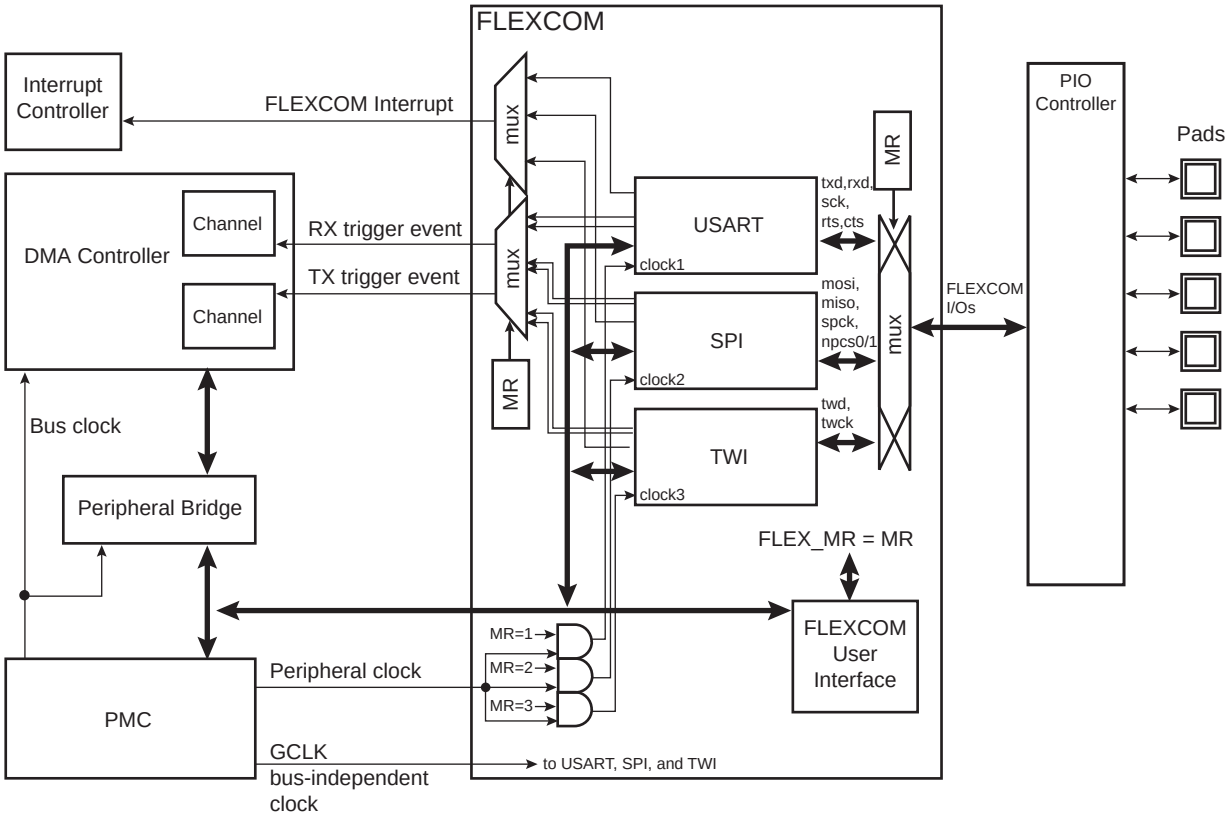
33.2.3. TWI/SMBus Characteristics

- 8-byte Transmit and Receive FIFOs
- Bit Rate can be Independent of the Processor/Peripheral Clock
- SMBus Support
- Compatible with I^2C Compatible Devices(1)
- One, Two or Three Bytes for Client Address
- Sequential Read/Write Operations
- General Call Supported in Client Mode
- Connection to Peripheral DMA Controller (PDC) Channels Optimizes Data Transfers
  - One channel for the receiver
  - One channel for the transmitter
- Functional Safety: Protection, Monitors and Reports
  - Register Write protection
  - Reports any write-protected access
- **Note:**
  a. See table TWI Compatibility with I2C Standard for further details.

© 2026 Microchip Technology Inc. and its subsidiaries

33.3. Block Diagram

Figure 33-1. FLEXCOM Block Diagram



33.4. I/O Lines Description

Table 33-2. I/O Lines Description

Name	Description			Type
	USART/UART	SPI	TWI	
FLEXCOM_IO0	TXD	MOSI	TWD	I/O
FLEXCOM_IO1	RXD	MISO	TWCK	I/O
FLEXCOM_IO2	SCK	SPCK	–	I/O
FLEXCOM_IO3	CTS	NPCS0/NSS	–	I/O
FLEXCOM_IO4	RTS	NPCS1	–	O

33.5. Product Dependencies

33.5.1. I/O Lines

The pins used for interfacing the FLEXCOM are multiplexed with the PIO lines. The programmer must first program the PIO controller to assign the desired FLEXCOM pins to their peripheral function. If I/O lines of the FLEXCOM are not used by the application, they can be used for other purposes by the PIO Controller.

33.5.2. Power Management

The peripheral clock is not continuously provided to the FLEXCOM. The programmer must first enable the FLEXCOM Clock in the Power Management Controller (PMC) before using the USART or SPI or TWI.

© 2026 Microchip Technology Inc. and its subsidiaries

To enable asynchronous partial wakeup for the FLEXCOM, the PMC must be configured first. The FLEXCOM peripheral clock can be automatically provided depending on the instructions (requests) provided by the FLEXCOM to the PMC.

33.5.3. Interrupt Sources

The FLEXCOM interrupt line is connected on one of the internal sources of the Interrupt Controller. Using the FLEXCOM interrupt requires the Interrupt Controller to be programmed first.

33.6. Register Accesses

Register accesses support 8-bit, 16-bit and 32-bit access, allowing, for example, an 8-bit part of a 32-bit register to be written in one access. To do so, the access must be done with the right size at the right address.

8-bit, 16-bit and 32-bit accesses are supported for register accesses. However, a field in a register cannot be partially written (for example, if a field is bigger than 8 bits, the whole field must be written).

This feature avoids a read-modify-write process if only a small part of the register is to be modified.

33.7. USART Functional Description

33.7.1. Baud Rate Generator

The baud rate generator provides the bit period clock named “baud rate clock” to both the receiver and the transmitter.

Configuring the USCLKS field in FLEX_US_MR selects the baud rate generator clock from one of the following sources:

- the peripheral clock
- a division of the peripheral clock, the divider being product dependent, but generally set to 8
- a fully programmable generic clock (GCLK) provided by PMC and independent of processor/peripheral clock
- the external clock, available on the SCK pin

The baud rate generator is based upon a 16-bit divider, which is programmed with the CD field of the Baud Rate Generator register (FLEX_US_BRGR). If a zero is written to CD, the baud rate generator does not generate any clock. If a one is written to CD, the divider is bypassed and becomes inactive.

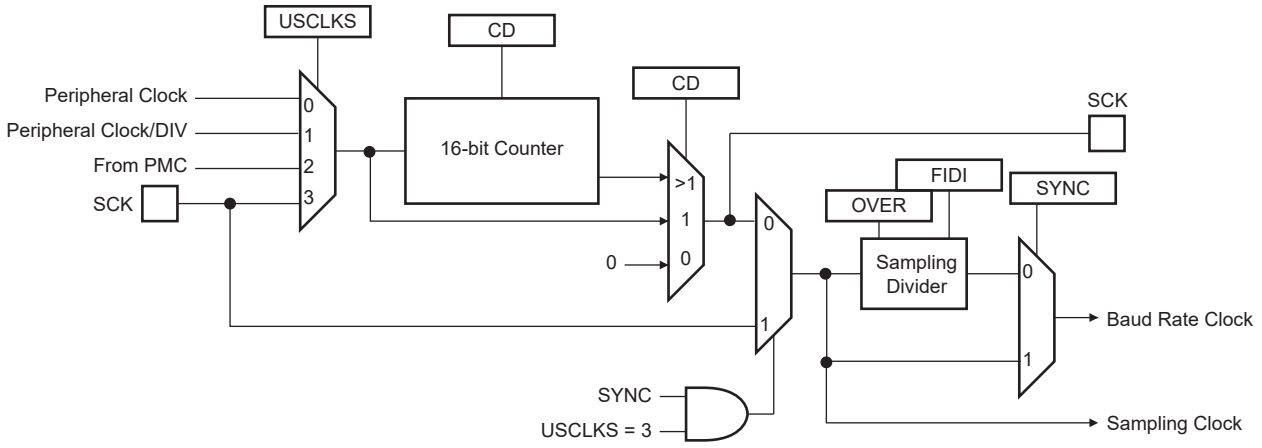
If the external SCK clock is selected, the duration of the low and high levels of the signal provided on the SCK pin must be longer than a peripheral clock period. The frequency of the signal provided on SCK must be at least three times lower than peripheral clock .

If GCLK is selected, the baud rate is independent of the processor/peripheral clock and thus processor/peripheral clock frequency can be changed without affecting the USART transfer. The GCLK frequency must be at least three times lower than peripheral clock frequency.

If GCLK is selected (USCLKS = 2) and the SCK pin is driven (CLKO = 1), the CD field must be greater than 1.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-2. Baud Rate Generator



33.7.1.1. Baud Rate in Asynchronous Mode

If the USART is programmed to operate in Asynchronous mode, the selected clock is first divided by CD, which is field-programmed in FLEX_US_BRGR. The resulting clock is provided to the receiver as a sampling clock and then divided by 16 or 8, depending on the programming of FLEX_US_MR.OVER.

If OVER is set, the receiver sampling is eight times higher than the baud rate clock. If OVER is cleared, the sampling is performed at 16 times the baud rate clock.

The baud rate is calculated as per the following formula:

$$\text{Baud rate} = \frac{\text{Selected Clock}}{(8(2 - \text{OVER})\text{CD})}$$

This gives a maximum baud rate of peripheral clock divided by 8, assuming that peripheral clock is the highest possible clock and that the OVER bit is set.

33.7.1.1.1. Baud Rate Calculation Example

The following table shows calculations of CD to obtain a baud rate at 38,400 bit/s for different source clock frequencies. It also shows the actual resulting baud rate and the error.

Table 33-3. Baud Rate Example (OVER = 0)

Source Clock (MHz)	Expected Baud Rate (bit/s)	Calculation Result	CD	Actual Baud Rate (bit/s)	Error
3,686,400	38,400	6.00	6	38,400.00	0.00%
4,915,200	38,400	8.00	8	38,400.00	0.00%
5,000,000	38,400	8.14	8	39,062.50	1.70%
7,372,800	38,400	12.00	12	38,400.00	0.00%
8,000,000	38,400	13.02	13	38,461.54	0.16%
12,000,000	38,400	19.53	20	37,500.00	2.40%
12,288,000	38,400	20.00	20	38,400.00	0.00%
14,318,180	38,400	23.30	23	38,908.10	1.31%
14,745,600	38,400	24.00	24	38,400.00	0.00%
18,432,000	38,400	30.00	30	38,400.00	0.00%
24,000,000	38,400	39.06	39	38,461.54	0.16%
24,576,000	38,400	40.00	40	38,400.00	0.00%
25,000,000	38,400	40.69	40	38,109.76	0.76%
32,000,000	38,400	52.08	52	38,461.54	0.16%
32,768,000	38,400	53.33	53	38,641.51	0.63%

© 2026 Microchip Technology Inc. and its subsidiaries

Table 33-3. Baud Rate Example (OVER = 0) (continued)

Source Clock (MHz)	Expected Baud Rate (bit/s)	Calculation Result	CD	Actual Baud Rate (bit/s)	Error
33,000,000	38,400	53.71	54	38,194.44	0.54%
40,000,000	38,400	65.10	65	38,461.54	0.16%
50,000,000	38,400	81.38	81	38,580.25	0.47%

The baud rate is calculated with the following formula:

Baud rate = MCK / CD × 16

The baud rate error is calculated with the following formula. It is not recommended to work with an error higher than 5%.

$$\text{Error} = 1 - \left(\frac{\text{Expected Baud Rate}}{\text{Actual Baud Rate}}\right)$$

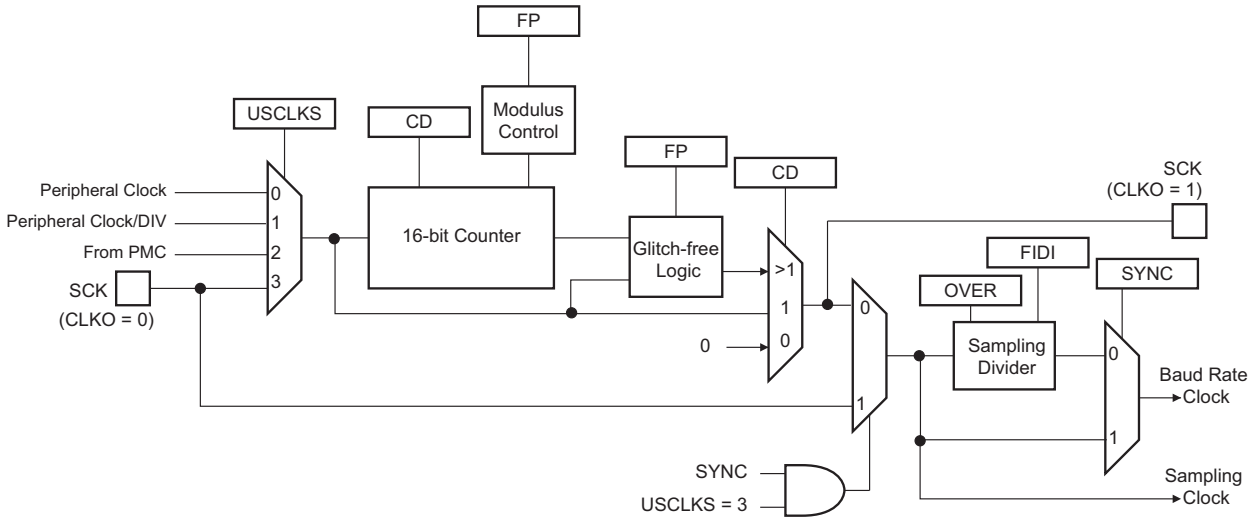
33.7.1.2. Fractional Baud Rate in Asynchronous Mode

The baud rate generator previously defined is subject to the following limitation: the output frequency changes by only integer multiples of the reference frequency. An approach to this problem is to integrate a fractional N clock generator that has a high resolution. The generator architecture is modified to obtain baud rate changes by a fraction of the reference source clock. This fractional part is programmed with the FP field in FLEX_US_BRGR. If FP is not 0, the fractional part is activated. The resolution is one eighth of the clock divider. The fractional baud rate is calculated using the following formula:

$$\text{Baud rate} = \frac{\text{Selected Clock}}{\left(8(2 - \text{OVER})\left(\text{CD} + \frac{\text{FP}}{8}\right)\right)}$$

The modified architecture is presented in the following figure.

Figure 33-3. Fractional Baud Rate Generator



WARNING When the value of field FP is greater than 0, the SCK (oversampling clock) generates nonconstant duty cycles. The SCK high duration is increased by "selected clock" period from time to time. The duty cycle depends on the value of the CD field.

© 2026 Microchip Technology Inc. and its subsidiaries

33.7.1.3. Baud Rate in Synchronous Mode

If the USART is programmed to operate in Synchronous mode, the selected clock is simply divided by the CD field in FLEX_US_BRGR:

$$\text{Baud rate} = \frac{\text{Selected Clock}}{\text{CD}}$$

In Synchronous mode, if the external clock is selected (USCLKS = 3) and CLKO = 0 (Client mode), the clock is provided directly by the signal on the USART SCK pin. No division is active. The value written in FLEX_US_BRGR has no effect. The external clock frequency must be at least three times lower than the system clock. In Synchronous mode host (USCLKS = 0 or 1, CLKO = 1), the receive part limits the SCK maximum frequency to $f_{\text{peripheral clock}}/3$.

When either the external clock SCK or the internal clock divided (peripheral clock/DIV or GCLK) is selected and if the user has to ensure a 50:50 mark/space ratio on the SCK pin, the value programmed in CD must be even. If the peripheral clock is selected and if the value programmed in CD is odd, the baud rate generator ensures a 50:50 duty cycle on the SCK pin.

33.7.1.4. Baud Rate in ISO 7816 Mode

The ISO7816 specification defines the bit rate with the following formula:

$$B = \frac{\text{Di}}{\text{Fi}} \times f$$

where:

- B is the bit rate
- Di is the bit rate adjustment factor
- Fi is the clock frequency division factor
- f is the ISO7816 clock frequency (Hz)

Di is a binary value encoded on a 4-bit field, named DI, as represented in the following table.

Table 33-4. Binary and Decimal Values for Di

DI field	0001	0010	0011	0100	0101	0110	1000	1001
Di (decimal)	1	2	4	8	16	32	12	20

Fi is a binary value encoded on a 4-bit field, named FI, as represented in the following table.

Table 33-5. Binary and Decimal Values for Fi

FI field	0000	0001	0010	0011	0100	0101	0110	1001	1010	1011	1100	1101
Fi (decimal)	372	372	558	744	1116	1488	1860	512	768	1024	1536	2048

The following table shows the resulting Fi/Di Ratio, which is the ratio between the ISO7816 clock and the baud rate clock.

Table 33-6. Possible Values for the Fi/Di Ratio

Fi/Di	372	558	744	1116	1488	1806	512	768	1024	1536	2048
1	372	558	744	1116	1488	1860	512	768	1024	1536	2048
2	186	279	372	558	744	930	256	384	512	768	1024
4	93	139.5	186	279	372	465	128	192	256	384	512
8	46.5	69.75	93	139.5	186	232.5	64	96	128	192	256
16	23.25	34.87	46.5	69.75	93	116.2	32	48	64	96	128
32	11.62	17.43	23.25	34.87	46.5	58.13	16	24	32	48	64
12	31	46.5	62	93	124	155	42.66	64	85.33	128	170.6
20	18.6	27.9	37.2	55.8	74.4	93	25.6	38.4	51.2	76.8	102.4

© 2026 Microchip Technology Inc. and its subsidiaries

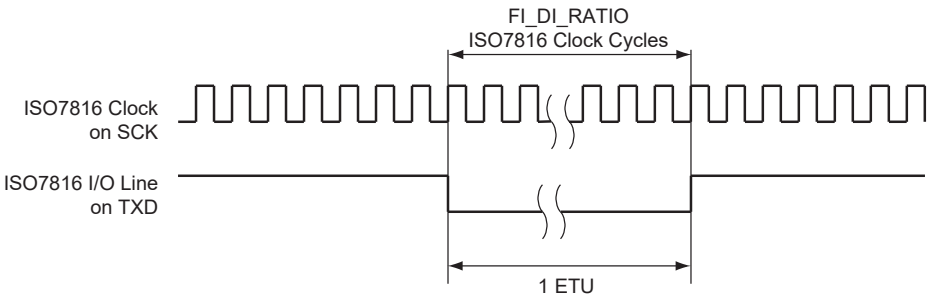
If the USART is configured in ISO7816 mode, the clock selected by the USCLKS field in FLEX_US_MR is first divided by the value programmed in field CD field in FLEX_US_BRGR. The resulting clock can be provided to the SCK pin to feed the smart card clock inputs. This means that FLEX_US_MR.CLKO can be set.

This clock is then divided by the value programmed in the FI_DI_RATIO field in the FI DI Ratio register (FLEX_US_FIDI). This is performed by the Sampling Divider, which performs a division by up to 2047 in ISO7816 mode. The noninteger values of the Fi/Di Ratio are not supported and the user must program the FI_DI_RATIO field to a value as close as possible to the expected value.

The FI_DI_RATIO field resets to the value 0x174 (372 in decimal) and is the most common divider between the ISO7816 clock and the bit rate (Fi = 372, Di = 1).

The following figure shows the relation between the Elementary Time Unit, corresponding to a bit time, and the ISO 7816 clock.

Figure 33-4. Elementary Time Unit (ETU)



33.7.2. Receiver and Transmitter Control

After reset, the receiver is disabled. The user must enable the receiver by setting the RXEN bit in the USART Control register (FLEX_US_CR). However, the receiver registers can be programmed before the receiver clock is enabled.

After reset, the transmitter is disabled. The user must enable it by setting the TXEN bit in FLEX_US_CR. However, the transmitter registers can be programmed before being enabled.

The receiver and the transmitter can be enabled together or independently.

At any time, the software can perform a reset on the receiver or the transmitter of the USART by setting the corresponding bit, RSTRX and RSTTX respectively, in FLEX_US_CR. The software resets clear the status flag and reset internal state machines but the user interface configuration registers hold the value configured prior to software reset. Regardless of what the receiver or the transmitter is performing, the communication is immediately stopped.

The user can also independently disable the receiver or the transmitter by setting RXDIS and TXDIS respectively in FLEX_US_CR. If the receiver is disabled during a character reception, the USART waits until the end of reception of the current character, then the reception is stopped. If the transmitter is disabled while it is operating, the USART waits the end of transmission of both the current character and character being stored in the USART Transmit Holding register (FLEX_US_THR). If a timeguard is programmed, it is handled normally.

33.7.3. Synchronous and Asynchronous Modes

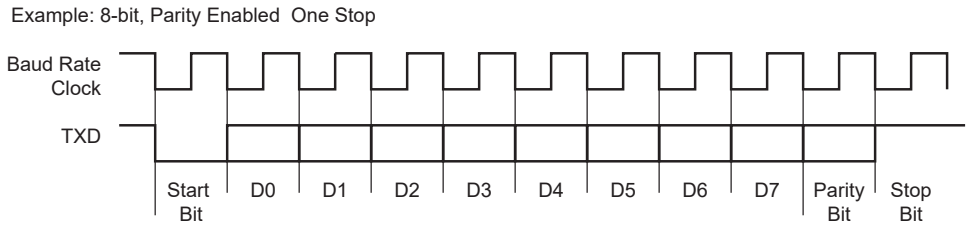
33.7.3.1. Transmitter Operations

The transmitter performs the same in both Synchronous and Asynchronous operating modes (SYNC = 0 or SYNC = 1). One start bit, up to 9 data bits, 1 optional parity bit and up to 2 stop bits are successively shifted out on the TXD pin at each falling edge of the programmed serial clock.

© 2026 Microchip Technology Inc. and its subsidiaries

The number of data bits is selected by the CHRL field and the MODE9 bit in FLEX_US_MR. Nine bits are selected by setting the MODE9 bit regardless of the CHRL field. The parity bit is set according to the PAR field in FLEX_US_MR. The even, odd, space, marked or none parity bit can be configured. The MSBF bit in FLEX_US_MR configures which data bit is sent first. If written to 1, the most significant bit is sent first. If written to 0, the less significant bit is sent first. The number of stop bits is selected by the NBSTOP field in FLEX_US_MR. The 1.5 stop bit is supported in Asynchronous mode only.

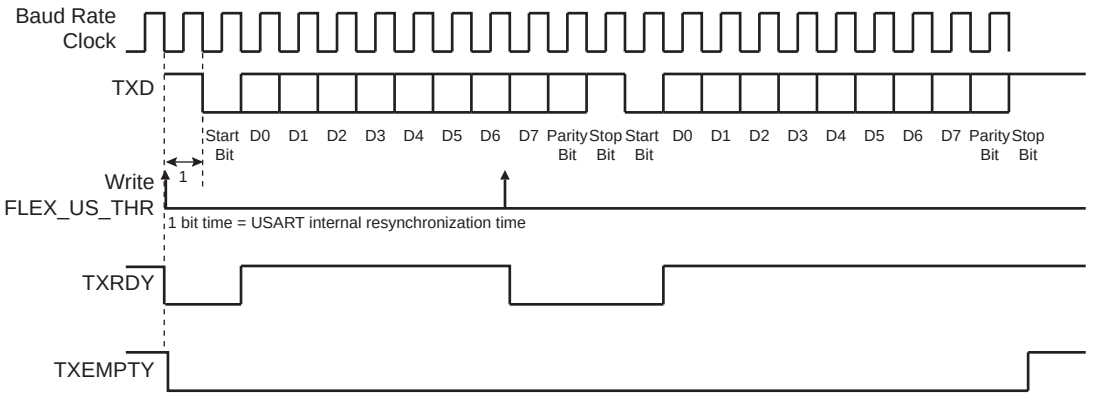
Figure 33-5. Character Transmit



The characters are sent by writing in FLEX_US_THR. The transmitter reports two status bits in the USART Channel Status register (FLEX_US_CSR): TXRDY (Transmitter Ready), which indicates that FLEX_US_THR is empty and TXEMPTY, which indicates that all the characters written in FLEX_US_THR have been processed. When the current character processing is completed, the last character written in FLEX_US_THR is transferred into the shift register of the transmitter and FLEX_US_THR is emptied, thus TXRDY rises.

Both TXRDY and TXEMPTY bits are low when the transmitter is disabled. Writing a character in FLEX_US_THR while TXRDY is low has no effect and the written character is lost.

Figure 33-6. Transmitter Status

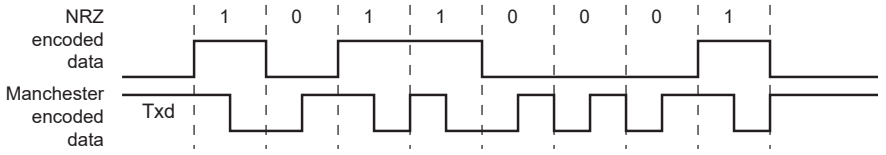


33.7.3.2. Manchester Encoder

When the Manchester encoder is in use, characters transmitted through the USART are encoded based on biphase Manchester II format. To enable this mode, set the FLEX_US_MR.MAN bit to 1. Depending on polarity configuration, a logic level (zero or one), is transmitted as a coded signal one-to-zero or zero-to-one. Thus, a transition always occurs at the midpoint of each bit time. It consumes more bandwidth than the original NRZ signal (2x) but the receiver has more error control since the expected input must show a change at the center of a bit cell. An example of Manchester encoded sequence is: the byte 0xB1 or 10110001 encodes to 10 01 10 10 01 01 01 10, assuming the default polarity of the encoder. The following figure illustrates this coding scheme.

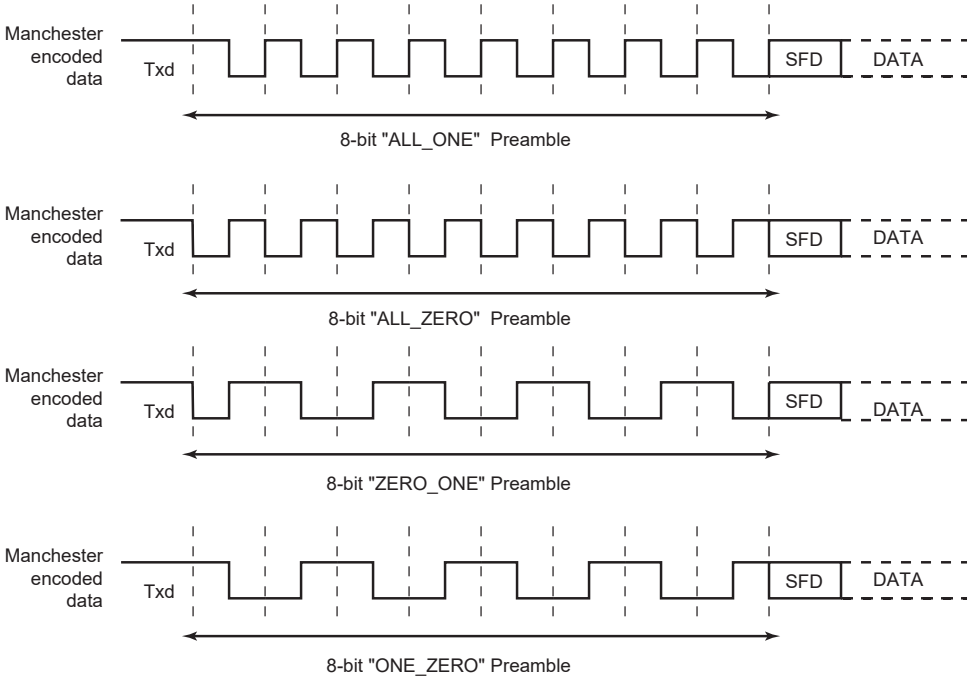
© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-7. NRZ to Manchester Encoding



The Manchester encoded character can also be encapsulated by adding both a configurable preamble and a start frame delimiter pattern. Depending on the configuration, the preamble is a training sequence, composed of a predefined pattern with a programmable length from 1 to 15 bit times. If the preamble length is set to 0, the preamble waveform is not generated prior to any character. The preamble pattern is chosen among the following sequences: ALL_ONE, ALL_ZERO, ONE_ZERO or ZERO_ONE, writing the FLEX_US_MAN.TX_PP field. The TX_PL field is used to configure the preamble length. The following figure illustrates and defines the valid patterns. To improve flexibility, the encoding scheme can be configured using the FLEX_US_MAN.TX_MPOL bit. If the TX_MPOL bit is set to zero (default), a logic zero is encoded with a zero-to-one transition and a logic one is encoded with a one-to-zero transition. If the TX_MPOL bit is set to one, a logic one is encoded with a one-to-zero transition and a logic zero is encoded with a zero-to-one transition.

Figure 33-8. Preamble Patterns, Default Polarity Assumed

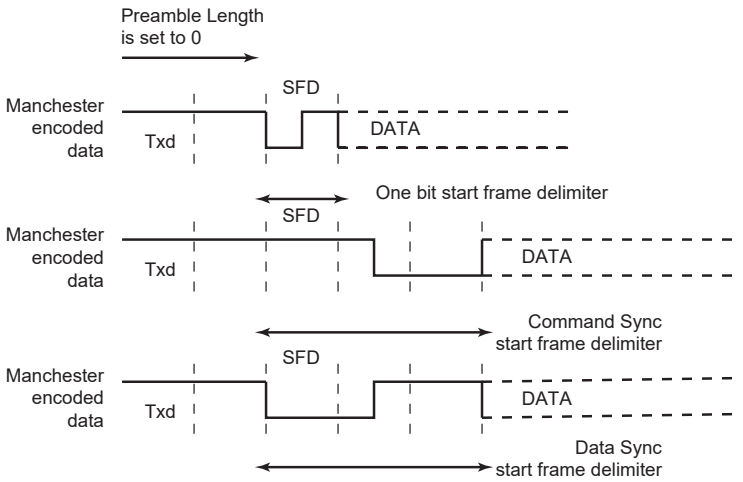


A start frame delimiter is to be configured using the FLEX_US_MR.ONEBIT bit. It consists of a user-defined pattern that indicates the beginning of a valid data. The following figure illustrates these patterns. If the start frame delimiter, also known as the start bit, is one bit, (ONEBIT = 1), a logic zero is Manchester encoded and indicates that a new character is being sent serially on the line. If the start frame delimiter is a synchronization pattern also referred to as sync (ONEBIT = 0), a sequence of three bit times is sent serially on the line to indicate the start of a new character. The sync waveform is in itself an invalid Manchester waveform as the transition occurs at the middle of the second bit time. Two distinct sync patterns are used: the command sync and the data sync. The command sync has a logic one level for one and a half bit times, then a transition to logic zero for the second one and a half bit times. If the FLEX_US_MR.MODSYNC bit is set to 1, the next character is a command. If it is set to 0, the next character is a data. When direct memory access is used, the MODSYNC bit can be immediately updated with a modified character located in memory. To enable

© 2026 Microchip Technology Inc. and its subsidiaries

this mode, the FLEX_US_MR.VAR_SYNC bit must be set. In this case, the FLEX_US_MR.MODSYNC bit is bypassed and the sync configuration is held in the FLEX_US_THR.TXSYNH bit. The USART character format is modified and includes sync information.

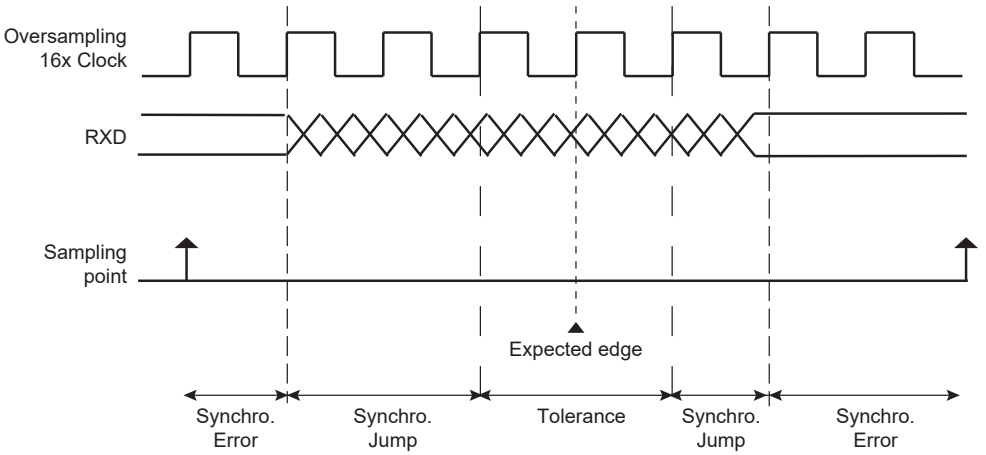
Figure 33-9. Start Frame Delimiter



33.7.3.2.1. Drift Compensation

Drift compensation is available only in 16X Oversampling mode. An hardware recovery system allows a larger clock drift. To enable the hardware system, the bit in the FLEX_US_MAN register must be set. If the RXD edge is one 16X clock cycle from the expected edge, this is considered as normal jitter and no corrective actions is taken. If the RXD event is between 4 and 2 clock cycles before the expected edge, then the current period is shortened by one clock cycle. If the RXD event is between 2 and 3 clock cycles after the expected edge, then the current period is lengthened by one clock cycle. These intervals are considered to be drift and so corrective actions are automatically taken.

Figure 33-10. Bit Resynchronization



33.7.3.3. Asynchronous Receiver

If the USART is programmed in Asynchronous operating mode (SYNC = 0), the receiver oversamples the RXD input line. The oversampling is either 16 or 8 times the baud rate clock, depending on the FLEX_US_MR.OVER bit.

The receiver samples the RXD line. If the line is sampled during one half of a bit time to 0, a start bit is detected and data, parity and stop bits are successively sampled on the bit rate clock.

© 2026 Microchip Technology Inc. and its subsidiaries

If the oversampling is 16 (OVER = 0), a start is detected at the eighth sample to 0. Data bits, parity bit and stop bit are assumed to have a duration corresponding to 16 oversampling clock cycles. If the oversampling is 8 (OVER = 1), a start bit is detected at the fourth sample to 0. Data bits, parity bit and stop bit are assumed to have a duration corresponding to 8 oversampling clock cycles.

The number of data bits, first bit sent and Parity mode are selected by the same fields and bits as the transmitter, i.e., respectively CHRL, MODE9, MSBF and PAR. For the synchronization mechanism only, the number of stop bits has no effect on the receiver as it considers only one stop bit, regardless of the NBSTOP field, so that resynchronization between the receiver and the transmitter can occur. Moreover, as soon as the stop bit is sampled, the receiver starts looking for a new start bit so that resynchronization can also be accomplished when the transmitter is operating with one stop bit.

The following figures illustrate start detection and character reception when USART operates in Asynchronous mode.

Figure 33-11. Asynchronous Start Detection

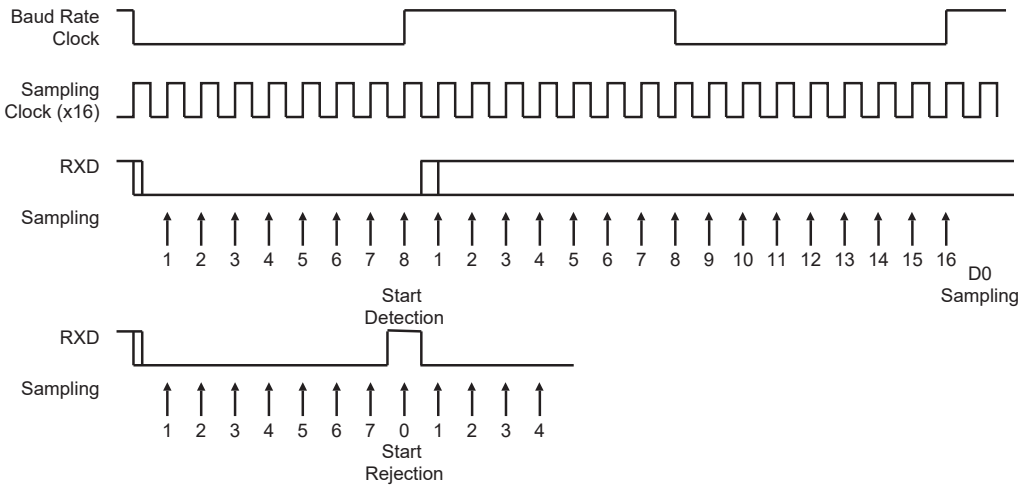
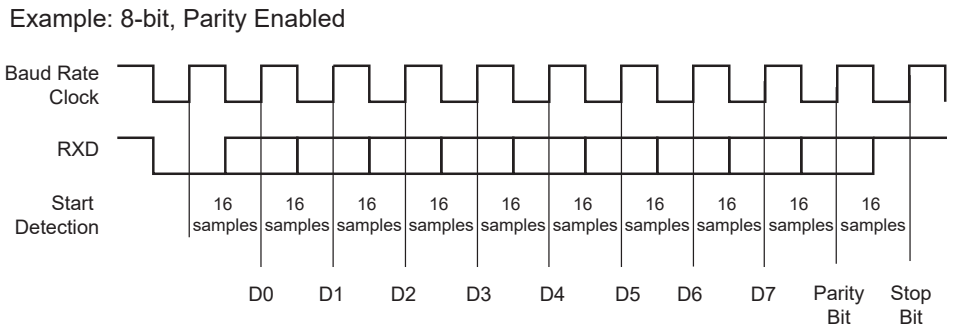


Figure 33-12. Asynchronous Character Reception



33.7.3.4. Manchester Decoder

When the FLEX_US_MR.MAN bit is set, the Manchester decoder is enabled. The decoder performs both preamble and start frame delimiter detection. One input line is dedicated to Manchester encoded input data.

An optional preamble sequence can be defined. Its length is user-defined and totally independent of the transmitter side. Use the FLEX_US_MAN.RX_PL field to configure the length of the preamble sequence. If the length is set to 0, no preamble is detected and the function is disabled. In addition, the polarity of the input stream is programmable with the FLEX_US_MAN.RX_MPOL bit. Depending on the desired application, the preamble pattern matching is to be defined via the

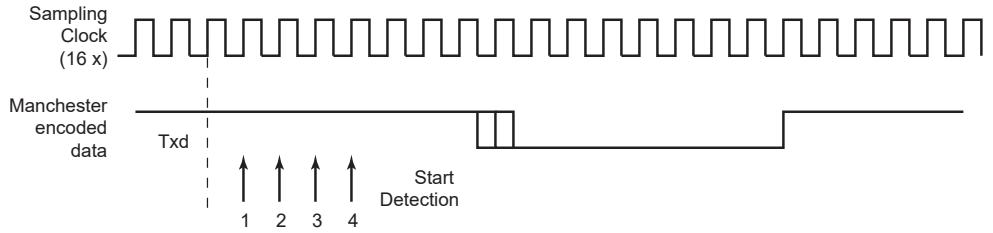
© 2026 Microchip Technology Inc. and its subsidiaries

FLEX_US_MAN.RX_PP field. See figure Preamble Patterns, Default Polarity Assumed for available preamble patterns.

Unlike preamble, the start frame delimiter is shared between Manchester Encoder and Decoder. So, if ONEBIT bit = 1, only a zero encoded Manchester can be detected as a valid start frame delimiter. If ONEBIT = 0, only a sync pattern is detected as a valid start frame delimiter. Decoder operates by detecting transition on incoming stream. If RXD is sampled during one quarter of a bit time to zero, a start bit is detected. See the following figure. The sample pulse rejection mechanism applies.

The FLEX_US_MAN.RXIDLEV bit informs the USART of the receiver line idle state value (receiver line inactive). The user must define RXIDLEV to ensure reliable synchronization. By default, RXIDLEV is set to one (receiver line is at level 1 when there is no activity).

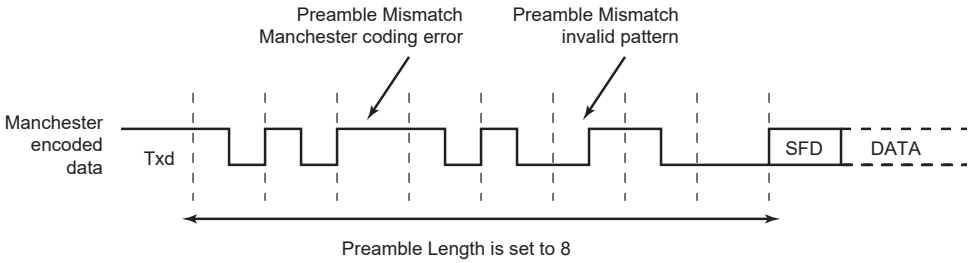
Figure 33-13. Asynchronous Start Bit Detection



The receiver is activated and starts preamble and frame delimiter detection, sampling the data at one quarter and then three quarters. If a valid preamble pattern or start frame delimiter is detected, the receiver continues decoding with the same synchronization. If the stream does not match a valid pattern or a valid start frame delimiter, the receiver resynchronizes on the next valid edge.The minimum time threshold to estimate the bit value is three quarters of a bit time.

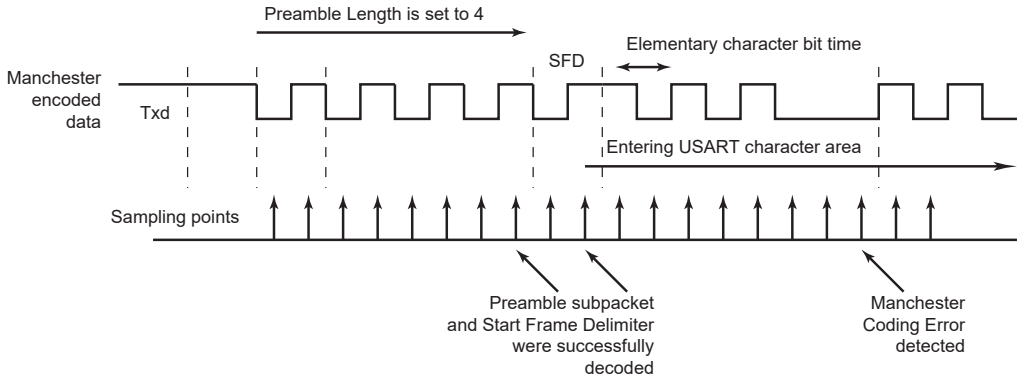
If a valid preamble (if used) followed with a valid start frame delimiter is detected, the incoming stream is decoded into NRZ data and passed to USART for processing. The following figure illustrates Manchester pattern mismatch. When incoming data stream is passed to the USART, the receiver is also able to detect Manchester code violation. A code violation is a lack of transition in the middle of a bit cell. In this case, the MANE flag in FLEX_US_CSR is raised. It is cleared by writing a one to FLEX_US_CR.RSTSTA. See figure "Manchester Error Flag" below for an example of Manchester error detection during the data phase.

Figure 33-14. Preamble Pattern Mismatch



© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-15. Manchester Error Flag



When the start frame delimiter is a sync pattern (ONEBIT = 0), both command and data delimiter are supported. If a valid sync is detected, the received character is written as RXCHR field in the Receive Holding register (FLEX_US_RHR) and the RXSYNH is updated. RXCHR is set to 1 when the received character is a command, and it is set to 0 if the received character is a data. This mechanism alleviates and simplifies the direct memory access as the character contains its own sync field in the same register.

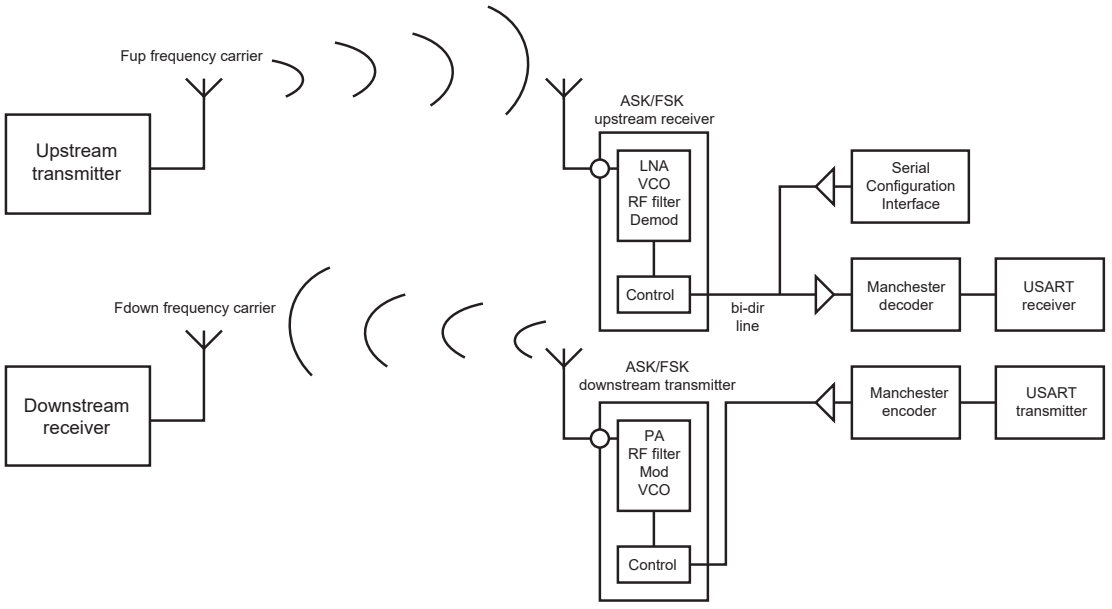
As the decoder is setup to be used in Unipolar mode, the first bit of the frame has to be a zero-to-one transition.

33.7.3.5. Radio Interface: Manchester Encoded USART Application

This section describes low data rate RF transmission systems and their integration with a Manchester encoded USART. These systems are based on transmitter and receiver ICs that support ASK and FSK modulation schemes.

The goal is to perform full-duplex radio transmission of characters using two different frequency carriers. See configuration in the following figure.

Figure 33-16. Manchester Encoded Characters RF Transmission



The USART peripheral is configured as a Manchester encoder/decoder. Looking at the downstream communication channel, Manchester encoded characters are serially sent to the RF transmitter. This may also include a user defined preamble and a start frame delimiter. Mostly, preamble is used in

© 2026 Microchip Technology Inc. and its subsidiaries

the RF receiver to distinguish between a valid data from a transmitter and signals due to noise. The Manchester stream is then modulated. See the following figure for an example of ASK modulation scheme. When a logic one is sent to the ASK modulator, the power amplifier, referred to as PA, is enabled and transmits an RF signal at downstream frequency. When a logic zero is transmitted, the RF signal is turned off. If the FSK modulator is activated, two different frequencies are used to transmit data. When a logic 1 is sent, the modulator outputs an RF signal at frequency F0 and switches to F1 if the data sent is a 0. See figure "FSK Modulator Output" below.

From the receiver side, another carrier frequency is used. The RF receiver performs a bit check operation examining demodulated data stream. If a valid pattern is detected, the receiver switches to Receiving mode. The demodulated stream is sent to the Manchester decoder. Because of bit checking inside RF IC, the data transferred to the microcontroller is reduced by a user-defined number of bits. The Manchester preamble length is to be defined in accordance with the RF IC configuration.

Figure 33-17. ASK Modulator Output

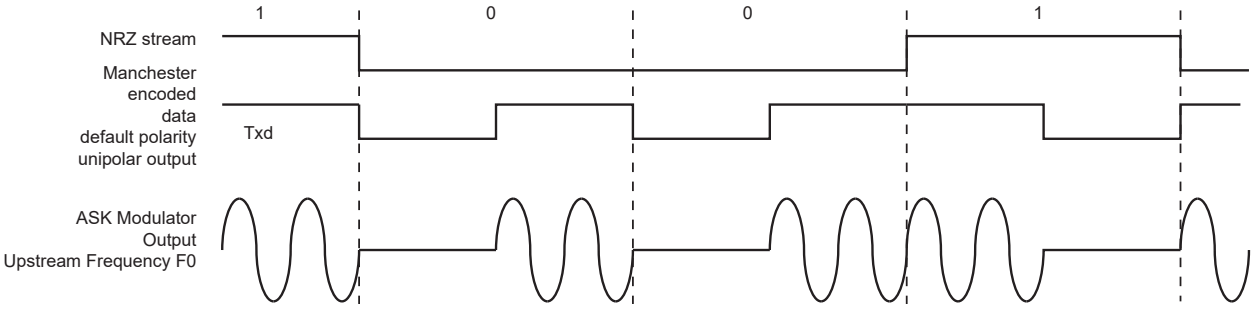
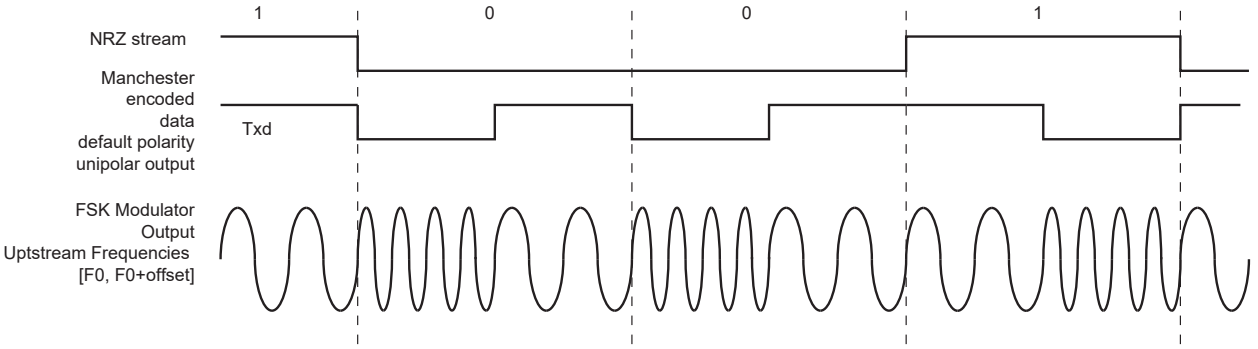


Figure 33-18. FSK Modulator Output



33.7.3.6. Synchronous Receiver

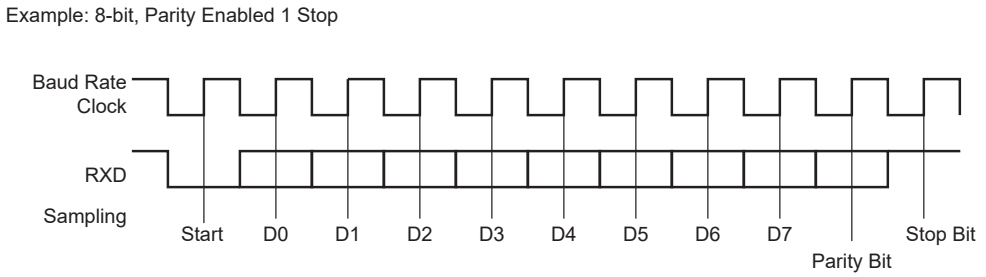
In Synchronous mode (SYNC = 1), the receiver samples the RXD signal on each rising edge of the baud rate clock. If a low level is detected, it is considered as a start. All data bits, the parity bit and the stop bits are sampled and the receiver waits for the next start bit. Synchronous mode operations provide a high-speed transfer capability.

Configuration fields and bits are the same as in Asynchronous mode.

The following figure illustrates a character reception in Synchronous mode.

© 2026 Microchip Technology Inc. and its subsidiaries

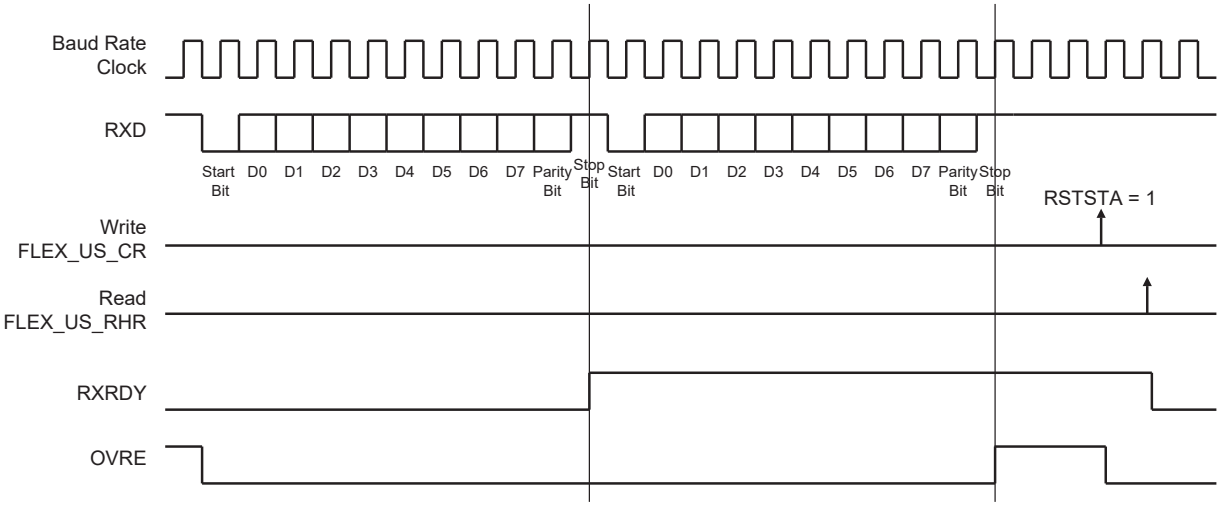
Figure 33-19. Synchronous Mode Character Reception



33.7.3.7. Receiver Operations

When a character reception is completed, it is transferred to the Receive Holding register (FLEX_US_RHR) and the FLEX_US_CSR.RXRDY bit is raised. If a character is completed while the RXRDY is set, the Overrun Error (OVRE) bit is set. The last character is transferred into FLEX_US_RHR and overwrites the previous one. The OVRE bit is cleared by writing a one to Reset Status bit FLEX_US_CR.RSTSTA.

Figure 33-20. Receiver Status



33.7.3.8. Parity

The USART supports five parity modes that are selected by writing to the FLEX_US_MR.PAR field. The PAR field also enables the Multidrop mode (see Multidrop Mode). Even and odd parity bit generation and error detection are supported.

If even parity is selected, the parity generator of the transmitter drives the parity bit to 0 if a number of 1s in the character data bit is even, and to 1 if the number of 1s is odd. Accordingly, the receiver parity checker counts the number of received 1s and reports a parity error if the sampled parity bit does not correspond. If odd parity is selected, the parity generator of the transmitter drives the parity bit to 1 if a number of 1s in the character data bit is even, and to 0 if the number of 1s is odd. Accordingly, the receiver parity checker counts the number of received 1s and reports a parity error if the sampled parity bit does not correspond. If the mark parity is used, the parity generator of the transmitter drives the parity bit to 1 for all characters. The receiver parity checker reports an error if the parity bit is sampled to 0. If the space parity is used, the parity generator of the transmitter drives the parity bit to 0 for all characters. The receiver parity checker reports an error if the parity bit is sampled to 1. If parity is disabled, the transmitter does not generate any parity bit and the receiver does not report any parity error.

© 2026 Microchip Technology Inc. and its subsidiaries

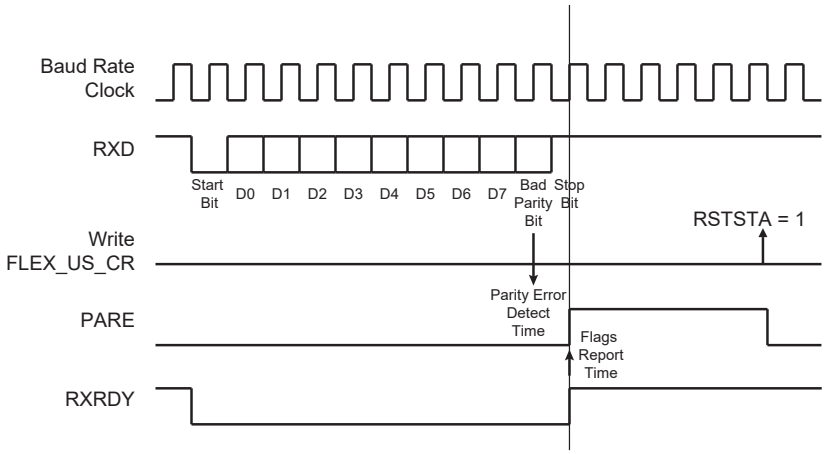
The following table shows an example of the parity bit for the character 0x41 (character ASCII “A”) depending on the configuration of the USART. Because there are two bits set to 1 in the character value, the parity bit is set to 1 when the parity is odd, or configured to 0 when the parity is even.

Table 33-7. Parity Bit Examples

Character	Hexadecimal	Binary	Parity Bit	Parity Mode
A	0x41	0100 0001	1	Odd
A	0x41	0100 0001	0	Even
A	0x41	0100 0001	1	Mark
A	0x41	0100 0001	0	Space
A	0x41	0100 0001	None	None

When the receiver detects a parity error, it sets the Parity Error bit FLEX_US_CSR.PARE. The PARE bit can be cleared by writing a one to the FLEX_US_CR.RSTSTA bit. The following figure illustrates the parity bit status setting and clearing.

Figure 33-21. Parity Error



33.7.3.9. Multidrop Mode

If the value 0x6 or 0x07 is written to the FLEX_US_MR.PAR field, the USART runs in Multidrop mode. This mode differentiates the data characters and the address characters. Data are transmitted with the parity bit to 0 and addresses are transmitted with the parity bit to 1.

If the USART is configured in Multidrop mode, the receiver sets the PARE parity error bit when the parity bit is high and the transmitter is able to send a character with the parity bit high when a one is written to the FLEX_US_CR.SENDA bit.

To handle parity error, the PARE bit is cleared by writing a one to the FLEX_US_CR.RSTSTA bit.

The transmitter sends an address byte (parity bit set) when the FLEX_US_CR.SENDA bit is written to 1. In this case, the next byte written to FLEX_US_THR is transmitted as an address. Any character written in FLEX_US_THR when the SENDA command is not written is transmitted normally with parity to 0.

33.7.3.10. Transmitter Timeguard

The timeguard feature enables the USART interface with slow remote devices.

The timeguard function enables the transmitter to insert an idle state on the TXD line between two characters. This idle state actually acts as a long stop bit.

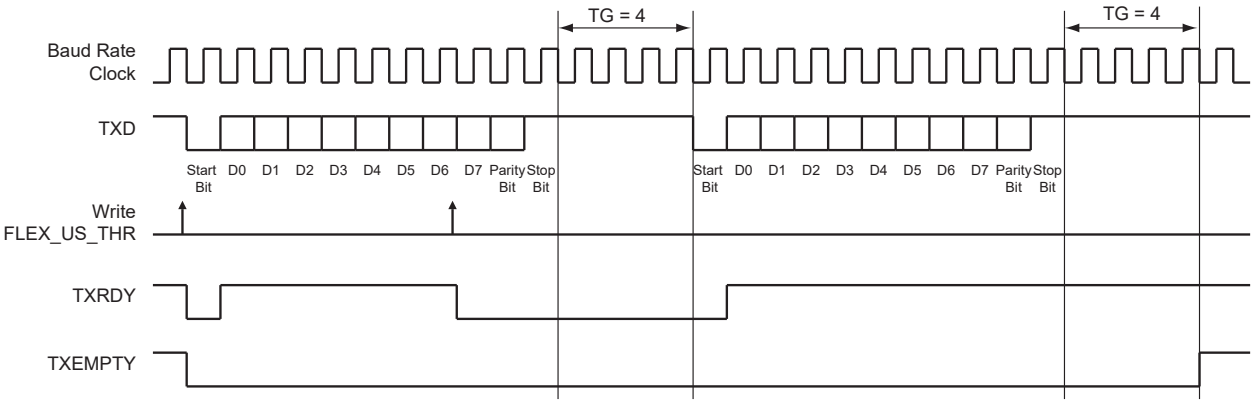
The duration of the idle state is programmed in the TG field of the Transmitter Timeguard register (FLEX_US_TTGR). When this field is written to zero, no timeguard is generated. Otherwise, the

© 2026 Microchip Technology Inc. and its subsidiaries

transmitter holds a high level on TXD after each transmitted byte during the number of bit periods programmed in TG in addition to the number of stop bits.

As illustrated in the following figure, the behavior of the TXRDY and TXEMPTY status bits is modified by the programming of a timeguard. TXRDY rises only when the start bit of the next character is sent, and thus remains to 0 during the timeguard transmission if a character has been written in FLEX_US_THR. TXEMPTY remains low until the timeguard transmission is completed as the timeguard is part of the current character being transmitted.

Figure 33-22. Timeguard Operations



The following table indicates the maximum length of a timeguard period that the transmitter can handle in relation to the function of the baud rate.

Table 33-8. Maximum Timeguard Length Depending on Baud Rate

Baud Rate (bit/s)	Bit Time (µs)	Timeguard (ms)
1,200	833	212.50
9,600	104	26.56
14,400	69.4	17.71
19,200	52.1	13.28
28,800	34.7	8.85
38,400	26	6.63
56,000	17.9	4.55
57,600	17.4	4.43
115,200	8.7	2.21

33.7.3.11. Receiver Timeout

The Receiver Timeout provides support in handling variable-length frames. This feature detects an idle condition on the RXD line. When a timeout is detected, the FLEX_US_CSR.TIMEOUT bit rises and can generate an interrupt, thus indicating to the driver an end of frame.

The timeout delay period (during which the receiver waits for a new character) is programmed in the TO field of the Receiver Timeout register (FLEX_US_RTOR). If the TO field is written to 0, the Receiver Timeout is disabled and no timeout is detected. The FLEX_US_CSR.TIMEOUT bit remains at 0. Otherwise, the receiver loads a 16-bit counter with the value programmed in TO. This counter is decremented at each bit period and reloaded each time a new character is received. If the counter reaches 0, the FLEX_US_CSR.TIMEOUT bit rises. Then, the user can either:

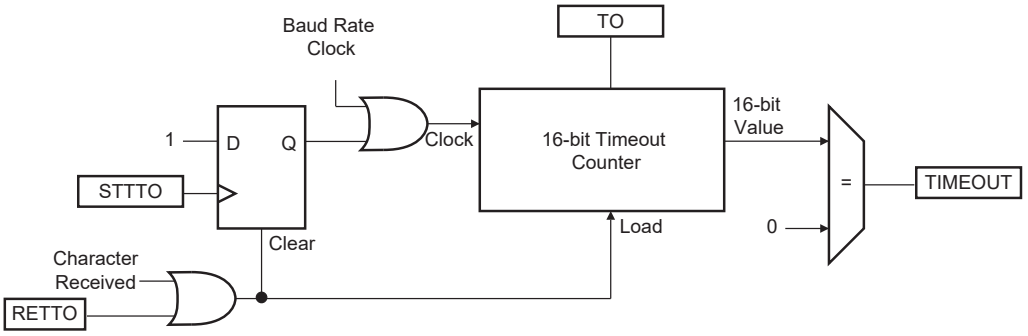
- Stop the counter clock until a new character is received. This is performed by writing a ‘1’ to FLEX_US_CR.STTTO. In this case, the idle state on RXD before a new character is received does not provide a timeout. This prevents having to handle an interrupt before a character is received and enables waiting for the next idle state on RXD after a frame is received.

© 2026 Microchip Technology Inc. and its subsidiaries

- Obtain an interrupt while no character is received. This is performed by writing a '1' to FLEX_US_CR.RETTO. In this case, the counter starts counting down immediately from the value TO. This generates a periodic interrupt so that a user timeout can be handled, for example when no key is pressed on a keyboard.

The following figure shows the block diagram of the Receiver Timeout feature.

Figure 33-23. Receiver Timeout Block Diagram



The following table gives the maximum timeout period for some standard baud rates.

Table 33-9. Maximum Timeout Period

Baud Rate (bit/s)	Bit Time (µs)	Timeout (ms)
600	1,667	109,225
1,200	833	54,613
2,400	417	27,306
4,800	208	13,653
9,600	104	6,827
14,400	69	4,551
19,200	52	3,413
28,800	35	2,276
38,400	26	1,704
56,000	18	1,170
57,600	17	1,138
200,000	5	328

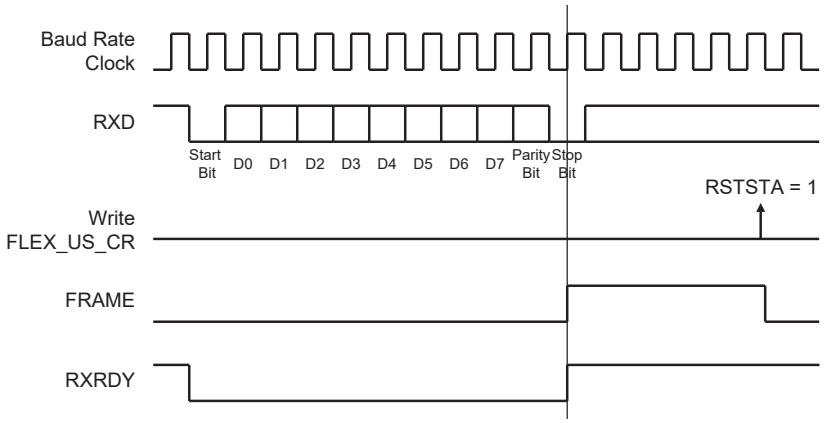
33.7.3.12. Framing Error

The receiver is capable of detecting framing errors. A framing error happens when the stop bit of a received character is detected at level 0. This can occur if the receiver and the transmitter are fully desynchronized.

A framing error is reported on the FLEX_US_CSR.FRAME bit. The FRAME bit is asserted in the middle of the stop bit as soon as the framing error is detected. It is cleared by writing a one to the FLEX_US_CR.RSTSTA bit.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-24. Framing Error Status



33.7.3.13. Transmit Break

The user can request the transmitter to generate a break condition on the TXD line. A break condition drives the TXD line low during at least one complete character. It appears the same as a 0x00 character sent with the parity and the stop bits to 0. However, the transmitter holds the TXD line at least during one character until the user requests the break condition to be removed.

A break is transmitted by setting the FLEX_US_CR.STTBRK bit. This can be done at any time, either while the transmitter is empty (no character in either the shift register or in FLEX_US_THR) or when a character is being transmitted. If a break is requested while a character is being shifted out, the character is first completed before the TXD line is held low.

Once the Start Break command is requested, further Start Break commands are ignored until the end of the break is completed.

The break condition is removed by setting the FLEX_US_CR.STPBRK bit. If the Stop Break command is requested before the end of the minimum break duration (one character, including start, data, parity and stop bits), the transmitter ensures that the break condition completes.

The transmitter considers the break as though it is a character, i.e., the Start Break and Stop Break commands are processed only if the FLEX_US_CSR.TXRDY bit = 1 and the start of the break condition clears the TXRDY and TXEMPTY bits as if a character was processed.

Setting both the FLEX_US_CR.STTBRK and FLEX_US_CR.STPBRK bits can lead to an unpredictable result. All Stop Break commands requested without a previous Start Break command are ignored. A byte written into the Transmit Holding register while a break is pending, but not started, is ignored.

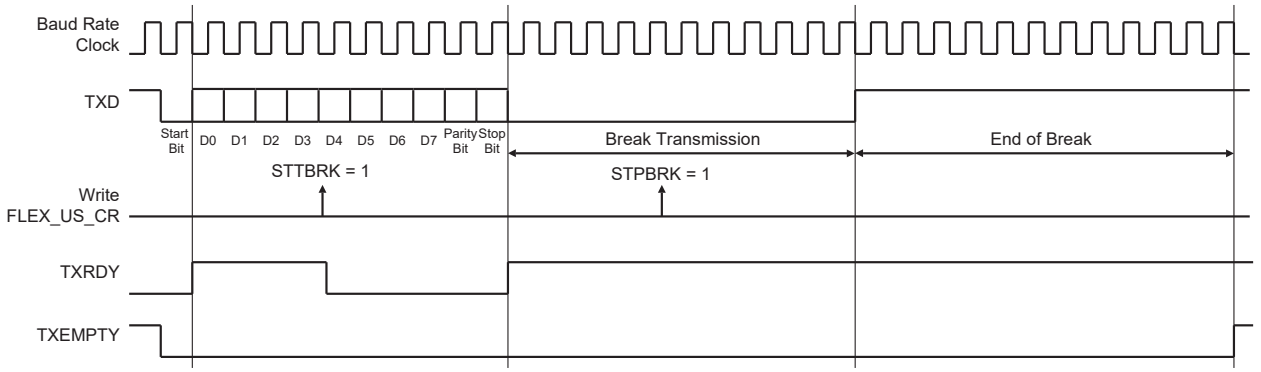
After the break condition, the transmitter returns the TXD line to 1 for a minimum of 12 bit times. Thus, the transmitter ensures that the remote receiver detects correctly the end of break and the start of the next character. If the timeguard is programmed with a value higher than 12, the TXD line is held high for the timeguard period.

After holding the TXD line for this period, the transmitter resumes normal operations.

The following figure illustrates the effect of both the Start Break (STTBRK) and Stop Break (STPBRK) commands on the TXD line.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-25. Break Transmission



33.7.3.14. Receive Break

The receiver detects a break condition when all data, parity and stop bits are low. This corresponds to detecting a framing error with data to 0x00, but FRAME remains low.

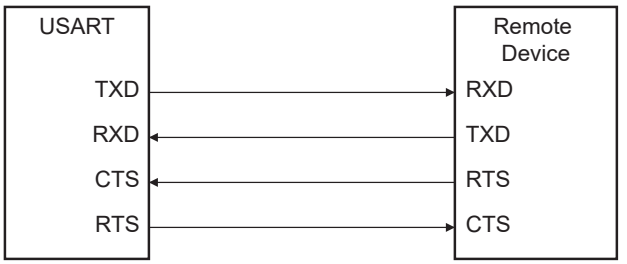
When the low stop bit is detected, the receiver asserts the FLEX_US_CSR.RXBRK bit. FLEX_US_CSR.RXBRK may be cleared by setting the FLEX_US_CR.RSTSTA bit.

An end of receive break is detected by a high level for at least 2/16ths of a bit period in Asynchronous operating mode or one sample at high level in Synchronous operating mode. The end of break detection also asserts the RXBRK bit.

33.7.3.15. Hardware Handshaking

The USART features a hardware handshaking out-of-band flow control. The RTS and CTS pins are used to connect with the remote device, as shown in the following figure.

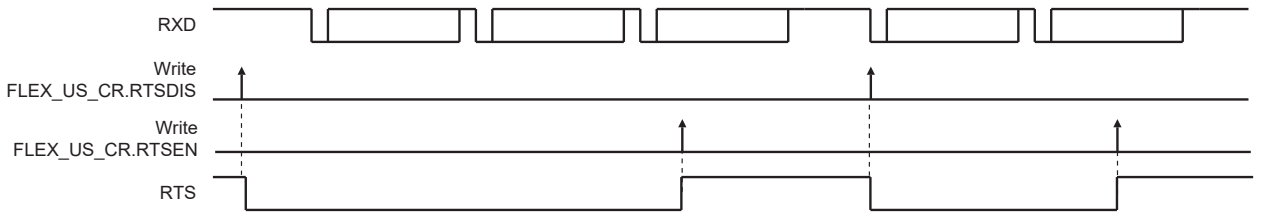
Figure 33-26. Connection with a Remote Device for Hardware Handshaking



Setting the USART to operate with hardware handshaking is performed by writing the FLEX_US_MR.USART_MODE field to the value 0x2.

The USART behavior when hardware handshaking is enabled is the same as the behavior in standard Synchronous or Asynchronous mode, except that the receiver drives the RTS pin as described below and the level on the CTS pin modifies the behavior of the transmitter as described below. Using this mode requires using the PDC channel for reception. The transmitter can handle hardware handshaking in any case.

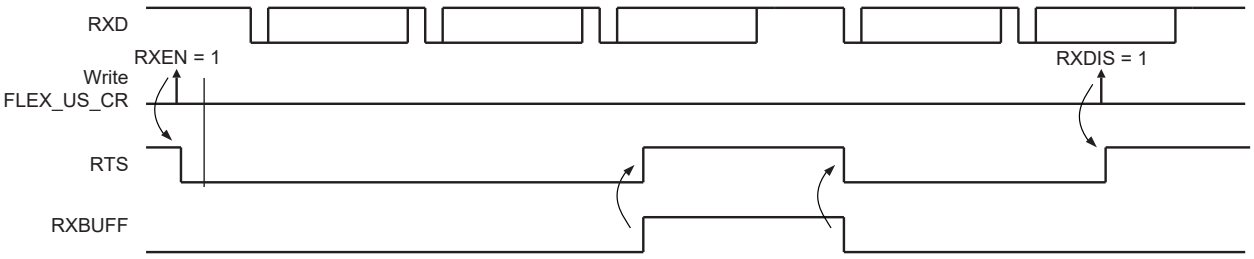
Figure 33-27. RTS Line Software Control when FLEX_US_MR.USART_MODE = 2



© 2026 Microchip Technology Inc. and its subsidiaries

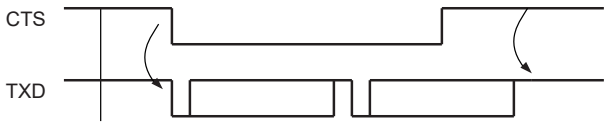
The following figure shows how the receiver operates if hardware handshaking is enabled. The RTS pin is driven high if the receiver is disabled and if the status RXBUFF (Receive Buffer Full) coming from the PDC channel is high. Normally, the remote device does not start transmitting while its CTS pin (driven by RTS) is high. As soon as the receiver is enabled, the RTS falls, indicating to the remote device that it can start transmitting. Defining a new buffer to the PDC clears the status bit RXBUFF and, as a result, asserts the pin RTS low.

Figure 33-28. Receiver Behavior when Operating with Hardware Handshaking



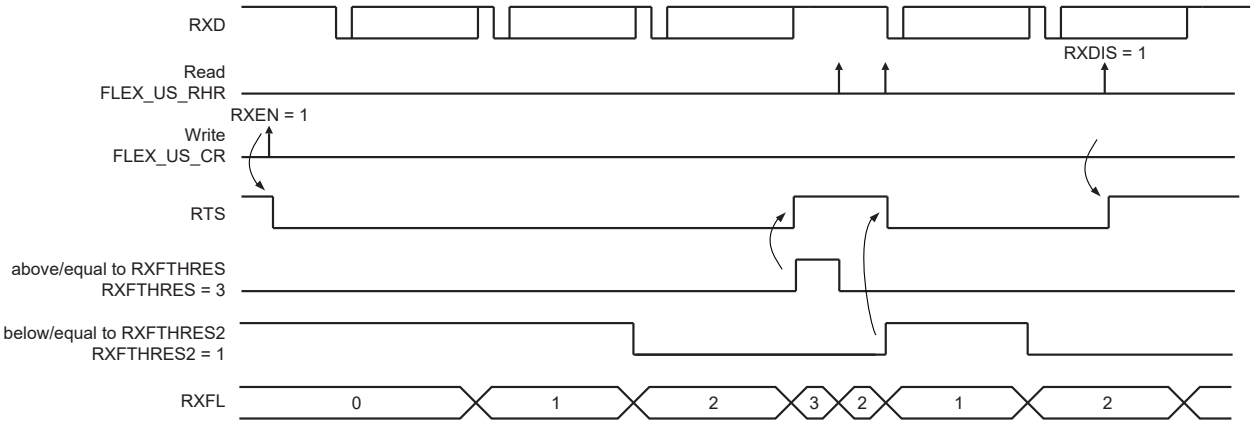
The following figure shows how the transmitter operates if hardware handshaking is enabled. The CTS pin disables the transmitter. If a character is being processed, the transmitter is disabled only after the completion of the current character and transmission of the next character happens as soon as the pin CTS falls.

Figure 33-29. Transmitter Behavior when Operating with Hardware Handshaking



If USART FIFOs are enabled (bit FLEX_US_CR.FIFOEN), the RTS pin can be controlled by the USART Receive FIFO thresholds. The RTS pin control through Receive FIFO thresholds can be activated with the FLEX_US_FMR.FRTSC bit. Once activated, the RTS pin will be controlled by Receive FIFO thresholds, set to level 1 each time RXFTHRES is reached and set to level '0' each time RXFTHRES2 is reached (and RXFTHRES is not reached).

Figure 33-30. Receiver Behavior When FIFO Enabled and FRTSC Set to '1'



Note: In this mode, RXFTHRES must be > RXFTHRES2.

33.7.4. ISO7816 Mode

The USART features an ISO7816-compatible operating mode. This mode permits interfacing with smart cards and Security Access Modules (SAM) communicating through an ISO7816 link. Both T = 0 and T = 1 protocols defined by the ISO7816 specification are supported.

© 2026 Microchip Technology Inc. and its subsidiaries

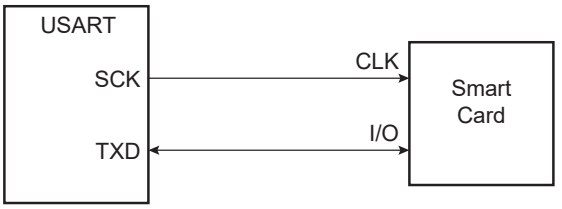
Setting the USART in ISO7816 mode is performed by writing the FLEX_US_MR.USART_MODE field to the value 0x4 for protocol T = 0 and to the value 0x6 for protocol T = 1.

33.7.4.1. ISO7816 Mode Overview

The ISO7816 is a half-duplex communication on only one bidirectional line. The baud rate is determined by a division of the clock provided to the remote device (see figure in section Baud Rate Generator).

The USART connects to a smart card as shown in the following figure. The TXD line becomes bidirectional and the baud rate generator feeds the ISO7816 clock on the SCK pin. As the TXD pin becomes bidirectional, its output remains driven by the output of the transmitter but only when the transmitter is active while its input is directed to the input of the receiver. The USART is considered as the host of the communication as it generates the clock.

Figure 33-31. Connection of a Smart Card to the USART



When operating in ISO7816, either in T = 0 or T = 1 modes, the character format is partially predefined. The configuration is forced to 8 data bits, and 1 or 2 stop bits, regardless of the values programmed in the CHRL, MODE9 and CHMODE fields. MSBF can be used to transmit LSB or MSB first. The bit INVDATA can be used to transmit in Normal or Inverse mode.

The USART cannot operate concurrently in both Receiver and Transmitter modes as the communication is unidirectional at a time. It has to be configured according to the required mode by enabling or disabling either the receiver or the transmitter as desired. Enabling both the receiver and the transmitter at the same time in ISO7816 mode may lead to unpredictable results.

The ISO7816 specification defines an inverse transmission format. Data bits of the character must be transmitted on the I/O line at their negative value.

33.7.4.2. Protocol T = 0

In T = 0 protocol, a character is made up of 1 start bit, 8 data bits, 1 parity bit and 1 guard time, which lasts two bit times. The transmitter shifts out the bits and does not drive the I/O line during the guard time.

If no parity error is detected, the I/O line remains at 1 during the guard time and the transmitter can continue with the transmission of the next character, as shown in the following figure.

If a parity error is detected by the receiver, it drives the I/O line to 0 during the guard time, as shown in figure "T = 0 Protocol with Parity Error" below. This error bit is also named NACK, for Non Acknowledge. In this case, the character lasts 1 bit time more, as the guard time length is the same and is added to the error bit time which lasts 1 bit time.

When the USART is the receiver and it detects an error, it does not load the erroneous character in the Receive Holding register (FLEX_US_RHR). It appropriately sets the PARE bit in the Status register (FLEX_US_CSR) so that the software can handle the error.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-32. T = 0 Protocol without Parity Error

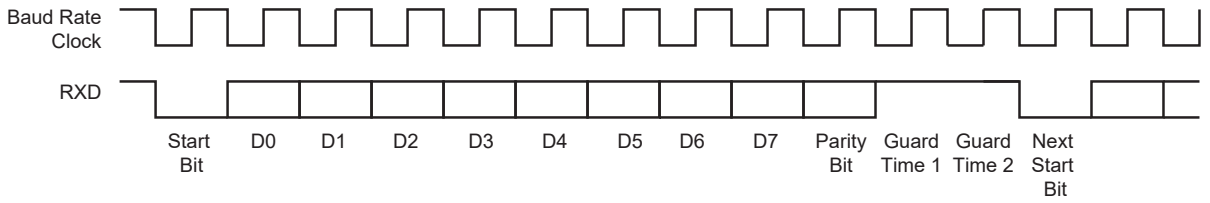
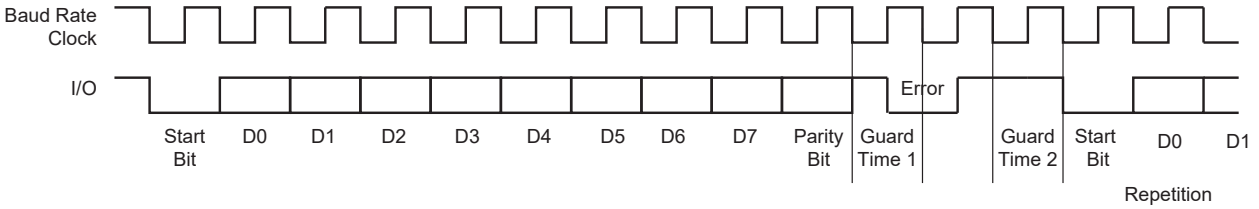


Figure 33-33. T = 0 Protocol with Parity Error



33.7.4.2.1. Receive Error Counter

The USART receiver also records the total number of errors. This can be read in the Number of Error (FLEX_US_NER) register. The NB_ERRORS field can record up to 255 errors. Reading FLEX_US_NER automatically clears the NB_ERRORS field.

33.7.4.2.2. Receive NACK Inhibit

The USART can be configured to inhibit an error. This is done by writing a '1' to FLEX_US_MR.INACK. In this case, no error signal is driven on the I/O line even if a parity bit is detected.

Moreover, if INACK = 1, the erroneous received character is stored in the Receive Holding register as if no error occurred, and the RXRDY bit rises.

33.7.4.2.3. Transmit Character Repetition

When the USART is transmitting a character and gets a NACK, it can automatically repeat the character before moving on to the next one. Repetition is enabled by writing the FLEX_US_MR.MAX_ITERATION field at a value higher than 0. Each character can be transmitted up to eight times: the first transmission plus seven repetitions.

If MAX_ITERATION does not equal zero, the USART repeats the character as many times as the value loaded in MAX_ITERATION.

When the USART repetition number reaches MAX_ITERATION, and the last repeated character is not acknowledged, the FLEX_US_CSR.ITER bit is set. If the repetition of the character is acknowledged by the receiver, the repetitions are stopped and the iteration counter is cleared.

The FLEX_US_CSR.ITER bit can be cleared by writing the FLEX_US_CR.RSTIT bit to 1.

33.7.4.2.4. Disable Successive Receive NACK

The receiver can limit the number of successive NACKs sent back to the remote transmitter. This is programmed by setting the FLEX_US_MR.DSNACK bit. The maximum number of NACKs transmitted is programmed in the MAX_ITERATION field. As soon as MAX_ITERATION is reached, no error signal is driven on the I/O line and the FLEX_US_CSR.ITER bit is set.

33.7.4.3. Protocol T = 1

When operating in ISO7816 protocol T = 1, the transmission is similar to an asynchronous format with only one stop bit. The parity is generated when transmitting and checked when receiving. Parity error detection sets the FLEX_US_CSR.PARE bit.

33.7.5. IrDA Mode

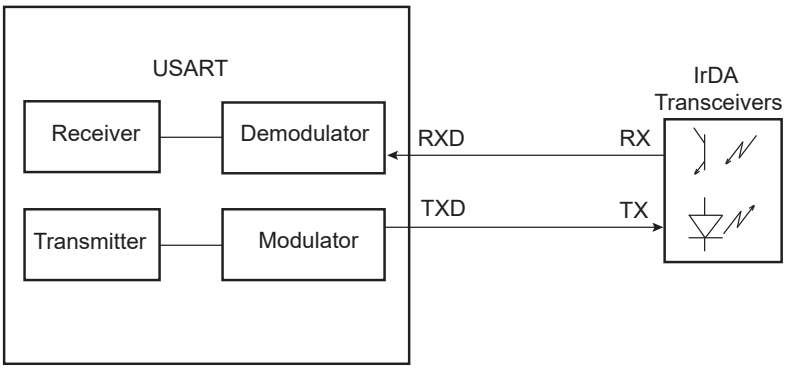
The USART features an IrDA mode supplying half-duplex point-to-point wireless communication. It embeds the modulator and demodulator which allows a glueless connection to the infrared

© 2026 Microchip Technology Inc. and its subsidiaries

transceivers, as shown in the following figure. The modulator and demodulator are compliant with the IrDA specification version 1.1 and support data transfer speeds ranging from 2.4 kbit/s to 115.2 kbit/s.

The USART IrDA mode is enabled by setting the FLEX_US_MR.USART_MODE field to the value 0x8. The IrDA Filter register (FLEX_US_IF) allows configuring the demodulator filter. The USART transmitter and receiver operate in a normal Asynchronous mode and all parameters are accessible. Note that the modulator and the demodulator are activated.

Figure 33-34. Connection to IrDA Transceivers



The receiver and the transmitter must be enabled or disabled according to the direction of the transmission to be managed.

To receive IrDA signals, the following needs to be done:

- Disable TX and Enable RX
- Configure the TXD pin as PIO and set it as an output to 0 (to avoid LED transmission). Disable the internal pullup (better for power consumption).
- Receive data

33.7.5.1. IrDA Modulation

For baud rates up to and including 115.2 kbit/s, the RZI modulation scheme is used. “0” is represented by a light pulse of 3/16th of a bit time. Some examples of signal pulse duration are shown in the following table.

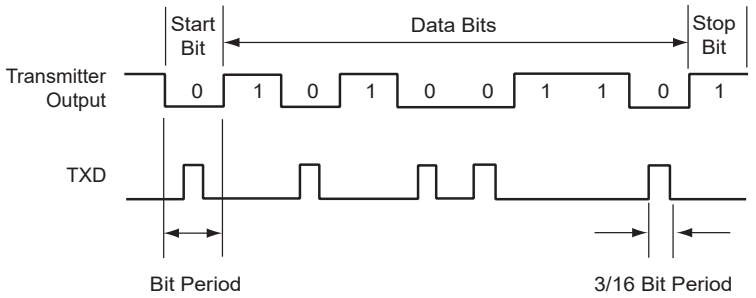
Table 33-10. IrDA Pulse Duration

Baud Rate	Pulse Duration (3/16)
2.4 kbit/s	78.13 µs
9.6 kbit/s	19.53 µs
19.2 kbit/s	9.77 µs
38.4 kbit/s	4.88 µs
57.6 kbit/s	3.26 µs
115.2 kbit/s	1.63 µs

The following figure shows an example of character transmission.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-35. IrDA Modulation



33.7.5.2. IrDA Baud Rate

The following table gives some examples of CD values, baud rate error and pulse duration. Note that the requirement on the maximum acceptable error of ±1.87% must be met.

Table 33-11. IrDA Baud Rate Error

Peripheral Clock	Baud Rate (bit/s)	CD	Baud Rate Error	Pulse Time (µs)
3,686,400	115,200	2	0.00%	1.63
20,000,000	115,200	11	1.38%	1.63
32,768,000	115,200	18	1.25%	1.63
40,000,000	115,200	22	1.38%	1.63
3,686,400	57,600	4	0.00%	3.26
20,000,000	57,600	22	1.38%	3.26
32,768,000	57,600	36	1.25%	3.26
40,000,000	57,600	43	0.93%	3.26
3,686,400	38,400	6	0.00%	4.88
20,000,000	38,400	33	1.38%	4.88
32,768,000	38,400	53	0.63%	4.88
40,000,000	38,400	65	0.16%	4.88
3,686,400	19,200	12	0.00%	9.77
20,000,000	19,200	65	0.16%	9.77
32,768,000	19,200	107	0.31%	9.77
40,000,000	19,200	130	0.16%	9.77
3,686,400	9,600	24	0.00%	19.53
20,000,000	9,600	130	0.16%	19.53
32,768,000	9,600	213	0.16%	19.53
40,000,000	9,600	260	0.16%	19.53
3,686,400	2,400	96	0.00%	78.13
20,000,000	2,400	521	0.03%	78.13
32,768,000	2,400	853	0.04%	78.13

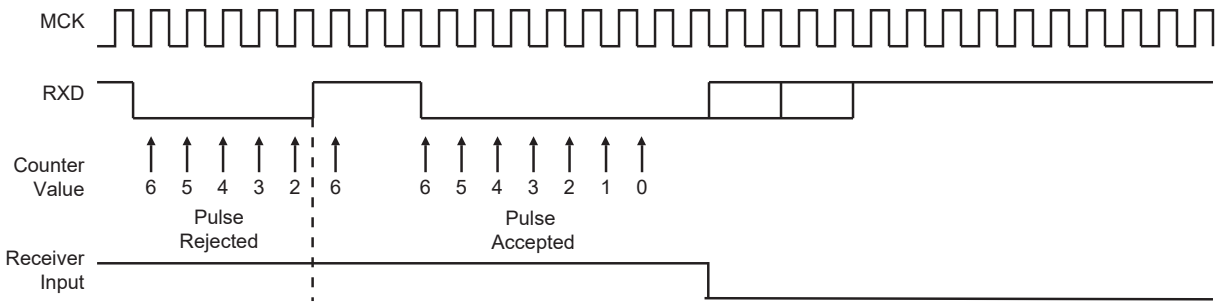
33.7.5.3. IrDA Demodulator

The demodulator is based on the IrDA Receive filter comprised of an 8-bit down counter which is loaded with the value programmed in FLEX_US_IF. When a falling edge is detected on the RXD pin, the Filter Counter starts counting down at the peripheral clock speed. If a rising edge is detected on the RXD pin, the counter stops and is reloaded with FLEX_US_IF. If no rising edge is detected when the counter reaches 0, the input of the receiver is driven low during one bit time.

The following figure illustrates the operations of the IrDA demodulator.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-36. IrDA Demodulator Operations



The programmed value in the FLEX_US_IF register must always meet the following criteria:

$t_{peripheral\ clock} \times (IRDA\_FILTER + 3) < 1.41\ \mu s$

As the IrDA mode uses the same logic as the ISO7816, note that the FLEX_US_FIDI.FI_DI_RATIO field must be set to a value higher than 0 to make sure IrDA communications operate correctly.

33.7.6. OOK Modulation and Demodulation

The USART has the capability to modulate the TXD line according to OOK modulation standard.

When the bit FLEX_US_MR.OOKEN=1 and the field FLEX_US_MR.USART_MODE is lower than 4, the TXD line is modulated if a logical 0 is transmitted, and not modulated if a logical 1 is transmitted.

The modulation frequency is adjusted by configuring the field FLEX_US_FIDI.OOKFREQ (see FLEX_US_FIDI) when the bit FLEX_US_MR.OOKEN=1.

The modulation frequency is calculated as per the following formula:

$$\text{OOK Frequency} = \frac{\text{Selected Clock}}{(\text{OOKFREQ} + 1)}$$

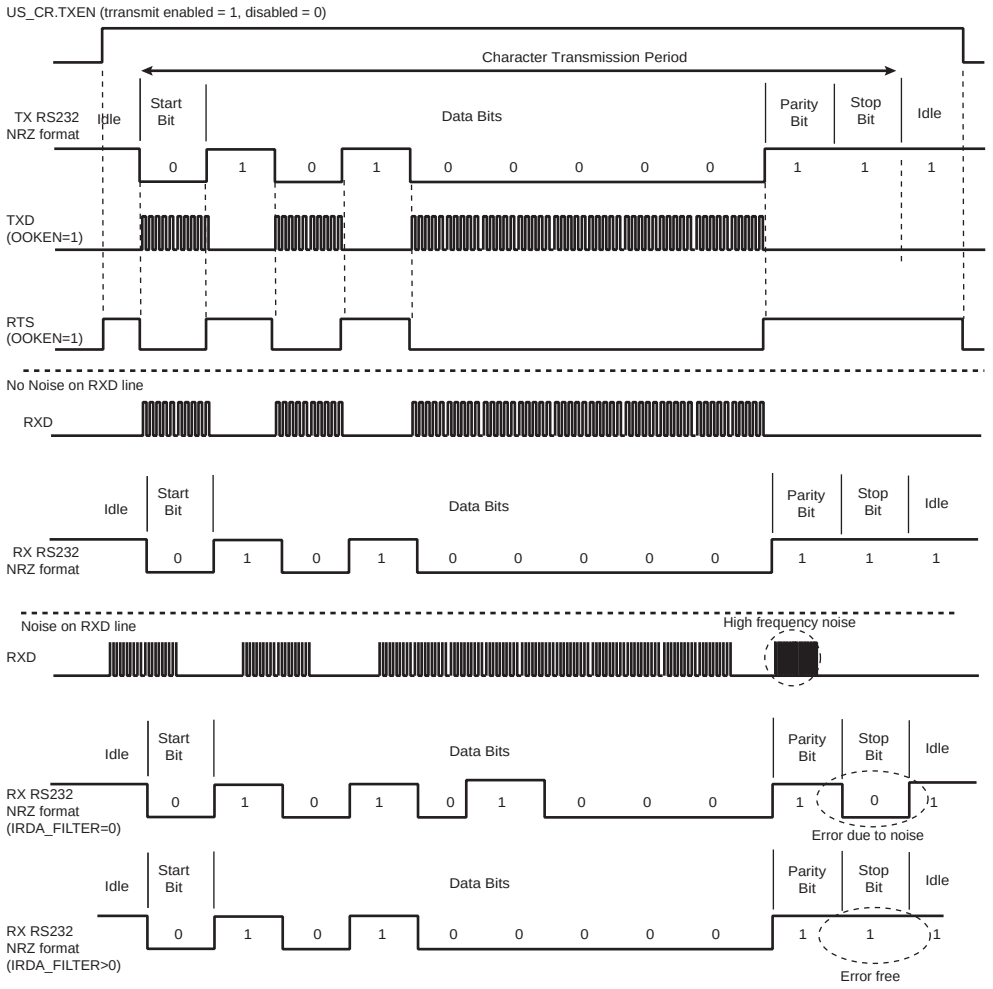
The selected clock can be defined in US_MR.USCLKS (see Baud Rate Generator).

When the bit FLEX_US_MR.OOKEN=1, the RXD line is demodulated. It is possible to configure a low-pass filter before entering the OOK demodulation logic. The low-pass filter is active when the field FLEX_US_IF.IRDA_FILTER is greater than 0 (see FLEX_US_IF).

If the bit US_MR.OOKEN=1, the RTS line is driven to 1 when a character is currently transmitted on the TXD line and the data bit being transmitted is 1. Therefore, during the stop bit the RTS line is driven to 1. It is driven to 0 when the internal state of the transmitter logic is in Idle mode.

© 2026 Microchip Technology Inc. and its subsidiaries

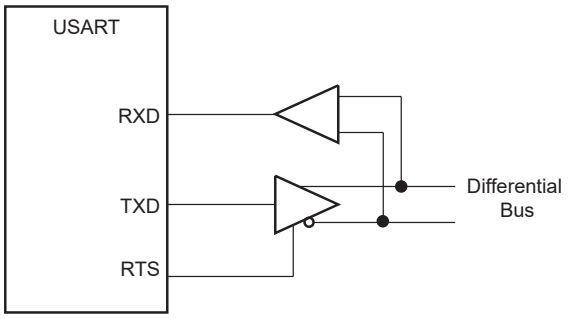
Figure 33-37. TXD Modulation and RXD Demodulation



33.7.7. RS485 Mode

The USART features the RS485 mode to enable line driver control. While operating in RS485 mode, the USART behaves as though in Asynchronous or Synchronous mode and configuration of all the parameters is possible. The difference is that the RTS pin is driven high when the transmitter is operating. The behavior of the RTS pin is controlled by the TXEMPTY bit. A typical connection of the USART to an RS485 bus is shown in the following figure.

Figure 33-38. Typical Connection to an RS485 Bus

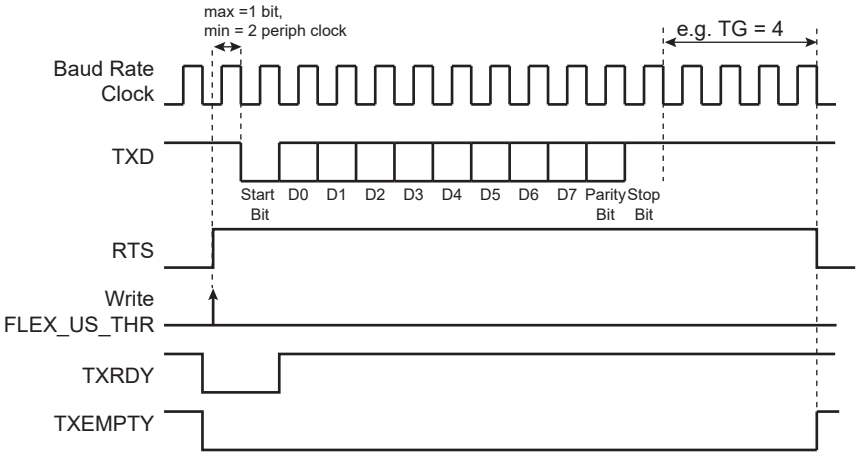


The USART is set to RS485 mode by writing the value 0x1 to the FLEX_US_MR.USART_MODE field.

© 2026 Microchip Technology Inc. and its subsidiaries

The RTS pin is at a level inverse to the TXEMPTY bit. Significantly, the RTS pin remains high when a timeguard is programmed, so that the line can remain driven after the last character completion. The following figure gives an example of the RTS waveform during a character transmission when the timeguard is enabled.

Figure 33-39. Example of RTS Drive with Timeguard



Note: In case the minimum period between RTS rising edge and START bit falling edge is not suitable for the application, it is possible to drive the RTS signal by software when FLEX_US_MR.USART_MODE= 0 or 2. FLEX_US_CR.RTSEN/RTSDIS can be configured to manage RTS.

33.7.8. USART Comparison Function on Received Character

The comparison function differs if the asynchronous partial wakeup is enabled or not.

If the asynchronous partial wakeup is disabled, the CMP flag in FLEX_US_CSR is set when the received character matches the conditions programmed in FLEX_US_CMPR. The CMP flag is set as soon as FLEX_US_RHR is loaded with the new received character. The CMP flag is cleared by writing a one to FLEX_US_CR.RSTSTA.

FLEX_US_CMPR can be programmed to provide different comparison methods:

- If VAL1 equals VAL2, then the comparison is performed on a single value and the flag is set to 1 if the received character equals VAL1.
- If VAL1 is strictly lower than VAL2, then any value between VAL1 and VAL2 sets the CMP flag.
- If VAL1 is strictly higher than VAL2, then the flag CMP is set to 1 if any received character equals VAL1 or VAL2.

When the FLEX_US_CMPR.CMPMODE bit is set to FLAG_ONLY (value 0), all received data are loaded in FLEX_US_RHR and the CMP flag provides the status of the comparison result.

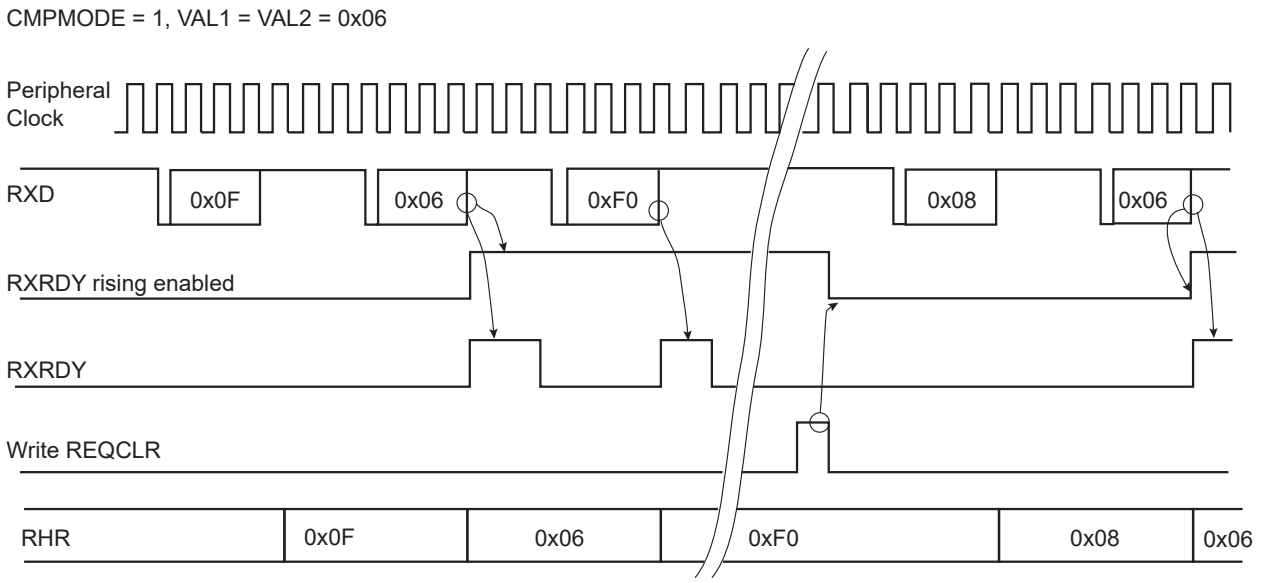
By programming the START_CONDITION.CMPMODE bit (value 1), the comparison function result triggers the start of the loading of FLEX_US_RHR (see the following figure). The trigger condition exists as soon as the received character value matches the condition defined by the programming of VAL1, VAL2 and CMPPAR in FLEX_US_CMPR. The comparison trigger event is restarted by writing a 1 to the FLEX_US_CR.REQCLR bit.

By setting the CMPMODE bit to FILTER (value 2), the comparison result triggers the start of the US_RHR loading. The trigger condition exists as soon as the received address byte value matches the conditions defined by VAL1, VAL2 in US_CMPR. The comparison trigger event is restarted automatically after the reception of the data byte. This comparison mode is only available when FLEX_US_MR.USART_MODE is set).

© 2026 Microchip Technology Inc. and its subsidiaries

The value programmed in the VAL1 and VAL2 fields must not exceed the maximum value of the received character (see CHRL field in FLEX_US_MR).

Figure 33-40. Receive Holding Register Management



33.7.9. USART Asynchronous and Partial Wakeup

Asynchronous and partial wakeup is a means of data preprocessing that qualifies an incoming event, thus allowing the USART to decide whether or not to wake up the system. This operating mode is used primarily when the system is in Wait mode (refer to the section “Power Management Controller (PMC)" for more details) but can also be enabled when the system is fully running. In any case, only the peripheral clock is modified and VDDCORE always remains active.

Asynchronous and partial wakeup requires the USART module to be programmed in UART (FLEX_US_MR.SYNC = 0).

The maximum baud rate that can be achieved when asynchronous and partial wakeup is enabled is 1200.

The FLEX_US_RHR register must be read before enabling the asynchronous and partial wakeup.

When asynchronous and partial wakeup is enabled for the USART (refer to the section “Power Management Controller (PMC)"), the PMC decodes a clock request from the USART. The request is generated as soon as there is a falling edge on the RXD line as this may indicate the beginning of a start bit. If the system is in Wait mode (processor and peripheral clocks switched off), the PMC restarts the fast RC oscillator and provides the clock only to the USART.

As soon as the clock is provided by the PMC, the USART processes the received frame and compares the received character with VAL1 and VAL2 in FLEX_US_CMPR.

The USART instructs the PMC to disable the clock if the received character value does not meet the conditions defined by VAL1 and VAL2 fields in FLEX_US_CMPR (see figure Asynchronous Event Generating Only Partial Wakeup).

If the received character value meets the conditions, the USART instructs the PMC to exit the system from Wait mode (see figure Asynchronous Wakeup Use Case Examples).

The VAL1 and VAL2 fields can be programmed to provide different comparison methods and thus matching conditions.

- If VAL1 equals VAL2, then the comparison is performed on a single value and the wakeup is triggered if the received character equals VAL1.

© 2026 Microchip Technology Inc. and its subsidiaries

- If VAL1 is strictly lower than VAL2, then any value between VAL1 and VAL2 wakes up the system.
- If VAL1 is strictly higher than VAL2, then the flag CMP is set to 1 if any received character equals VAL1 or VAL2.
- If VAL1 = 0 and VAL2 = 511, the wakeup is triggered as soon as a character is received.

The matching condition can be configured to include the parity bit (FLEX_US_CMPR.CMPPAR). Thus, if the received data matches the comparison condition defined by VAL1 and VAL2 but a parity error is encountered, the matching condition is cancelled and the UART instructs the PMC to disable the clock (see figure Asynchronous Event Generating Only Partial Wakeup).

If the processor and peripherals are running, the USART can be configured in Asynchronous and Partial Wakeup mode by enabling the PMC_SLPWK_ER (refer to the section “Power Management Controller (PMC)"). When activity is detected on the receive line, the USART requests the clock from the PMC and the comparison is performed. If there is a comparison match, the USART continues to request the clock. If there is no match, the clock is switched off for the USART only, until a new activity is detected.

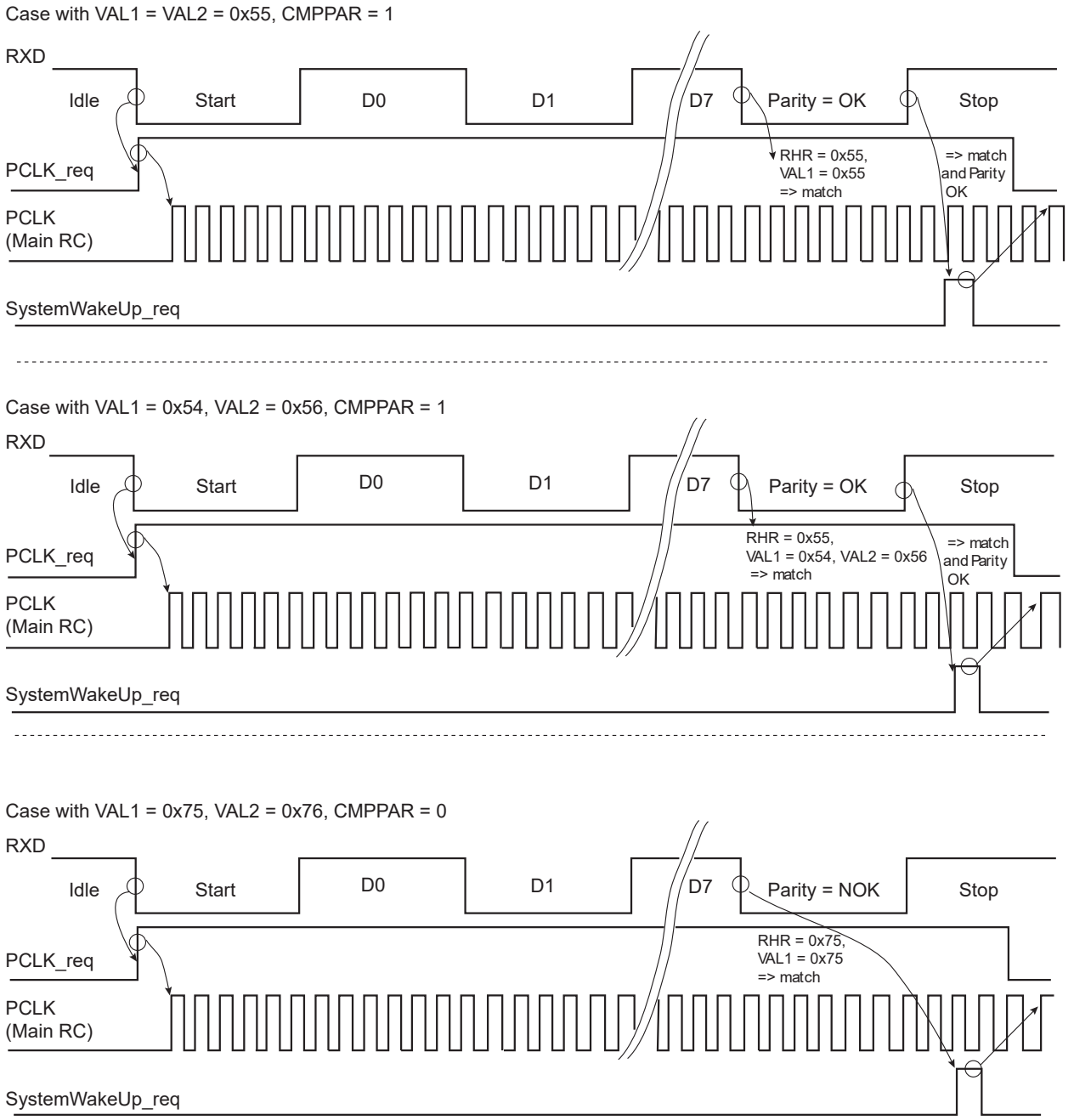
The CMPMODE configuration has no effect when Asynchronous and Partial Wakeup mode is enabled for the USART (refer to PMC_SLPWK_ER in the section “Power Management Controller (PMC)").

When the system is in Active mode and the USART enters Asynchronous and Partial Wakeup mode, the flag RXRDY must be programmed as the unique source of the USART interrupt.

When the system exits Wait mode as the result of a matching condition, the RXRDY flag is used to determine if the USART is the source for the exit from Wait mode.

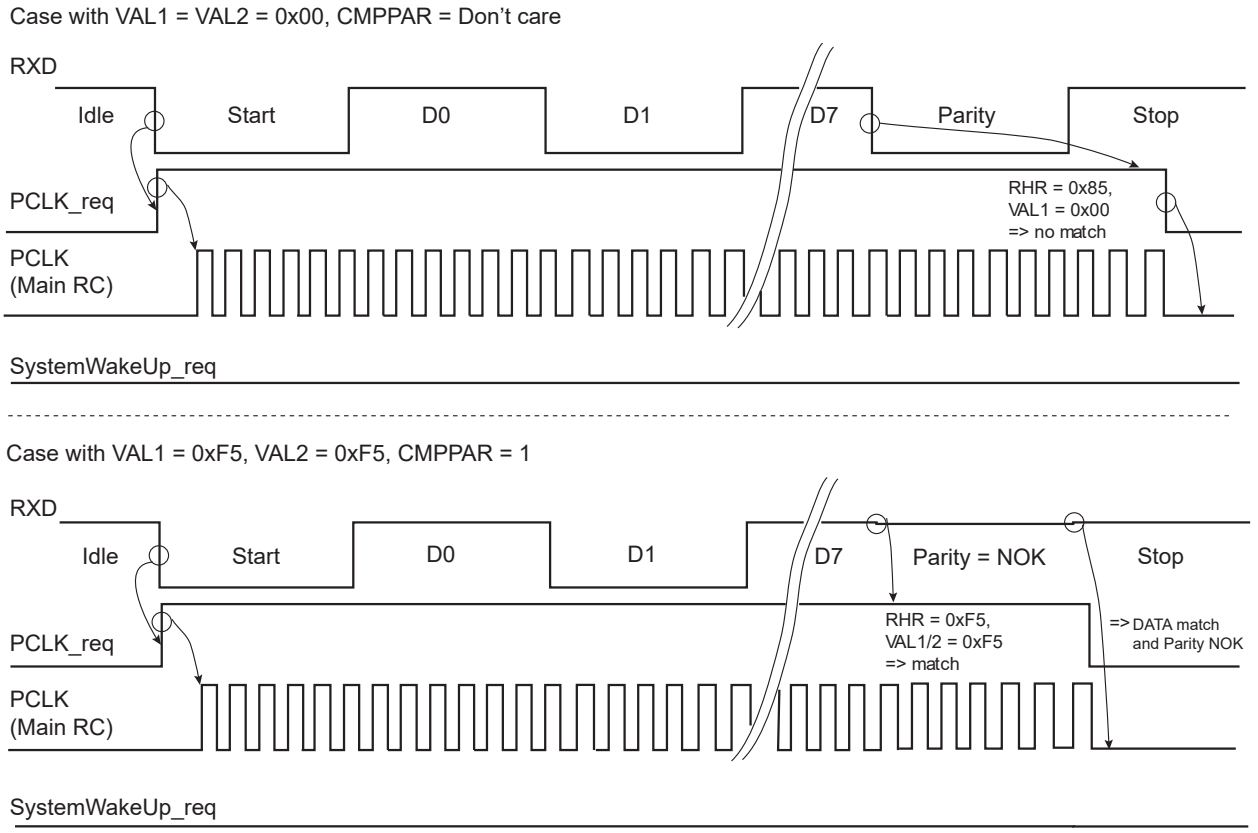
© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-41. Asynchronous Wakeup Use Case Examples



© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-42. Asynchronous Event Generating Only Partial Wakeup



33.7.10. LIN Mode

The LIN mode provides host node and client node connectivity on a LIN bus.

The LIN (Local Interconnect Network) is a serial communication protocol which efficiently supports the control of mechatronic nodes in distributed automotive applications.

The main properties of the LIN bus are:

- Single host/multiple clients concept
- Low-cost silicon implementation based on common UART/SCI interface hardware, an equivalent in software, or as a pure state machine.
- Self synchronization without quartz or ceramic resonator in the client nodes
- Deterministic signal transmission
- Low cost single-wire implementation
- Speed up to 20 kbit/s

LIN provides cost efficient bus communication where the bandwidth and versatility of CAN are not required.

The LIN mode enables processing LIN frames with a minimum of action from the microprocessor.

33.7.10.1. Modes of Operation

The USART can act either as a LIN host node or as a LIN client node.

The node configuration is chosen by setting the USART_MODE field in the USART Mode register (FLEX_US_MR):

- LIN host node (USART_MODE = 0xA)

© 2026 Microchip Technology Inc. and its subsidiaries

- LIN client node (USART_MODE = 0xB)

In order to avoid unpredictable behavior, any change of the LIN node configuration must be followed by a software reset of the transmitter and of the receiver (except the initial node configuration after a hardware reset). See Receiver and Transmitter Control.

33.7.10.2. Baud Rate Configuration

See Baud Rate in Asynchronous Mode.

- LIN host node: The baud rate is configured in FLEX_US_BRGR.
- LIN client node: The initial baud rate is configured in FLEX_US_BRGR. This configuration is automatically copied in the LIN Baud Rate register (FLEX_US_LINBRR) when writing FLEX_US_BRGR. After the synchronization procedure, the baud rate is updated in FLEX_US_LINBRR.

33.7.10.3. Receiver and Transmitter Control

See Receiver and Transmitter Control.

33.7.10.4. Character Transmission

See Transmitter Operations.

33.7.10.5. Character Reception

See Receiver Operations.

33.7.10.6. Header Transmission (Host Node Configuration)

All LIN frames start with a header sent by the host node and consisting of a Synch Break Field, a Synch Field and an Identifier Field.

So in host node configuration, the frame handling starts with the sending of the header.

The header is transmitted as soon as the identifier is written in the LIN Identifier register (FLEX_US_LINIR). At this moment, the flag TXRDY falls.

The Break Field, the Synch Field and the Identifier Field are sent automatically one after the other.

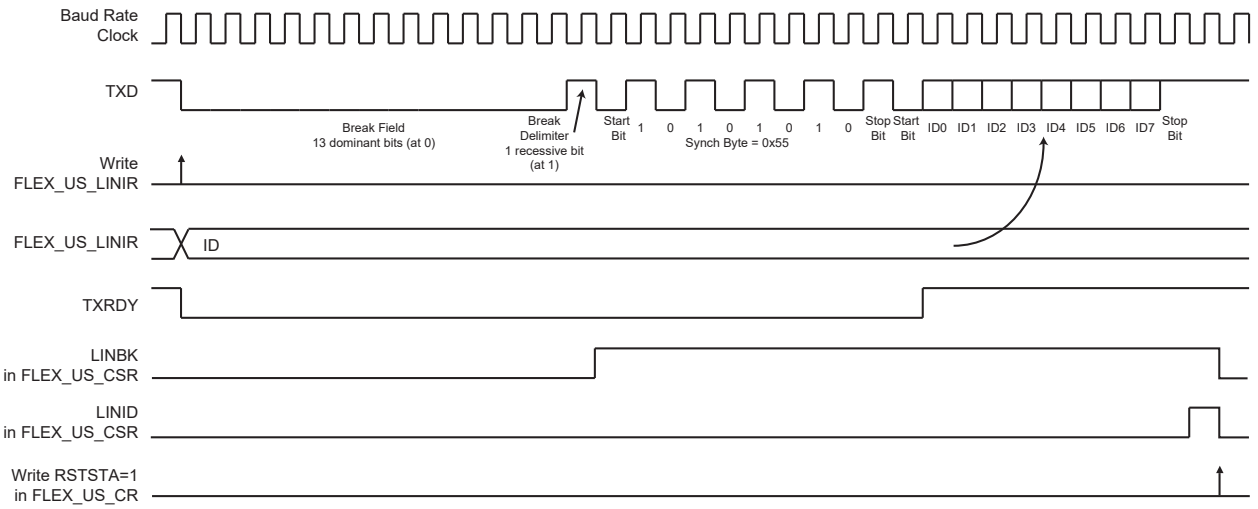
The Break Field consists of 13 dominant bits and 1 recessive bit, the Synch Field is the character 0x55 and the Identifier corresponds to the character written in the LIN Identifier register (FLEX_US_LINIR). The Identifier parity bits can be automatically computed and sent (see Identifier Parity).

The flag TXRDY rises when the identifier character is transferred into the shift register of the transmitter.

As soon as the Synch Break Field is transmitted, the FLEX_US_CSR.LINBK flag bit is set. Likewise, as soon as the Identifier Field is sent, the FLEX_US_CSR.LINID flag bit is set. These flags are reset by writing a one to the FLEX_US_CR.RSTSTA bit.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-43. Header Transmission



33.7.10.7. Header Reception (Client Node Configuration)

All the LIN frames start with a header which is sent by the host node and consists of a Synch Break Field, Synch Field and Identifier Field.

In client node configuration, the frame handling starts with the reception of the header.

The USART uses a break detection threshold of 11 nominal bit times at the actual baud rate. At any time, if 11 consecutive recessive bits are detected on the bus, the USART detects a Break Field. As long as a Break Field has not been detected, the USART stays idle and the received data are not taken in account.

When a Break Field has been detected, the FLEX_US_CSR.LINBK flag is set and the USART expects the Synch Field character to be 0x55. This field is used to update the actual baud rate in order to remain synchronized (see Client Node Synchronization). If the received Synch character is not 0x55, an Inconsistent Synch Field error is generated (see LIN Errors).

After receiving the Synch Field, the USART expects to receive the Identifier Field.

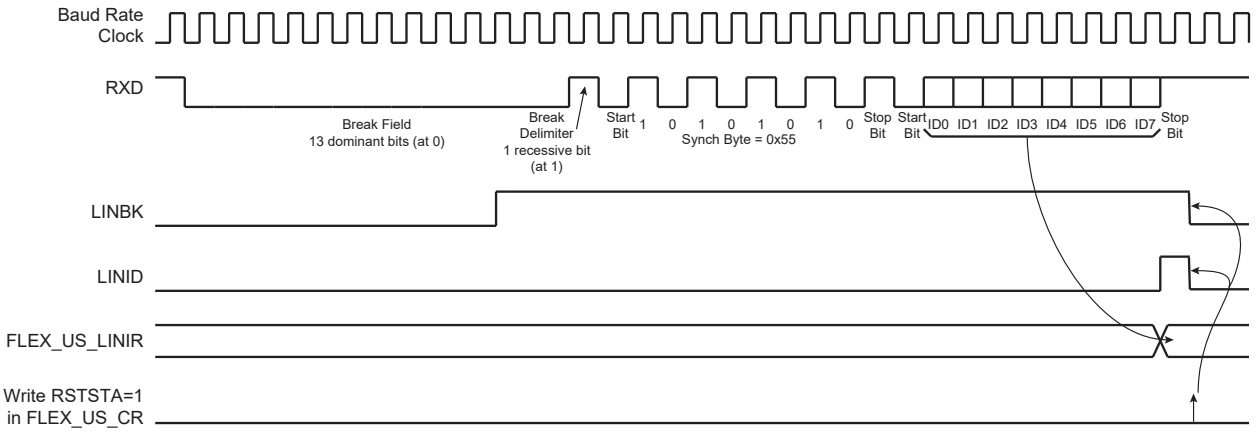
When the Identifier Field has been received, the FLEX_US_CSR.LINID flag bit is set. At this moment, the IDCHR field in the LIN Identifier register (FLEX_US_LINIR) is updated with the received character. The Identifier parity bits can be automatically computed and checked (see Identifier Parity).

If the header is not entirely received within the time given by the maximum length of the header $t_{Header\_Maximum}$, the FLEX_US_CSR.LINHTE error flag bit is set.

The flag bits LINID, LINBK and LINHTE are reset by writing a one to the FLEX_US_CR.RSTSTA bit.

© 2026 Microchip Technology Inc. and its subsidiaries

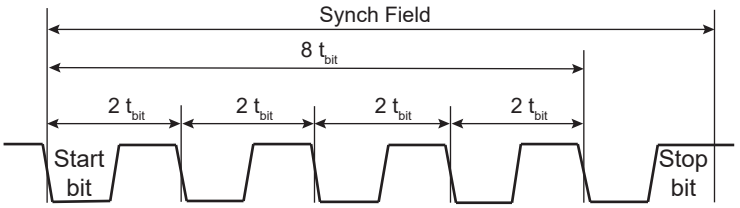
Figure 33-44. Header Reception



33.7.10.8. Client Node Synchronization

The synchronization is done only in client node configuration. The procedure is based on time measurement between the falling edges of the Synch Field. The falling edges are available in distances of 2, 4, 6 and 8 bit times.

Figure 33-45. Synch Field



The time measurement is made by a 19-bit counter driven by the sampling clock (see Baud Rate Generator).

When the start bit of the Synch Field is detected, the counter is reset. Then during the next eight t_{bit} of the Synch Field, the counter is incremented. At the end of these eight t_{bit}, the counter is stopped. At this moment, the 16 most significant bits of the counter (value divided by 8) give the new clock divider (LINCD) and the 3 least significant bits of this value (the remainder) give the new fractional part (LINFP).

Once the Synch Field has been entirely received, the clock divider (LINCD) and the fractional part (LINFP) are updated in the LIN Baud Rate register (FLEX_US_LINBRR) with the computed values, if the Synchronization is not disabled by the SYNCDIS bit in the LIN Mode register (FLEX_US_LINMR).

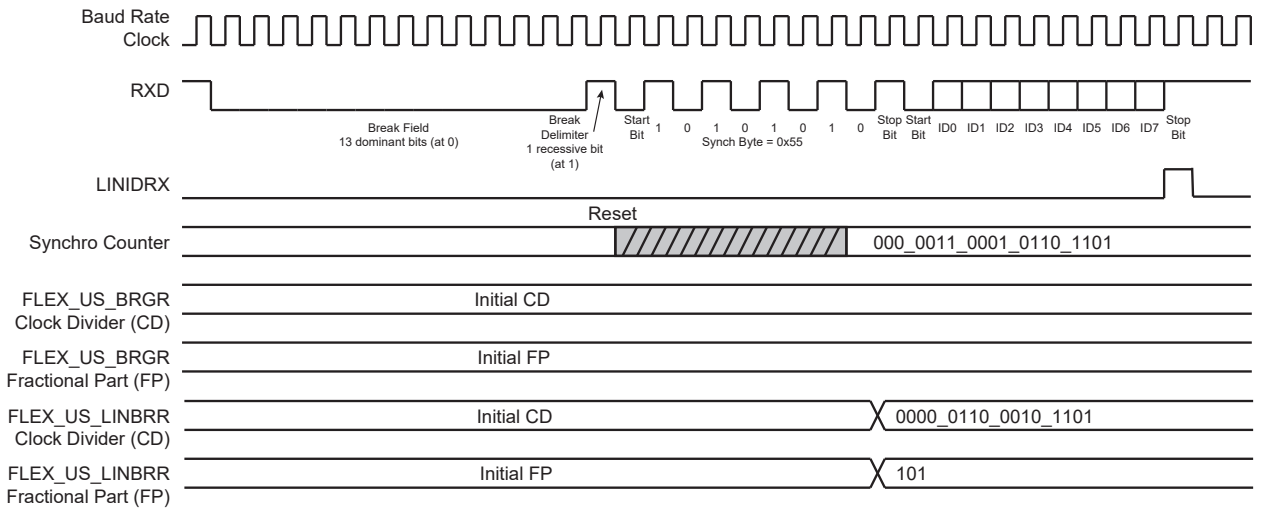
After reception of the Synch Field:

- If it appears that the computed baud rate deviation compared to the initial baud rate is superior to the maximum tolerance FTol_Unsynch (±15%), then the clock divider (LINCD) and the fractional part (LINFP) are not updated, and the FLEX_US_CSR.LINSTE error flag bit is set.
- If it appears that the sampled Synch character is not equal to 0x55, then the clock divider (LINCD) and the fractional part (LINFP) are not updated, and the FLEX_US_CSR.LINISFE error flag bit is set.

Flags LINSTE and LINISFE are reset by writing a one to the FLEX_US_CR.RSTSTA bit.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-46. Client Node Synchronization



The synchronization accuracy depends on several parameters:

- The nominal clock frequency (f_{Nom}) (the theoretical client node clock frequency)
- The baud rate
- The oversampling (OVER = 0 => 16X or OVER = 1 => 8X)

The following formula is used to compute the deviation of the client bit rate relative to the host bit rate after synchronization (f_{CLIENT} is the real client node clock frequency).

$$\text{Baud rate deviation} = \left(100 \times \frac{[\alpha \times 8 \times (2 - \text{Over}) + \beta] \times \text{Baud rate}}{8 \times f_{CLIENT}}\right)\%$$

$$\text{Baud rate deviation} = \left(100 \times \frac{[\alpha \times 8 \times (2 - \text{Over}) + \beta] \times \text{Baud rate}}{8 \times \left(\frac{f_{TOL\_UNSYNCH}}{100}\right) \times f_{Nom}}\right)\%$$

$-0.5 \leq \alpha \leq +0.5 \quad -1 < \beta < +1$

$f_{TOL\_UNSYNCH}$ is the deviation of the real client node clock from the nominal clock frequency. The LIN Standard imposes that it must not exceed ±15%. The LIN Standard imposes also that for communication between two nodes, their bit rate must not differ by more than ±2%. This means that the baud rate deviation must not exceed ±1%.

Therefore, a minimum value for the nominal clock frequency can be computed as follows:

$$f_{Nom}(\text{min}) = \left(100 \times \frac{[0.5 \times 8 \times (2 - \text{Over}) + 1] \times \text{Baud rate}}{8 \times \left(\frac{-15}{100} + 1\right) \times 1\%}\right)\text{Hz}$$

Examples:

- Baud rate = 20 kbit/s, OVER = 0 (Oversampling 16X) => f_{Nom}(min) = 2.64 MHz
- Baud rate = 20 kbit/s, OVER = 1 (Oversampling 8X) => f_{Nom}(min) = 1.47 MHz
- Baud rate = 1 kbit/s, OVER = 0 (Oversampling 16X) => f_{Nom}(min) = 132 kHz

© 2026 Microchip Technology Inc. and its subsidiaries

- Baud rate = 1 kbit/s, OVER = 1 (Oversampling 8X) => f_{Nom}(min) = 74 kHz

33.7.10.9. Identifier Parity

A protected identifier consists of two subfields: the identifier and the identifier parity. Bits 0 to 5 are assigned to the identifier, and bits 6 and 7 are assigned to the parity.

The USART interface can generate/check these parity bits, but this feature can also be disabled. The user can choose between two modes via the FLEX_US_LINMR.PARDIS bit:

- PARDIS = 0:
  - During header transmission, the parity bits are computed and sent with the six least significant bits of the IDCHR field of the LIN Identifier register (FLEX_US_LINIR). Bits 6 and 7 of this register are discarded.
  - During header reception, the parity bits of the identifier are checked. If the parity bits are wrong, an Identifier Parity error occurs (see Parity). Only the six least significant bits of the IDCHR field are updated with the received Identifier. Bits 6 and 7 are stuck to 0.
- PARDIS = 1:
  - During header transmission, all the bits of the IDCHR field of the LIN Identifier register (FLEX_US_LINIR) are sent on the bus.
  - During header reception, all the bits of the IDCHR field are updated with the received Identifier.

33.7.10.10. Node Action

Depending on the identifier, the node is affected—or not—by the LIN response. Consequently, after sending or receiving the identifier, the USART must be configured. There are three possible configurations:

- PUBLISH: the node sends the response.
- SUBSCRIBE: the node receives the response.
- IGNORE: the node is not concerned by the response, it does not send and does not receive the response.

This configuration is made by the LIN Node Action (NACT) field in USART LIN Mode Register (FLEX_US_LINMR).

Example: a LIN cluster that contains a host and two clients:

- Data transfer from the host to client 1 and to client 2:

NACT(host) = PUBLISH

NACT(client 1) = SUBSCRIBE

NACT(client 2) = SUBSCRIBE

- Data transfer from the host to client 1 only:

NACT(host) = PUBLISH

NACT(client 1) = SUBSCRIBE

NACT(client 2) = IGNORE

- Data transfer from client 1 to the host:

NACT(host) = SUBSCRIBE

NACT(client 1) = PUBLISH

NACT(client 2) = IGNORE

- Data transfer from client 1 to client 2:

© 2026 Microchip Technology Inc. and its subsidiaries

NACT(host) = IGNORE

NACT(client 1) = PUBLISH

NACT(client 2) = SUBSCRIBE

- Data transfer from client 2 to the host and to client 1:

NACT(host) = SUBSCRIBE

NACT(client 1) = SUBSCRIBE

NACT(client 2) = PUBLISH

33.7.10.11. Response Data Length

The LIN response data length is the number of data fields (bytes) of the response excluding the checksum.

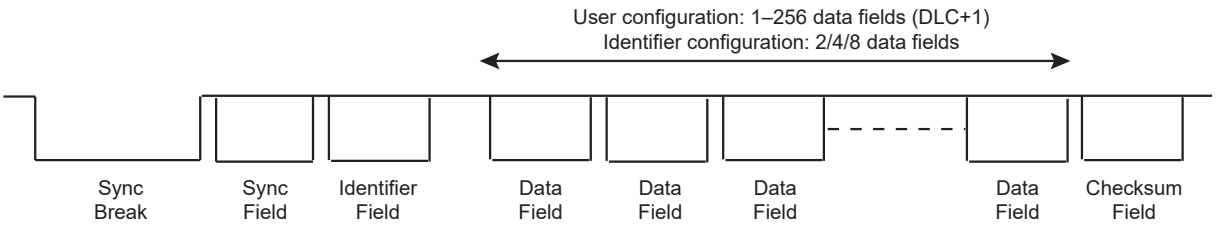
The response data length can either be configured by the user or be defined automatically by bits 4 and 5 of the Identifier (compatibility to LIN Specification 1.1). The user can choose between these two modes by the FLEX_US_LINMR.DLM bit:

- DLM = 0: The response data length is configured by the user via the FLEX_US_LINMR.DLC field. The response data length is equal to (DLC + 1) bytes. DLC can be programmed from 0 to 255, so the response can contain from 1 data byte up to 256 data bytes.
- DLM = 1: The response data length is defined by the Identifier (IDCHR in FLEX_US_LINIR) according to the table below. The FLEX_US_LINMR.DLC field is discarded. The response can contain 2 or 4 or 8 data bytes.

Table 33-12. Response Data Length if DLM = 1

IDCHR[5]	IDCHR[4]	Response Data Length (bytes)
0	0	2
0	1	2
1	0	4
1	1	8

Figure 33-47. Response Data Length



33.7.10.12. Checksum

The last field of a frame is the checksum. The checksum contains the inverted 8- bit sum with carry, over all data bytes or all data bytes and the protected identifier. Checksum calculation over the data bytes only is called classic checksum and it is used for communication with LIN 1.3 clients. Checksum calculation over the data bytes and the protected identifier byte is called enhanced checksum and it is used for communication with LIN 2.0 clients.

The USART can be configured to:

- Send/Check an Enhanced checksum automatically (CHKDIS = 0 & CHKTYP = 0)
- Send/Check a Classic checksum automatically (CHKDIS = 0 & CHKTYP = 1)
- Not send/check a checksum (CHKDIS = 1)

© 2026 Microchip Technology Inc. and its subsidiaries

This configuration is made by the Checksum Type (CHKTYP) and Checksum Disable (CHKDIS) bits of FLEX_US_LINMR.

If the checksum feature is disabled, the user can send it manually all the same, by considering the checksum as a normal data byte and by adding 1 to the response data length (see Response Data Length).

33.7.10.13. Frame Slot Mode

This mode is useful only for host nodes. It respects the following rule: each frame slot shall be longer than or equal to $t_{Frame\_Maximum}$.

If the Frame Slot mode is enabled (FSDIS = 0) and a frame transfer has been completed, the TXRDY flag is set again only after $t_{Frame\_Maximum}$ delay, from the start of frame. So the host node cannot send a new header if the frame slot duration of the previous frame is inferior to $t_{Frame\_Maximum}$.

If the Frame Slot mode is disabled (FSDIS = 1) and a frame transfer has been completed, the TXRDY flag is set again immediately.

The $t_{Frame\_Maximum}$ is calculated as follows:

If the Checksum is sent (CHKDIS = 0):

- $t_{Header\_Nominal} = 34 \times t_{bit}$
- $t_{Response\_Nominal} = 10 \times (NData + 1) \times t_{bit}$
- $t_{Frame\_Maximum} = 1.4 \times (t_{Header\_Nominal} + t_{Response\_Nominal} + 1)$(1)
- $t_{Frame\_Maximum} = 1.4 \times (34 + 10 \times (DLC + 1 + 1) + 1) \times t_{bit}$
- $t_{Frame\_Maximum} = (77 + 14 \times DLC) \times t_{bit}$

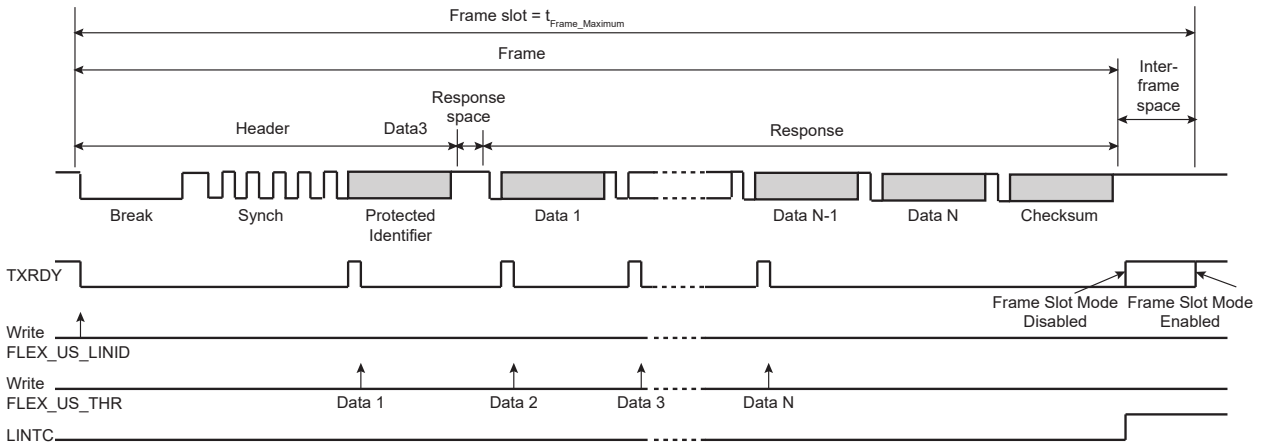
If the Checksum is not sent (CHKDIS = 1):

- $t_{Header\_Nominal} = 34 \times t_{bit}$
- $t_{Response\_Nominal} = 10 \times NData \times t_{bit}$
- $t_{Frame\_Maximum} = 1.4 \times (t_{Header\_Nominal} + t_{Response\_Nominal} + 1)$(1)
- $t_{Frame\_Maximum} = 1.4 \times (34 + 10 \times (DLC + 1) + 1) \times t_{bit}$
- $t_{Frame\_Maximum} = (63 + 14 \times DLC) \times t_{bit}$

Note:

1. The term “+1” leads to an integer result for $t_{Frame\_Maximum}$ (LIN Specification 1.3).

Figure 33-48. Frame Slot Mode



© 2026 Microchip Technology Inc. and its subsidiaries

33.7.10.14. LIN Errors

33.7.10.14.1. Bit Error

This error is generated in host or client node configuration, when the USART is transmitting and if the transmitted value on the Tx line is different from the value sampled on the Rx line. If a bit error is detected, the transmission is aborted at the next byte border.

This error is reported by the FLEX_US_CSR.LINBE flag.

33.7.10.14.2. Inconsistent Synch Field Error

This error is generated in client node configuration, if the Synch Field character received is other than 0x55.

This error is reported by the FLEX_US_CSR.LINISFE flag.

33.7.10.14.3. Identifier Parity Error

This error is generated in client node configuration, if the parity of the identifier is wrong. This error can be generated only if the parity feature is enabled (PARDIS = 0).

This error is reported by the FLEX_US_CSR.LINIPE flag.

33.7.10.14.4. Checksum Error

This error is generated in host or client node configuration, if the received checksum is wrong. This flag can be set to 1 only if the checksum feature is enabled (CHKDIS = 0).

This error is reported by the FLEX_US_CSR.LINCE flag.

33.7.10.14.5. Client Not Responding Error

This error is generated in host or client node configuration, when the USART expects a response from another node (NACT = SUBSCRIBE) but no valid message appears on the bus within the time given by the maximum length of the message frame, $t_{Frame\_Maximum}$ (see Frame Slot Mode). This error is disabled if the USART does not expect any message (NACT = PUBLISH or NACT = IGNORE).

This error is reported by the FLEX_US_CSR.LINSNRE.

33.7.10.14.6. Synch Tolerance Error

This error is generated in client node configuration if, after the clock synchronization procedure, it appears that the computed baud rate deviation compared to the initial baud rate is superior to the maximum tolerance FTol_Unsynch (±15%).

This error is reported by the FLEX_US_CSR.LINSTE flag.

33.7.10.14.7. Header Timeout Error

This error is generated in client node configuration, if the header is not entirely received within the time given by the maximum length of the header, $t_{Header\_Maximum}$.

This error is reported by the FLEX_US_CSR.LINHTE flag.

33.7.10.15. LIN Frame Handling

33.7.10.15.1. Host Node Configuration

- Write FLEX_US_CR.TXEN and FLEX_US_CR.RXEN to enable both the transmitter and the receiver.
- Write FLEX_US_MR.USART_MODE to select the LIN mode and the host node configuration.
- Write FLEX_US_BRGR.CD and FLEX_US_BRGR.FP to configure the baud rate.
- Write NACT, PARDIS, CHKDIS, CHKTYPE, DLCM, FSDIS and DLC in FLEX_US_LINMR to configure the frame transfer.
- Check that FLEX_US_CSR. TXRDY is set to 1.
- Write FLEX_US_LINIR.IDCHR to send the header.

What comes next depends on the NACT configuration:

© 2026 Microchip Technology Inc. and its subsidiaries

- Case 1: NACT = PUBLISH, the USART sends the response.
  - Wait until FLEX_US_CSR. TXRDY rises.
  - Write FLEX_US_THR.TCHR to send a byte.
  - If all the data have not been written, repeat the two previous steps.
  - Wait until FLEX_US_CSR.LINTC rises.
  - Check the LIN errors.
- Case 2: NACT = SUBSCRIBE, the USART receives the response.
  - Wait until FLEX_US_CSR.RXRDY rises.
  - Read FLEX_US_RHR.RCHR.
  - If all the data have not been read, repeat the two previous steps.
  - Wait until FLEX_US_CSR.LINTC rises.
  - Check the LIN errors.
- Case 3: NACT = IGNORE, the USART is not concerned by the response.
  - Wait until FLEX_US_CSR.LINTC rises.
  - Check the LIN errors.

Figure 33-49. Host Node Configuration, NACT = PUBLISH

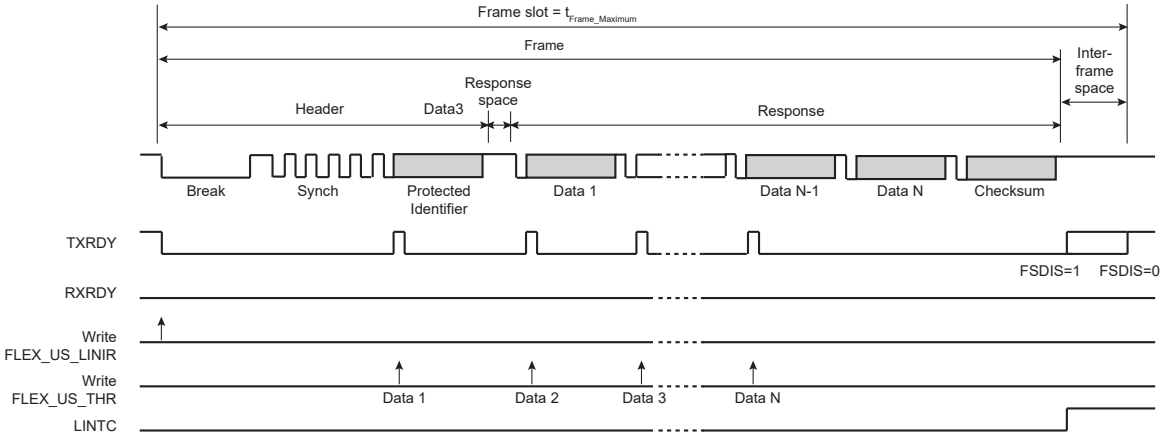
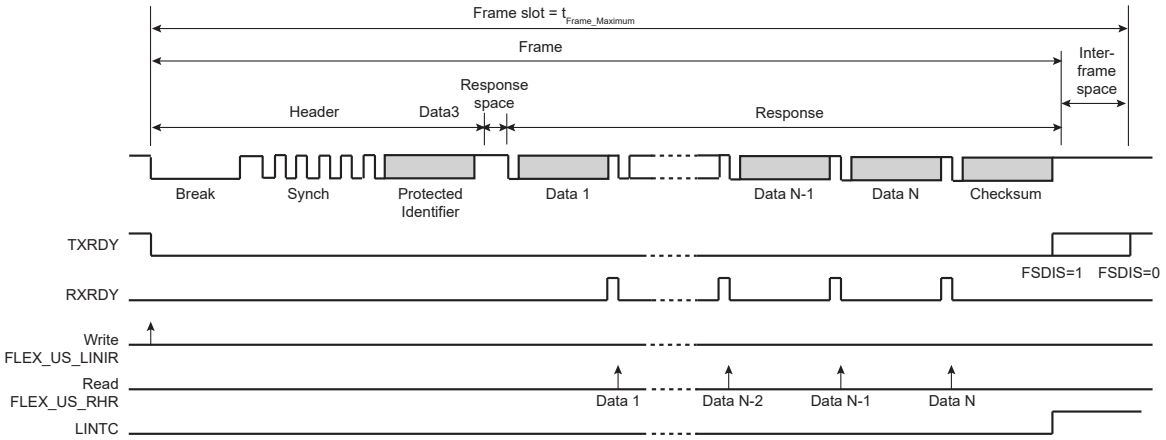
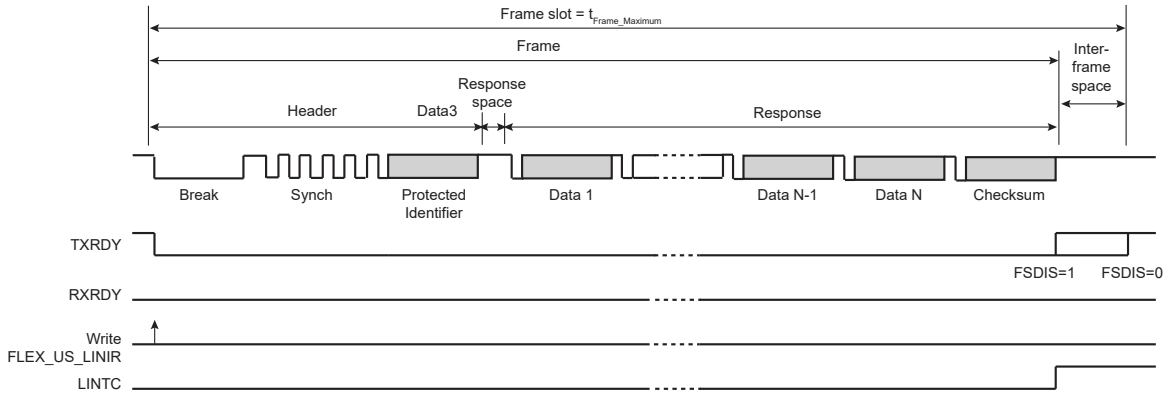


Figure 33-50. Host Node Configuration, NACT = SUBSCRIBE



© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-51. Host Node Configuration, NACT = IGNORE



33.7.10.15.2. Client Node Configuration

- Write FLEX_US_CR.TXEN and FLEX_US_CR.RXEN to enable both the transmitter and the receiver.
- Write FLEX_US_MR.USART_MODE to select the LIN mode and the client node configuration.
- Write FLEX_US_BRGR.CD and FLEX_US_BRGR.FP to configure the baud rate.
- Wait until FLEX_US_CSR.LINID rises.
- Check LINISFE and LINPE errors.
- Read FLEX_US_RHR.IDCHR.
- Write NACT, PARDIS, CHKDIS, CHKTYPE, DLCM and DLC in FLEX_US_LINMR to configure the frame transfer.

IMPORTANT: If the NACT configuration for this frame is PUBLISH, FLEX_US_LINMR must be written with NACT = PUBLISH even if this field is already correctly configured, in order to set the TXREADY flag and the corresponding write transfer request.

What comes next depends on the NACT configuration:

- Case 1: NACT = PUBLISH, the LIN controller sends the response.
  - Wait until FLEX_US_CSR.TXRDY rises.
  - Write FLEX_US_THR.TCHR to send a byte.
  - If all the data have not been written, repeat the two previous steps.
  - Wait until FLEX_US_CSR. LINTC rises.
  - Check the LIN errors.
- Case 2: NACT = SUBSCRIBE, the USART receives the response.
  - Wait until FLEX_US_CSR.RXRDY rises.
  - Read FLEX_US_RHR.RCHR.
  - If all the data have not been read, repeat the two previous steps.
  - Wait until FLEX_US_CSR.LINTC rises.
  - Check the LIN errors.
- Case 3: NACT = IGNORE, the USART is not concerned by the response.
  - Wait until FLEX_US_CSR.LINTC rises.
  - Check the LIN errors.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-52. Client Node Configuration, NACT = PUBLISH

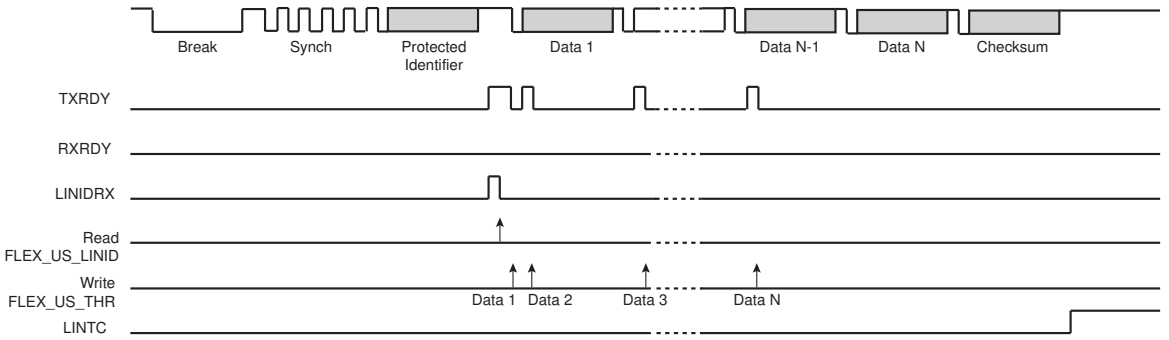


Figure 33-53. Client Node Configuration, NACT = SUBSCRIBE

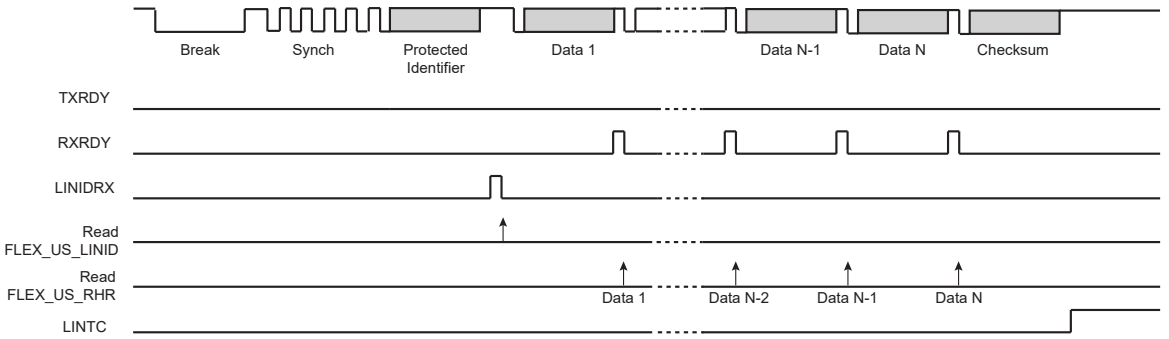
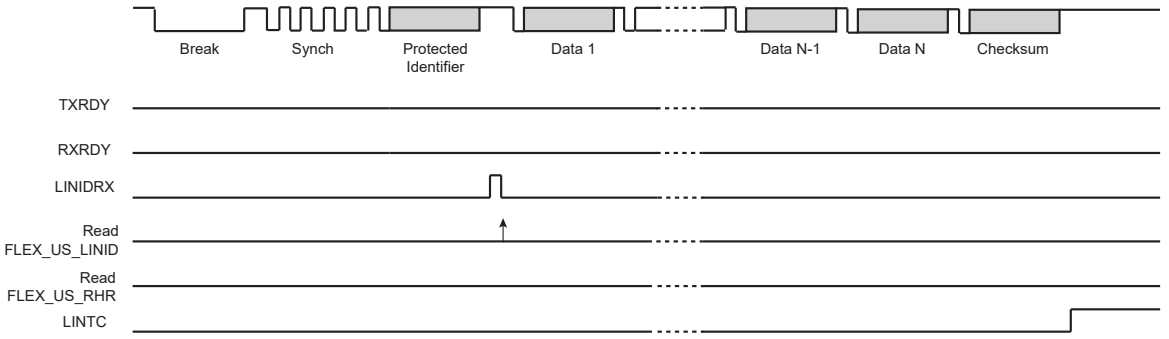


Figure 33-54. Client Node Configuration, NACT = IGNORE



33.7.10.16. LIN Frame Handling with the PDC

The USART can be used in association with the PDC in order to transfer data directly into/from the on- and off-chip memories without any processor intervention.

The PDC uses the trigger flags, TXRDY and RXRDY, to write or read into the USART. The PDC always writes in the Transmit Holding register (FLEX_US_THR) and it always reads in the Receive Holding register (FLEX_US_RHR). The size of the data written or read by the PDC in the USART is always a byte.

33.7.10.16.1. Host Node Configuration

The user can choose between two PDC modes by configuring the FLEX_US_LINMR.PDCM bit:

- PDCM = 1: The LIN configuration is stored in the WRITE buffer and it is written by the PDC in the Transmit Holding register FLEX_US_THR (instead of the LIN Mode register FLEX_US_LINMR). Because the PDC transfer size is limited to a byte, the transfer is split into two accesses. During the first access, the NACT, PARDIS, CHKDIS, CHKTYP, DLM and FSDIS bits are written. During the second access, the 8-bit DLC field is written.

© 2026 Microchip Technology Inc. and its subsidiaries

- PDCM = 0: The LIN configuration is not stored in the WRITE buffer and it must be written by the user in FLEX_US_LINMR.

The WRITE buffer also contains the Identifier and the data, if the USART sends the response (NACT = PUBLISH).

The READ buffer contains the data if the USART receives the response (NACT = SUBSCRIBE).

Figure 33-55. Host Node with PDC (PDCM = 1)

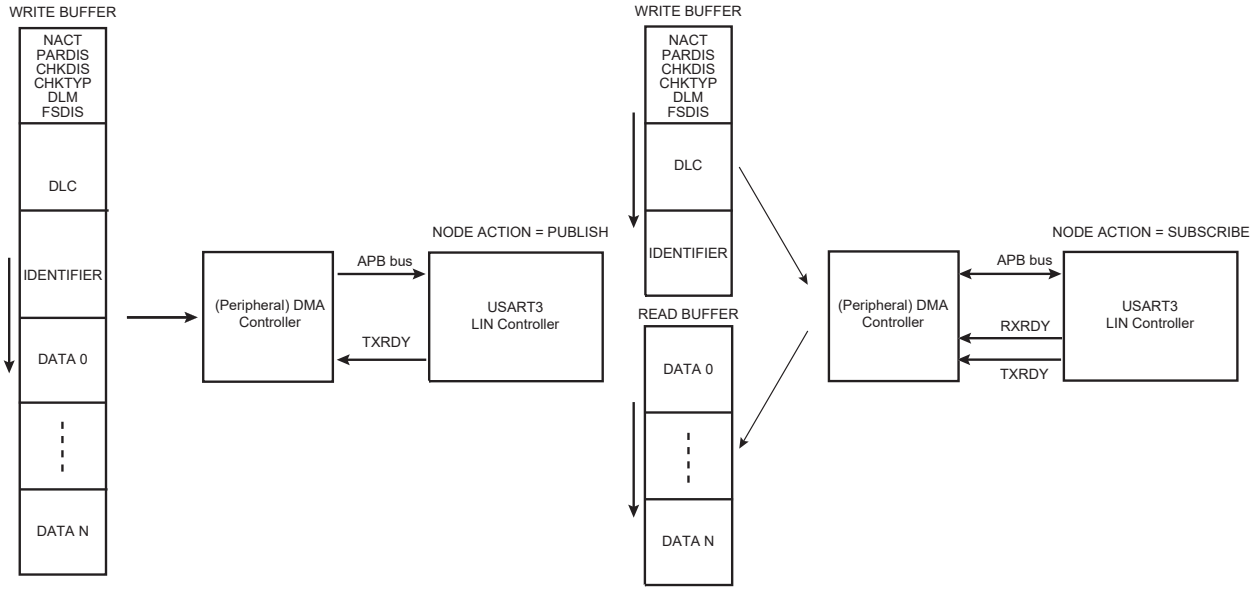
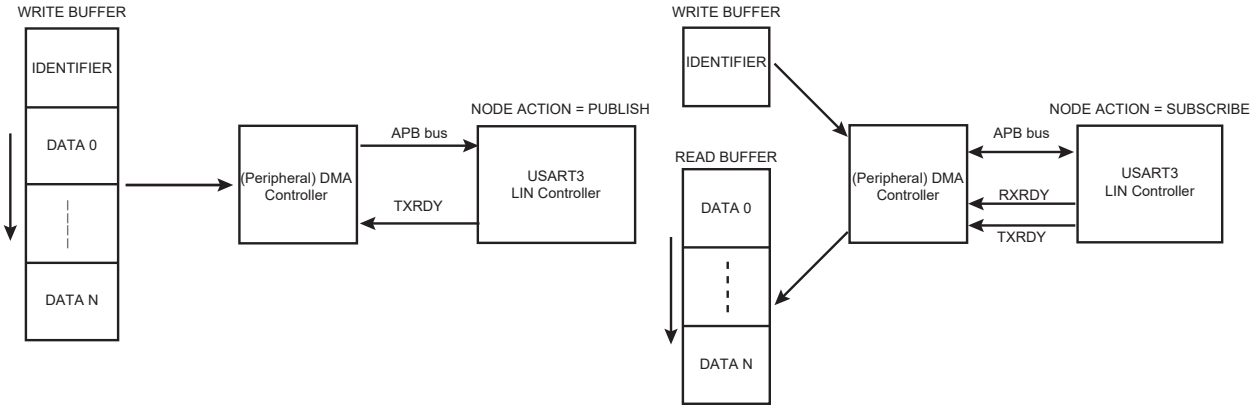


Figure 33-56. Host Node with PDC (PDCM = 0)



33.7.10.16.2. Client Node Configuration

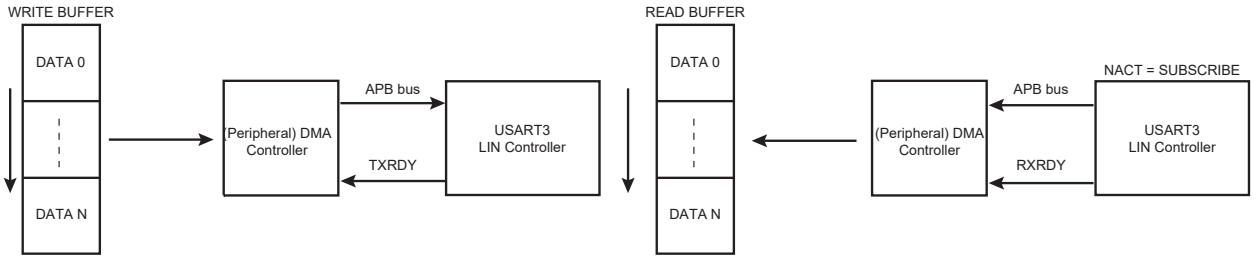
In this configuration, the PDC transfers only the data. The identifier must be read by the user in the LIN Identifier register (FLEX_US_LINIR). The LIN mode must be written by the user in FLEX_US_LINMR.

The WRITE buffer contains the data if the USART sends the response (NACT = PUBLISH).

The READ buffer contains the data if the USART receives the response (NACT = SUBSCRIBE).

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-57. Client Node with PDC



33.7.10.17. Wakeup Request

Any node in a sleeping LIN cluster may request a wakeup.

In the LIN 2.0 specification, the wakeup request is issued by forcing the bus to the dominant state from 250 µs to 5 ms. For this, it is necessary to send the character 0xF0 in order to impose five successive dominant bits. Whatever the baud rate is, this character respects the specified timings.

- Baud rate min = 1 kbit/s -> t_{bit} = 1 ms -> 5 t_{bit} = 5 ms
- Baud rate max = 20 kbit/s -> t_{bit} = 50 µs -> 5 t_{bit} = 250 µs

In the LIN 1.3 specification, the wakeup request should be generated with the character 0x80 in order to impose eight successive dominant bits.

Using the FLEX_US_LINMR.WKUPTYP bit, the user can choose to send either a LIN 2.0 wakeup request (WKUPTYP = 0) or a LIN 1.3 wakeup request (WKUPTYP = 1).

A wakeup request is transmitted by writing the FLEX_US_CR.LINWKUP bit to 1. Once the transfer is completed, the LINTC flag is asserted in the Status register (FLEX_US_CSR). It is cleared by writing a one to the FLEX_US_CR.RSTSTA bit.

33.7.10.18. Bus Idle Timeout

If the LIN bus is inactive for a certain duration, the client nodes shall automatically enter in Sleep mode. In the LIN 2.0 specification, this timeout is defined as 4 seconds. In the LIN 1.3 specification, it is defined as 25,000 t_{bit}.

In client node configuration, the receiver timeout detects an idle condition on the RXD line. When a timeout is detected, the FLEX_US_CSR.TIMEOUT bit rises and can generate an interrupt, thus indicating to the driver to go into Sleep mode.

The timeout delay period (during which the receiver waits for a new character) is programmed in the FLEX_US_RTOR.TO field. If a zero is written to the TO field, the Receiver Timeout is disabled and no timeout is detected. The FLEX_US_CSR.TIMEOUT bit remains at 0. Otherwise, the receiver loads a 17-bit counter with the value programmed in TO. This counter is decremented at each bit period and reloaded each time a new character is received. If the counter reaches 0, the FLEX_US_CSR.TIMEOUT bit rises.

If STTTO is performed, the counter clock is stopped until a first character is received.

If RETTO is performed, the counter starts counting down immediately from the value TO.

Table 33-13. Receiver Timeout Programming

LIN Specification	Baud Rate	Timeout period	TO
2.0	1,000 bit/s	4s	4,000
	2,400 bit/s		9,600
	9,600 bit/s		38,400
	19,200 bit/s		76,800
	20,000 bit/s		80,000

© 2026 Microchip Technology Inc. and its subsidiaries

Table 33-13. Receiver Timeout Programming (continued)

LIN Specification	Baud Rate	Timeout period	TO
1.3	–	25,000 t_{bit}	25,000

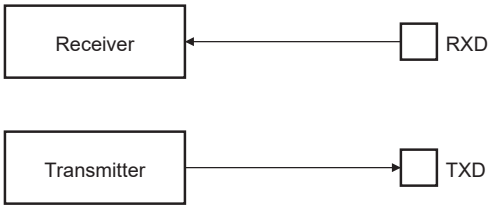
33.7.11. Test Modes

The USART can be programmed to operate in three different test modes. The internal loopback capability allows on-board diagnostics. In Loopback mode, the USART interface pins are disconnected or not and reconfigured for loopback internally or externally.

33.7.11.1. Normal Mode

Normal mode connects the RXD pin on the receiver input and the transmitter output on the TXD pin.

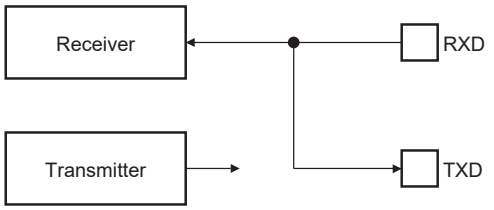
Figure 33-58. Normal Mode Configuration



33.7.11.2. Automatic Echo Mode

Automatic Echo mode allows bit-by-bit retransmission. When a bit is received on the RXD pin, it is sent to the TXD pin, as shown in the following figure. Programming the transmitter has no effect on the TXD pin. The RXD pin is still connected to the receiver input, thus the receiver remains active.

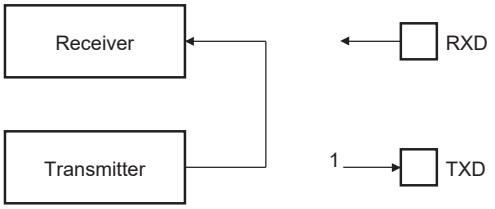
Figure 33-59. Automatic Echo Mode Configuration



33.7.11.3. Local Loopback Mode

Local Loopback mode connects the output of the transmitter directly to the input of the receiver, as shown in the following figure. The TXD and RXD pins are not used. The RXD pin has no effect on the receiver and the TXD pin is continuously driven high, as in idle state.

Figure 33-60. Local Loopback Mode Configuration

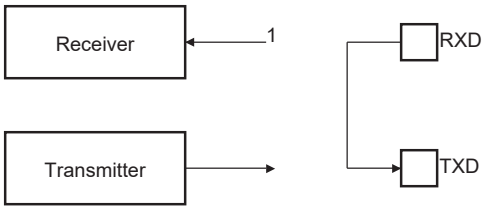


33.7.11.4. Remote Loopback Mode

Remote Loopback mode directly connects the RXD pin to the TXD pin, as shown in the following figure. The transmitter and the receiver are disabled and have no effect. This mode allows bit-by-bit retransmission.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-61. Remote Loopback Mode Configuration



33.7.12. USART FIFOs

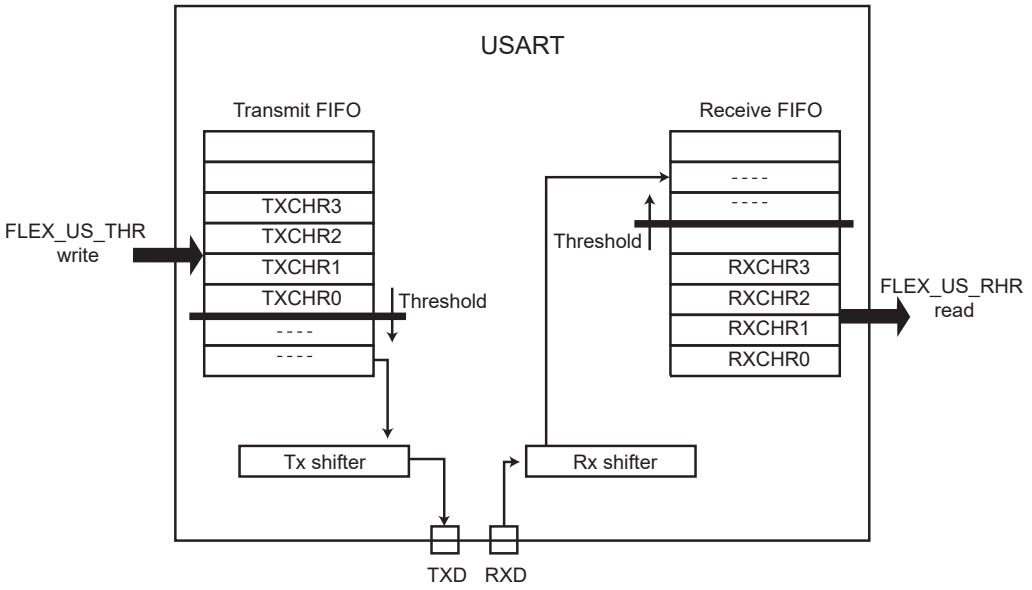
33.7.12.1. Overview

The USART includes two FIFOs which can be enabled/disabled using FLEX_US_CR.FIFOEN/FIFODIS. Both the transmitter and the receiver must be disabled before enabling or disabling the FIFOs, using the FLEX_US_CR.TXDIS/RXDIS bits.

Writing FLEX_US_CR.FIFOEN to '1' enables a 8-data Transmit FIFO and a 8-data Receive FIFO.

When the FIFO is enabled, it is possible to write or to read single data (5-bit to 9-bit data) or multiple data (5-bit to 8-bit data) in the same access to FLEX_US_THR/RHR. See FIFO Single Data Access and FIFO Multiple Data Access.

Figure 33-62. USART FIFOs Block Diagram



33.7.12.2. Sending Data with FIFO Enabled

When the Transmit FIFO is enabled, write access to FLEX_US_THR loads the Transmit FIFO.

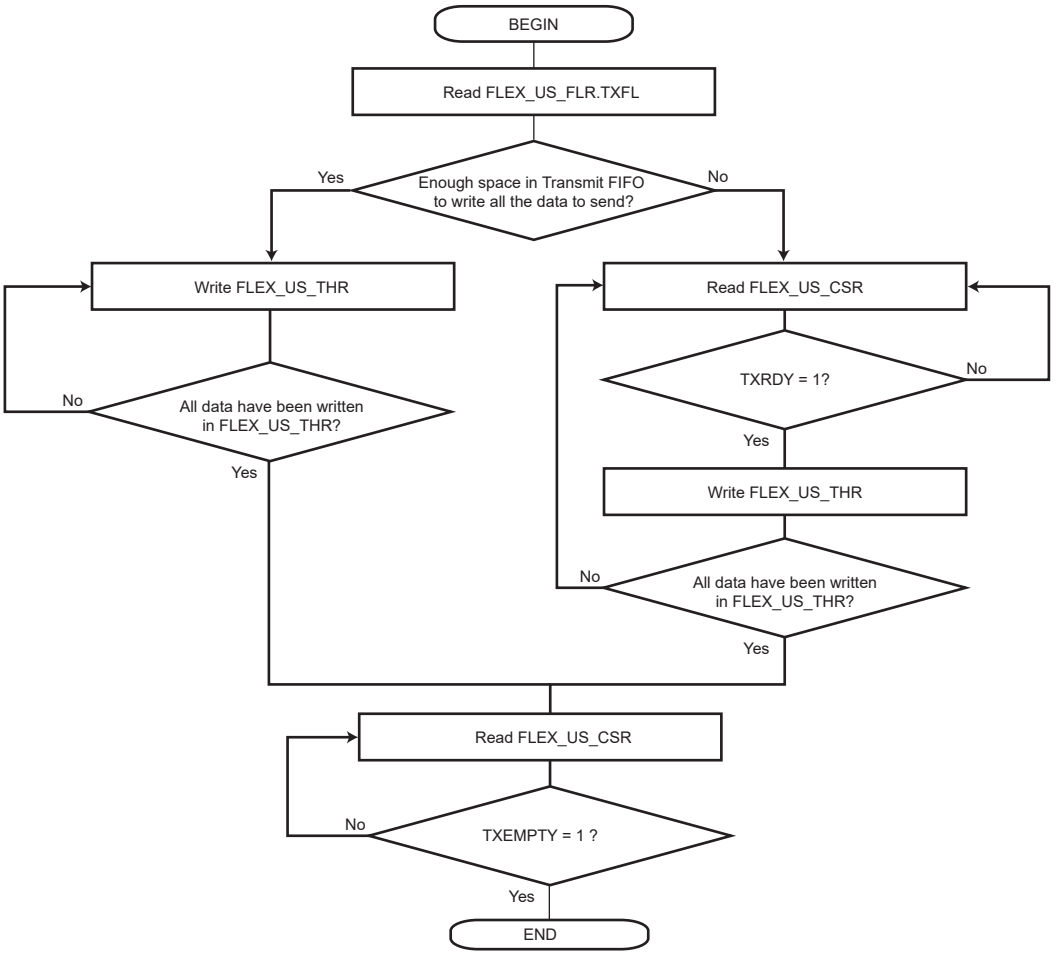
The FIFO level is provided in FLEX_US_FLR.TXFL. If the FIFO can accept the number of data to be transmitted, there is no need to monitor FLEX_US_CSR.TXRDY and the data can be successively written in FLEX_US_THR.

If the FIFO cannot accept the data due to insufficient space, wait for the TXRDY flag to be set before writing the data in FLEX_US_THR.

When the space in the FIFO allows only a portion of the data to be written, the TXRDY flag must be monitored before writing the remaining data.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-63. Sending Data with FIFO Enabled



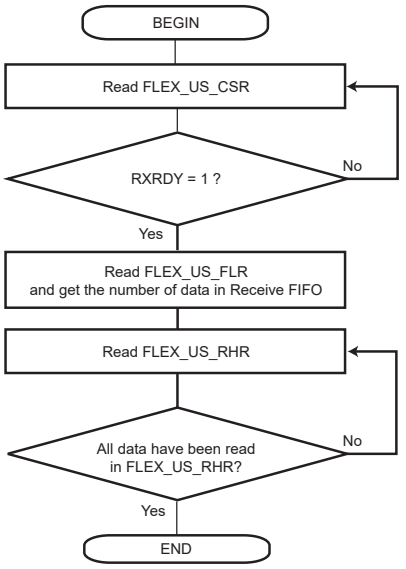
33.7.12.3. Receiving Data with FIFO Enabled

When the Receive FIFO is enabled, FLEX_US_RHR access reads the FIFO.

When data are present in the Receive FIFO (RXRDY flag set to ‘1’), the exact number of data can be checked with FLEX_US_FLR.RXFL. All the data can be read successively in FLEX_US_RHR without checking the RXRDY flag between each access.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-64. Receiving Data with FIFO Enabled



33.7.12.4. Clearing/Flushing FIFOs

Each FIFO can be cleared/flushed using FLEX_US_CR.TXFCLR/RXFCLR.

33.7.12.5. TXEMPTY, TXRDY and RXRDY Behavior

FLEX_US_CSR.TXEMPTY, FLEX_US_CSR.TXRDY and FLEX_US_CSR.RXRDY flags display a specific behavior when FIFOs are enabled.

The TXEMPTY flag is cleared as long as there are characters in the Transmit FIFO or in the internal shift register. TXEMPTY is set when there are no characters in the Transmit FIFO and in the internal shift register.

TXRDY indicates if a data can be written in the Transmit FIFO. Thus the TXRDY flag is set as long as the Transmit FIFO can accept new data. See figure TXRDY in Single Data Mode and TXRDYM = 0.

RXRDY indicates if an unread data is present in the Receive FIFO. Thus the RXRDY flag is set as soon as one unread data is in the Receive FIFO. See figure RXRDY in Single Data Mode and RXRDYM = 0 below.

TXRDY and RXRDY behavior can be modified using the TXRDYM and RXRDYM fields in the USART FIFO Mode register (FLEX_US_FMR) .

See FLEX_US_FMR for the FIFO configuration.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-65. TXRDY Behavior for Single Data Access and TXRDYM = 0

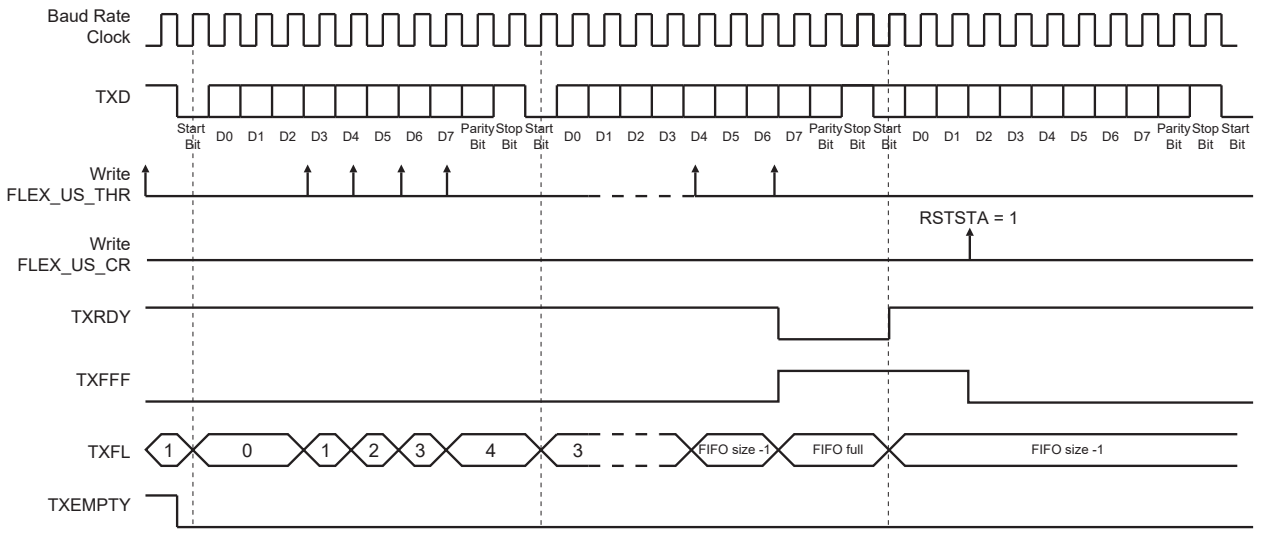
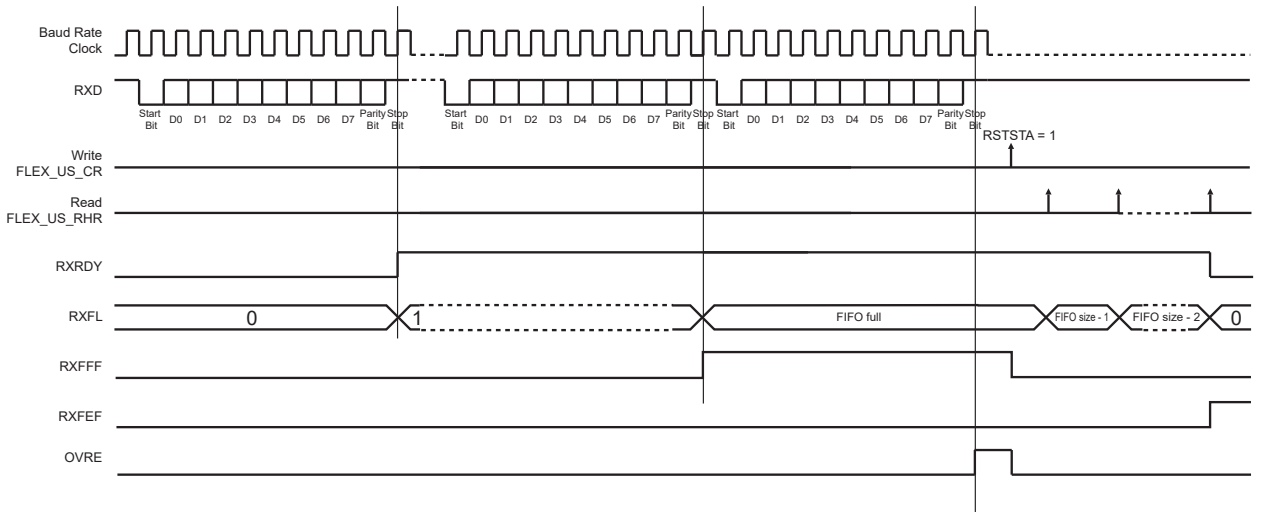


Figure 33-66. RXRDY Behavior for Single Data Access and RXRDYM = 0



33.7.12.6. FIFO Single Data Access

When FIFO is enabled and a byte access is performed in FLEX_US_THR (5-bit to 8-bit data size), a single data is written in FIFO. The similar behavior applies for FLEX_US_RHR.

If FLEX_US_MR.MODE9 is set (9-bit data), or if FLEX_US_MR.USART_MODE is configured to operate in LIN Host mode or LIN Client mode, or if FLEX_US_MR.MAN is set, any type of access to FLEX_US_THR/RHR writes/reads a single data.

See USART Receive Holding Register (FLEX_US_RHR) and USART Transmit Holding Register (FLEX_US_THR).

However, for some configurations it is possible to write/read multiple data each time FLEX_US_THR/FLEX_US_RHR is accessed. See FIFO Multiple Data Access.

33.7.12.6.1. PDC

The PDC transfer type must be configured in bytes or halfwords when FIFOs operate in Single Data mode (the same applies when FIFOs are disabled).

© 2026 Microchip Technology Inc. and its subsidiaries

33.7.12.7. FIFO Multiple Data Access

For some operating modes, it is possible to reduce the number of accesses to/from FLEX_US_THR/FLEX_US_RHR required to transfer an amount of data, by concatenating multiple data (5-bit to 8-bit) when FIFO is enabled (FLEX_US_CR.FIFOEN=1) and 5- to 8-bit data characters are transferred (FLEX_US_MR.MODE9=0).

Up to four data (5-bit to 8-bit) can be written/read in one FLEX_US_THR/FLEX_US_RHR access.

When the FIFO is enabled, the number of data to write/read is defined by the type of access in the holding register. If the access is a byte, only one data is written/read (single data access), if the access is a halfword or a word a multiple data access is performed. If the access is a halfword, then two data are written/read and if the access is a word, four data are written/read.

Written/read data are always right-aligned, as described in USART Receive Holding Register (FIFO Multi Data) and USART Transmit Holding Register (FIFO Multi Data).

Multiple data access cannot be used for the following configurations:

- If FLEX_US_MR.MODE9 is set
- If FLEX_US_MR.USART_MODE is configured to operated in LIN Host mode or LIN Client mode
- FLEX_US_MR.MAN is set

As an example of multiple data access, if the Transmit FIFO is empty and there are six data to send, any of the following write accesses may be performed:

- six FLEX_US_THR-byte write accesses
- three FLEX_US_THR-halfword write accesses
- one FLEX_US_THR word write access and one FLEX_US_THR halfword write access

With a Receive FIFO containing six data, any of the following read accesses may be performed:

- six FLEX_US_RHR-byte read accesses
- three FLEX_US_RHR-halfword read accesses
- one FLEX_US_RHR-word read access and one FLEX_US_RHR-halfword read access

33.7.12.7.1. TXRDY and RXRDY Configuration

The TXRDY flag indicates if one or more data can be written in the FIFO depending on the configuration of FLEX_US_FMR.TXRDYM/RXRDYM.

As an example, if a word (32-bit) is written in FLEX_US_THR, the TXRDYM field must be configured so that the TXRDY flag is at '1' only when at least four data can be written in the Transmit FIFO.

In the same way, if a word (32-bit) is read in FLEX_US_RHR, the RXRDYM field must be configured so that the RXRDY flag is at '1' only when at least four unread data are in the Receive FIFO.

33.7.12.7.2. PDC

The PDC transfer type must be configured according to the FLEX_US_FMR.TXRDYM/RXRDYM settings.

As an example, FLEX_US_FMR.TXRDYM/RXRDYM=0 is not compatible with DMAC_PDC transfers in word (32-bit).

33.7.12.8. Transmit FIFO Lock

- LIN Mode:

If a frame is aborted using the Abort LIN Transmission bit (FLEX_US_CR.LINABT), a lock is set on the Transmit FIFO, preventing any new frame from being sent until it is cleared. This allows clearing the FIFO if needed, resetting PDC channels, etc., without any risk.

The TXFLOCK bit in the USART FIFO Event Status register (FLEX_US_FESR) is used to check the state of the Transmit FIFO lock.

© 2026 Microchip Technology Inc. and its subsidiaries

The Transmit FIFO lock can be cleared by setting FLEX_US_CR.TXFLCLR to '1'.

33.7.12.9. FIFO Pointer Error

A FIFO overflow is reported in FLEX_US_FESR.

If the Transmit FIFO is full and a write access is performed on FLEX_US_THR, it generates a Transmit FIFO pointer error and sets FLEX_US_FESR.TXFPTEF.

If the number of data written in FLEX_US_THR (according to the register access size) is greater than the free space in the Transmit FIFO, a Transmit FIFO pointer error is generated and FLEX_US_FESR.TXFPTEF is set.

A FIFO underflow is reported in FLEX_US_FESR.

If the number of data read in FLEX_US_RHR (according to the register access size) is greater than the number of unread data in the Receive FIFO, a Receive FIFO pointer error is generated and FLEX_US_FESR.RXFPTEF is set.

No pointer error occurs if the FIFO state/level is checked before writing/reading in FLEX_US_THR/FLEX_US_RHR. The FIFO state/level can be checked either with TXRDY, RXRDY, TXFL or RXFL. When a pointer error occurs, other FIFO flags may not behave as expected; their states should be ignored.

If a Transmit pointer error occurs, a transmitter reset must be performed using FLEX_US_CR.RSTTX. If a Receive pointer error occurs, a receiver reset must be performed using FLEX_US_CR.RSTRX.

33.7.12.10. FIFO Thresholds

Each Transmit and Receive FIFO includes a threshold feature used to set a flag and an interrupt when a FIFO threshold is crossed. Thresholds are defined as a number of data in the FIFO, and the FIFO state (TXFL or RXFL) represents the number of data currently in the FIFO.

The Transmit FIFO threshold can be set using the field FLEX_US_FMR.TXFTHRES. Each time the Transmit FIFO level goes from 'above threshold' to 'equal to or below threshold', the flag FLEX_US_FESR.TXFTHF is set. The application is warned that the Transmit FIFO has reached the defined threshold and that it can be reloaded.

The Receive FIFO threshold can be set using the field FLEX_US_FMR.RXFTHRES. Each time the Receive FIFO level goes from 'below threshold' to 'equal to or above threshold', the flag FLEX_US_FESR.RXFTHF is set. The application is warned that the Receive FIFO has reached the defined threshold and that it can be read to prevent an underflow.

The Receive FIFO threshold 2 can be set using the field FLEX_US_FMR.RXFTHRES2. Each time the Receive FIFO level goes from 'above threshold 2' to 'equal to or below threshold 2', the flag FLEX_US_FESR.RXFTHF2 is set. The application is warned that the Receive FIFO has reached the defined threshold and that it can be read to prevent an underflow.

The TXFTHF, RXFTHF and RXTHF2 flags can be configured to generate an interrupt using FLEX_US_FIER and FLEX_US_FIDR.

33.7.12.11. FIFO Flags

FIFOs come with a set of flags which can be configured to generate interrupts through FLEX_US_FIER and FLEX_US_FIDR.

FIFO flags state can be read in FLEX_US_FESR. They are cleared by writing FLEX_US_CR.RSTSTA to '1'.

33.7.13. 16-bit Data Protocol Support

When configuring 0xC in FLEX_US_MR.USART_MODE, the transmitter sends a 16-bit data frame and the receiver expects an 8-bit data frame. The number of stop bits is defined in the field NBSTOP. The transmitter and/or receiver must operate in asynchronous mode (FLEX_US_MR.SYNC must be cleared).

When configuring 0xD in FLEX_US_MR.USART_MODE, the transmitter sends an 8-bit frame whereas the receiver expects a 16-bit frame.

© 2026 Microchip Technology Inc. and its subsidiaries

The FIFO mode must be enabled by setting FLEX_US_CR.FIFOEN to '1'.

A 16-bit frame starts as soon as two 8-bit characters are written in FLEX_US_THR (assuming 0xC is written in the field USART_MODE).

Note: When FLEX_US_MR.USART_MODE =0xC or 0xD, there must be a parity bit in the frame (FLEX_US_MR.USART_MODE must not be equal to 4).

33.7.14. USART Register Write Protection

The FLEXCOM operating mode (FLEX_MR.OPMODE) must be set to FLEX_MR_OPMODE_USART to enable access to the write protection registers.

To prevent any single software error from corrupting USART behavior, certain registers in the address space can be write-protected by setting the WPEN (Write Protection Enable), WPITEN (Write Protection Interrupt Enable), and/or WPCREN (Write Protection Control Enable) bits in the USART Write Protection Mode Register (FLEX_US_WPMR).

If a write access to a write-protected register is detected, the Write Protection Violation Status (WPVS) flag in the USART Write Protection Status Register (FLEX_US_WPSR) is set and the Write Protection Violation Source (WPVSRC) field indicates the register in which the write access has been attempted.

The WPVS bit is automatically cleared after reading FLEX_US_WPSR.

The following registers can be write-protected when WPEN is set:

- USART Mode Register
- USART Baud Rate Generator Register
- USART Receiver Timeout Register
- USART Transmitter Timeguard Register
- USART FI DI RATIO Register
- USART IrDA FILTER Register
- USART Manchester Configuration Register
- USART Comparison Register

The following register(s) can be write-protected when WPITEN is set:

- USART Interrupt Enable Register
- USART Interrupt Disable Register

The following register(s) can be write-protected when WPCREN is set:

- USART Control Register

33.8. SPI Functional Description

33.8.1. Modes of Operation

The SPI operates in Host mode or in Client mode.

- The SPI operates in Host mode by writing a 1 to the MSTR bit in the SPI Mode register (FLEX_SPI_MR):
  - The pins NPCS0 to NPCS1 are all configured as outputs.
  - The SPCK pin is driven.
  - The MISO line is wired on the receiver input.
  - The MOSI line is driven as an output by the transmitter.
- The SPI operates in Client mode if the MSTR bit in FLEX_SPI_MR is written to 0:

© 2026 Microchip Technology Inc. and its subsidiaries

- The MISO line is driven by the transmitter output.
- The MOSI line is wired on the receiver input.
- The SPCK pin is driven by the transmitter to synchronize the receiver.
- The NPCS0 pin becomes an input, and is used as a client select signal (NSS).
- Pin NPCS1 is not driven and can be used for other purposes.

The data transfers are identically programmable for both modes of operation. The bit rate generator is activated only in Host mode.

33.8.2. Data Transfer

Four combinations of polarity and phase are available for data transfers. The clock polarity is programmed with the CPOL bit in the SPI Chip Select register (FLEX_SPI_CSR). The clock phase is programmed with the NCPHA bit. These two parameters determine the edges of the clock signal on which data are driven and sampled. Each of the two parameters has two possible states, resulting in four possible combinations that are incompatible with one another. Consequently, a host/client pair must use the same parameter pair values to communicate. If multiple clients are connected and require different configurations, the host must reconfigure itself each time it needs to communicate with a different client.

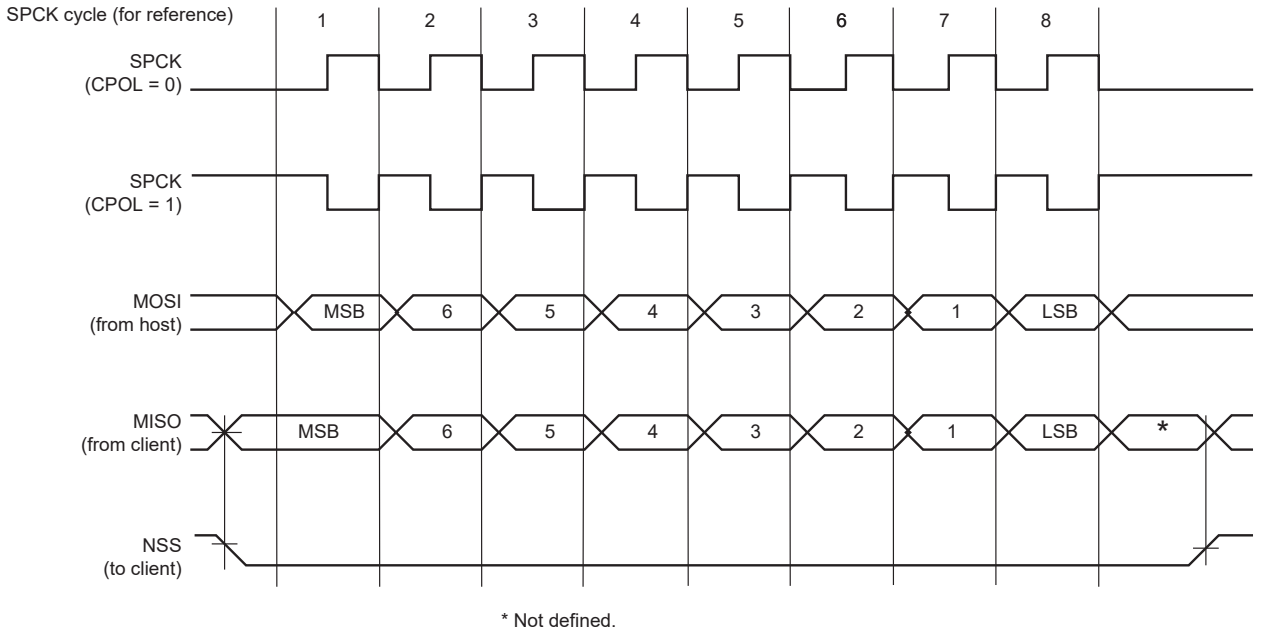
The following table shows the four modes and corresponding parameter settings.

Table 33-14. SPI Bus Protocol Mode

SPI Mode	CPOL	NCPHA	Host Shift SPCK Edge	Client Capture SPCK Edge	SPCK Inactive Level
0	0	1	Falling	Rising	Low
1	0	0	Rising	Falling	Low
2	1	1	Rising	Falling	High
3	1	0	Falling	Rising	High

The following figures show examples of data transfers.

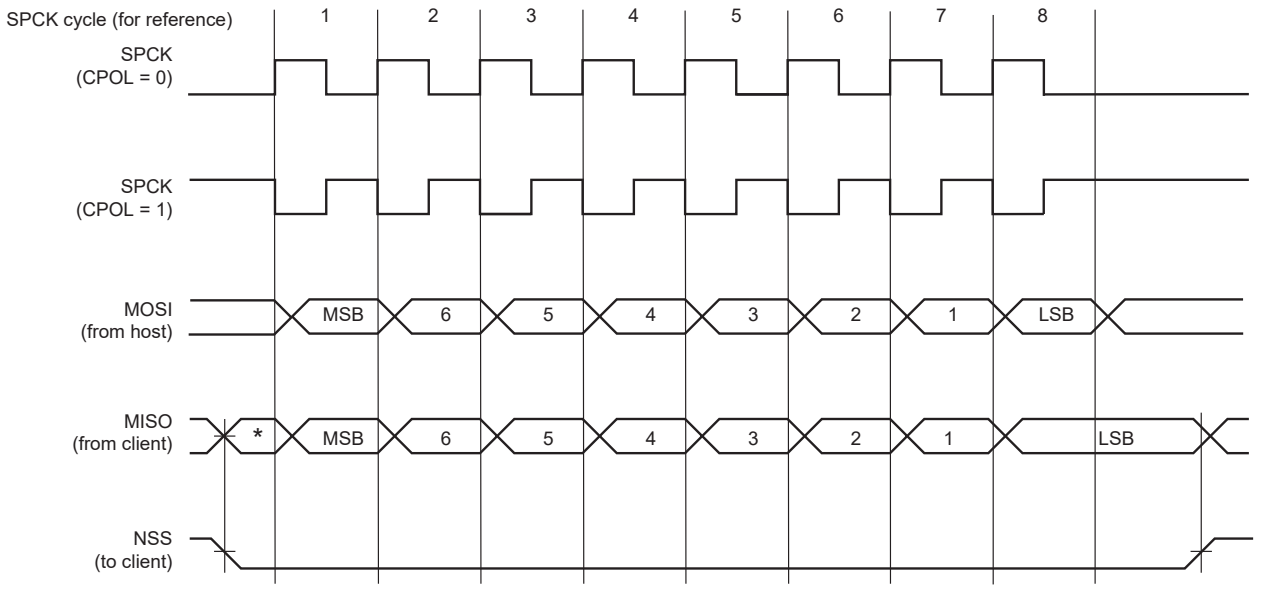
Figure 33-67. SPI Transfer Format (NCPHA = 1, 8 bits per transfer) Mode 0 and 2



* Not defined.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-68. SPI Transfer Format (NCPHA = 0, 8 bits per transfer) Mode 1 and 3



* Not defined.

33.8.3. Host Mode Operations

When configured in Host mode, the SPI operates on the clock generated by the internal programmable bit rate generator. It fully controls the data transfers to and from the client(s) connected to the SPI bus. The SPI drives the chip select line to the client and the serial clock signal (SPCK).

The SPI features two holding registers, the Transmit Data register (FLEX_SPI_TDR) and the Receive Data register (FLEX_SPI_RDR), and a single shift register. The holding registers maintain the data flow at a constant rate.

After enabling the SPI, a data transfer starts when the processor writes to FLEX_SPI_TDR. The written data are immediately transferred in the shift register and the transfer on the SPI bus starts. While the data in the shift register is shifted on the MOSI line, the MISO line is sampled and shifted in the shift register. Data cannot be loaded in FLEX_SPI_RDR without transmitting data. If there is no data to transmit, a dummy data can be used (FLEX_SPI_TDR filled with ones). When the WDRBT bit is set, a new data cannot be transmitted if FLEX_SPI_RDR has not been read. If Receiving mode is not required, for example when communicating with a client receiver only (such as an LCD), the receive status flags in the SPI Status register (FLEX_SPI_SR) can be discarded.

Before writing the TDR, the FLEX_SPI_MR.PCS field must be set in order to select a client.

If new data are written in FLEX_SPI_TDR during the transfer, it is kept in FLEX_SPI_TDR until the current transfer is completed. Then, the received data are transferred from the shift register to FLEX_SPI_RDR, the data in FLEX_SPI_TDR is loaded in the shift register and a new transfer starts.

As soon as the FLEX_SPI_TDR is written, the Transmit Data Register Empty (TDRE) flag in FLEX_SPI_SR is cleared. When the data written in FLEX_SPI_TDR is loaded into the shift register, the FLEX_SPI_SR.TDRE flag is set. The TDRE bit is used to trigger the Transmit PDC channel (see figure below).

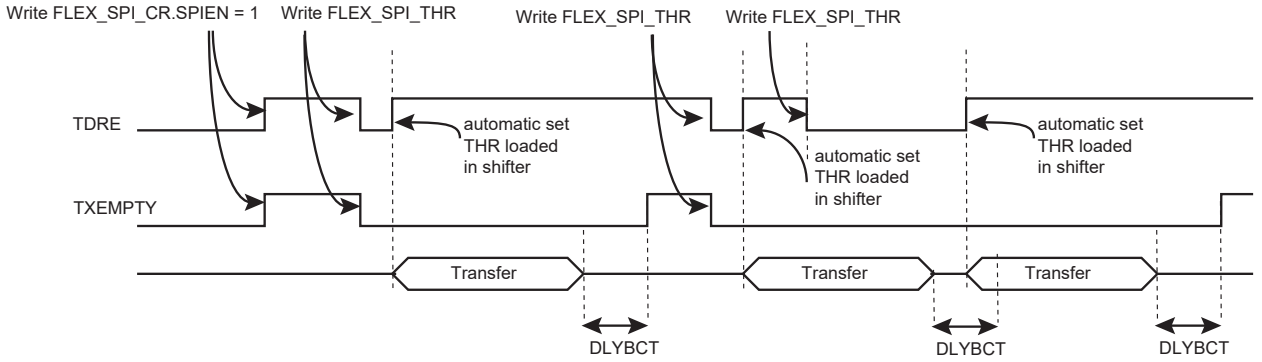
© 2026 Microchip Technology Inc. and its subsidiaries

The end of transfer is indicated by FLEX_SPI_SR.TXEMPTY. If a transfer delay (DLYBCT) is greater than 0 for the last transfer, TXEMPTY is set after the completion of this delay. The peripheral clock can be switched off at this time.

Notes:

1. When the SPI is enabled, the TDRE and TXEMPTY flags are set.
2. The TXEMPTY flag alone cannot be used to detect the end of the buffer PDC transfer.

Figure 33-69. TDRE and TXEMPTY Flag Behavior



The transfer of received data from the shift register to FLEX_SPI_RDR is indicated by the Receive Data Register Full (RDRF) bit in FLEX_SPI_SR. When the received data are read, the RDRF bit is cleared.

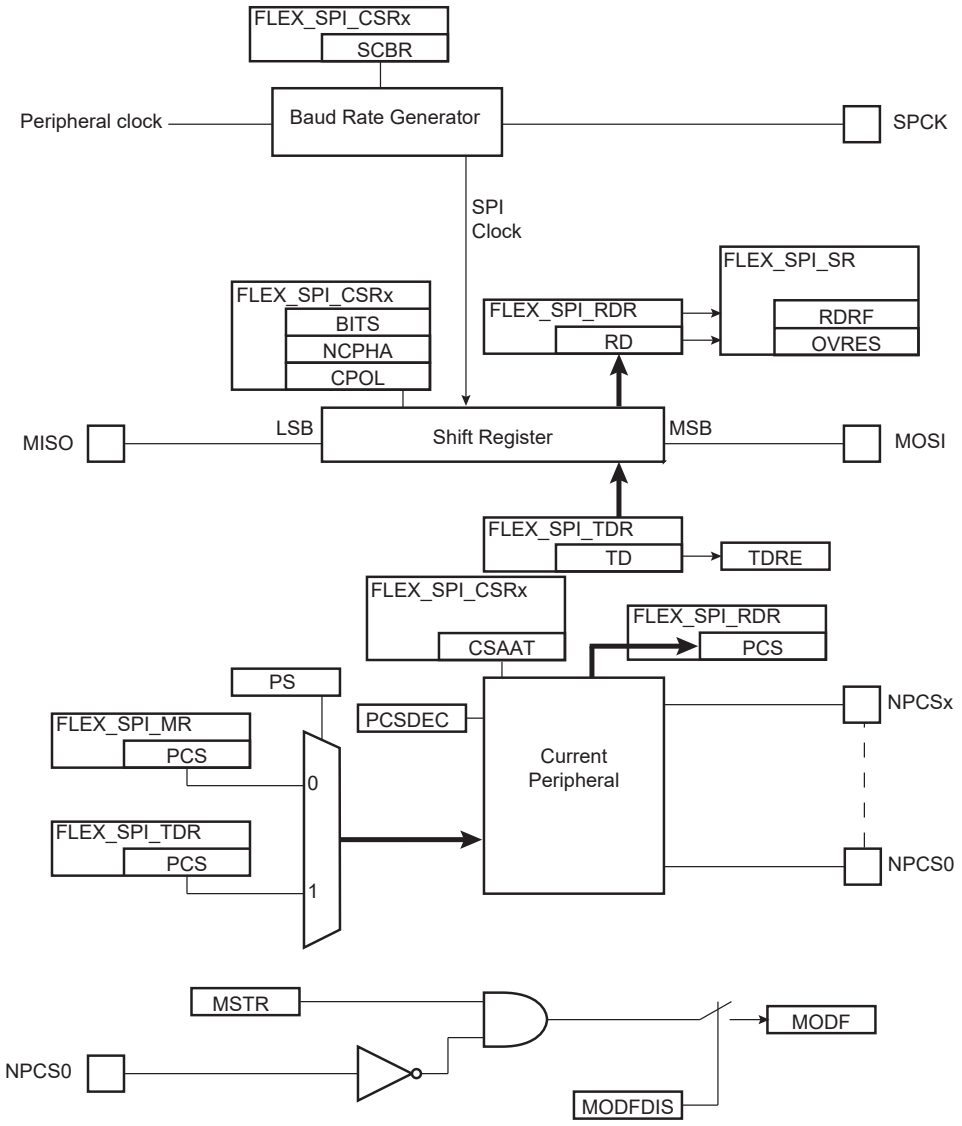
If FLEX_SPI_RDR has not been read before new data are received, the Overrun Error bit (OVRES) in FLEX_SPI_SR is set. As long as this flag is set, data are loaded in FLEX_SPI_RDR. The user has to read the status register to clear the OVRES bit.

The following figures show, respectively, a block diagram of the SPI when operating in Host mode and a flow chart describing how transfers are handled.

© 2026 Microchip Technology Inc. and its subsidiaries

33.8.3.1. Host Mode Block Diagram

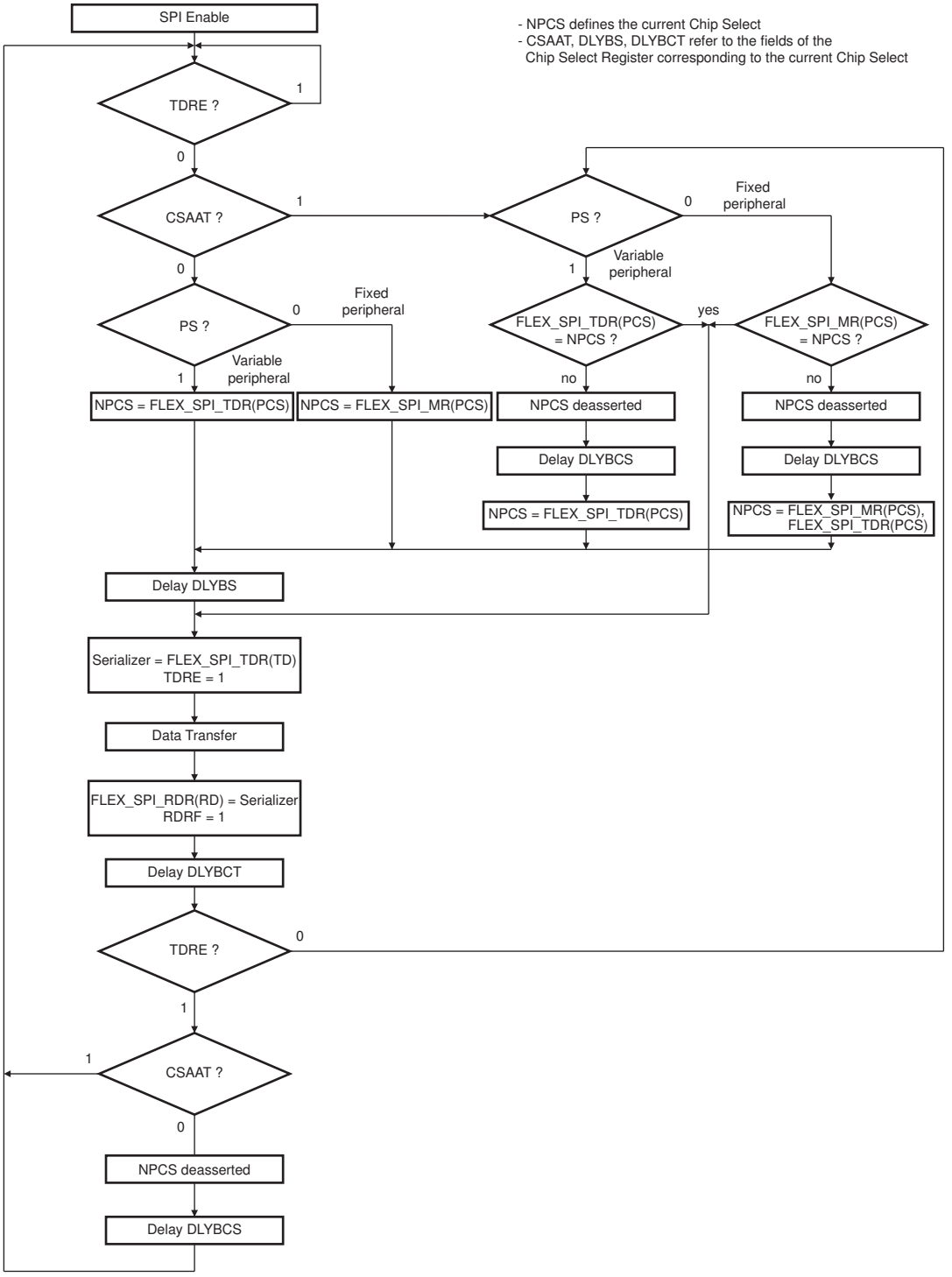
Figure 33-70. Host Mode Block Diagram



© 2026 Microchip Technology Inc. and its subsidiaries

33.8.3.2. Host Mode Flowchart

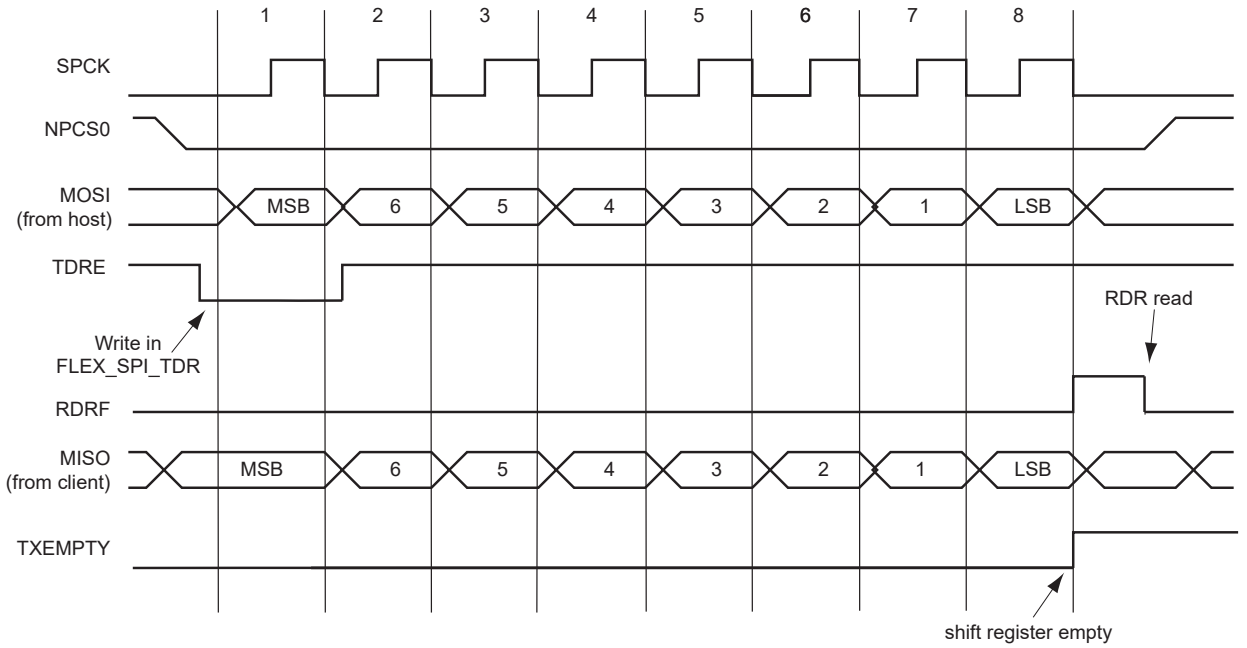
Figure 33-71. Host Mode



The following figure shows the behavior of Transmit Data Register Empty (TDRE), Receive Data Register (RDRF) and Transmission Register Empty (TXEMPTY) status flags within FLEX_SPI_SR during an 8-bit data transfer in Fixed mode without the PDC involved.

© 2026 Microchip Technology Inc. and its subsidiaries

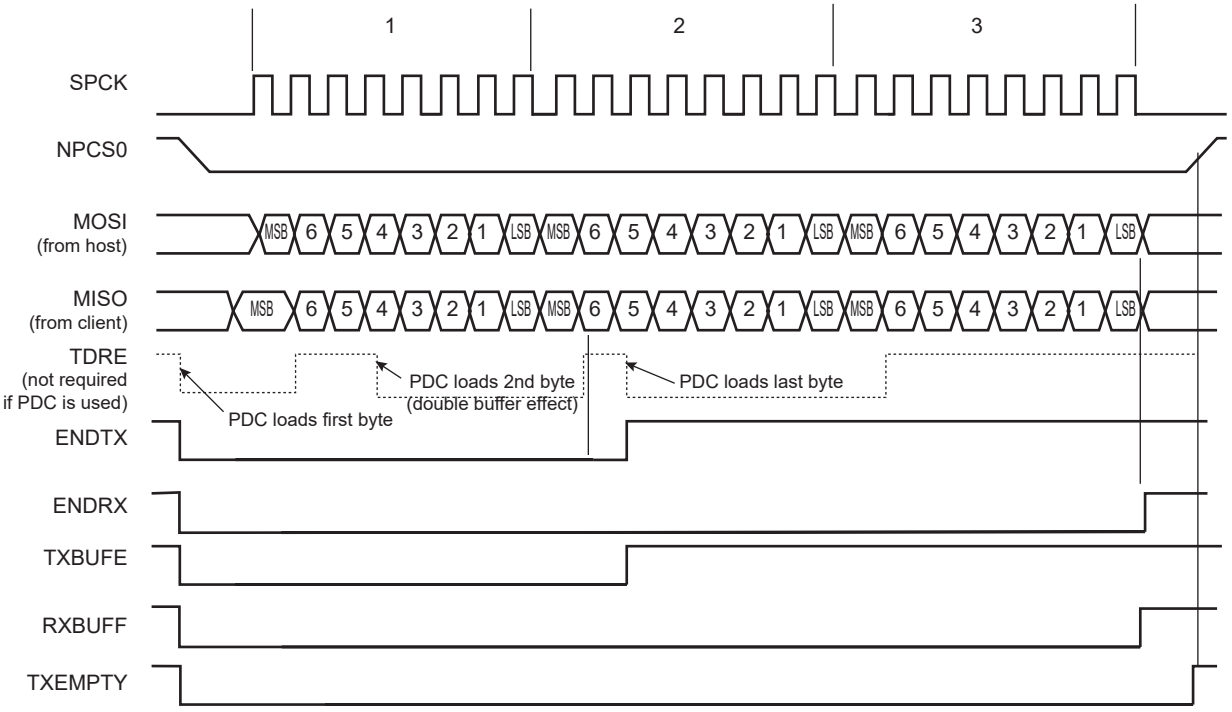
Figure 33-72. Status Register Flags Behavior



The following figure shows the behavior of Transmission Register Empty (TXEMPTY), End of RX buffer (ENDRX), End of TX buffer (ENDTX), RX Buffer Full (RXBUFF) and TX Buffer Empty (TXBUFE) status flags within FLEX_SPI_SR during an 8-bit data transfer in Fixed mode with the Peripheral Data Controller involved. The PDC is programmed to transfer and receive three data. The next pointer and counter are not used. The RDRF and TDRE are not shown because these flags are managed by the PDC when using the PDC.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-73. PDC Status Register Flags Behavior



33.8.3.3. Clock Generation

The SPI bit rate clock is generated by dividing a source clock which can be the peripheral clock or a programmable clock from the GCLK. The divider can be a value between 1 and 255.

If the SCBR field is programmed to 1 and the clock source is GCLK, the operating bit rate is peripheral clock (refer to the section “Electrical Characteristics” for the SPCK maximum frequency). Triggering a transfer while SCBR is at 0 can lead to unpredictable results.

At reset, SCBR is 0 and the user has to program it to a valid value before performing the first transfer.

The divisor can be defined independently for each chip select, as it has to be programmed in the FLEX_SPI_CSR.SCBR field. This allows the SPI to automatically adapt the bit rate for each interfaced peripheral without reprogramming.

If GCLK is selected as source clock (FLEX_SPI_MR.BRSRCCLK = 1), the bit rate is independent of the processor/bus clock. Thus, the processor clock can be changed while SPI is enabled. The processor clock frequency changes must be performed only by programming the PMC_MCKR.PRES field (refer to the section “Power Management Controller (PMC)"). Any other method to modify the processor/bus clock frequency (PLL multiplier, etc.) is forbidden when SPI is enabled.

The peripheral clock frequency must be at least three times higher than GCLK.

33.8.3.4. Transfer Delays

The figure below shows a chip select transfer change and consecutive transfers on the same chip select. Three delays can be programmed to modify the transfer waveforms:

- The delay between the chip selects. It is programmable only once for all chip selects by writing the FLEX_SPI_MR.DLYBCS field. The SPI client device deactivation delay is managed through DLYBCS. If there is only one SPI client device connected to the host, the DLYBCS field does

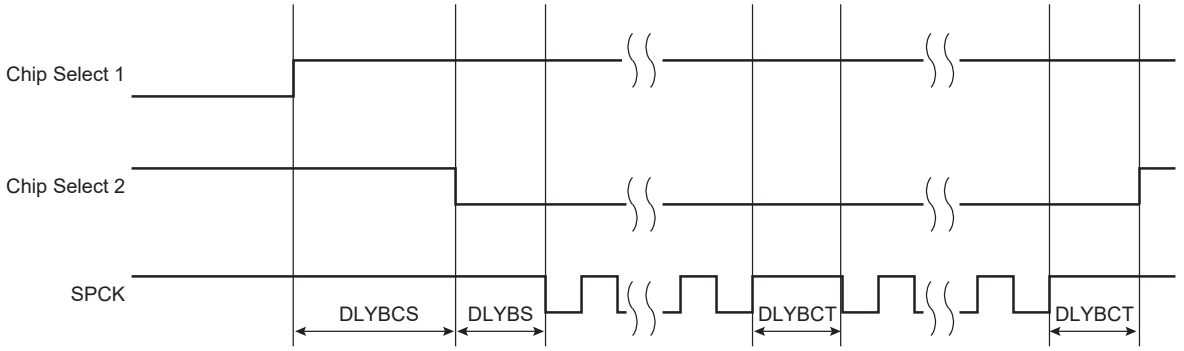
© 2026 Microchip Technology Inc. and its subsidiaries

not need to be configured. If several client devices are connected to a host, DLYBCS must be configured depending on the highest deactivation delay. Refer to “SPI Timings” in the section “Electrical Characteristics”.

- The delay before SPCK, independently programmable for each chip select by writing the DLYBS field. The SPI client device activation delay is managed through DLYBS. Refer to “SPI Timings” in the section “Electrical Characteristics” to define DLYBS.
- The delay between consecutive transfers, independently programmable for each chip select by writing the DLYBCT field. The time required by the SPI client device to process received data is managed through DLYBCT. This time depends on the SPI client system activity.

These delays allow the SPI to be adapted to the interfaced peripherals and their speed and bus release time.

Figure 33-74. Programmable Delays



33.8.3.5. Peripheral Selection

The serial peripherals are selected through the assertion of the NPCS0 to NPCS1 signals. By default, all NPCS signals are high before and after each transfer.

- Fixed Peripheral Select Mode: SPI exchanges data with only one peripheral.
  Fixed Peripheral Select mode is enabled by writing the FLEX_SPI_MR.PS bit to zero. In this case, the current peripheral is defined by the FLEX_SPI_MR.PCS field, and the FLEX_SPI_TDR. PCS field has no effect.
- Variable Peripheral Select Mode: Data can be exchanged with more than one peripheral without having to reprogram FLEX_SPI_MR.PCS.
  Variable Peripheral Select Mode is enabled by setting the FLEX_SPI_MR.PS bit to one. The FLEX_SPI_TDR.PCS field is used to select the current peripheral. This means that the peripheral selection can be defined for each new data. The value must be written in a single access to FLEX_SPI_TDR in the following format:

  [xxxxxxx(7-bit) + LASTXFER(1-bit)(1)+ xxxx(4-bit) + PCS (4-bit) + TD (8 to 16-bit data)]
  with LASTXFER at 0 or 1 depending on the CSAAT bit, and PCS equal to the chip select to assert, as defined in SPI Transmit Data Register (FLEX_SPI_TDR).
  Note: 1. Optional

  The CSAAT, LASTXFER and CSNAAT bits are discussed in Peripheral Deselection with PDC.

  If LASTXFER is used, the command must be issued after writing the last character. Instead of LASTXFER, the user can use the SPIDIS command. After the end of the PDC transfer, it is necessary to wait for the TXEMPTY flag and then write SPIDIS into the SPI Control register (FLEX_SPI_CR). This does not change the configuration register values). The NPCS is disabled after the last character transfer. Then, another PDC transfer can be started if the FLEX_SPI_CR.SPIEN bit has previously been written.

© 2026 Microchip Technology Inc. and its subsidiaries

33.8.3.6. SPI Peripheral DMA Controller (PDC)

In both Fixed and Variable Peripheral Select modes, the Peripheral DMA Controller (PDC) can be used to reduce processor overhead.

The fixed peripheral selection allows buffer transfers with a single peripheral. Using the PDC is an optimal means, as the size of the data transfer between the memory and the SPI is either 8 bits or 16 bits. However, if the peripheral selection is modified, FLEX_SPI_MR must be reprogrammed.

The variable peripheral selection allows buffer transfers with multiple peripherals without reprogramming FLEX_SPI_MR. Data written in FLEX_SPI_TDR is 32 bits wide and defines the real data to be transmitted and the destination peripheral. Using the PDC in this mode requires 32-bit wide buffers, with the data in the LSBs and the PCS and LASTXFER fields in the MSBs. However, the SPI still controls the number of bits (8 to16) to be transferred through MISO and MOSI lines with the chip select registers (FLEX_SPI_CSRx). This is not the optimal means in terms of memory size for the buffers, but it provides a very effective means to exchange data with several peripherals without any intervention of the processor.

33.8.3.6.1. Transfer Size

Depending on the data size to transmit, from 8 to 16 bits, the PDC manages automatically the type of pointer size it has to point to. The PDC performs the following transfer, depending on the mode and number of bits per data.

- Fixed mode:
  - 8-bit data:
    1-byte transfer,
    PDC pointer address = address + 1 byte,
    PDC counter = counter - 1
  - 9-bit to 16-bit data:
    2-byte transfer. n-bit data transfer with don't care data (MSB) filled with zeros,
    PDC pointer address = address + 2 bytes,
    PDC counter = counter - 1
- Variable mode:
  - In Variable mode, PDC pointer address = address + 4 bytes and PDC counter = counter - 1 for 8 to 16-bit transfer size.
  - When using the PDC, the TDRE and RDRF flags are handled by the PDC. The user's application does not have to check these bits. Only End of RX Buffer (ENDRX), End of TX Buffer (ENDTX), Buffer Full (RXBUFF), TX Buffer Empty (TXBUFE) are significant. For further details about the Peripheral DMA Controller and user interface, refer to the section "Peripheral DMA Controller (PDC)".

33.8.3.7. Peripheral Chip Select Decoding

The user can program the SPI to operate with up to 3 client peripherals by decoding the two chip select lines, NPCS0 to NPCS1 with an external decoder/demultiplexer (see the following figure). This can be enabled by setting the FLEX_SPI_MR.PCSDEC bit.

When operating without decoding, the SPI makes sure that in any case only one chip select line is activated, i.e., one NPCS line driven low at a time. If two bits are defined low in a PCS field, only the lowest numbered chip select is driven low.

When operating with decoding, the SPI directly outputs the value defined by the PCS field on the NPCS lines of either FLEX_SPI_MR or FLEX_SPI_TDR (depending on PS).

As the SPI sets a default value of 0x3 on the chip select lines (i.e., all chip select lines at 1) when not processing any transfer, only 3 peripherals can be decoded.

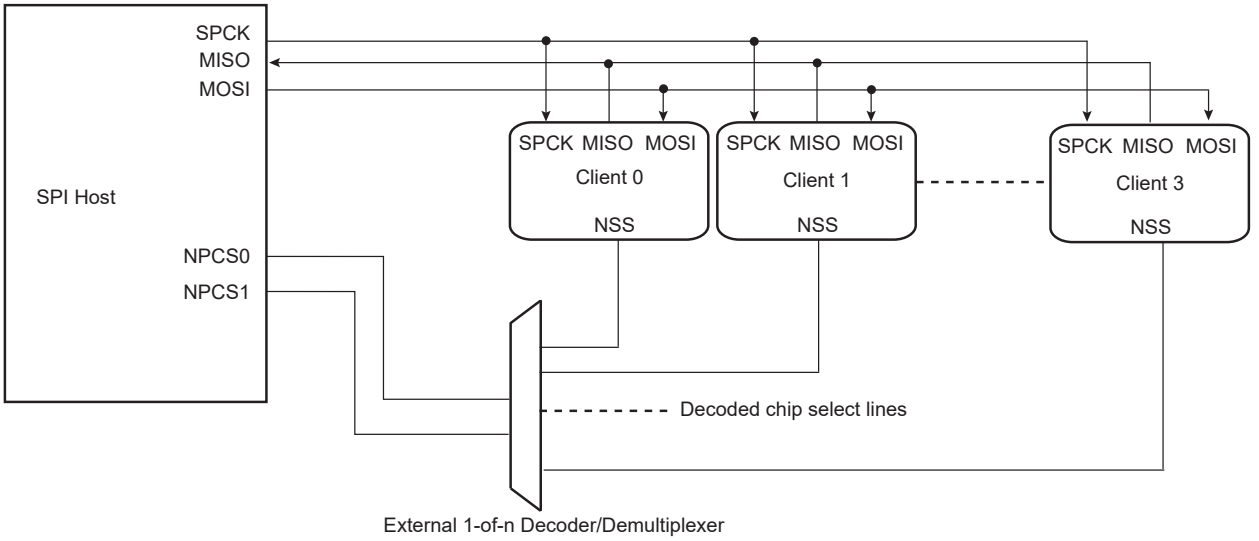
The SPI has only two Chip Select registers. As a result, when external decoding is activated, each NPCS chip select defines the characteristics of up to two peripherals. As an example, FLEX_SPI_CRS0 defines the characteristics of the externally decoded peripherals 0 to 1, corresponding to the PCS

© 2026 Microchip Technology Inc. and its subsidiaries

values 0x0 to 0x1. Consequently, the user has to make sure to connect compatible peripherals on the decoded chip select lines 0 to 1 and 2. The following figure shows this type of implementation.

If the CSAAT bit is used, with or without the PDC, the mode fault detection for NPCS0 line must be disabled. This is not required for all other chip select lines since mode fault detection is only on NPCS0.

Figure 33-75. Chip Select Decoding Application Block Diagram: Single Host/Multiple Client Implementation



33.8.3.8. Peripheral Deselection without PDC

During a transfer of more than one data on a Chip Select without the PDC, FLEX_SPI_TDR is loaded by the processor, the TDRE flag rises as soon as the content of FLEX_SPI_TDR is transferred into the internal shift register. When this flag is detected high, FLEX_SPI_TDR can be reloaded. If this reload by the processor occurs before the end of the current transfer, and if the next transfer is performed on the same chip select as the current transfer, the Chip Select is not deasserted between the two transfers. But depending on the application software handling the SPI status register flags (by interrupt or polling method) or servicing other interrupts or other tasks, the processor may not reload FLEX_SPI_TDR in time to keep the chip select active (low). A null DLYBCT value (delay between consecutive transfers) in FLEX_SPI_CSR, gives even less time for the processor to reload FLEX_SPI_TDR. With some SPI client peripherals, if the chip select line must remain active (low) during a full set of transfers, communication errors can occur.

To facilitate interfacing with such devices, the Chip Select registers [CSR0...CSR1] can be programmed with the Chip Select Active After Transfer (CSAAT) bit to 1. This allows the chip select lines to remain in their current state (low = active) until a transfer to another chip select is required. Even if FLEX_SPI_TDR is not reloaded, the chip select remains active. To de-assert the chip select line at the end of the transfer, the Last Transfer (LASTXFER) bit in FLEX_SPI_CR must be set after writing the last data to transmit into FLEX_SPI_TDR.

33.8.3.9. Peripheral Deselection with PDC

PDC provides faster reloads of FLEX_SPI_TDR compared to software. However, depending on the system activity, it is never sure that FLEX_SPI_TDR is written with the next data before the end of the current transfer. Consequently, a data can be lost by the deassertion of the NPCS line for SPI client

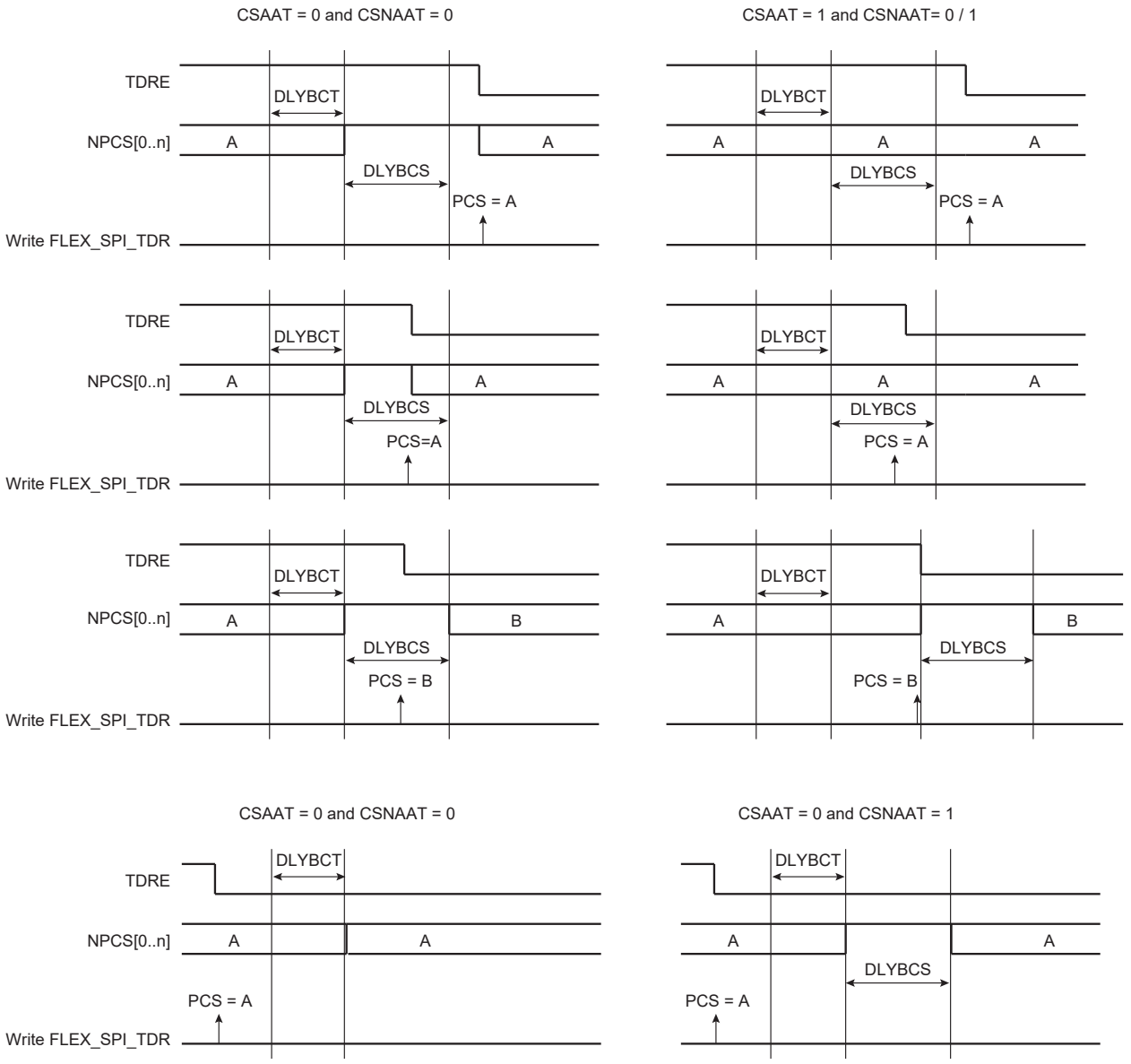
© 2026 Microchip Technology Inc. and its subsidiaries

peripherals requiring the chip select line to remain active between two transfers. The only way to ensure a safe transfer in this case is the use of the CSAAT and LASTXFER bits.

When the CSAAT bit is cleared, the NPCS does not rise in all cases between two transfers on the same peripheral. During a transfer on a Chip Select, the TDRE flag rises as soon as the content of FLEX_SPI_TDR is transferred into the internal shift register. When this flag is detected, FLEX_SPI_TDR can be reloaded. If this reload occurs before the end of the current transfer and if the next transfer is performed on the same chip select as the current transfer, the Chip Select is not deasserted between the two transfers. This can lead to difficulties to interface with some serial peripherals requiring the chip select to be deasserted after each transfer. To facilitate interfacing with such devices, FLEX_SPI_CSR can be programmed with the Chip Select Not Active After Transfer (CSNAAT) bit to 1. This allows the chip select lines to be deasserted systematically during a time “DLYBCS” (the value of the CSNAAT bit is processed only if the CSAAT bit is cleared for the same chip select).

The following figure shows different peripheral deselection cases and the effect of the CSAAT and CSNAAT bits.

Figure 33-76. Peripheral Deselection



© 2026 Microchip Technology Inc. and its subsidiaries

33.8.3.10. Mode Fault Detection

The SPI has the capability to operate in multi-host environment. Consequently, the NPCS0/NSS line must be monitored. If one of the hosts on the SPI bus is currently transmitting, the NPCS0/NSS line is low and the SPI must not transmit a data. A mode fault is detected when the SPI is programmed in Host mode and a low level is driven by an external host on the NPCS0/NSS signal. In multi-host environment, NPCS0, MOSI, MISO and SPCK pins must be configured in open drain (through the PIO controller). When a mode fault is detected, the FLEX_SPI_SR.MODF bit is set until FLEX_SPI_SR is read and the SPI is automatically disabled until it is re-enabled by writing the FLEX_SPI_CR.SPIEN bit to 1.

By default, the mode fault detection is enabled. The user can disable it by setting the FLEX_SPI_MR.MODFDIS bit.

33.8.4. SPI Client Mode

When operating in Client mode, the SPI processes data bits on the clock provided on the SPI clock pin (SPCK).

The SPI waits until NSS goes active before receiving the serial clock from an external host. When NSS falls, the clock is validated and the data are loaded in FLEX_SPI_RDR according to the configuration value of the FLEX_SPI_CSR0.BITS field. These bits are processed following a phase and a polarity defined respectively by the FLEX_SPI_CSR0.NCPHA and FLEX_SPI_CSR0.CPOL bits. Note that the BITS field, CPOL bit and NCPHA bit of the other Chip Select registers have no effect when the SPI is programmed in Client mode.

The bits are shifted out on the MISO line and sampled on the MOSI line.

Note: For more information on the BITS field, see also the note below the FLEX_SPI_CSRx register bitmap in section SPI Chip Select Register

When all bits are processed, the received data are transferred in FLEX_SPI_RDR and the RDRF bit rises. If FLEX_SPI_RDR has not been read before new data are received, the Overrun Error bit (OVRES) in FLEX_SPI_SR is set. As long as this flag is set, data are loaded in FLEX_SPI_RDR. The user must read FLEX_SPI_SR to clear the OVRES bit.

When a transfer starts, the data shifted out is the data present in the shift register. If no data has been written in FLEX_SPI_TDR, the last data received is transferred. If no data has been received since the last reset, all bits are transmitted low, as the shift register resets to 0.

When a first data is written in FLEX_SPI_TDR, it is transferred immediately in the shift register and the TDRE flag rises. If new data is written, it remains in FLEX_SPI_TDR until a transfer occurs, i.e., NSS falls and there is a valid clock on the SPCK pin. When the transfer occurs, the last data written in FLEX_SPI_TDR is transferred in the shift register and the TDRE flag rises. This enables frequent updates of critical variables with single transfers.

Then, a new data is loaded in the shift register from FLEX_SPI_TDR. If no character is ready to be transmitted, i.e., no character has been written in FLEX_SPI_TDR since the last load from FLEX_SPI_TDR to the shift register, FLEX_SPI_TDR is retransmitted. In this case the Underrun Error Status flag (UNDES) is set in FLEX_SPI_SR.

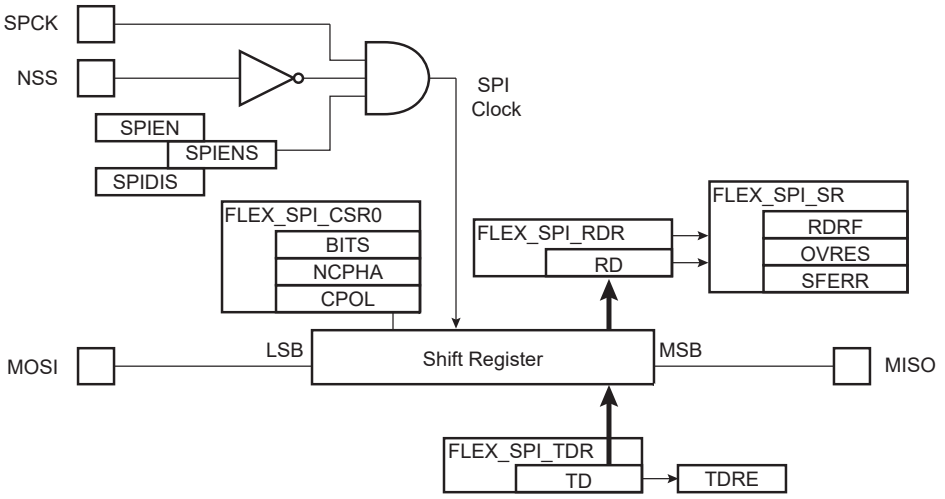
If NSS rises between two characters, it must be kept high for two MCK clock periods or more and the next SPCK capture edge must not occur less than four MCK periods after NSS rise.

In Client mode, if the NSS line rises and the received character length does not match the configuration defined in FLEX_SPI_CSR0.BITS, the SFERR flag is set in FLEX_SPI_SR.

The following figure shows a block diagram of the SPI when operating in Client mode.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-77. Client Mode Functional Block Diagram



33.8.5. SPI Comparison Function on Received Character

The comparison is only relevant for SPI Client mode (MSTR = 0 in FLEX_US_MR).

The effect of a comparison match changes if the system is in Wait or Active mode.

In Wait mode, if asynchronous partial wakeup is enabled, a system wakeup is performed (see SPI Asynchronous and Partial Wake-up).

In Active mode, the CMP flag in FLEX_SPI_SR is raised. It is set when the received character matches the conditions programmed in the SPI Comparison register (FLEX_SPI_CMPR). The CMP flag is set as soon as FLEX_SPI_RDR is loaded with the new received character. The CMP flag is cleared by reading FLEX_SPI_SR.

The SPI Comparison register can be programmed to provide different comparison methods. These are listed below:

- If VAL1 equals VAL2, then the comparison is performed on a single value and the flag is set to 1 if the received character equals VAL1.
- If VAL1 is strictly lower than VAL2, then any value between VAL1 and VAL2 sets the CMP flag.
- If VAL1 is strictly higher than VAL2, then the flag CMP is set to 1 if any received character equals VAL1 or VAL2.

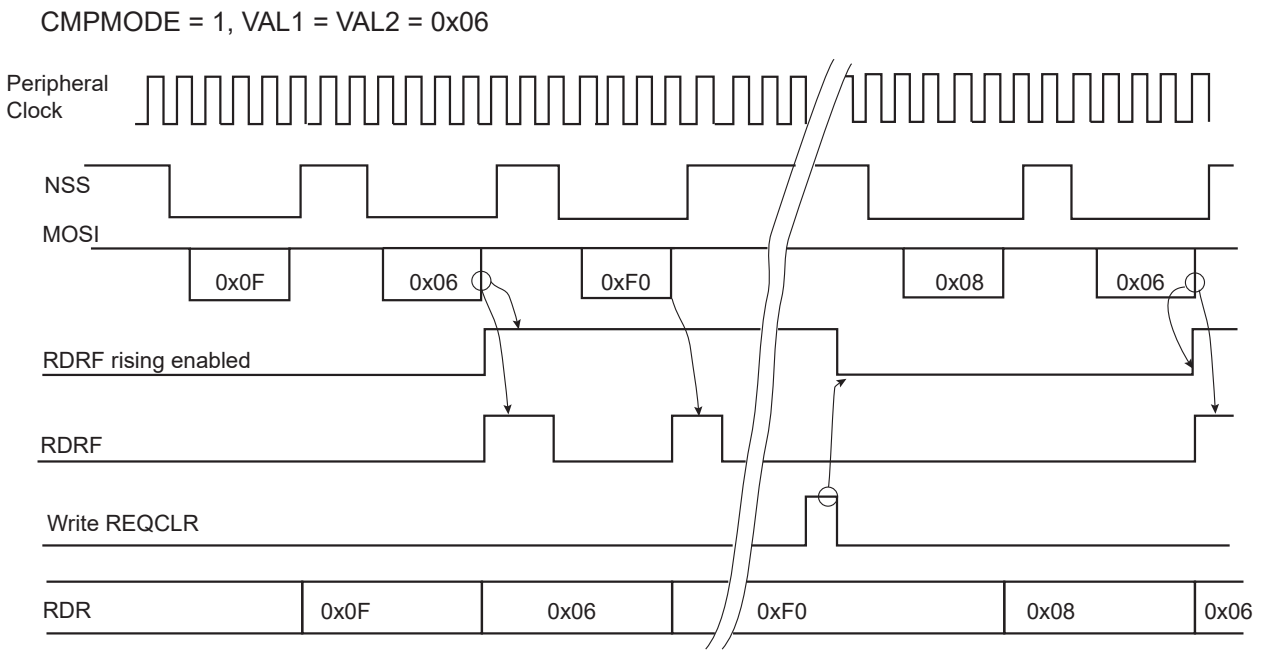
When FLEX_SPI_MR.CMPMODE is cleared, all received data is loaded in FLEX_SPI_RDR and the CMP flag provides the status of the comparison result.

By setting the CMPMODE bit, the comparison result triggers the start of FLEX_SPI_RDR loading (see the figure below). The trigger condition exists as soon as the received character value matches the conditions defined by VAL1 and VAL2 in FLEX_SPI_CMPR. The comparison trigger event is restarted by writing a 1 to the FLEX_SPI_CR.REQCLR bit.

The value programmed in VAL1 and VAL2 fields must not exceed the maximum value of the received character (see BITS field in SPI Chip Select register (FLEX_SPI_CSR)).

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-78. Receive Data Register Management



33.8.6. SPI Asynchronous and Partial Wake-up

This operating mode is a means of data preprocessing that qualifies an incoming event, thus allowing the SPI to decide whether or not to wake up the system. Asynchronous and partial wakeup is mainly used when the system is in Wait mode (refer to the section “Power Management Controller (PMC)" for further details). It can also be enabled when the system is fully running. In any case, only the peripheral clock is modified and VDDCORE always remains active.

Asynchronous and partial wake-up can be used only when SPI is configured in Client mode (FLEX_SPI_MR.MSTR is cleared).

The maximum SPI clock (SPCK) frequency that can be provided by the SPI host is bounded by the peripheral clock frequency. The SPCK frequency must be lower than or equal to the peripheral clock. The NSS line must be deasserted by the SPI host between two characters. The NSS deassertion duration time must be greater than or equal to six peripheral clock periods. The time between the assertion of NSS line (falling edge) and the first edge of the SPI clock must be higher than 15 µs.

The FLEX_SPI_RDR register must be read before enabling the asynchronous and partial wake-up.

When asynchronous and partial wake-up is enabled for the SPI (refer to the section “Power Management Controller (PMC)"), the PMC decodes a clock request from the SPI. The request is generated as soon as there is a falling edge on the NSS line as this may indicate the beginning of a frame. If the system is in Wait mode (processor and peripheral clocks switched off), the PMC restarts the fast RC oscillator and provides the clock only to the SPI.

The SPI processes the received frame and compares the received character with VAL1 and VAL2 in FLEX_SPI_CMPR.

The SPI instructs the PMC to disable the peripheral clock if the received character value does not meet the conditions defined by VAL1 and VAL2 fields in FLEX_SPI_CMPR (see figure Asynchronous Wake-up Use Case Example).

© 2026 Microchip Technology Inc. and its subsidiaries

If the received character value meets the conditions, the SPI instructs the PMC to exit the system from Wait mode (see figure Asynchronous Event Generating Only Partial Wake-up).

The VAL1 and VAL2 fields can be programmed to provide different comparison methods and thus matching conditions.

- If VAL1 equals VAL2, then the comparison is performed on a single value and the wake-up is triggered if the received character equals VAL1.
- If VAL1 is strictly lower than VAL2, then any value between VAL1 and VAL2 wakes up the system.
- If VAL1 is strictly higher than VAL2, the wake-up is triggered if any received character equals VAL1 or VAL2.
- If VAL1 = 0 and VAL2 = 65535, the wake-up is triggered as soon as a character is received.

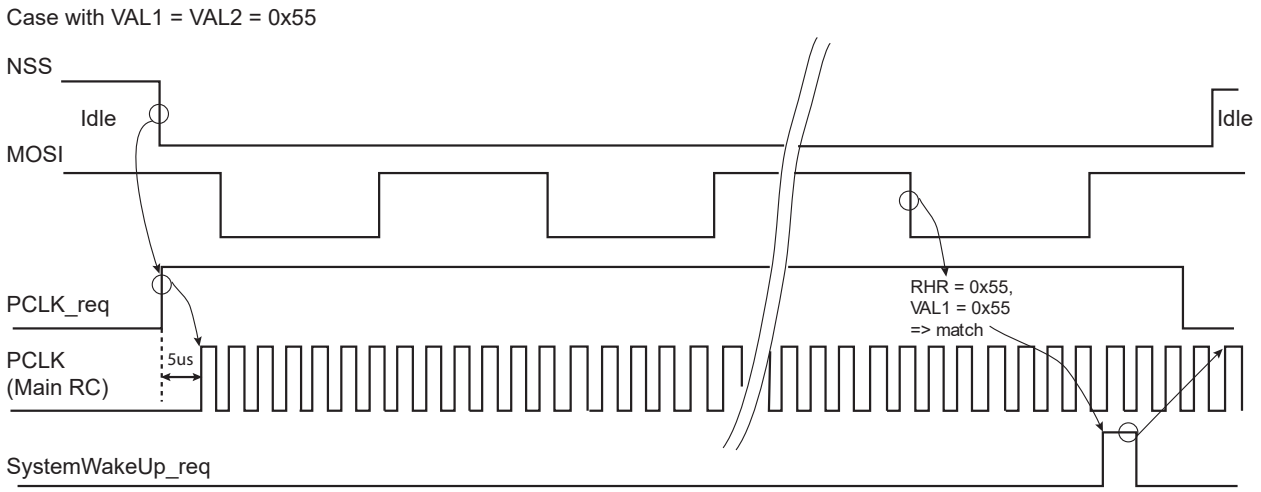
If the processor and peripherals are running, the SPI can be configured in Asynchronous and Partial Wake-up mode by enabling the PMC_SLPWK_ER (refer to the section “Power Management Controller (PMC)"). When activity is detected on the receive line, the SPI requests the clock from the PMC and the comparison is performed. If there is a comparison match, the SPI continues to request the clock. If there is no match, the clock is switched off for the SPI only, until a new activity is detected.

The CMPMODE configuration has no effect when Asynchronous and Partial Wake-up mode is enabled for the SPI (refer to PMC_SLPWK_ER in the section “Power Management Controller (PMC)").

When the system is in Active mode and the SPI enters Asynchronous and Partial Wake-up mode, the flag RDRF must be programmed as the unique source of the SPI interrupt.

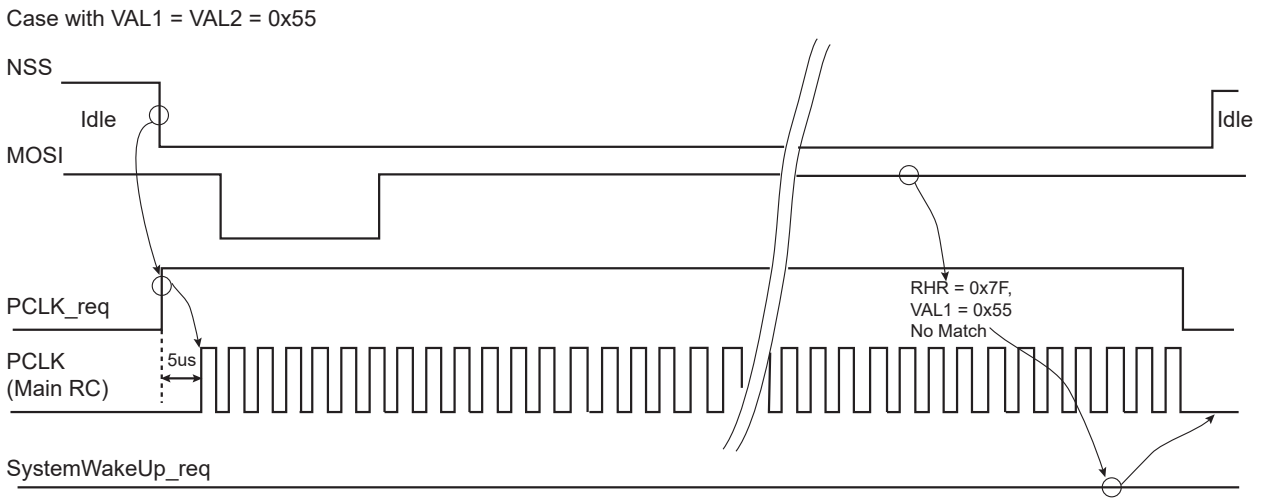
When the system exits Wait mode as the result of a matching condition, the RDRF flag is used to determine if the SPI is the source for the exit from Wait mode.

Figure 33-79. Asynchronous Wake-up Use Case Example



© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-80. Asynchronous Event Generating Only Partial Wake-up



33.8.7. SPI FIFOs

33.8.7.1. Overview

The SPI includes two FIFOs which can be enabled/disabled using the FLEX_SPI_CR.FIFOEN/FIFODIS. The SPI module must be disabled before enabling or disabling the SPI FIFOs (FLEX_SPI_CR.SPIDIS).

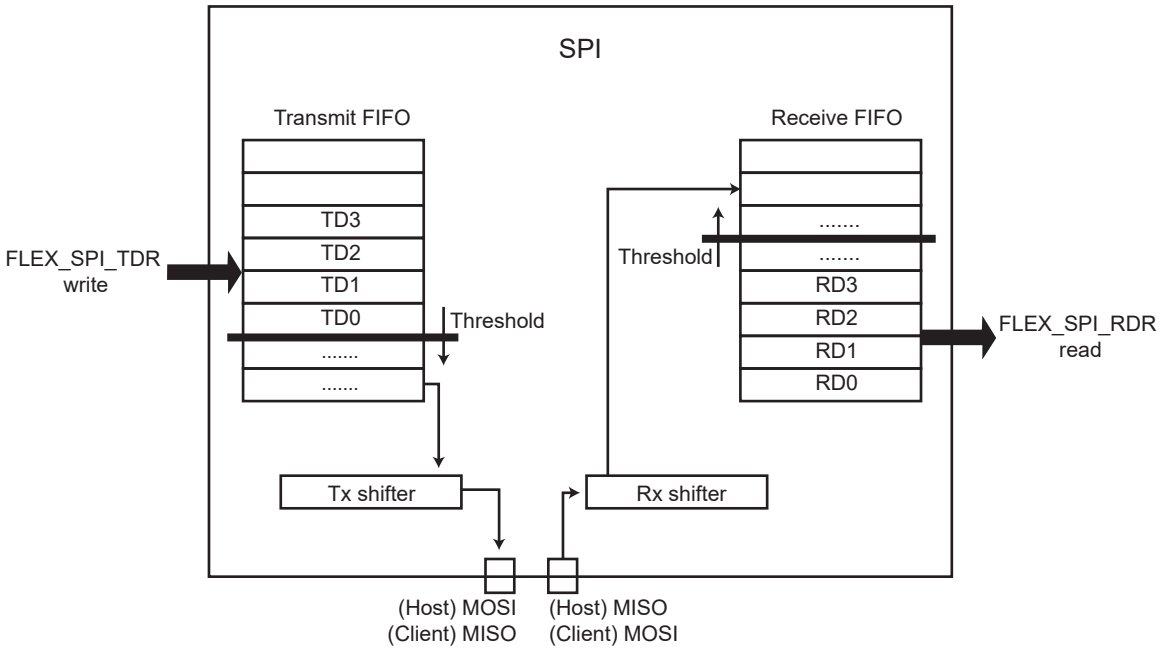
Writing FLEX_SPI_CR.FIFOEN to '1' enables a 8-data Transmit FIFO and a 8-data Receive FIFO.

The size of a data (8-bit to 16-bit) is determined by the value configured in FLEX_SPI_CSRx.BITS.

It is possible to write or to read single or multiple data in the same access to FLEX_SPI_TDR/RDR. See SPI Single Data Access and SPI Multiple Data Access.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-81. SPI FIFOs Block Diagram



33.8.7.2. Sending Data with FIFO Enabled

When the Transmit FIFO is enabled, write access to FLEX_SPI_TDR loads the Transmit FIFO.

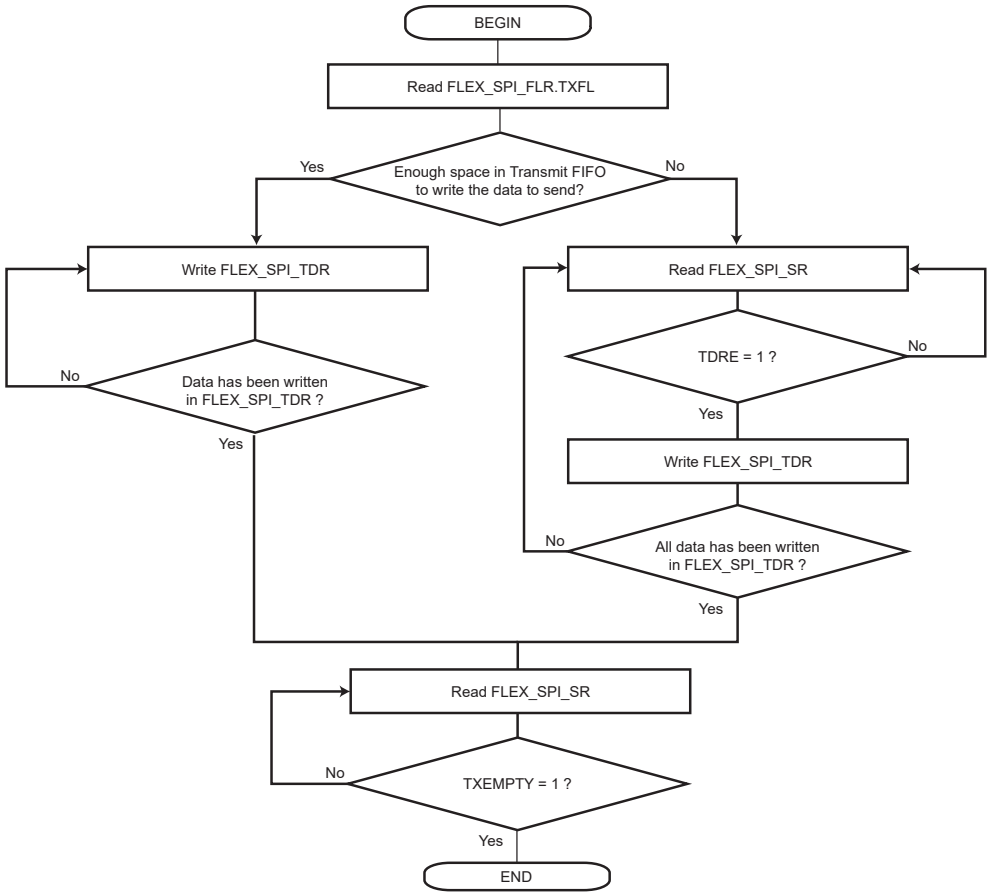
The FIFO level is provided in FLEX_SPI_FLR.TXFL. If the FIFO can accept the number of data to be transmitted, there is no need to monitor FLEX_SPI_SR.TDRE and the data can be successively written in FLEX_SPI_TDR.

If the FIFO cannot accept the data due to insufficient space, wait for the TDRE flag to be set before writing the data in FLEX_SPI_TDR.

When the space in the FIFO allows only a portion of the data to be written, the TDRE flag must be monitored before writing the remaining data.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-82. Sending Data with FIFO Enabled



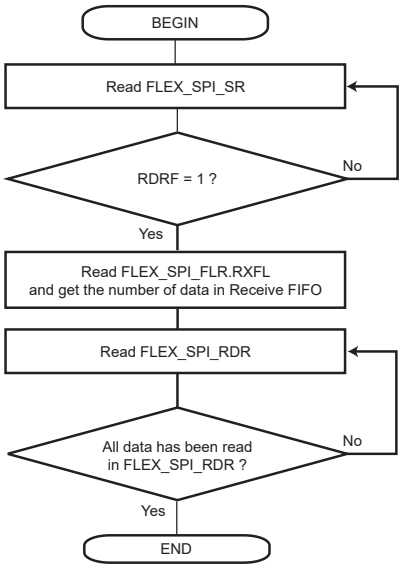
33.8.7.3. Receiving Data with FIFO Enabled

When the Receive FIFO is enabled, FLEX_SPI_RDR access reads the FIFO.

When data are present in the Receive FIFO (RDRF flag set to ‘1’), the exact number of data can be checked with FLEX_SPI_FLR.RXFL. All the data can be read successively in FLEX_SPI_RDR without checking the RDRF flag between each access.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-83. Receiving Data with FIFO Enabled



33.8.7.4. Clearing/Flushing FIFOs

Each FIFO can be cleared/flushed using FLEX_SPI_CR.TXFCLR/RXFCLR.

33.8.7.5. TXEMPTY, TDRE and RDRF Behavior

FLEX_SPI_SR.TXEMPTY, FLEX_SPI_SR.TDRE and FLEX_SPI_SR.RDRF flags display a specific behavior when FIFOs are enabled.

The TXEMPTY flag is cleared as long as there are characters in the Transmit FIFO or in the internal shift register. TXEMPTY is set when there are no characters in the Transmit FIFO and in the internal shift register.

TDRE indicates if a data can be written in the Transmit FIFO. Thus the TDRE flag is set as long as the Transmit FIFO can accept new data. See figure TDRE Behavior for Single Data Access and TXRDYM = 0.

RDRF indicates if an unread data is present in the Receive FIFO. Thus the RDRF flag is set as soon as one unread data is in the Receive FIFO. See figure RDRF Behavior in Single Data Access and RXRDYM = 0.

TDRE and RDRF behavior can be modified using the TXRDYM and RXRDYM fields in the SPI FIFO Mode register (FLEX_SPI_FMR) to reduce the number of accesses to FLEX_SPI_TDR/RDR. However, for some configurations, the following constraints apply:

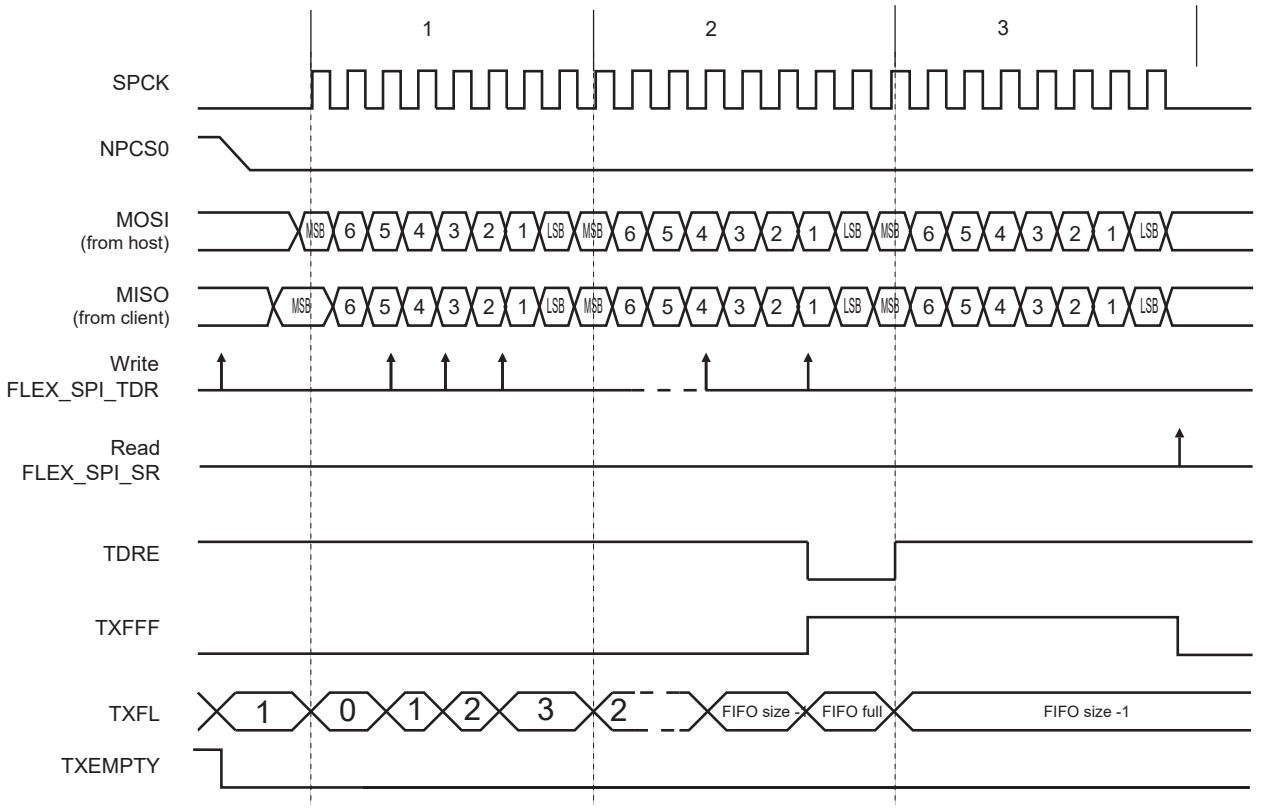
- When Variable Peripheral Select mode is used (FLEX_SPI_MR.PS=1), TXRDYM/RXRDYM must be cleared.
- In Host mode (FLEX_SPI_MR.MSTR=1), RXRDYM must be cleared.

As an example, in Host mode, the Transmit FIFO can be loaded with multiple data in the same access by configuring TXRDYM>0.

See SPI FIFO Mode register (FLEX_SPI_FMR) for the FIFO configuration.

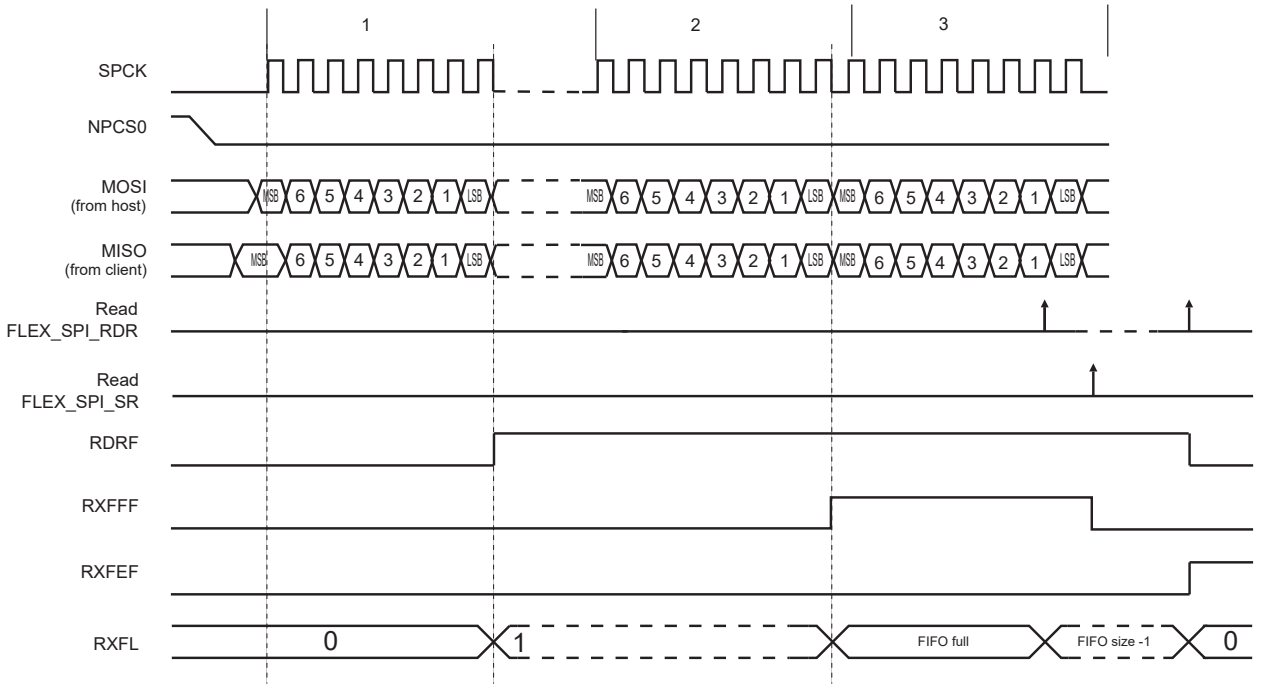
© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-84. TDRE Behavior for Single Data Access and TXRDYM = 0



© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-85. RDRF Behavior for Single Data Access and RXRDYM = 0



33.8.7.6. SPI Single Data Access

When FIFO is enabled and a byte or a halfword access (8-bit to 16-bit data) is performed in FLEX_SPI_TDR, a single data is written in FIFO each time FLEX_SPI_TDR is accessed. The similar behavior applies for FLEX_SPI_RDR.

If Host mode is used (FLEX_SPI_MR.MSTR=1) or if Variable Peripheral Select mode is used (FLEX_SPI_MR.PS=1), each access to FLEX_SPI_RDR must be read a single data.

See SPI Transmit Data Register and SPI Receive Data Register.

However, for some configurations it is possible to write/read multiple data each time FLEX_SPI_TDR/FLEX_SPI_RDR is accessed. See SPI Multiple Data Access.

33.8.7.6.1. PDC

When FIFOs operate in Single Data mode, the PDC transfer type must be configured either in bytes, halfwords or words depending on FLEX_SPI_MR.PS bit value and FLEX_SPI_CSRx.BITS field value.

The same applies when FIFOs are disabled.

33.8.7.7. SPI Multiple Data Access

For some operating modes, it is possible to reduce the number of accesses to FLEX_SPI_TDR/FLEX_SPI_RDR required to transfer an amount of data, by concatenating multiple data (8-bit or 9-bit to 16-bit) when the FIFO is enabled (FLEX_SPI_CR.FIFOEN=1) and fixed peripheral select is used (FLEX_SPI_MR.PS=0).

Up to two data can be written in one FLEX_SPI_TDR write access.

Up to four data can be read in one FLEX_SPI_RDR access.

When the FIFO is enabled, the number of data written in a single access to FLEX_SPI_TDR is only defined by the type of access.

© 2026 Microchip Technology Inc. and its subsidiaries

Table 33-15. Number of Data Written for Each Access to FLEX_SPI_TDR

Config/Access	Byte	Halfword	Word
8-bit to 16-bit	1	1	2

When the FIFO is enabled, the number of data read in a single access to FLEX_SPI_RDR is defined by the type of access and the configuration of FLEX_SPI_CSR0.BITS.

Table 33-16. Number of Data Read for Each Access to FLEX_SPI_RDR

Config/Access	Byte	Halfword	Word
FLEX_SPI_CSR0.BITS=0 8-bit data	1	2	4
FLEX_SPI_CSR0.BITS>0 9-bit to 16-bit data	1	1	2

Multiple data can be read from the Receive FIFO only in Client mode (FLEX_SPI_MR.MSTR=0).

The Transmit FIFO can be loaded with multiple data in the same FLEX_SPI_TDR access when FLEX_SPI_MR.PS=0.

Written/read data are always right-aligned, as described in sections SPI Receive Data Register (FIFO Multiple Data, 8-bit), SPI Receive Data Register (FIFO Multiple Data, 16-bit) and SPI Transmit Data Register (FIFO Multiple Data, 8- to 16-bit).

As an example, if the Transmit FIFO is empty and there are six data to send, either of the following write accesses may be performed:

- six FLEX_SPI_TDR-byte write accesses
- three FLEX_SPI_TDR-halfword write accesses

With a Receive FIFO containing six data, any of the following read accesses may be performed:

- six FLEX_SPI_RDR-byte read accesses
- three FLEX_SPI_RDR-halfword read accesses
- one FLEX_SPI_RDR-word read access and one FLEX_SPI_RDR-halfword read access

33.8.7.7.1. TDRE and RDRF Configuration

The TDRE flag indicates if one or more data can be written in the FIFO depending on the configuration of FLEX_SPI_FMR.TXRDYM/RXRDYM.

As an example, if two data are written each time in FLEX_SPI_TDR, the TXRDYM field can be configured so that the TDRE flag is at '1' only when at least two data can be written in the Transmit FIFO.

Similarly, if four data are read each time in FLEX_SPI_RDR, the RXRDYM field can be configured so that the RDRF flag is at '1' only when at least four unread data are in the Receive FIFO.

33.8.7.7.2. PDC

It is mandatory to configure PDC channel size (byte, halfword or word) according to the FLEX_SPI_FMR.TXRDYM/RXRDYM configuration. See constraints in SPI Multiple Data Access.

As an example, when FIFO is enabled, FLEX_SPI_FMR.TXRDYM/RXRDYM=0 configuration is not compatible with DMAC_PDC transfers in word (32-bit).

33.8.7.8. FIFO Pointer Error

A FIFO overflow is reported in FLEX_SPI_SR.

If the Transmit FIFO is full and a write access is performed on FLEX_SPI_TDR, it generates a Transmit FIFO pointer error and sets FLEX_SPI_SR.TXFPTEF.

© 2026 Microchip Technology Inc. and its subsidiaries

If the number of data written in FLEX_SPI_TDR (according to the register access size) is greater than the free space in the Transmit FIFO, a Transmit FIFO pointer error is generated and FLEX_SPI_SR.TXFPTEF is set.

A FIFO underflow is reported in FLEX_SPI_SR.

If the number of data read in FLEX_SPI_RDR (according to the register access size) is greater than the number of unread data in the Receive FIFO, a Receive FIFO pointer error is generated and FLEX_SPI_SR.RXFPTEF is set.

No pointer error occurs if the FIFO state/level is checked before writing/reading in FLEX_SPI_TDR/SPI_RDR. The FIFO state/level can be checked either with TXRDY, RXRDY, TXFL or RXFL. When a pointer error occurs, other FIFO flags may not behave as expected; their states should be ignored.

If a pointer error occurs, a software reset must be performed using FLEX_SPI_CR.SWRST (configuration will be lost).

33.8.7.9. FIFO Thresholds

Each Transmit and Receive FIFO includes a threshold feature used to set a flag and an interrupt when a FIFO threshold is crossed. Thresholds are defined as a number of data in the FIFO, and the FIFO state (TXFL or RXFL) represents the number of data currently in the FIFO.

The Transmit FIFO threshold can be set using the field FLEX_SPI_FMR.TXFTHRES. Each time the Transmit FIFO level goes from 'above threshold' to 'equal to or below threshold', the flag FLEX_SPI_SR.TXFTHF is set. The application is warned that the Transmit FIFO has reached the defined threshold and that it can be reloaded.

The Receive FIFO threshold can be set using the field FLEX_SPI_FMR.RXFTHRES. Each time the Receive FIFO level goes from 'below threshold' to 'equal to or above threshold', the flag FLEX_SPI_SR.RXFTHF is set. The application is warned that the Receive FIFO has reached the defined threshold and that it can be read to prevent an underflow.

The TXFTHF and RXFTHF flags can be configured to generate an interrupt using FLEX_SPI_IER and FLEX_SPI_IDR.

33.8.7.10. FIFO Flags

FIFOs come with a set of flags which can be configured to generate interrupts through FLEX_SPI_IER and FLEX_SPI_IDR.

FIFO flags state can be read in FLEX_SPI_SR. They are cleared when FLEX_SPI_SR is read.

33.8.8. SPI Register Write Protection

The FLEXCOM operating mode (FLEX_MR.OPMODE) must be set to FLEX_MR_OPMODE_SPI to enable access to the write protection registers.

To prevent any single software error from corrupting SPI behavior, certain registers in the address space can be write-protected by setting the WPEN (Write Protection Enable), WPITEN (Write Protection Interrupt Enable), and/or WPCREN (Write Protection Control Enable) bits in the SPI Write Protection Mode Register (FLEX_SPI_WPMR).

If a write access to a write-protected register is detected, the Write Protection Violation Status (WPVS) flag in the SPI Write Protection Status Register (FLEX_SPI_WPSR) is set and the Write Protection Violation Source (WPVSRC) field indicates the register in which the write access has been attempted.

The WPVS bit is automatically cleared after reading FLEX_SPI_WPSR.

The following registers can be write-protected when WPEN is set:

- SPI Mode Register
- SPI Chip Select Register

© 2026 Microchip Technology Inc. and its subsidiaries

- SPI Comparison Register

The following registers can be write-protected when WPITEN is set:

- SPI Interrupt Enable Register
- SPI Interrupt Disable Register

The following register can be write-protected when WPCREN is set:

- SPI Control Register

33.8.9. Local Loopback Test Mode

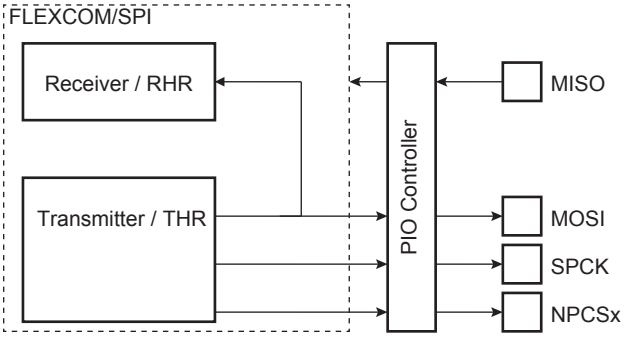
Local Loopback mode connects the output of the transmitter directly to the input of the receiver, as shown in the figure below. The MISO pin has no effect on the receiver and the MOSI pin is driven as in Normal mode.

The MOSI, SPCK and NPCSx are normally transmitted unless the PIO is configured to drive logical values to prevent the SPI external target device from starting.

Local Loopback mode and allows a quick and easy verification of the transmitter and receiver logic.

Local Loopback mode is enabled when FLEX_SPI_MR.LLB=1, FLEX_SPI_MR.MASTER=1 and the FLEX_MR.OPMODE=2.

Figure 33-86. Local Loopback Mode Configuration

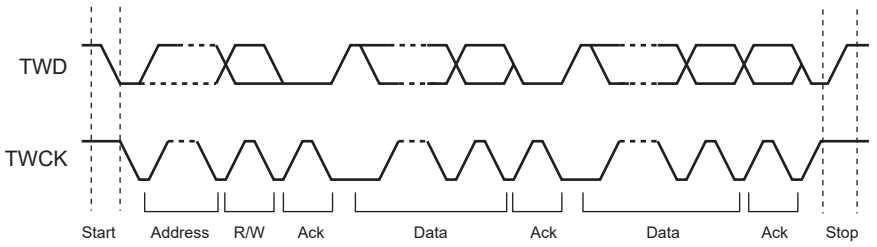


33.9. TWI Functional Description

33.9.1. Transfer Format

The data put on the TWD line must be 8 bits long. Data are transferred MSB first; each byte must be followed by an acknowledgement. The number of bytes per transfer is unlimited (see figure below).

Figure 33-87. Transfer Format

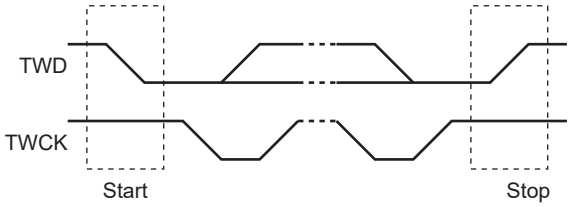


Each transfer begins with a START condition and terminates with a STOP condition (see figure below).

© 2026 Microchip Technology Inc. and its subsidiaries

- A high-to-low transition on the TWD line while TWCK is high defines the START condition.
- A low-to-high transition on the TWD line while TWCK is high defines a STOP condition.

Figure 33-88. START and STOP Conditions



33.9.1.1. Digital Filter

The TWI features digital filters on data and clock lines that can be configured by software via the TWI Filter register (FLEX_TWI_FILTR).

In Standard, Fast and Fast Plus modes, the digital filter must be enabled (FILT=1) and a pulse width threshold defined (THRES > 0).

The field THRES must be set according to the peripheral clock to suppress spikes below 50 ns. The recommended value is calculated using the formula:

$THRES > 50\ ns/t_{peripheral\_clock(ns)}$

33.9.2. Modes of Operation

The TWI has different modes of operation:

- Host Transmitter mode
- Host Receiver mode
- Multi-host Transmitter mode
- Multi-host Receiver mode
- Client Transmitter mode
- Client Receiver mode

These modes are described in the following sections.

33.9.3. Host Mode

33.9.3.1. Definition

The host is the device that starts a transfer, generates a clock and stops it. Host mode is not available if High-speed mode is selected.

33.9.3.2. Programming Host Mode

The following fields must be programmed before entering Host mode:

1. DADR (+ IADRSZ + IADR if a 10-bit device is addressed): The device address is used to access client devices in Read or Write mode.
2. CWGR + CKDIV + CHDIV + CLDIV: Clock waveform.
3. SVDIS: Disables Client mode.
4. MSEN: Enables Host mode.

Note: If the TWI is already in Host mode, the device address (DADR) can be configured without disabling Host mode.

© 2026 Microchip Technology Inc. and its subsidiaries

33.9.3.3. Transfer Speed/Bit Rate

The TWI speed is defined in FLEX_TWI_CWGR. The TWI bit rate can be based either on the peripheral clock if the BRSRCCLK bit value is 0 or on a programmable clock source provided by the GCLK if the BRSRCCLK bit value is 1.

If BRSRCCLK = 1, the bit rate is independent of the processor/peripheral clock and thus processor/peripheral clock frequency can be changed without affecting the TWI transfer rate.

The GCLK frequency must be at least three times lower than the peripheral clock frequency.

33.9.3.4. Host Transmitter Mode

This operating mode is not available if High-speed mode is selected.

After the host initiates a START condition when writing into the Transmit Holding register FLEX_TWI_THR, it sends a 7-bit client address, configured in the Host Mode register (DADR in FLEX_TWI_MMR), to notify the client device. The bit following the client address indicates the transfer direction, 0 in this case (FLEX_TWI_MMR.MREAD = 0).

The TWI transfers require the client to acknowledge each received byte. During the acknowledge clock pulse (ninth pulse), the host releases the data line (HIGH), enabling the client to pull it down in order to generate the acknowledge. If the client does not acknowledge the byte, then the Not Acknowledge flag (NACK) is set in the TWI Status register (FLEX_TWI_SR) of the host and a STOP condition is sent. Alternatively, if the FLEX_TWI_MMR.NOAP bit is set, no stop condition will be sent and a START or STOP condition must be triggered manually through the FLEX_TWI_CR.START or FLEX_TWI_CR.STOP bit once the software is ready for the transmission of the condition. The NACK flag must be cleared by reading the TWI Status register (FLEX_TWI_SR) before the next write into the TWI Transmit Holding register (FLEX_TWI_THR). As with the other status bits, an interrupt can be generated if enabled in the Interrupt Enable register (FLEX_TWI_IER). If the client acknowledges the byte, the data written in FLEX_TWI_THR is then shifted in the internal shifter and transferred. When an acknowledge is detected, the TXRDY bit is set until a new write in FLEX_TWI_THR.

TXRDY is used as transmit ready for the PDC transmit channel.

Note: To clear the TXRDY flag in Host mode, write the FLEX_TWI_CR.MSDIS bit to 1, then write the FLEX_TWI_CR.MSEN bit to 1.

While no new data is written in FLEX_TWI_THR, the serial clock line is tied low. When new data is written in FLEX_TWI_THR, the SCL is released and the data is sent. To generate a STOP event, the STOP command must be performed by writing in the STOP field of the TWI Control register (FLEX_TWI_CR).

After a host write transfer, the Serial Clock line is stretched (tied low) while no new data is written in FLEX_TWI_THR or until a STOP command is performed.

See the following figures.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-89. Host Write with One Data Byte

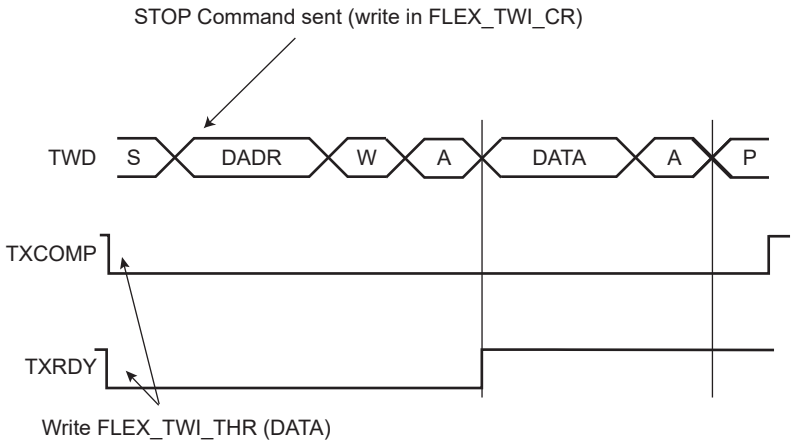
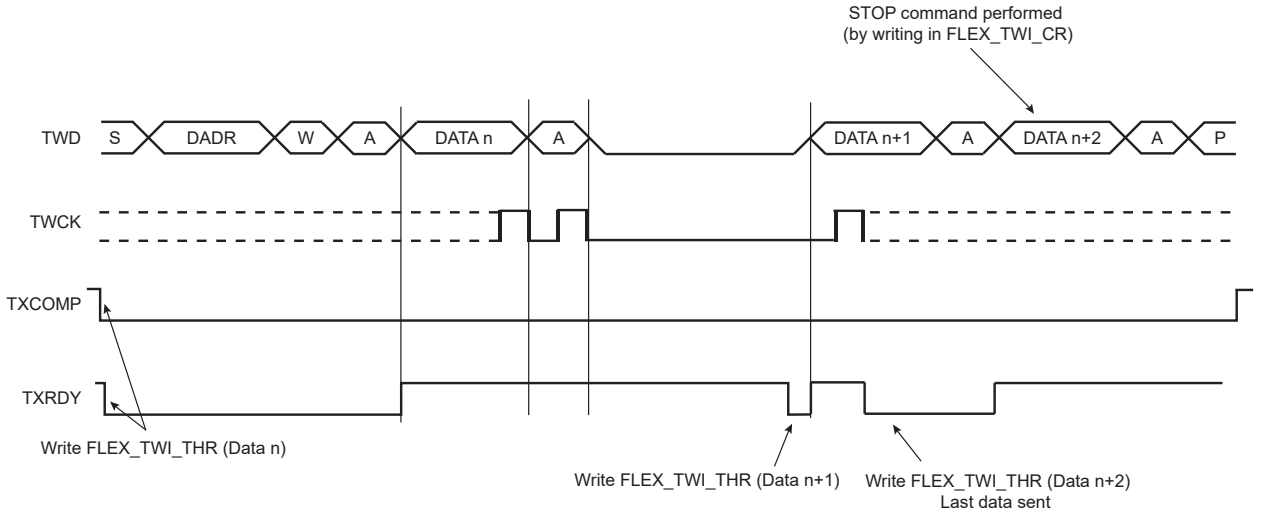
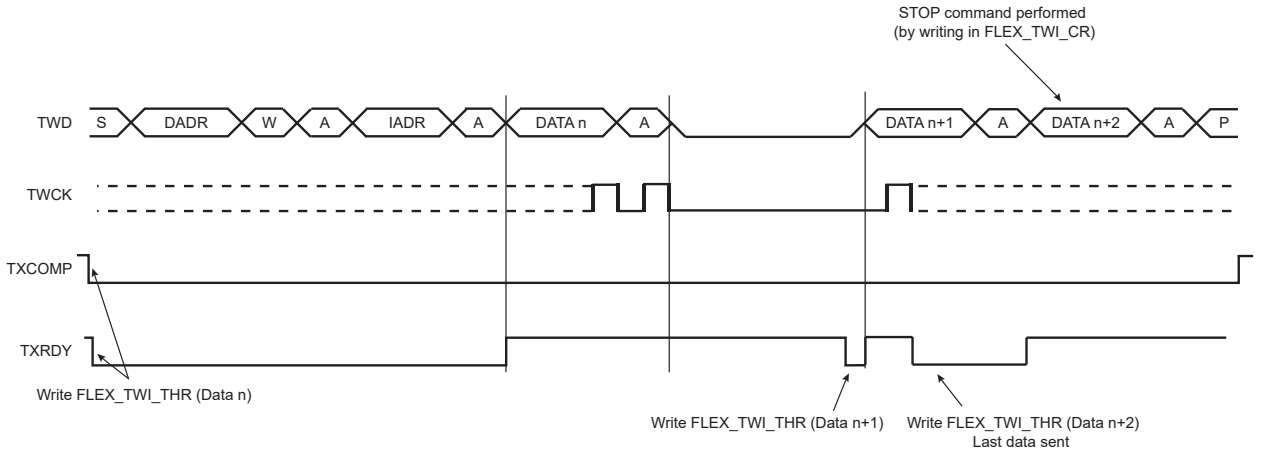


Figure 33-90. Host Write with Multiple Data Bytes



© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-91. Host Write with One Byte Internal Address and Multiple Data Bytes



33.9.3.5. Host Receiver Mode

Host Receiver mode is not available if High-speed mode is selected.

The read sequence begins by setting the START bit. After the start condition has been sent, the host sends a 7-bit client address to notify the client device. The bit following the client address indicates the transfer direction, 1 in this case (FLEX_TWI_MMR.MREAD = 1). During the acknowledge clock pulse (9th pulse), the host releases the data line (HIGH), enabling the client to pull it down in order to generate the acknowledge. The host polls the data line during this clock pulse and sets the FLEX_TWI_SR.NACK bit if the client does not acknowledge the byte.

If an acknowledge is received, the host is then ready to receive data from the client. After data has been received, the host sends an acknowledge condition to notify the client that the data has been received except for the last data (see figure "Host Read with One Data Byte" below). When the FLEX_TWI_SR.RXRDY bit is set, a character has been received in the Receive Holding register (FLEX_TWI_RHR). The RXRDY bit is reset when reading FLEX_TWI_RHR.

When a single data byte read is performed, with or without internal address (IADR), the START and STOP bits must be set at the same time. See figure "Host Read with One Data Byte" below. When a multiple data byte read is performed, with or without internal address (IADR), the STOP bit must be set after the next-to-last data received (same condition applies for START bit to generate a repeated start). See figure "Host Read with Multiple Data Bytes" below. For internal address usage, see Internal Address.

If FLEX_TWI_RHR is full (RXRDY high) and the host is receiving data, the serial clock line will be tied low before receiving the last bit of the data and until FLEX_TWI_RHR is read. Once FLEX_TWI_RHR is read, the host will stop stretching the serial clock line and end the data reception. See figure "Host Read Clock Stretching with Multiple Data Bytes" below.

⚠ WARNING When receiving multiple bytes in Host Read mode, if the next-to-last access is not read (the RXRDY flag remains high), the last access will not be completed until FLEX_TWI_RHR is read. The last access stops on the next-to-last bit (clock stretching). When FLEX_TWI_RHR is read there is only half a bit period to send the STOP bit (or START bit) command, else another read access might occur (spurious access). A possible workaround is to set the STOP bit (or START bit) before reading FLEX_TWI_RHR on the next-to-last access (within IT handler).

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-92. Host Read with One Data Byte

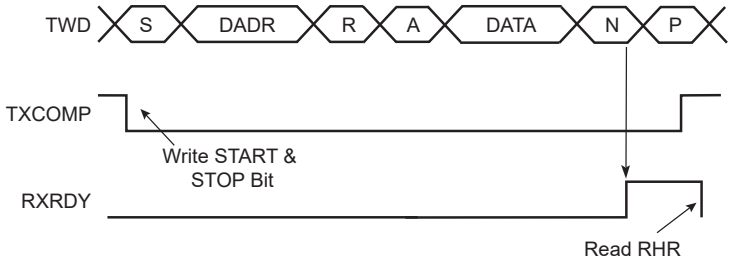


Figure 33-93. Host Read with Multiple Data Bytes

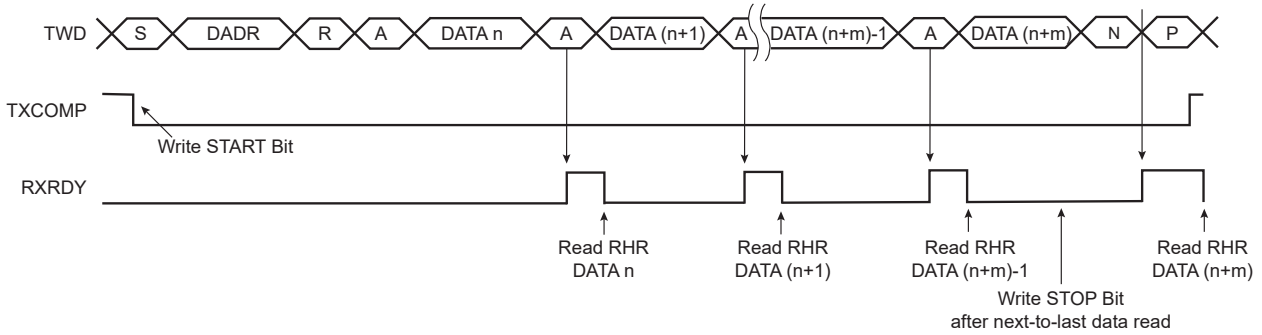
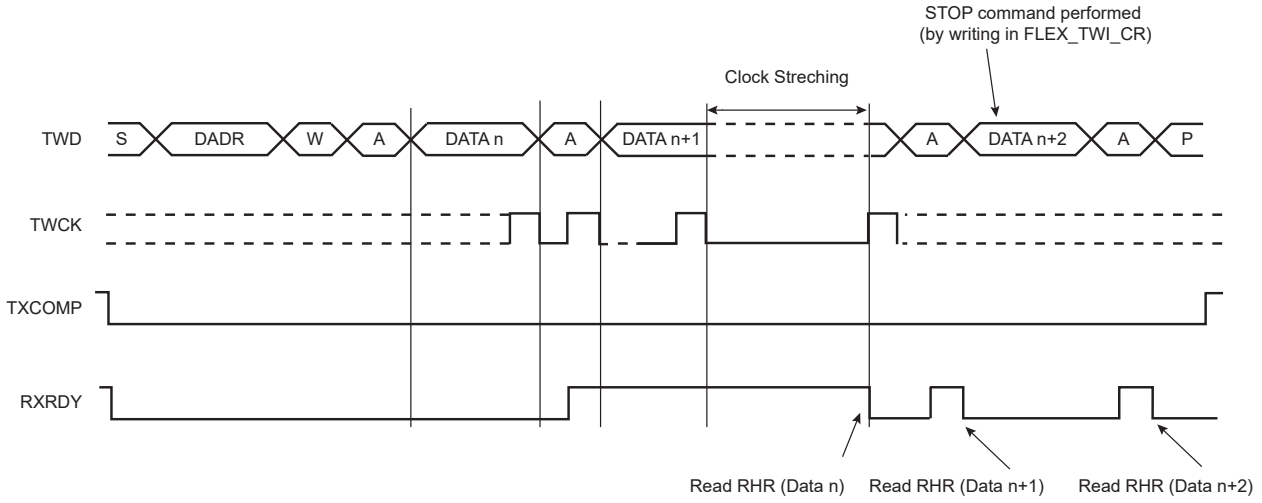


Figure 33-94. Host Read Clock Stretching with Multiple Data Bytes



RXRDY is used as receive ready trigger event for the PDC receive channel.

33.9.3.6. Internal Address

The TWI interface can perform transfers with 7-bit client address devices and with 10-bit client address devices.

33.9.3.6.1. 7-bit Client Addressing

When addressing 7-bit client devices, the internal address bytes are used to perform random address (read or write) accesses to reach one or more data bytes, for example, within a memory page location in a serial memory. When performing read operations with an internal address, the TWI performs a write operation to set the internal address into the client device, and then switch to Host Receiver mode. Note that the second start condition (after sending the IADR) is sometimes called “repeated start” (Sr) in I2C fully-compatible devices. See figure Host Read with One, Two or Three Bytes Internal Address and One Data Byte.

© 2026 Microchip Technology Inc. and its subsidiaries

See figures Host Write with One, Two or Three Bytes Internal Address and One Data Byte and Internal Address Usage for the host write operation with internal address.

The three internal address bytes are configurable through the Host Mode register (FLEX_TWI_MMR).

If the client device supports only a 7-bit address, that is, no internal address, IADRSZ must be configured to 0.

The abbreviations listed below are used in the following figures:

S	Start
Sr	Repeated Start
P	Stop
W	Write
R	Read
A	Acknowledge
N	Not Acknowledge
DADR	Device Address
IADR	Internal Address

Figure 33-95. Host Write with One, Two or Three Bytes Internal Address and One Data Byte

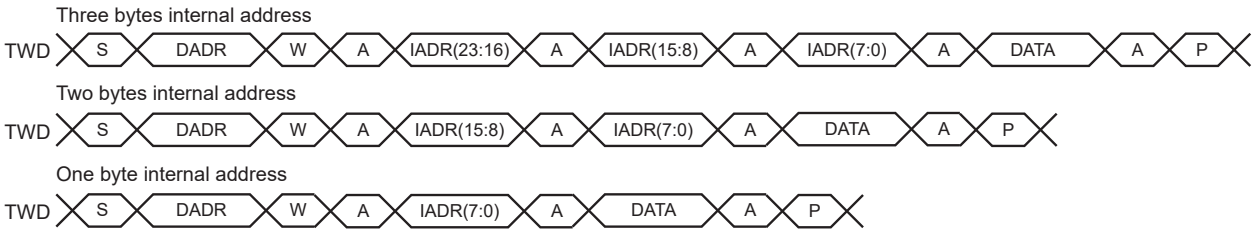
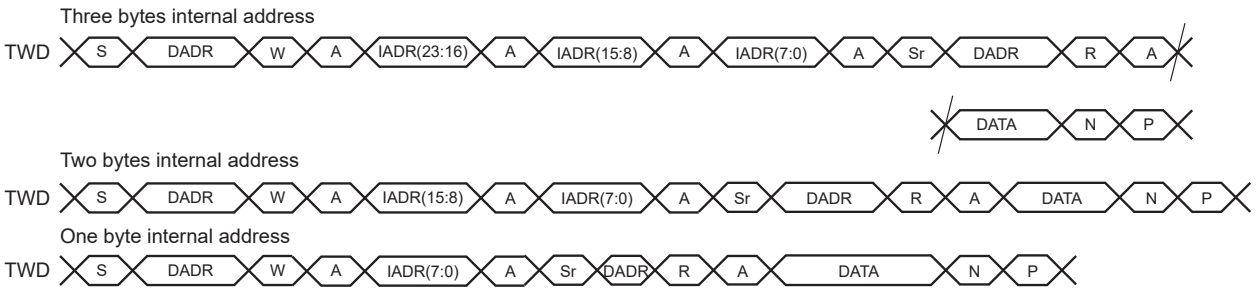


Figure 33-96. Host Read with One, Two or Three Bytes Internal Address and One Data Byte



33.9.3.6.2. 10-bit Client Addressing

For a client address higher than seven bits, the user must configure the address size (IADRSZ) and set the other client address bits in the Internal Address register (FLEX_TWI_IADR). The two remaining internal address bytes, IADR[15:8] and IADR[23:16], can be used the same way as in 7-bit client addressing.

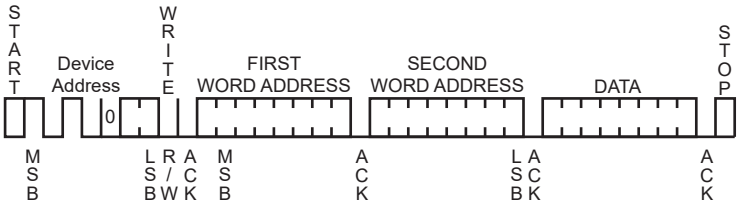
Example: Address a 10-bit device (10-bit device address is b1 b2 b3 b4 b5 b6 b7 b8 b9 b10)

© 2026 Microchip Technology Inc. and its subsidiaries

1. Program IADRSZ = 1.
2. Program DADR with 1 1 1 1 0 b1 b2 (b1 is the MSB of the 10-bit address, b2, etc.).
3. Program FLEX_TWI_IADR with b3 b4 b5 b6 b7 b8 b9 b10 (b10 is the LSB of the 10-bit address).

The following figure shows a byte write to a TWI EEPROM. This demonstrates the use of internal addresses to access the device.

Figure 33-97. Internal Address Usage



33.9.3.7. Repeated Start

In addition to Internal Address mode, repeated start (Sr) can be generated manually by writing the START bit at the end of a transfer instead of the STOP bit. In such case the parameters of the next transfer (direction, SADR, etc.) will need to be set before writing the START bit at the end of the previous transfer.

See Read/Write Flowcharts.

33.9.3.8. Bus Clear Command

The TWI interface can perform a Bus Clear command:

1. Configure Host mode (DADR, CKDIV, etc).
2. Read FLEX_TWI_SR.SDA/SCL flags and check the TWD (SDA) and TWCK (SCL) lines hold a high level.
3. Send the Bus Clear command by setting FLEX_TWI_CR.CLEAR.

Note: When TWD (SDA)=0, the Bus Clear command must be performed via the PIO. When TWCK (SCL)=0, no Bus Clear command can be issued.

Note: If an alternative command is used (ACMEN bit = 1), the DATAL field must be cleared.

33.9.3.9. Using the Peripheral DMA Controller (PDC) in Host Mode

The use of the PDC significantly reduces the CPU load.

To ensure correct implementation, respect the following programming sequences:

33.9.3.9.1. Data Transmit with the PDC in Host Mode

The PDC transfer size must be defined with the buffer size minus 1. The remaining character must be managed without PDC to ensure that the exact number of bytes are transmitted regardless of system bus latency conditions during the end of the buffer transfer period.

If Alternative Command mode is disabled (ACMEN bit set = 0):

1. Initialize the transmit PDC (memory pointers, transfer size - 1).
2. Configure Host mode (DADR, CKDIV, MREAD = 0, etc.).
3. Start the transfer by setting the PDC TXTEN bit.
4. Wait for the PDC ENDTX flag either by using the polling method or ENDTX interrupt.
5. Disable the PDC by setting the PDC TXTDIS bit.

© 2026 Microchip Technology Inc. and its subsidiaries

6. Wait for the FLEX_TWI_SR.TXRDY flag.
7. Set the STOP command in FLEX_TWI_CR.
8. Write the last character in FLEX_TWI_THR.
9. (Optional) Wait for the FLEX_TWI_SR.TXCOMP flag before disabling the peripheral clock if required.

If Alternative Command mode is enabled (ACMEN bit = 1):

1. Initialize the transmit PDC (memory pointers, transfer size).
2. Configure Host mode (DADR, CKDIV, etc.).
3. Start the transfer by setting the PDC TXTEN bit.
4. Wait for the PDC ENDTX flag by using either the polling method or the ENDTX interrupt.
5. Disable the PDC by setting the PDC TXTDIS bit.
6. (Optional) Wait for the FLEX_TWI_SR.TXCOMP flag before disabling the peripheral clock if required.

33.9.3.9.2. Data Receive with the PDC in Host Mode

The PDC transfer size must be defined with the buffer size minus 2. The two remaining characters must be managed without PDC to ensure that the exact number of bytes are received regardless of system bus latency conditions during the end of the buffer transfer period.

If Alternative Command mode is disabled (ACMEN bit set = 0):

1. Initialize the receive PDC (memory pointers, transfer size - 2).
2. Configure Host mode (DADR, CKDIV, MREAD = 1, etc.).
3. Set the PDC RXTEN bit.
4. (Host only) Write the START bit in FLEX_TWI_CR to start the transfer.
5. Wait for the PDC ENDRX flag either by using polling method or ENDRX interrupt.
6. Disable the PDC by setting the PDC RXTDIS bit.
7. Wait for the FLEX_TWI_SR.RXRDY flag.
8. Set the STOP command in FLEX_TWI_CR to end the transfer.
9. Read the penultimate character in FLEX_TWI_RHR.
10. Wait for the FLEX_TWI_SR.RXRDY flag.
11. Read the last character in FLEX_TWI_RHR.
12. (Optional) Wait for the FLEX_TWI_SR.TXCOMP flag before disabling the peripheral clock if required.

If Alternative Command mode is enabled (ACMEN bit = 1):

1. Initialize the transmit PDC (memory pointers, transfer size).
2. Configure Host mode (DADR, CKDIV, etc.).
3. Set the PDC RXTEN bit.
4. (Host only) Write the FLEX_TWI_CR.START bit to start the transfer.
5. Wait for the PDC ENDTX Flag by using either the polling method or the ENDTX interrupt.
6. Disable the PDC by setting the PDC TXTDIS bit.
7. (Optional) Wait for the FLEX_TWI_SR.TXCOMP flag before disabling the peripheral clock if required.

© 2026 Microchip Technology Inc. and its subsidiaries

33.9.3.10. SMBus Mode

SMBus mode is enabled when the FLEX_TWI_CR.SMBEN bit is written to one. SMBus mode operation is similar to I²C operation with the following exceptions:

1. Only 7-bit addressing can be used.
2. The SMBus standard describes a set of timeout values to ensure progress and throughput on the bus. These timeout values must be programmed into FLEX_TWI_SMBTR.
3. Transmissions can optionally include a CRC byte, called Packet Error Check (PEC).
4. A set of addresses has been reserved for protocol handling, such as alert response address (ARA) and host header (HH) address. Address matching on these addresses can be enabled by configuring FLEX_TWI_CR appropriately.

33.9.3.10.1. Packet Error Checking

Each SMBus transfer can optionally end with a CRC byte, called the PEC byte. Writing the FLEX_TWI_CR.PECEN bit to one enables automatic PEC handling in the current transfer. Transfers with and without PEC can freely be intermixed in the same system, since some clients may not support PEC. The PEC LFSR is always updated on every bit transmitted or received, so that PEC handling on combined transfers will be correct.

In Host Transmitter mode, the host calculates a PEC value and transmits it to the client after all data bytes have been transmitted. Upon reception of this PEC byte, the client will compare it to the PEC value it has computed itself. If the values match, the data was received correctly, and the client will return an ACK to the host. If the PEC values differ, data was corrupted, and the client will return a NACK value. Some clients may not be able to check the received PEC in time to return a NACK if an error occurred. In this case, the client should always return an ACK after the PEC byte, and some other mechanism must be implemented to verify that the transmission was received correctly.

In Host Receiver mode, the client calculates a PEC value and transmits it to the host after all data bytes have been transmitted. Upon reception of this PEC byte, the host will compare it to the PEC value it has computed itself. If the values match, the data was received correctly. If the PEC values differ, data was corrupted, and the FLEX_TWI_SR.PECERR bit is set. In Host Receiver mode, the PEC byte is always followed by a NACK transmitted by the host, since it is the last byte in the transfer.

In combined transfers, the PECRQ bit should only be set in the last of the combined transfers. If Alternative Command mode is enabled, only the NPEC bit should be set.

Consider the following transfer:

S, ADR+W, COMMAND_BYTE, ACK, SR, ADR+R, DATA_BYTE, ACK, PEC_BYTE, NACK, P

See Read/Write Flowcharts for detailed flowcharts.

33.9.3.10.2. Timeouts

The FLEX_TWI_SMBTR.TLOWS/TLOWM fields configure the SMBus timeout values. If a timeout occurs, the host transmits a STOP condition and leaves the bus. Furthermore, the FLEX_TWI_SR.TOUT bit is set.

33.9.3.11. SMBus Quick Command (Host Mode Only)

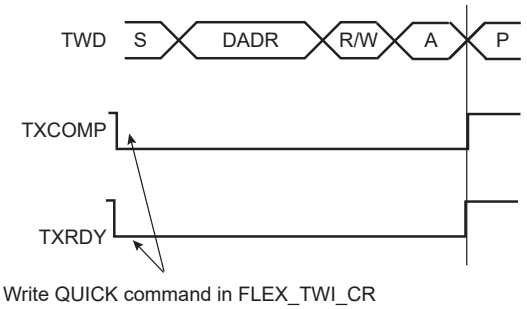
The TWI interface can perform a quick command:

1. Configure Host mode (DADR, CKDIV, etc).
2. Write the FLEX_TWI_MMR.MREAD bit at the value of the one-bit command to be sent.
3. Start the transfer by setting the FLEX_TWI_CR.QUICK bit.

Note: If an alternative command is used (ACMEN bit = 1), the DATAL field must be cleared.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-98. SMBus Quick Command



33.9.3.12. Alternative Command

Another way to configure the transfer is to enable the Alternative Command mode with the ACMEN bit of the TWI Control Register.

In this mode, the transfer is configured through the TWI Alternative Command Register. It is possible to define a simple read or write transfer or a combined transfer with a repeated start.

In order to set a simple transfer, the DATAL field and the DIR field of the TWI Alternative Command Register must be filled accordingly and the NDATAL field must be cleared. To begin the transfer, either set the START bit in the TWI Control Register in case of a read transfer, or write the TWI Transmit Holding Register in case of a write transfer.

For a combined transfer linked by a repeated start, the NDATAL field must be filled with the length of the second transfer and NDIR with the corresponding direction.

The PEC and NPEC bits are used to set a PEC field. In the case of a single transfer with PEC, the PEC bit must be set. In the case of a combined transfer, the NPEC bit must be set.

Note: If the Alternative Command mode is used, the FLEX_TWI_MMR.IADRSZ field must be set to 0.

See Read/Write Flowcharts for detailed flowcharts.

33.9.3.13. Handling Errors in Alternative Command

In case of NACK generated by a client device or SMBus timeout error, the TWI stops immediately the frame, but the PDC transfer may still be active. To prevent a new frame to be restarted with the remaining PDC data (transmit), the TWI prevents any start of frame until the FLEX_TWI_SR.LOCK flag is cleared.

The FLEX_TWI_SR.LOCK bit indicates the state of the TWI (locked or not locked).

When the TWI is locked, no transfer can begin until the LOCK is cleared using the FLEX_TWI_CR.LOCKCLR bit and until the error flags are cleared reading FLEX_TWI_SR.

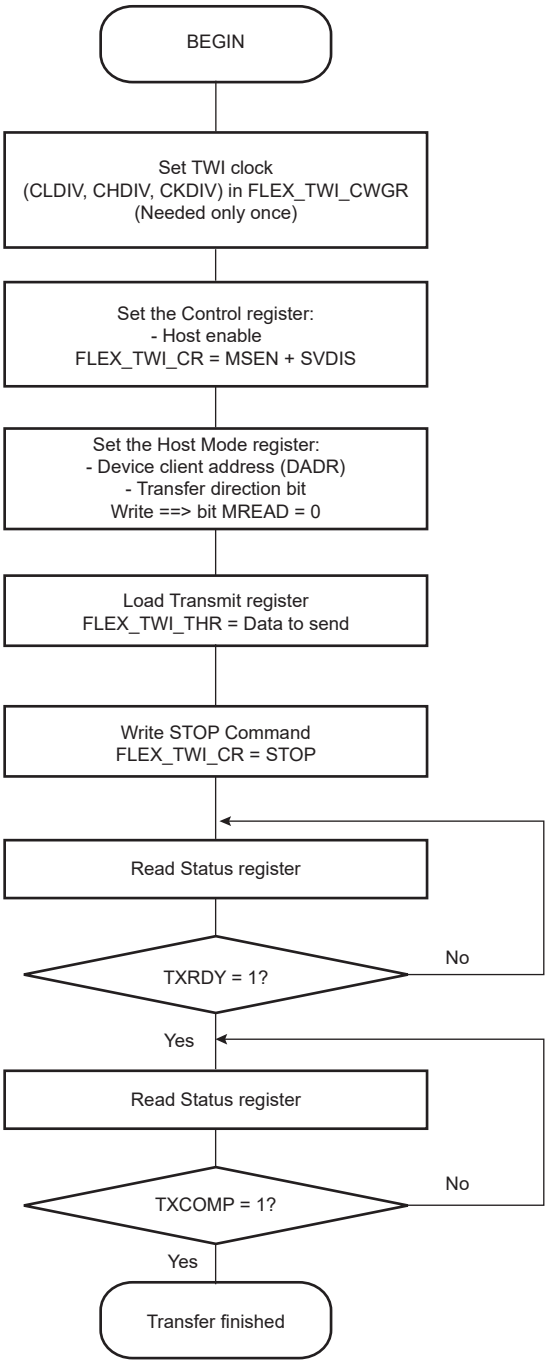
In case of error, FLEX_TWI_THR may have been loaded with a new data. The FLEX_TWI_CR.THRCLR bit can be used to flush FLEX_TWI_THR. If the THRCLR bit is set, the TXRDY and TXCOMP flags are set.

33.9.3.14. Read/Write Flowcharts

The flowcharts shown in this section provide examples for read and write operations. A polling or interrupt method can be used to check the status bits. The interrupt method requires that the Interrupt Enable register (FLEX_TWI_IER) be configured first.

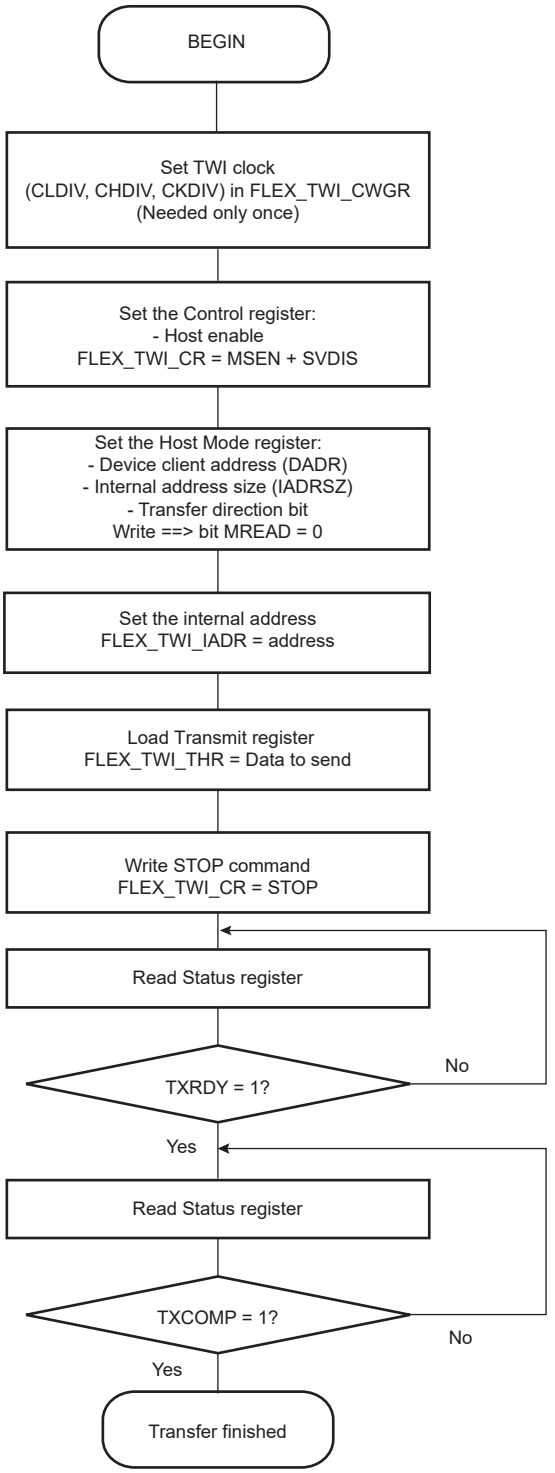
© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-99. TWI Write Operation with Single Data Byte without Internal Address



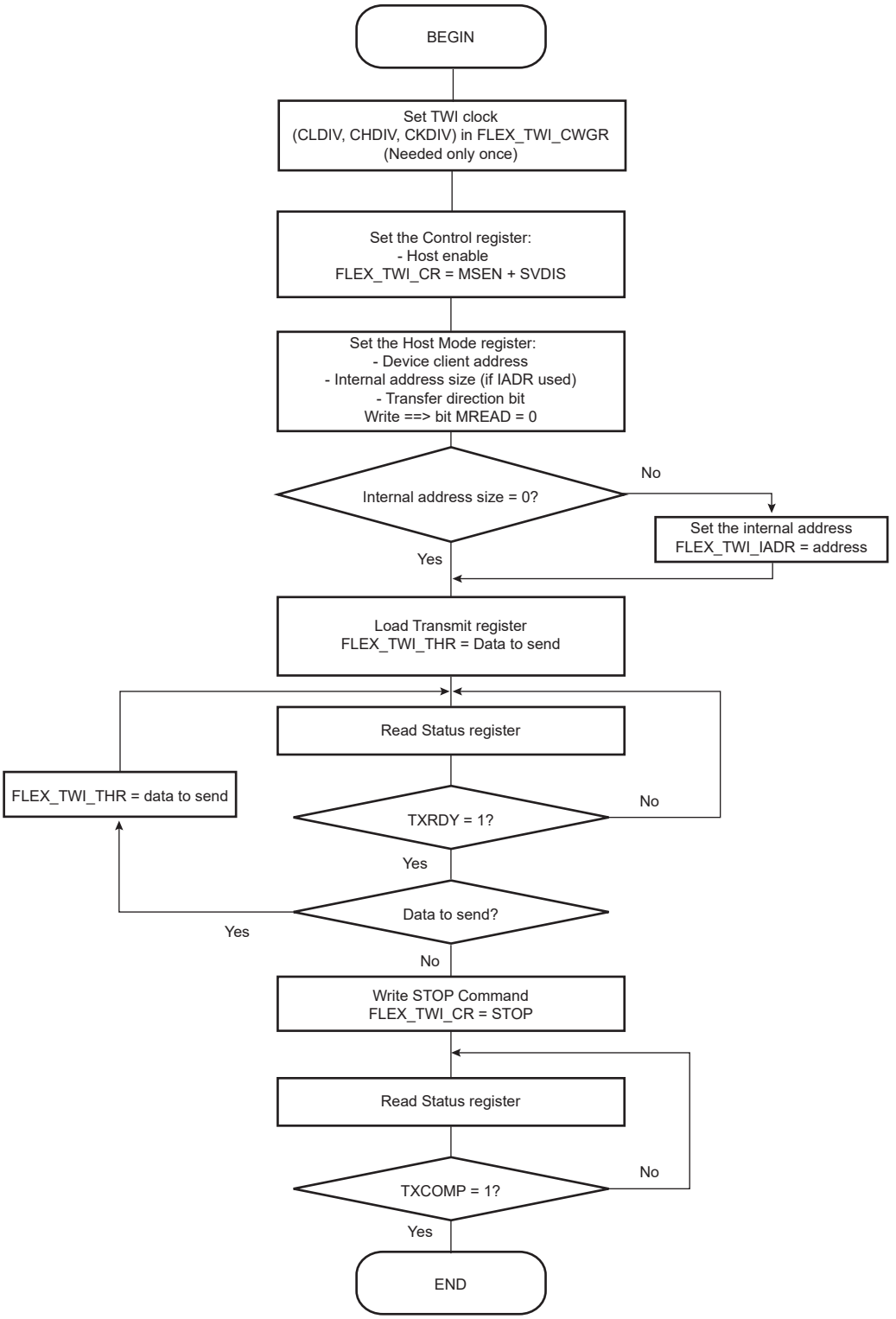
© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-100. TWI Write Operation with Single Data Byte and Internal Address



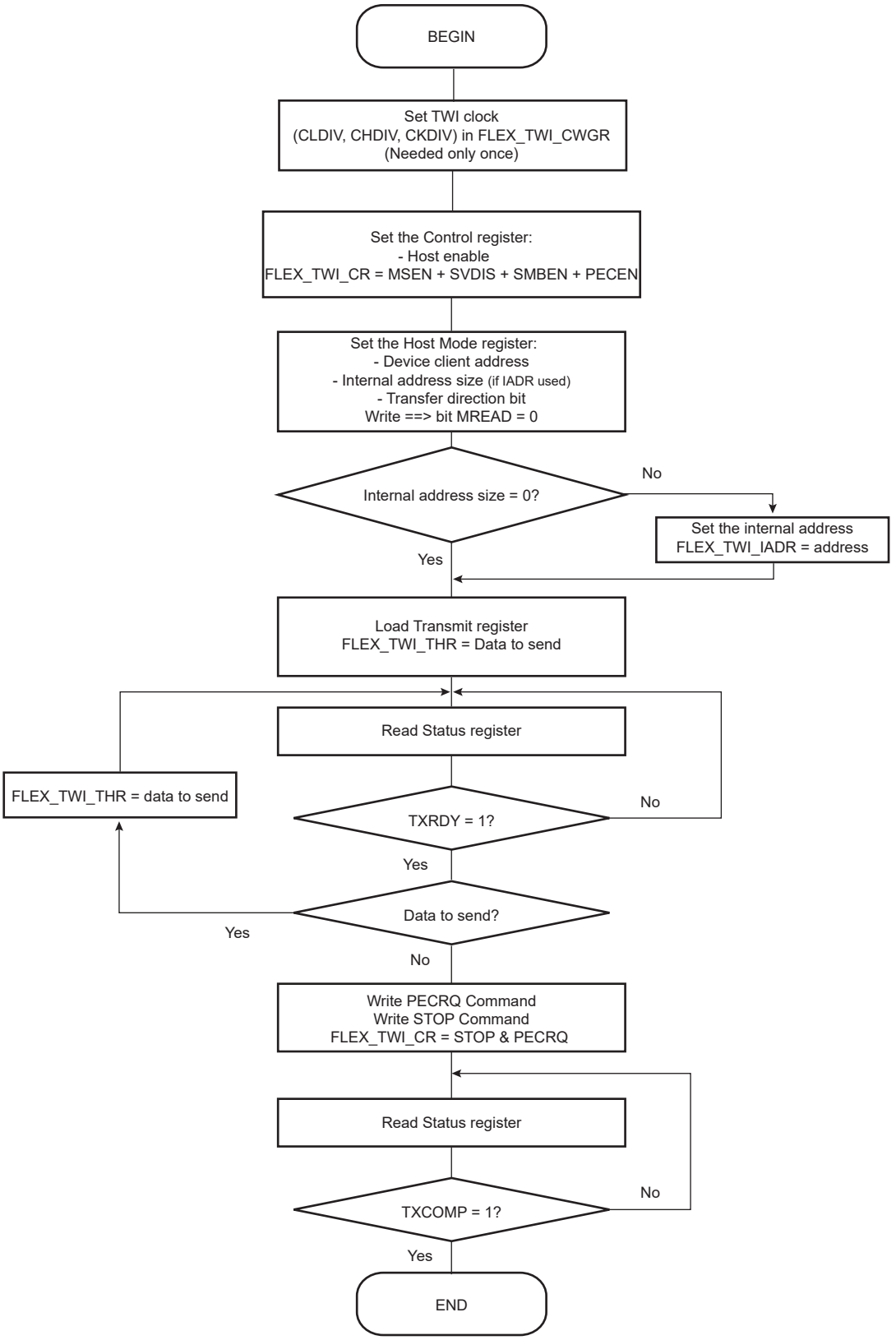
© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-101. TWI Write Operation with Multiple Data Bytes with or without Internal Address



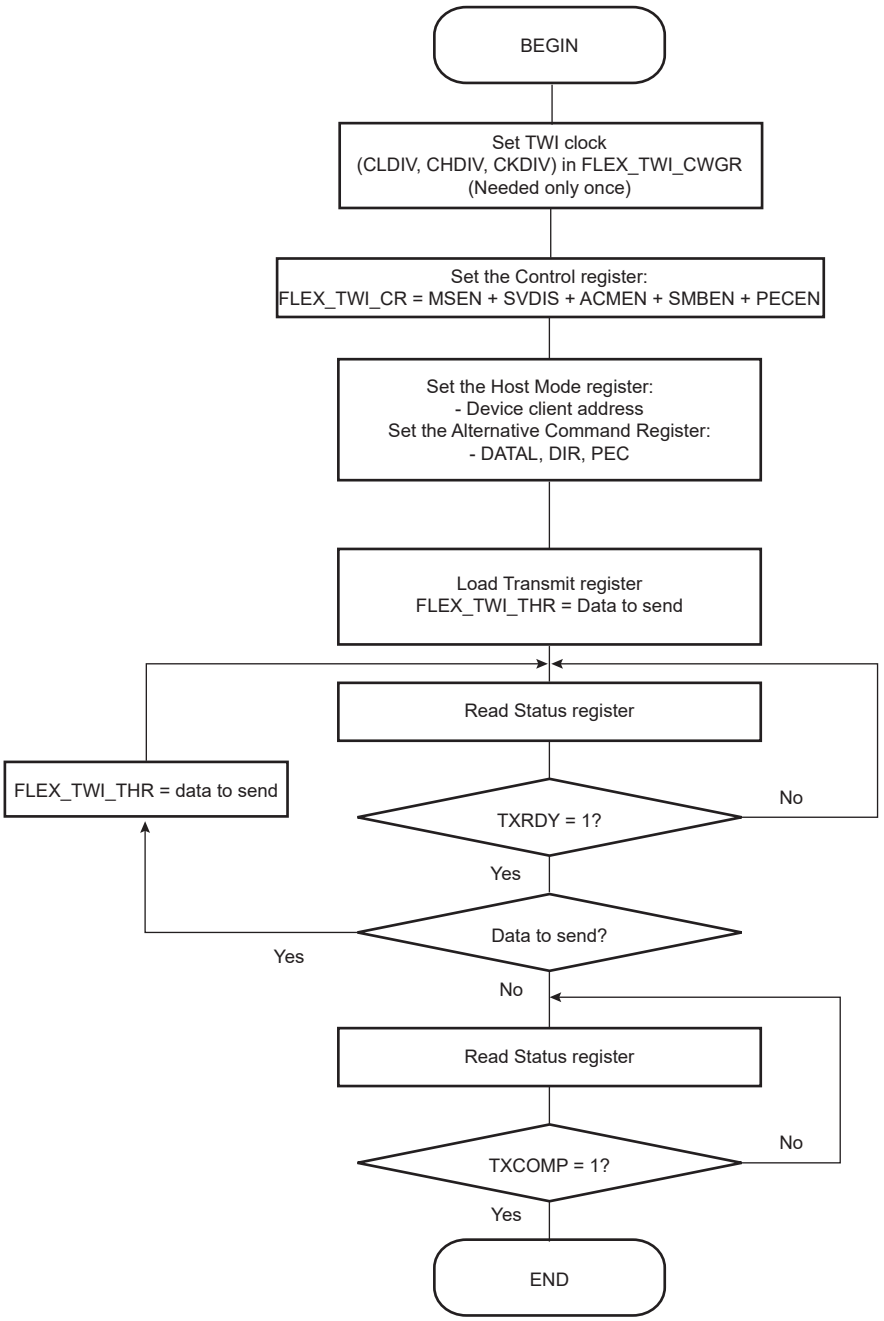
© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-102. SMBus Write Operation with Multiple Data Bytes with or without Internal Address and PEC Sending



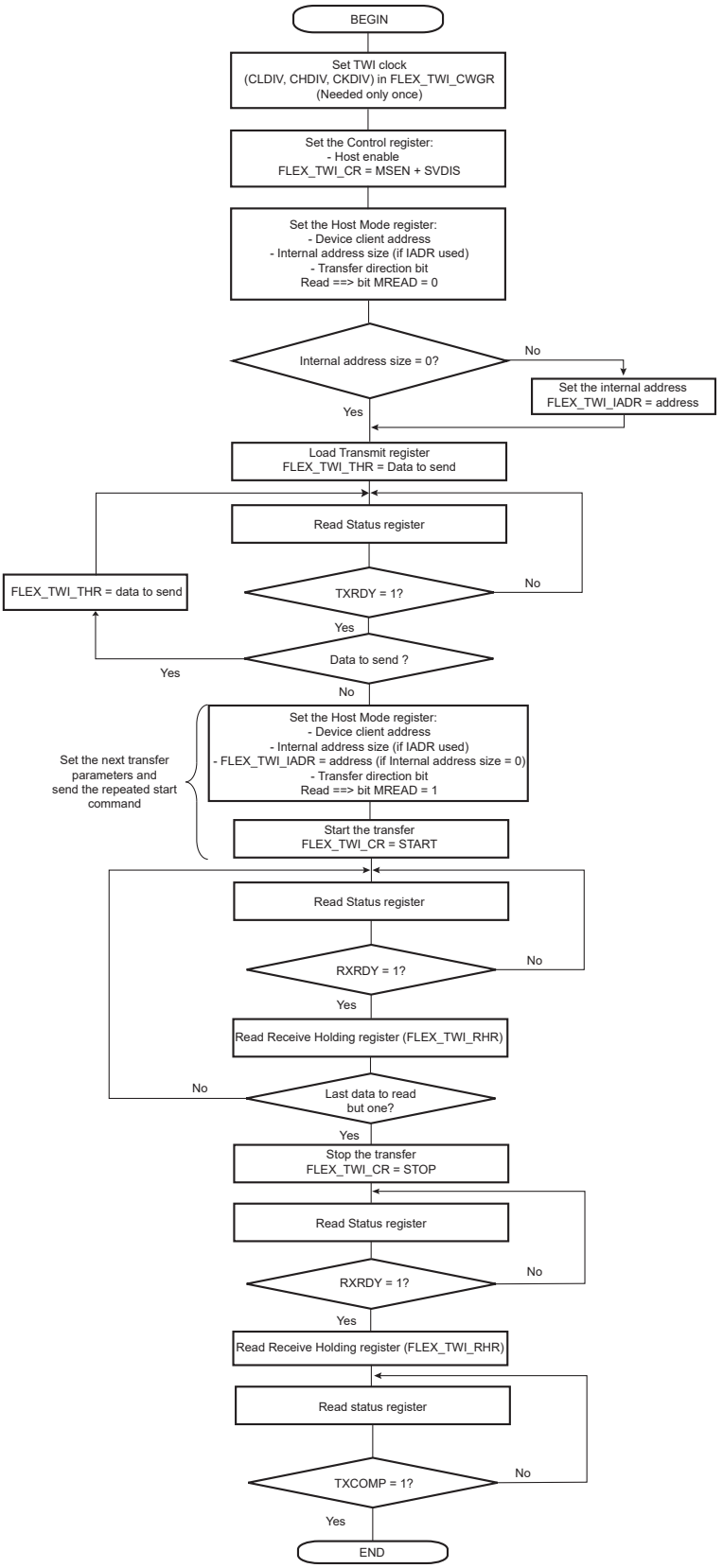
© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-103. SMBus Write Operation with Multiple Data Bytes with PEC and Alternative Command Mode



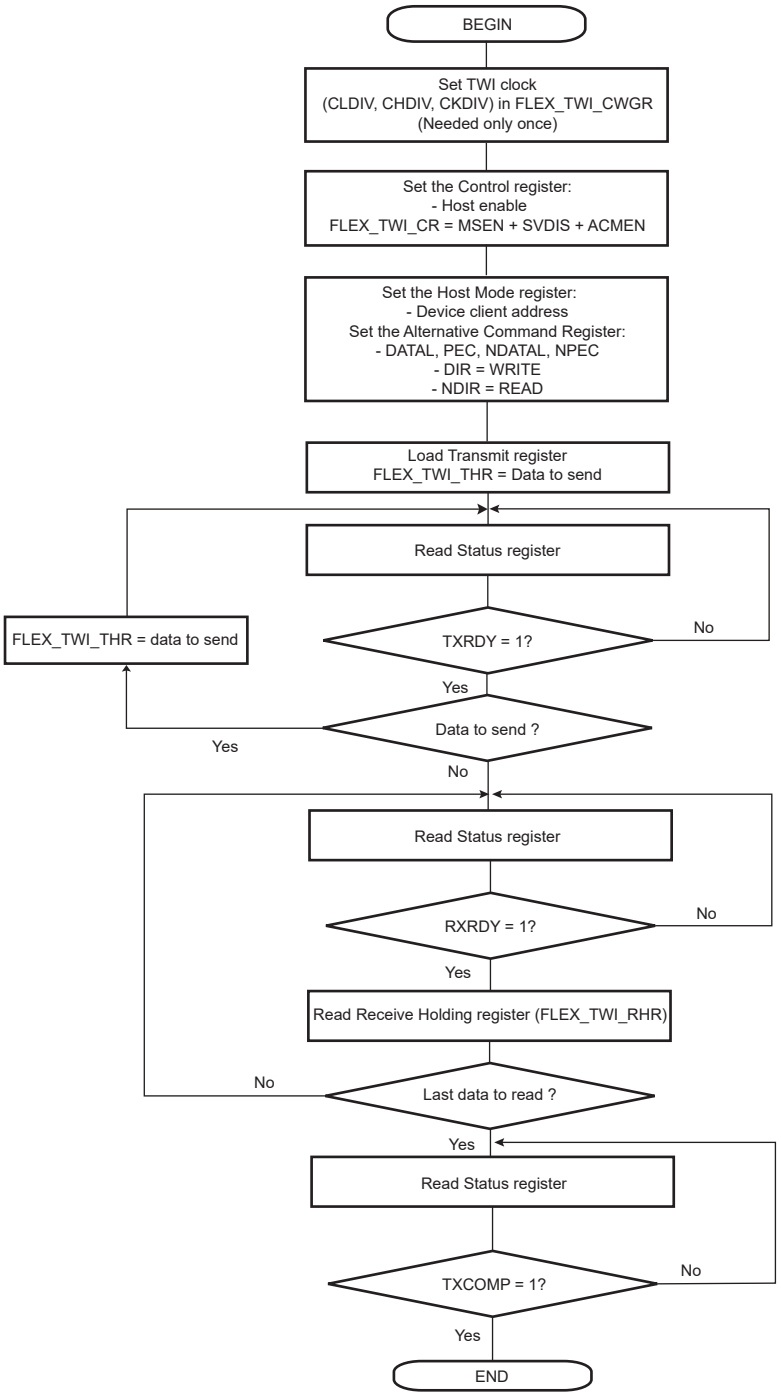
© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-104. TWI Write Operation with Multiple Data Bytes and Read Operation with Multiple Data Bytes (Sr)



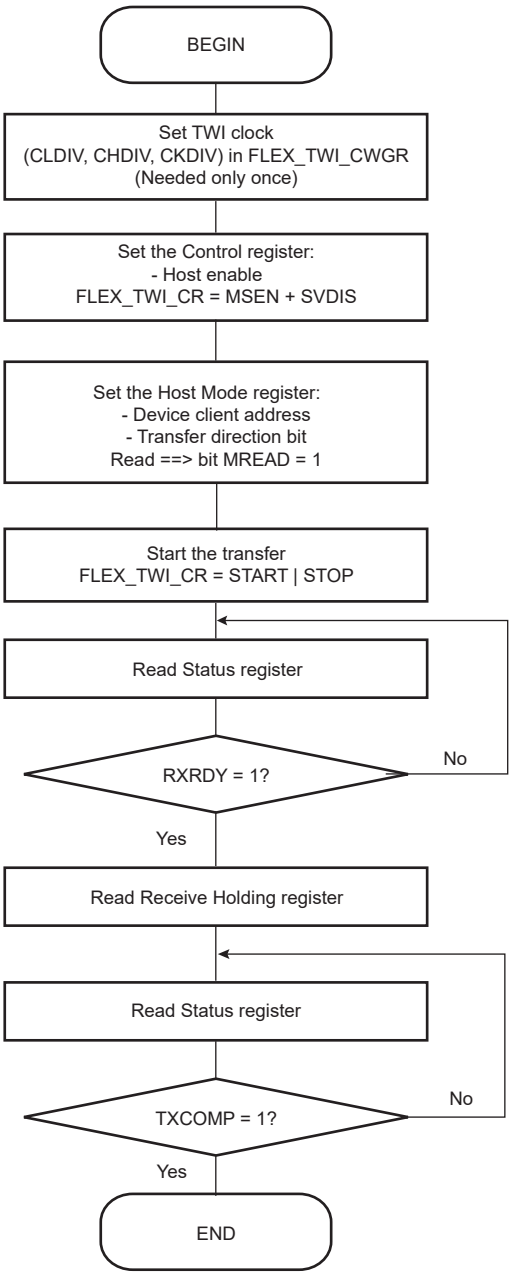
© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-105. TWI Write Operation with Multiple Data Bytes + Read Operation and Alternative Command Mode + PEC



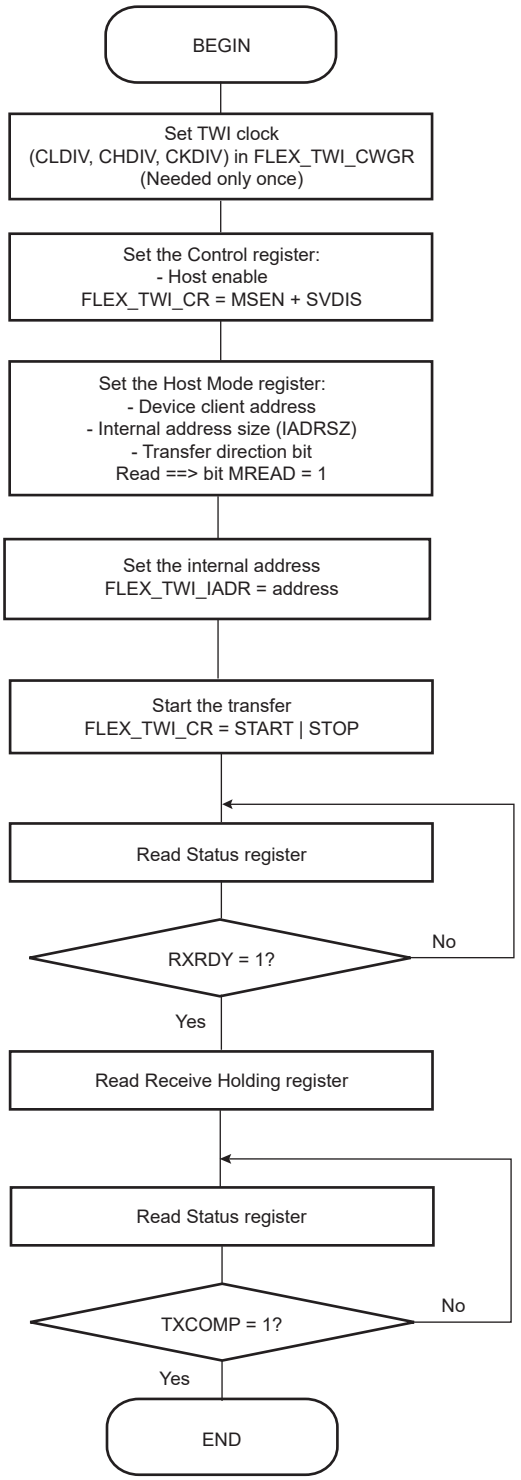
© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-106. TWI Read Operation with Single Data Byte without Internal Address



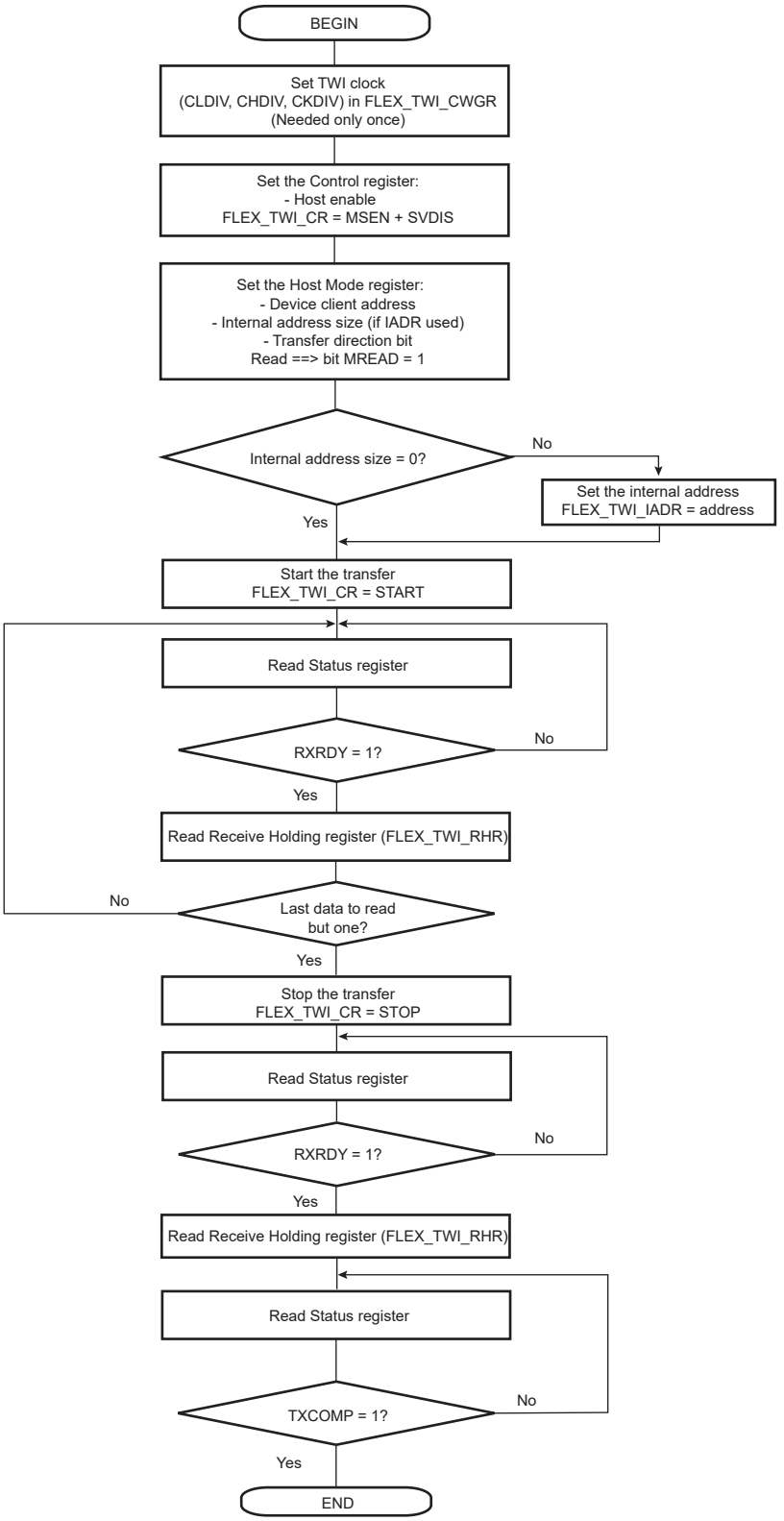
© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-107. TWI Read Operation with Single Data Byte and Internal Address



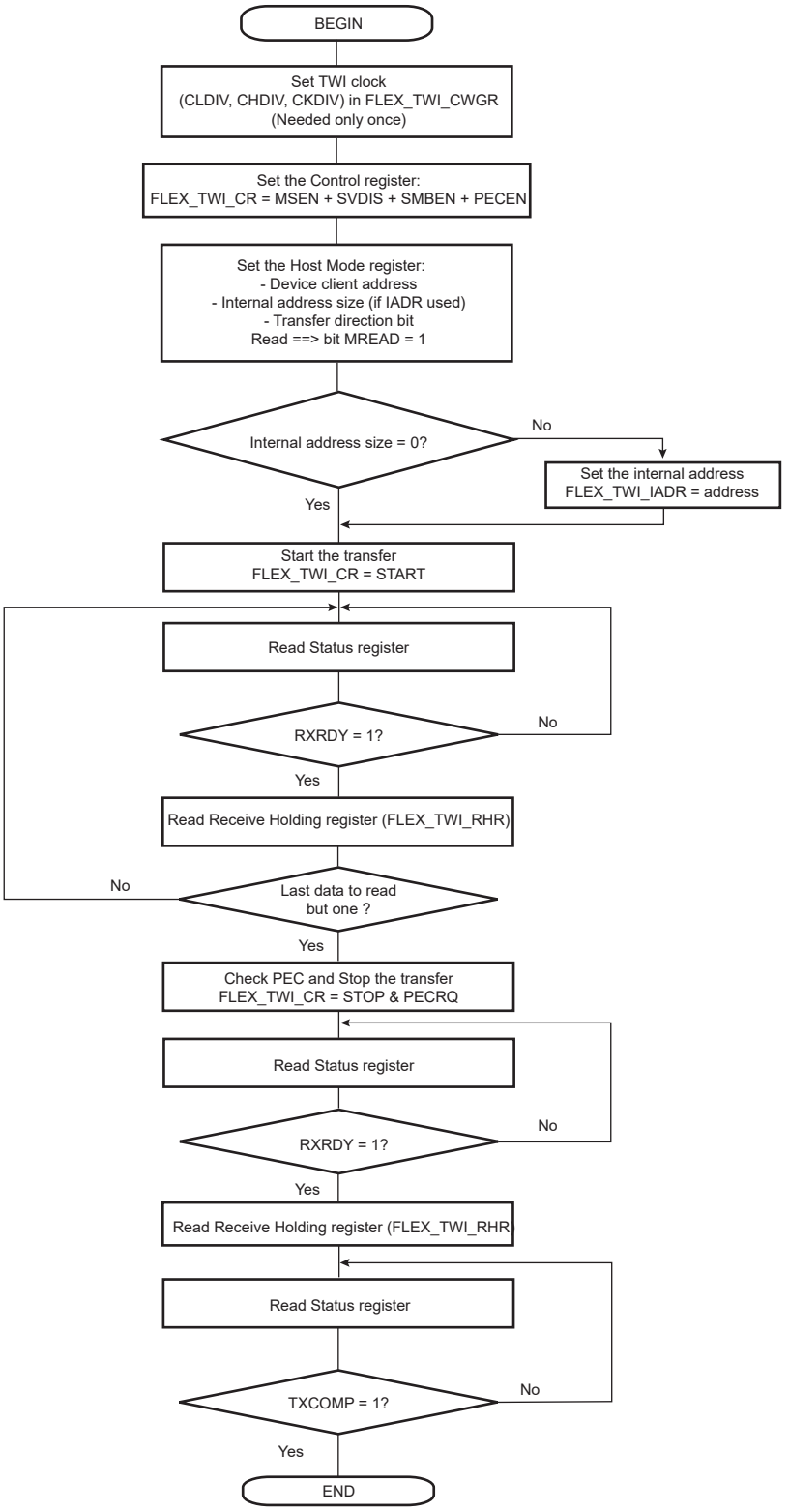
© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-108. TWI Read Operation with Multiple Data Bytes with or without Internal Address



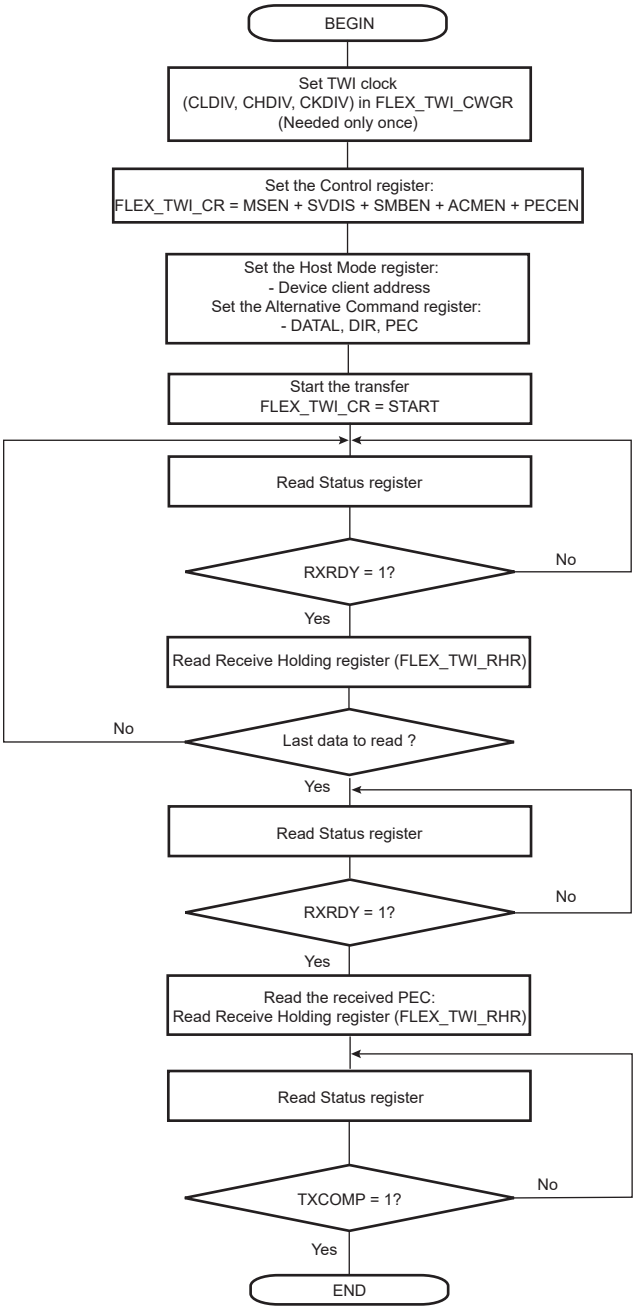
© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-109. TWI Read Operation with Multiple Data Bytes with or without Internal Address with PEC



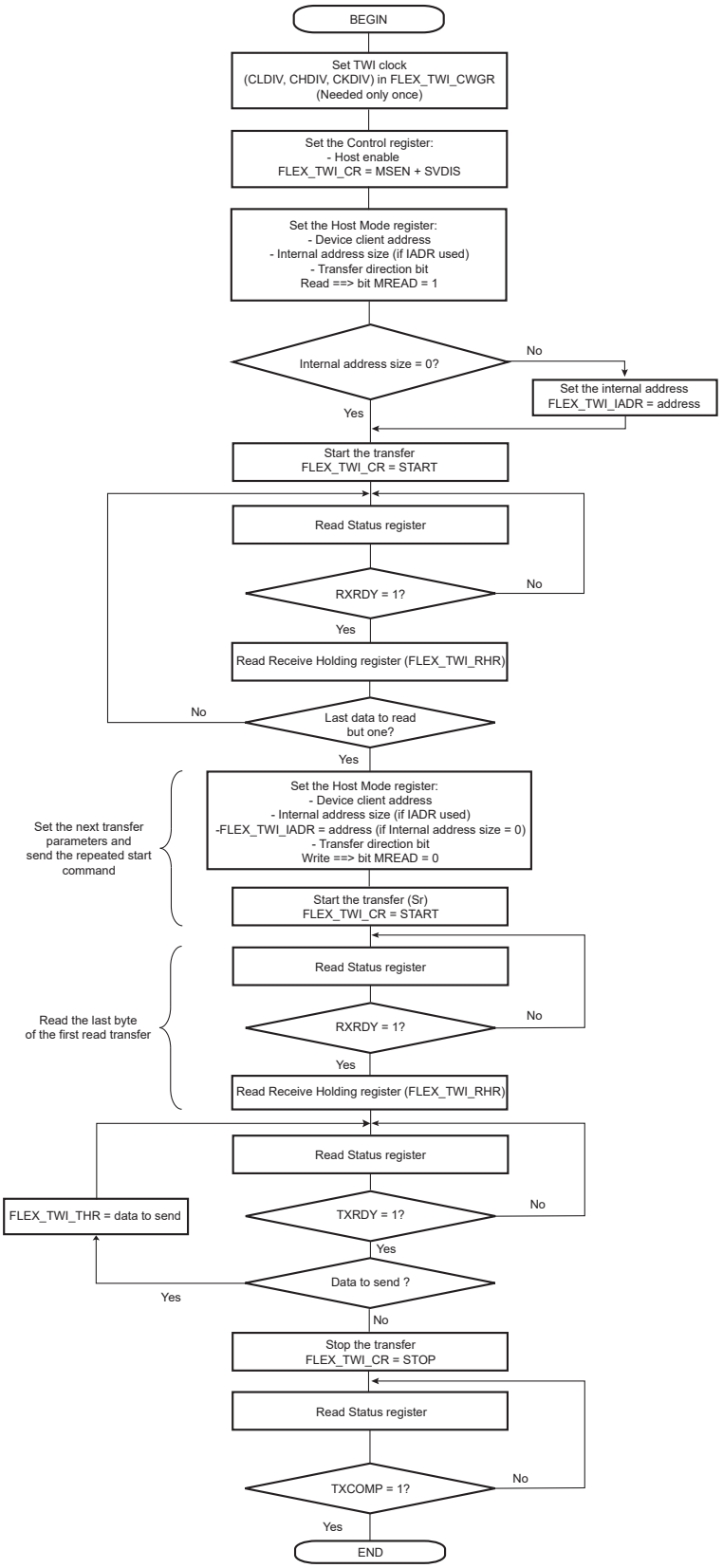
© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-110. TWI Read Operation with Multiple Data Bytes with Alternative Command Mode with PEC



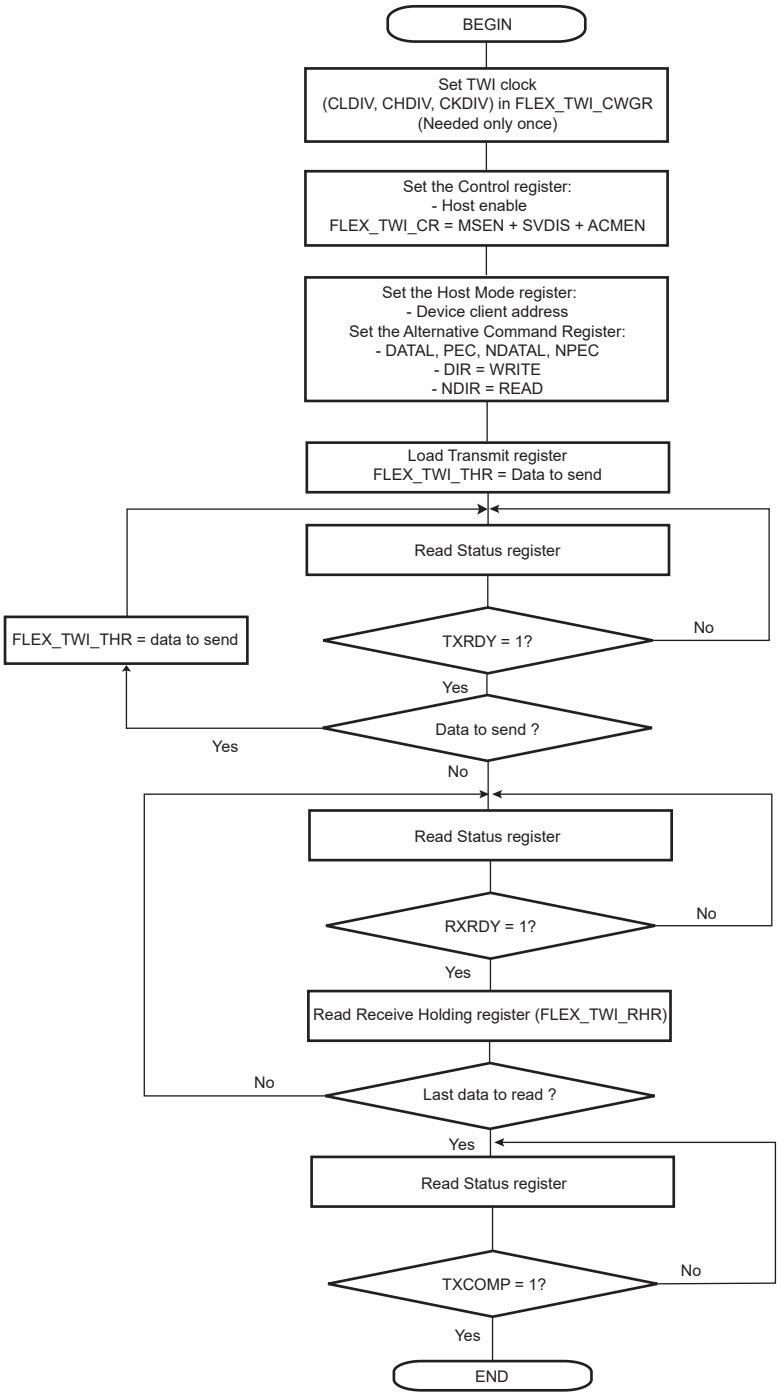
© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-111. TWI Read Operation with Multiple Data Bytes + Write Operation with Multiple Data Bytes (Sr)



© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-112. TWI Read Operation with Multiple Data Bytes + Write with Alternative Command Mode with PEC



33.9.4. Multi-Host Mode

33.9.4.1. Definition

In Multi-Host mode, more than one host may handle the bus at the same time without data corruption by using arbitration.

Arbitration starts as soon as two or more hosts place information on the bus at the same time, and stops (arbitration is lost) for the host that intends to send a logical one while the other host sends a logical zero.

© 2026 Microchip Technology Inc. and its subsidiaries

As soon as arbitration is lost by a host, it stops sending data and listens to the bus in order to detect a STOP. When the STOP is detected, the host that has lost arbitration may put its data on the bus by respecting arbitration.

Arbitration is illustrated in figure "Arbitration Cases" below.

33.9.4.2. Different Multi-Host Modes

Two Multi-Host modes are available:

- TWI as Host Only—TWI is considered as a host only and will never be addressed.
- TWI as Host or Client—TWI may be either a host or a client and may be addressed.

Note: Arbitration in supported in both Multi-Host modes.

33.9.4.2.1. TWI as Host Only

In this mode, the TWI is considered as a host only (MSEN is always at one) and must be driven like a host with the ARBLST (ARBitration Lost) flag in addition.

If arbitration is lost (ARBLST = 1), the user must reinitiate the data transfer.

If the user starts a transfer (ex.: DADR + START + W + Write in THR) and if the bus is busy, the TWI automatically waits for a STOP condition on the bus to initiate the transfer (see figure "User Sends Data While the Bus is Busy" below).

Note: The state of the bus (busy or free) is not indicated in the user interface.

33.9.4.2.2. TWI as Host or Client

The automatic reversal from host to client is not supported in case of a lost arbitration.

Then, in the case where TWI may be either a host or a client, the user must manage the pseudo Multi-Host mode described in the steps below:

1. Program the TWI in Client mode (SADR + MSDIS + SVEN) and perform a client access (if TWI is addressed).
2. If the TWI has to be set to Host mode, wait until TXCOMP flag is at 1.
3. Program Host mode (DADR + SVDIS + MSEN) and start the transfer (ex: START + Write in THR).
4. As soon as Host mode is enabled, the TWI scans the bus in order to detect if it is busy or free. When the bus is considered as free, the TWI initiates the transfer.
5. As soon as the transfer is initiated and until a STOP condition is sent, the arbitration becomes relevant and the user must monitor the ARBLST flag.
6. If the arbitration is lost (ARBLST is = 1), the user must program the TWI in Client mode in case the host that won the arbitration needs to access the TWI.
7. If the TWI has to be set to Client mode, wait until TXCOMP flag is at 1 and then program Client mode.
   Note: In case the arbitration is lost and the TWI is addressed, the TWI will not acknowledge even if it is programmed in Client mode as soon as ARBLST = 1. Then the host must repeat SADR.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-113. User Sends Data While the Bus is Busy

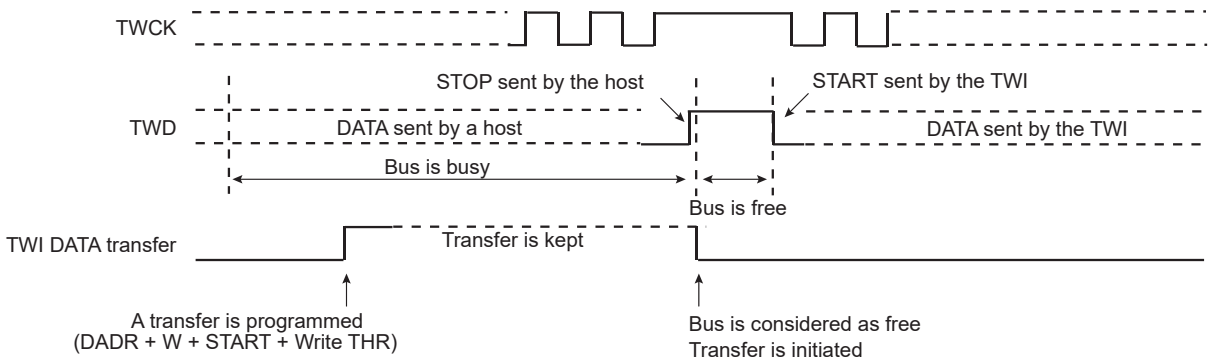
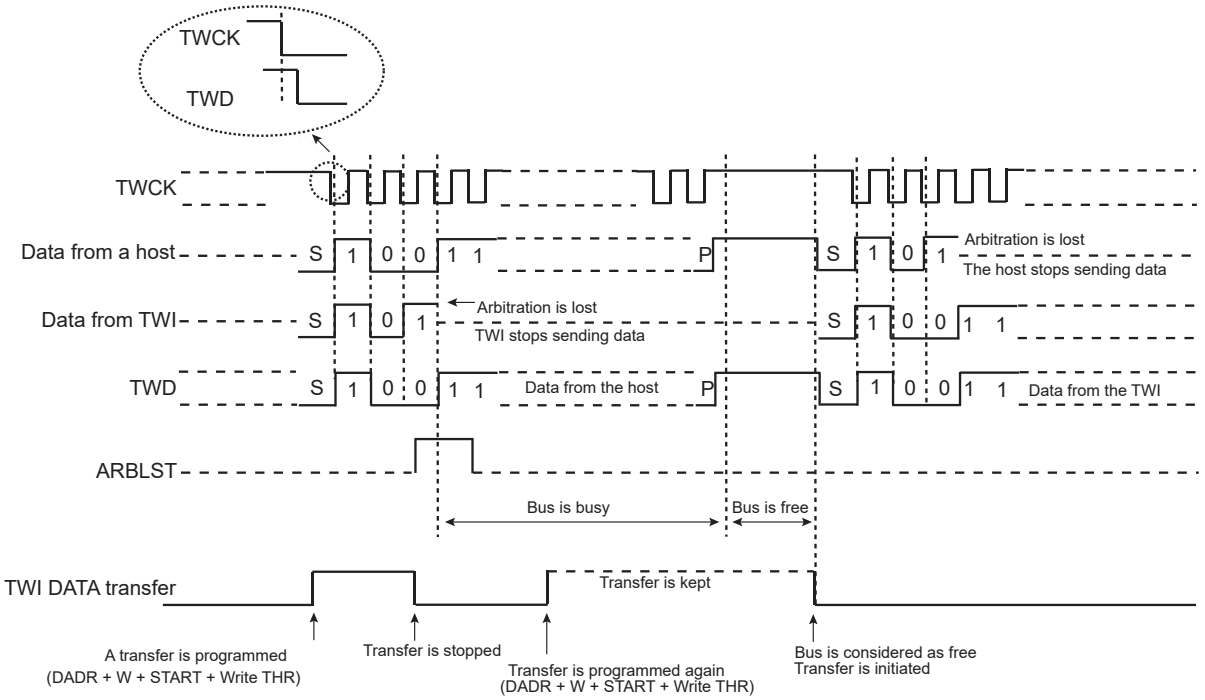


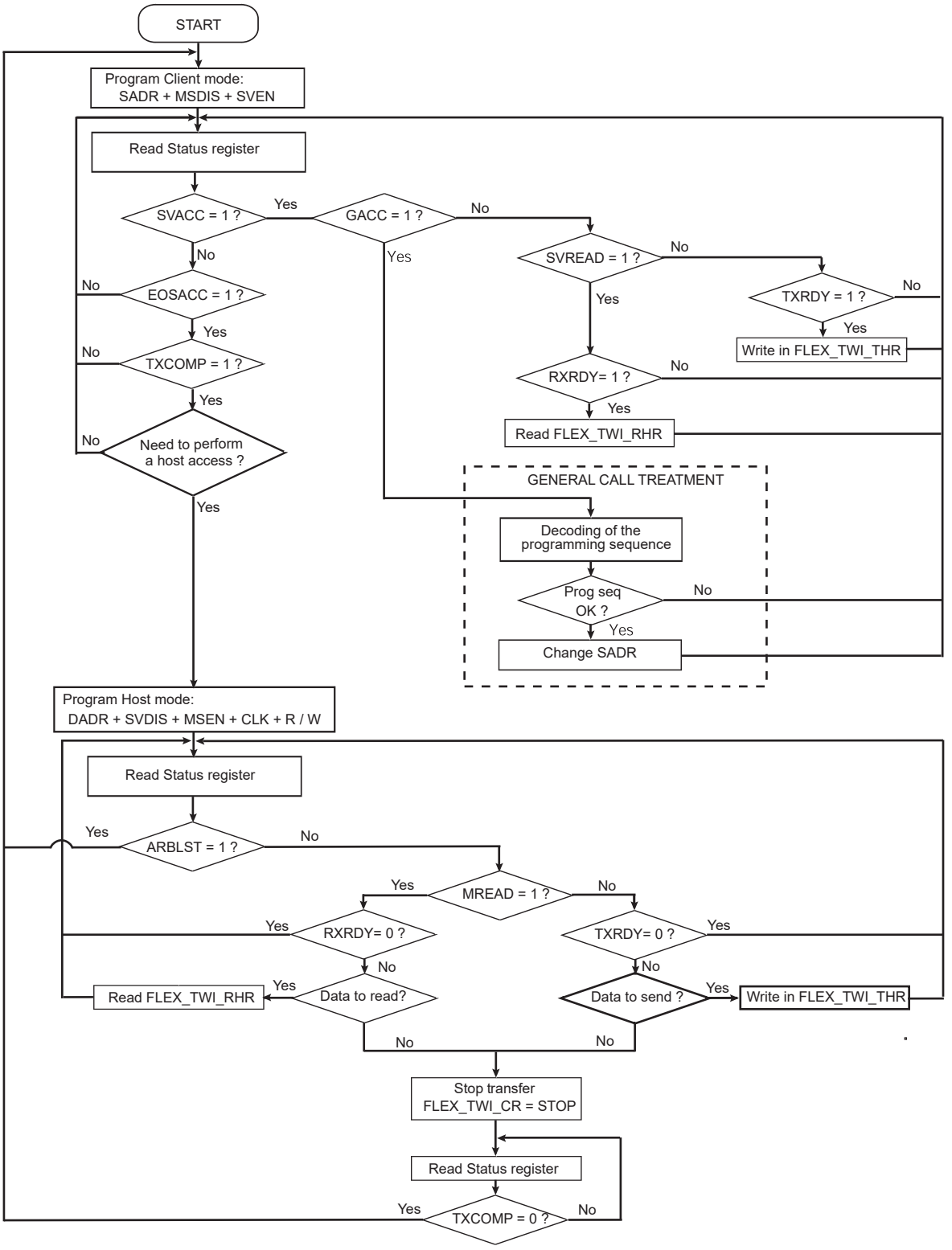
Figure 33-114. Arbitration Cases



The flowchart shown in the following figure gives an example of read and write operations in Multi-Host mode.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-115. Multi-Host Mode



© 2026 Microchip Technology Inc. and its subsidiaries

33.9.5. Client Mode

33.9.5.1. Definition

Client mode is defined as a mode where the device receives the clock and the address from another device called the host.

In this mode, the device never initiates and never completes the transmission (START, REPEATED_START and STOP conditions are always provided by the host).

33.9.5.2. Programming Client Mode

The following fields must be programmed before entering Client mode:

1. FLEX_TWI_SMR.SADR: The client device address is used in order to be accessed by host devices in Read or Write mode.
2. (Optional) FLEX_TWI_SMR.MASK can be set to mask some SADR address bits and thus allow multiple address matching.
3. FLEX_TWI_CR.MSDIS: Disables Host mode.
4. FLEX_TWI_CR.SVEN: Enables Client mode.

As the device receives the clock, values written in FLEX_TWI_CWGR are not processed.

33.9.5.3. Receiving Data

After a START or repeated START condition is detected, and if the address sent by the host matches the client address programmed in the SADR (Client Address) field, the SVACC (Client Access) flag is set and SVREAD (Client Read) indicates the direction of the transfer.

SVACC remains high until a STOP condition or a repeated START is detected. When such a condition is detected, EOSACC (End Of Client Access) flag is set.

33.9.5.3.1. Read Sequence

In the case of a read sequence (SVREAD is high), the TWI transfers data written in FLEX_TWI_THR (TWI Transmit Holding register) until a STOP condition or a REPEATED_START + an address different from SADR is detected. Note that at the end of the read sequence TXCOMP (Transmission Complete) flag is set and SVACC is reset.

As soon as data is written in FLEX_TWI_THR, the TXRDY (Transmit Holding Register Ready) flag is reset, and it is set when the internal shifter is empty and the sent data acknowledged or not. If the data is not acknowledged, the NACK flag is set.

Note that a STOP or a repeated START always follows a NACK.

See figure "Read Access Ordered by a Host" below.

Note: To clear the TXRDY flag in Client mode, write the FLEX_TWI_CR.SVDIS bit to 1, then write the FLEX_TWI_CR.SVEN bit to 1.

33.9.5.3.2. Write Sequence

In the case of a write sequence (SVREAD is low), the RXRDY (Receive Holding Register Ready) flag is set as soon as a character has been received in FLEX_TWI_RHR (TWI Receive Holding register). RXRDY is reset when reading FLEX_TWI_RHR.

TWI continues receiving data until a STOP condition or a REPEATED_START + an address different from SADR is detected. Note that at the end of the write sequence TXCOMP flag is set and SVACC reset.

See figure "Write Access Ordered by a Host" below.

33.9.5.3.3. Clock Stretching Sequence

If FLEX_TWI_THR or FLEX_TWI_RHR is not written/read in time, the TWI performs a clock stretching.

Clock stretching information is given by the SCLWS (Clock Wait State) bit.

© 2026 Microchip Technology Inc. and its subsidiaries

See figures "Clock Stretching in Read Mode" and "Clock Stretching in Write Mode" below.

Note: Clock stretching can be disabled by configuring the FLEX_TWI_SMR.SCLWSDIS bit. In that case, UNRE and OVRE flags will indicate underrun (when FLEX_TWI_THR is not filled on time) or overrun (when FLEX_TWI_RHR is not read on time).

33.9.5.3.4. General Call

In the case where a GENERAL CALL is performed, the GACC (General Call Access) flag is set.

After GACC is set, it is up to the user to interpret the meaning of the GENERAL CALL and to decode the new address programming sequence.

See figure "Host Performs a General Call" below.

33.9.5.4. Data Transfer

33.9.5.4.1. Read Operation

Read mode is defined as a data requirement from the host.

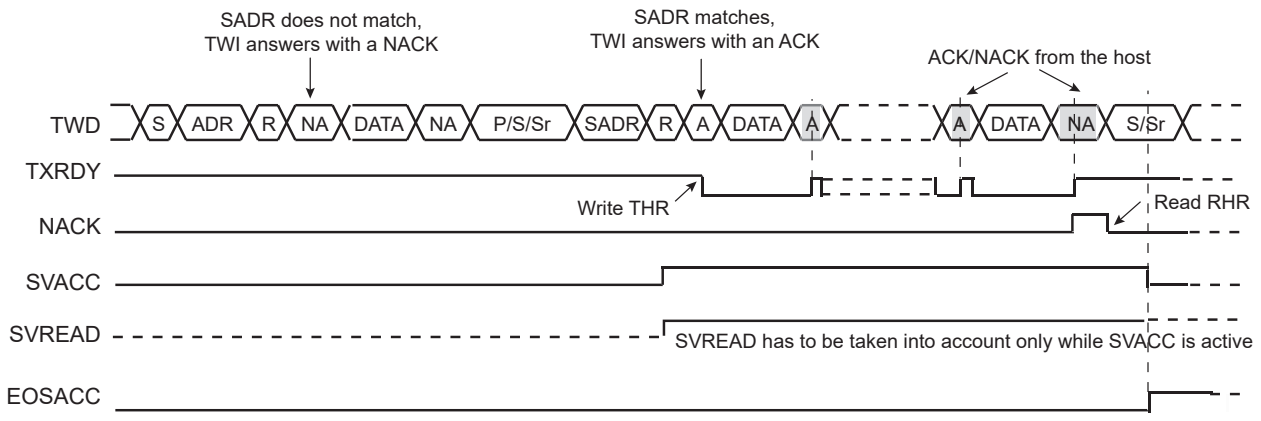
After a START or a REPEATED START condition is detected, the decoding of the address starts. If the client address (SADR) is decoded, SVACC is set and SVREAD indicates the direction of the transfer.

Until a STOP or REPEATED START condition is detected, TWI continues sending data loaded in FLEX_TWI_THR.

If a STOP condition or a REPEATED START + an address different from SADR is detected, SVACC is reset.

The following figure describes the read operation.

Figure 33-116. Read Access Ordered by a Host



Notes:

1. When SVACC is low, the state of SVREAD becomes irrelevant.
2. TXRDY is reset when data has been transmitted from FLEX_TWI_THR to the internal shifter and set when this data has been acknowledged or non acknowledged.

33.9.5.4.2. Write Operation

The Write mode is defined as a data transmission from the host.

After a START or a REPEATED START, the decoding of the address starts. If the client address is decoded, SVACC is set and SVREAD indicates the direction of the transfer (SVREAD is low in this case).

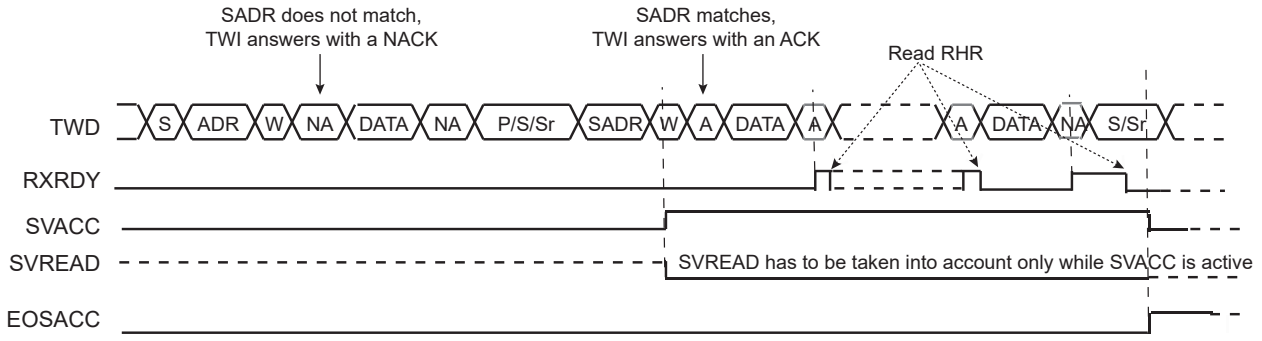
© 2026 Microchip Technology Inc. and its subsidiaries

Until a STOP or REPEATED START condition is detected, TWI stores the received data in FLEX_TWI_RHR.

If a STOP condition or a REPEATED START + an address different from SADR is detected, SVACC is reset.

The following figure describes the write operation.

Figure 33-117. Write Access Ordered by a Host



Notes:

1. When SVACC is low, the state of SVREAD becomes irrelevant.
2. RXRDY is set when data has been transmitted from the internal shifter to FLEX_TWI_RHR, and reset when this data is read.

33.9.5.4.3. General Call

The general call is performed in order to change the address of the client.

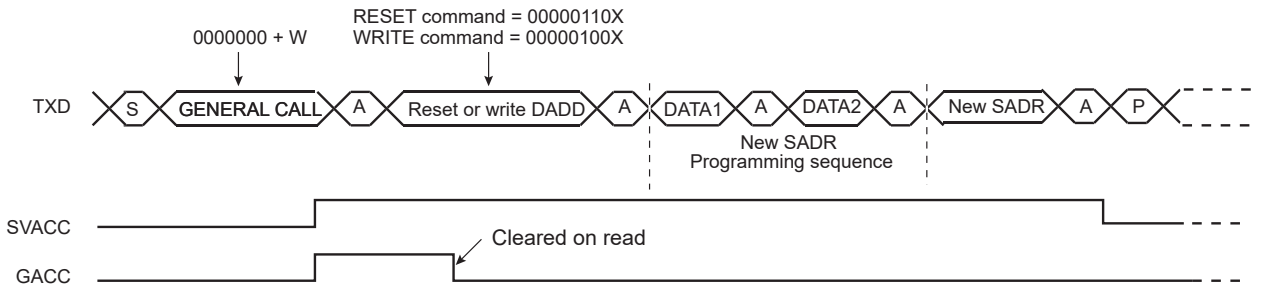
If a GENERAL CALL is detected, GACC is set.

After the detection of general call, it is up to the user to decode the commands which follow.

In case of a WRITE command, the user has to decode the programming sequence and program a new SADR if the programming sequence matches.

The following figure describes the general call access.

Figure 33-118. Host Performs a General Call



Note: This method enables to create a user-specific programming sequence by choosing the number of programming bytes. The programming sequence has to be provided to the host.

© 2026 Microchip Technology Inc. and its subsidiaries

33.9.5.4.4. Clock Stretching

In both Read and Write modes, it may happen that the FLEX_TWI_THR/FLEX_TWI_RHR buffer is not filled/emptied before the transmission/reception of a new character. In this case, to avoid sending/receiving undesired data, a clock stretching mechanism is implemented.

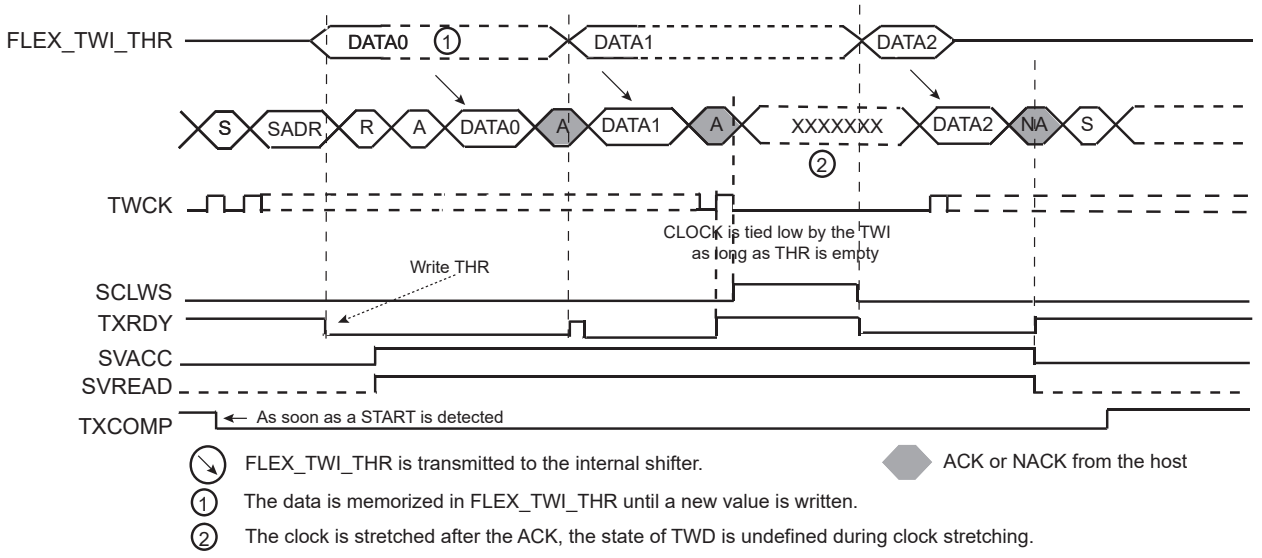
Note: Clock stretching can be disabled by setting the FLEX_TWI_SMR.SCLWSDIS bit. In that case, the UNRE and OVRE flags indicate an underrun (when FLEX_TWI_THR is not filled on time) or an overrun (when FLEX_TWI_RHR is not read on time).

— Clock Stretching in Read Mode

The clock is tied low if the internal shifter is empty and if a STOP or REPEATED START condition was not detected. It is tied low until the internal shifter is loaded.

The following figure describes clock stretching in Read mode.

Figure 33-119. Clock Stretching in Read Mode



Notes:

1. TXRDY is reset when data has been written in FLEX_TWI_THR to the internal shifter, and set when this data has been acknowledged or non acknowledged.
2. At the end of the read sequence, TXCOMP is set after a STOP or after a REPEATED_START + an address different from SADR.
3. SCLWS is automatically set when the clock stretching mechanism is started.

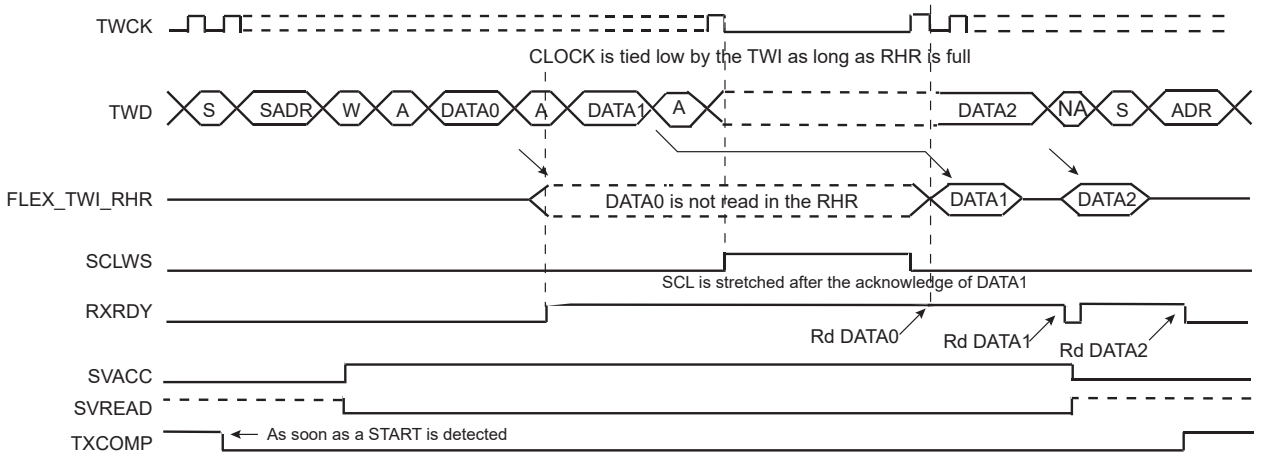
— Clock Stretching in Write Mode

The clock is tied low if the internal shifter and FLEX_TWI_RHR are full. If a STOP or REPEATED_START condition was not detected, it is tied low until FLEX_TWI_RHR is read.

The following figure describes the clock stretching in Write mode.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-120. Clock Stretching in Write Mode



Notes:

1. At the end of the read sequence, TXCOMP is set after a STOP or after a REPEATED_START + an address different from SADR.
2. SCLWS is automatically set when the clock stretching mechanism is started and automatically reset when the mechanism is finished.

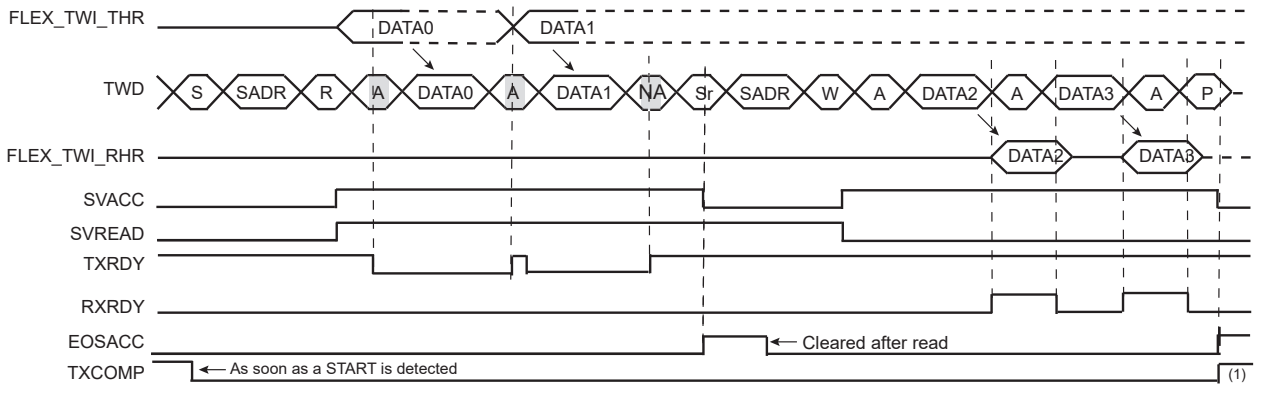
33.9.5.4.5. Reversal after a Repeated Start

— Reversal of Read to Write

The host initiates the communication by a read command and finishes it by a write command.

The following figure describes the repeated start and the reversal from Read mode to Write mode.

Figure 33-121. Repeated Start and Reversal from Read Mode to Write Mode



Note:
1. TXCOMP is only set at the end of the transmission because after the repeated start, SADR is detected again.

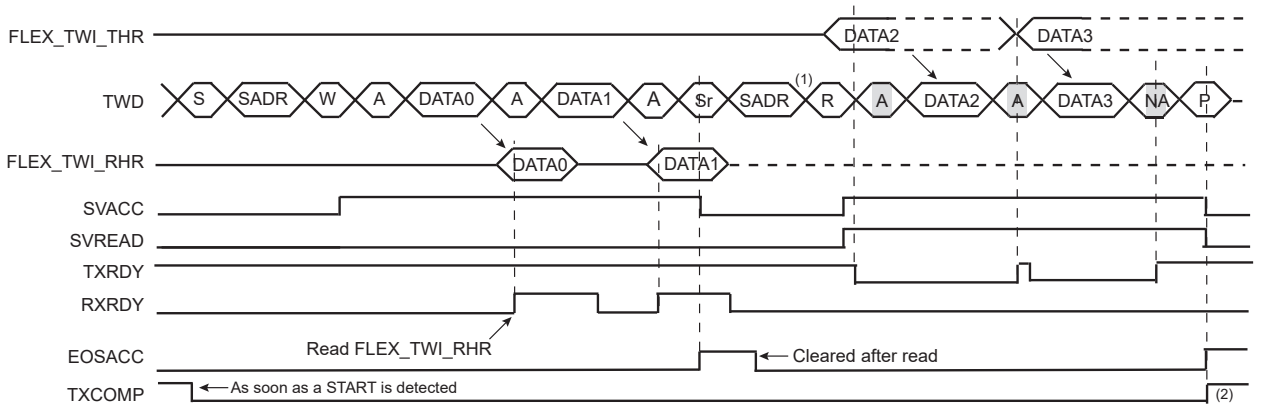
© 2026 Microchip Technology Inc. and its subsidiaries

— Reversal of Write to Read

The host initiates the communication by a write command and finishes it by a read command.

The following figure describes the repeated start and the reversal from Write mode to Read mode.

Figure 33-122. Repeated Start and Reversal from Write Mode to Read Mode



Notes:
1. In this case, if FLEX_TWI_THR has not been written at the end of the read command, the clock is automatically stretched before the ACK.
2. TXCOMP is only set at the end of the transmission because after the repeated start, SADR is detected again.

33.9.5.4.6. Using the Peripheral DMA Controller (PDC) in Client Mode

The use of the PDC significantly reduces the CPU load.

— Data Transmit with the PDC in Client Mode

The following procedure shows an example to transmit data with PDC.

1. Initialize the transmit PDC (memory pointers, transfer size).
2. Start the transfer by setting the PDC TXTEN bit.
3. Wait for the PDC ENDTX Flag by using either the polling method or the ENDTX interrupt.
4. Disable the PDC by setting the PDC TXTDIS bit.
5. (Optional) Wait for the FLEX_TWI_SR.TXCOMP flag before disabling the peripheral clock if required.

— Data Receive with the PDC in Client Mode

The following procedure shows an example to transmit data with PDC where the number of characters to receive is known.

1. Initialize the receive PDC (memory pointers, transfer size).
2. Set the PDC RXTEN bit.
3. Wait for the PDC ENDRX flag by using either the polling method or the ENDRX interrupt.
4. Disable the PDC by setting the PDC RXTDIS bit.
5. (Optional) Wait for the FLEX_TWI_SR.TXCOMP flag before disabling the peripheral clock if required.

33.9.5.4.7. SMBus Mode

SMBus mode is enabled when the FLEX_TWI_CR.SMEN bit is written to one. SMBus mode operation is similar to I²C operation with the following exceptions:

1. Only 7-bit addressing can be used.

© 2026 Microchip Technology Inc. and its subsidiaries

2. The SMBus standard describes a set of timeout values to ensure progress and throughput on the bus. These timeout values must be programmed into FLEX_TWI_SMBTR.
3. Transmissions can optionally include a CRC byte, called Packet Error Check (PEC).
4. A set of addresses have been reserved for protocol handling, such as alert response address (ARA) and host header (HH) address. Address matching on these addresses can be enabled by configuring FLEX_TWI_CR appropriately.

— Packet Error Checking

Each SMBus transfer can optionally end with a CRC byte, called the PEC byte. Writing the FLEX_TWI_CR.PECEN bit to one will send/check the FLEX_TWI_ACR.PEC field in the current transfer. The PEC generator is always updated on every bit transmitted or received, so that PEC handling on following linked transfers will be correct.

In Client Receiver mode, the host calculates a PEC value and transmits it to the client after all data bytes have been transmitted. Upon reception of this PEC byte, the client will compare it to the PEC value it has computed itself. If the values match, the data was received correctly, and the client will return an ACK to the host. If the PEC values differ, data was corrupted, and the client will return a NACK value. The FLEX_TWI_SR.PECERR bit is set automatically if a PEC error occurred.

In Client Transmitter mode, the client calculates a PEC value and transmits it to the host after all data bytes have been transmitted. Upon reception of this PEC byte, the host will compare it to the PEC value it has computed itself. If the values match, the data was received correctly. If the PEC values differ, data was corrupted, and the host must take appropriate action.

See Client Read/Write Flowcharts for detailed flowcharts.

— Timeouts

The TWI SMBus Timing register (FLEX_TWI_SMBTR) configures the SMBus timeout values. If a timeout occurs, the client leaves the bus. Furthermore, the FLEX_TWI_SR.TOUT bit is set.

33.9.5.5. High-Speed Client Mode

High-speed mode is enabled when the FLEX_TWI_CR.HSEN bit is written to one. Furthermore, the analog pad filter must be enabled, the FLEX_TWI_FILTR.PADFEN bit must be written to one and the FLEX_TWI_FILTR.FILT bit must be cleared. TWI High-speed mode operation is similar to TWI operation with the following exceptions:

1. A host code is received first at normal speed before entering High-speed mode period.
2. When TWI High-speed mode is active, clock stretching is only allowed after acknowledge (ACK), not-acknowledge (NACK), START (S) or repeated START (Sr) (asa consequence, OVF may happen).

TWI High-speed mode allows transfers of up to 3.4 Mbit/s.

The TWI client in High-speed mode requires that the peripheral clock runs at a minimum of 14 MHz if client clock stretching is enabled (SCLWSDIS bit at ‘0’). If client clock stretching is disabled (SCLWSDIS bit at ‘1’), the peripheral clock must run at a minimum of 11 MHz (assuming the system has no latency).

Notes:

1. When client clock stretching is disabled, FLEX_TWI_RHR must always be read before receiving the next data (write access frame generated by the host). It is strongly recommended to use either the polling method on the FLEX_TWI_SR.RXRDY flag, or the PDC. If the receive is managed by an interrupt, the TWI interrupt priority must be set to the right level and its latency minimized to avoid receive overrun.
2. When client clock stretching is disabled, FLEX_TWI_THR must be filled with the first data to send before the beginning of the frame (read access frame generated by the host). It is strongly recommended to use either the polling method on the FLEX_TWI_SR.TXRDY flag, or the PDC. If the transmit is managed by an interrupt, the TWI interrupt priority must be set to the right level and its latency minimized to avoid transmit underrun.

© 2026 Microchip Technology Inc. and its subsidiaries

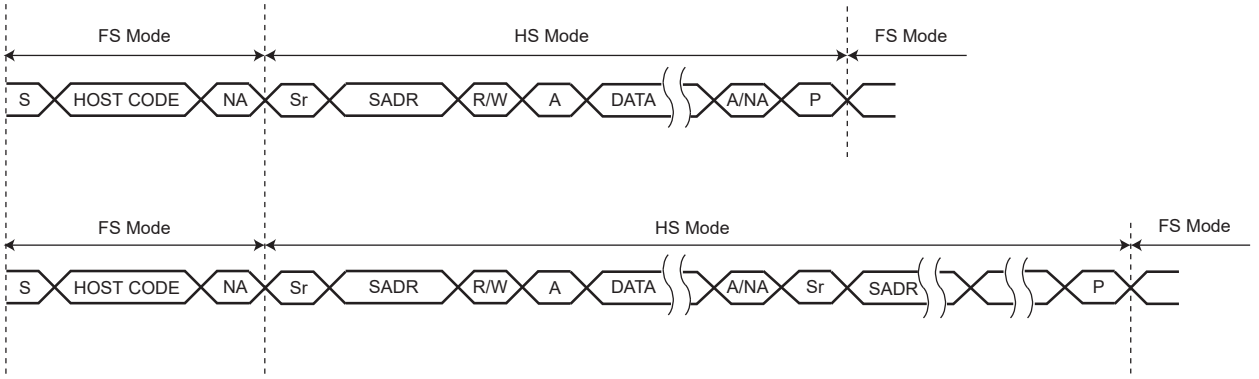
33.9.5.5.1. Read/Write Operation

A TWI high-speed frame always begins with the following sequence:

1. START condition (S)
2. Host Code (0000 1XXX)
3. Not-acknowledge (NACK)

When the TWI is programmed in Client mode and TWI High-speed mode is activated, host code matching is activated and internal timings are set to match the TWI High-speed mode requirements.

Figure 33-123. High-Speed Mode Read/Write



33.9.5.5.2. Usage

TWI High-speed mode usage is the same as the standard TWI (see Read/Write Flowcharts).

33.9.5.6. Alternative Command

In Client mode, the Alternative Command mode is used when the SMBus mode is enabled to send or check the PEC byte.

The Alternative Command mode is enabled by setting the ACMEN bit of the TWI Control register, and the transfer is configured in FLEX_TWI_ACR.

For a combined transfer with PEC, only the NPEC bit in FLEX_TWI_ACR must be set as the PEC byte is sent once at the end of the frame.

See Client Read/Write Flowcharts for detailed flowcharts.

33.9.5.7. TWI Asynchronous and Partial Wakeup

The TWI module includes an asynchronous start condition detector, capable of waking the device up from a Sleep mode upon an address match (and optionally an additional data match), including Sleep modes where the TWI peripheral clock is stopped. It can also be enabled when the system is fully running. In any case, only the peripheral clock is modified and VDDCORE always remains active.

FLEX_TWI_RHR must be read before enabling the asynchronous and partial wakeup.

After detecting the START condition on the bus, the TWI will stretch TWCK until the TWI peripheral clock has started. The time required for starting the TWI peripheral depends on which Sleep mode the device is in. After the TWI peripheral clock has started, the TWI releases its TWCK stretching and receives one byte of data (client address) on the bus. At this time, only a limited part of the device, including the TWI module, receives a clock, thus saving power. If the address phase causes a TWIS address match (and optionally if the first data byte causes data match as well), the entire device is wakened and normal TWI address matching actions are performed. Normal TWI transfer then follows. If the TWI module is not addressed (or if the optional data match fails), the TWI peripheral clock is automatically stopped and the device returns to its original Sleep mode.

© 2026 Microchip Technology Inc. and its subsidiaries

The TWI module has the capability to match on more than one address. The FLEX_TWI_SMR.SADR1EN/SADR2EN/SADR3EN bits enable address matching on additional addresses which can be configured through the FLEX_TWI_SWMR.SADR1/SADR2/SADR3 fields. The matching process can be extended to the first received data byte if the FLEX_TWI_SMR.DATAMEN bit is set. In that case, a complete matching includes address matching and first received data matching. The FLEX_TWI_SWMR.DATAM field can be used to configure the data to match on the first received byte.

When the system is in Active mode and the TWI enters asynchronous partial Wakeup mode, the flag SVACC must be programmed as the unique source of the TWI interrupt and the data match comparison must be disabled.

When the system exits Wait mode as the result of a matching condition, the SVACC flag is used to determine if the TWI is the source of the exit from Wait mode.

Figure 33-124. Address Match and Data Matching Disabled

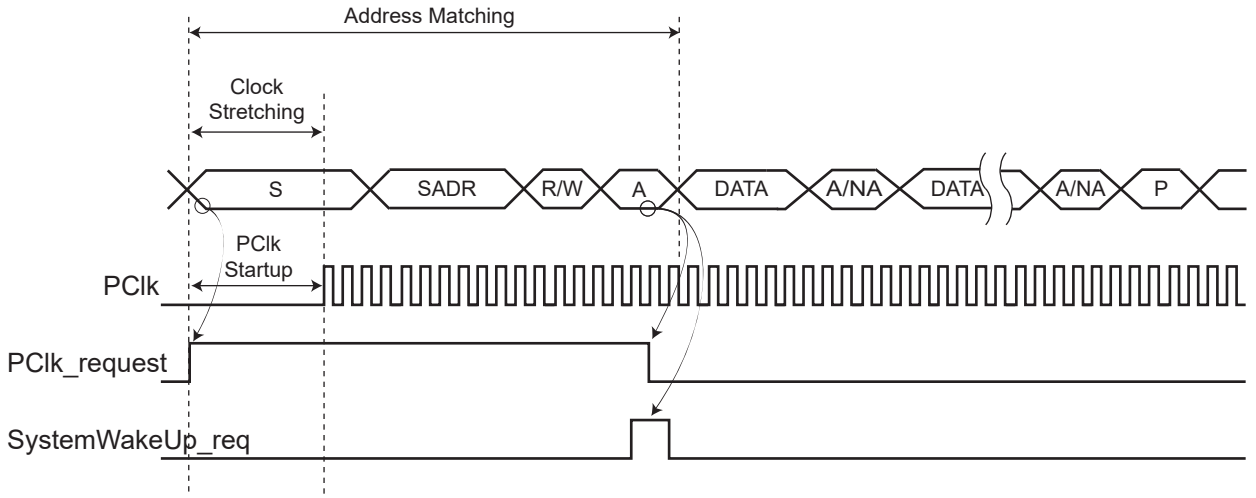
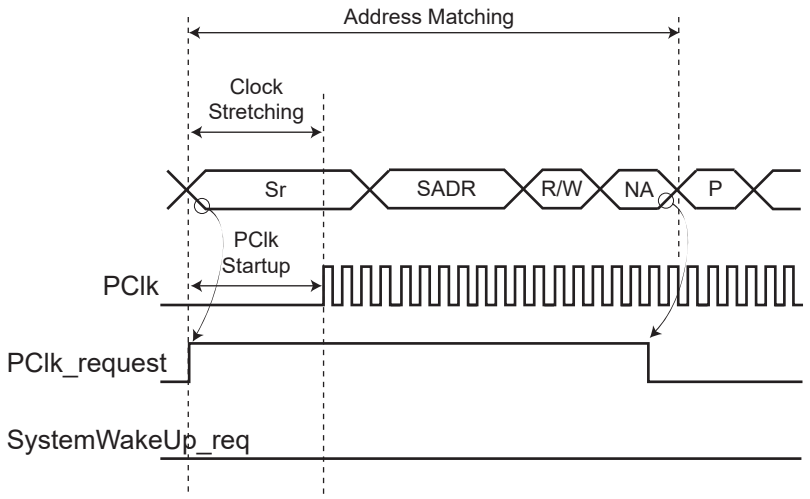


Figure 33-125. Address Does Not Match and Data Matching Disabled



© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-126. Address and Data Match (Data Matching Enabled)

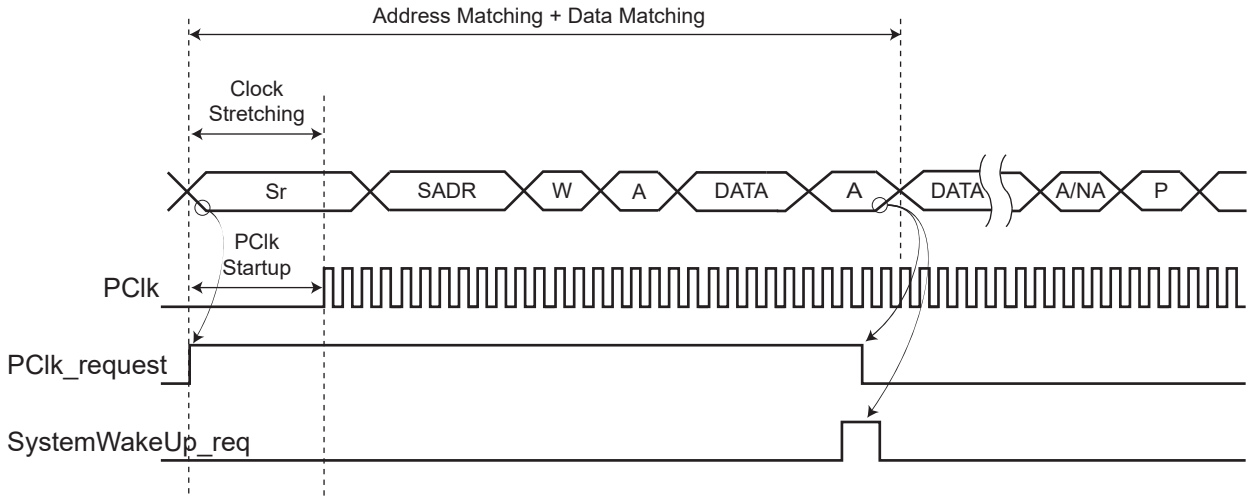
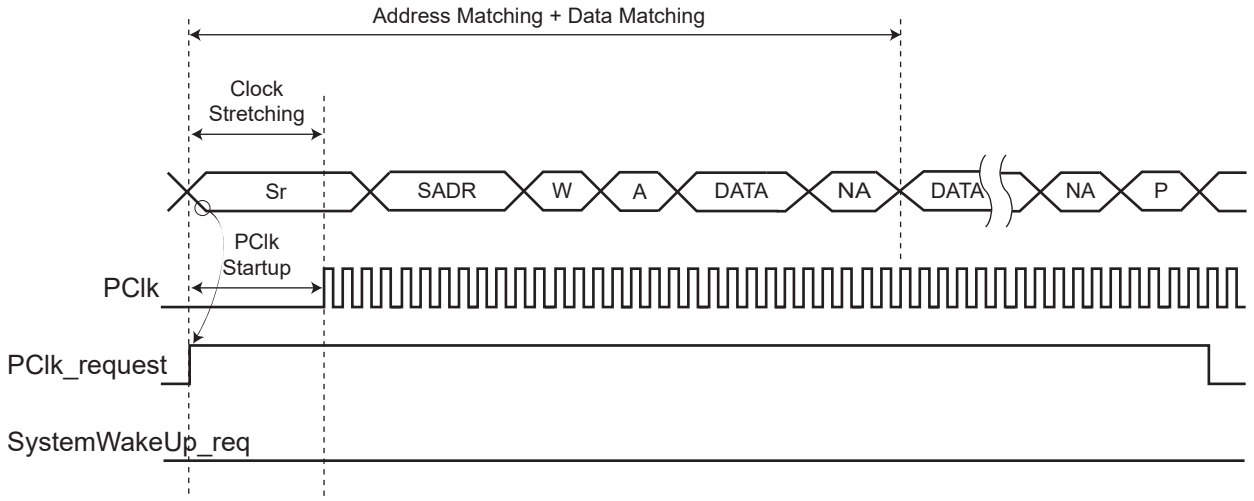


Figure 33-127. Address Matches and Data Do Not Match (Data Matching Enabled)

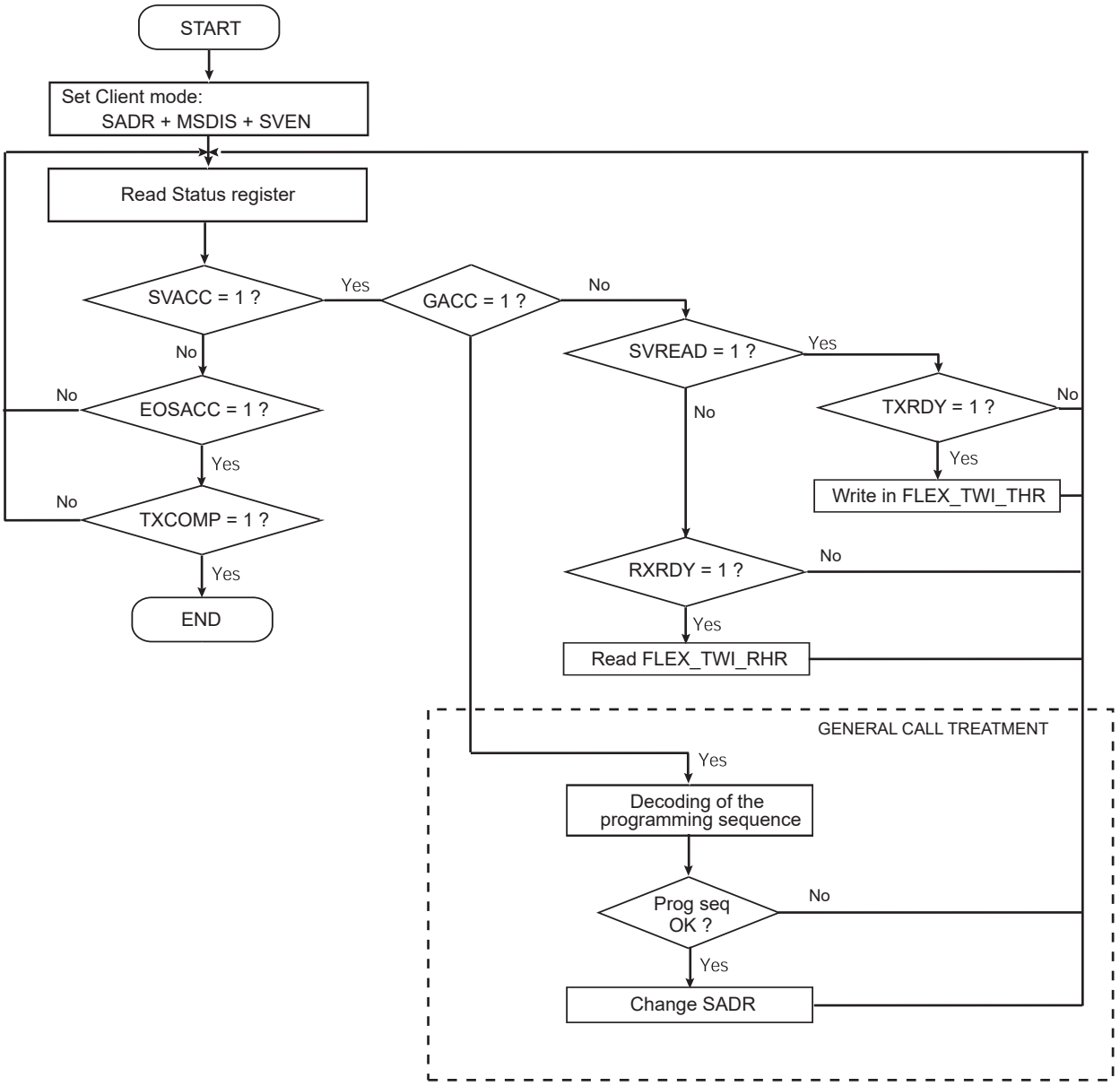


33.9.5.8. Client Read/Write Flowcharts

The flowchart shown in the following figure gives an example of read and write operations in Client mode. A polling or interrupt method can be used to check the status bits. The interrupt method requires that the Interrupt Enable register (FLEX_TWI_IER) be configured first.

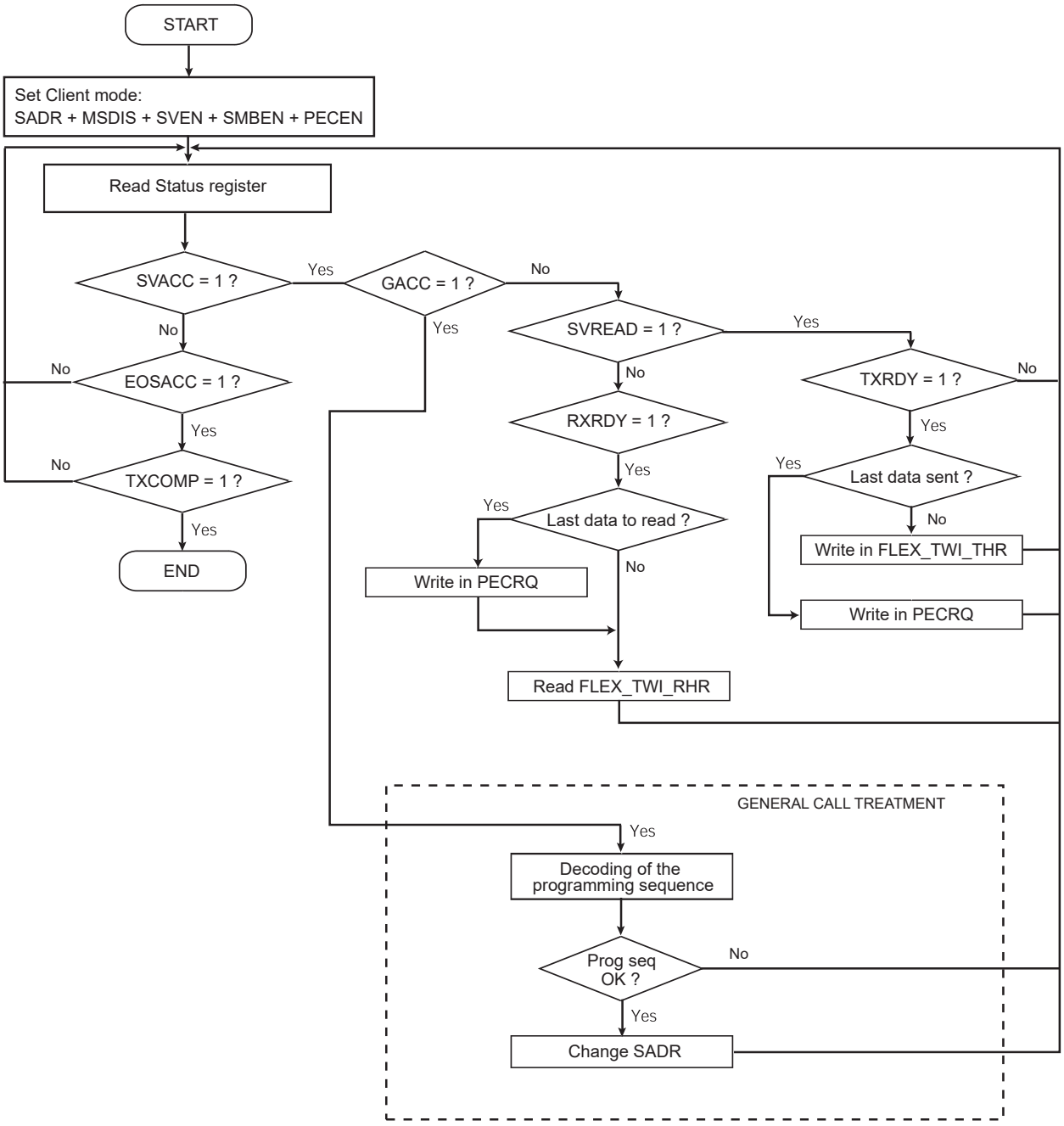
© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-128. Read/Write in Client Mode



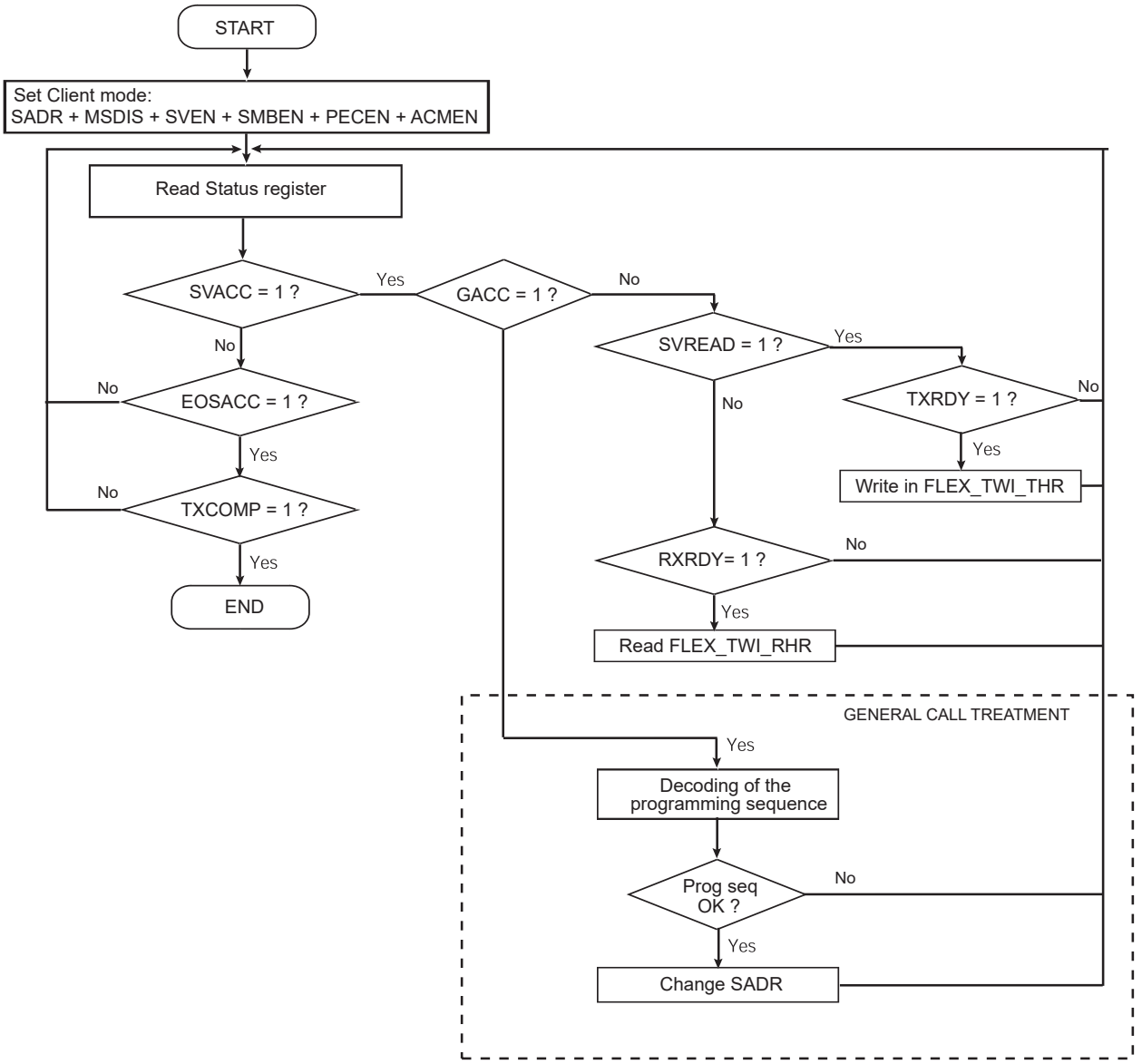
© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-129. Read/Write in Client Mode with SMBus PEC



© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-130. Read/Write in Client Mode with SMBus PEC and Alternative Command Mode



33.9.6. TWI FIFOs

33.9.6.1. Overview

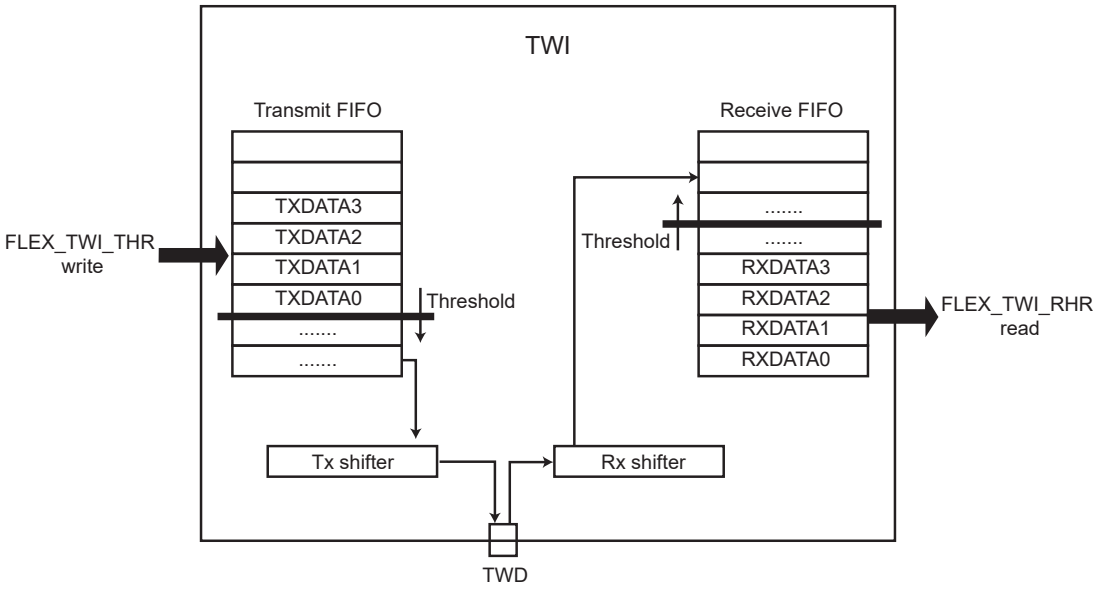
The TWI includes two FIFOs which can be enabled/disabled using FLEX_TWI_CR.FIFOEN/FIFODIS. Both Host and Client modes must be disabled before enabling or disabling the FIFOs (FLEX_TWI_CR.MSDIS/SVDIS).

Writing FLEX_TWI_CR.FIFOEN to '1' enables a 8-byte Transmit FIFO and a 8-byte Receive FIFO.

It is possible to write or to read single or multiple bytes in the same access to FLEX_TWI_THR/RHR, depending on FLEX_TWI_FMR.TXRDYM/RXRDYM settings.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-131. TWI FIFOs Block Diagram



33.9.6.2. Sending Data with FIFO Enabled

When the Transmit FIFO is enabled, write access to FLEX_TWI_THR loads the Transmit FIFO.

The Transmit FIFO level is provided in FLEX_TWI_FLR.TXFL. If the FIFO can accept the number of bytes to be transmitted, there is no need to monitor FLEX_TWI_SR.TXRDY and the bytes can be successively written in FLEX_TWI_THR.

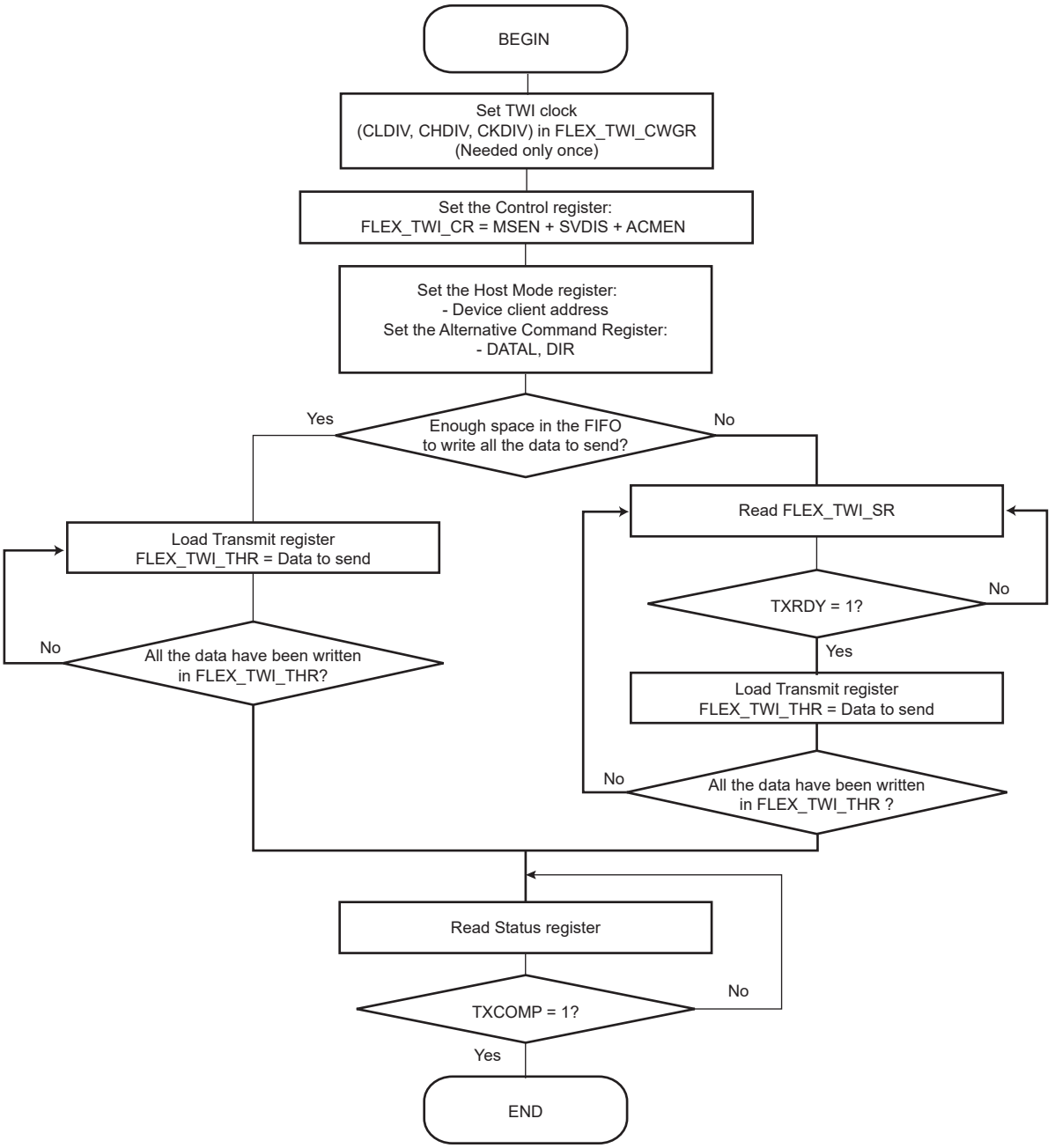
If the FIFO cannot accept the bytes due to insufficient space, wait for the TXRDY flag to be set before writing the bytes in FLEX_TWI_THR.

When the space in the FIFO allows only a portion of the data to be written, the TXRDY flag must be monitored before writing the remaining data.

See figures Sending Data with FIFO Enabled in Host Mode and Sending/Receiving Data with FIFO Enabled in Client Mode.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-132. Sending Data with FIFO Enabled in Host Mode



33.9.6.3. Receiving Data with FIFO Enabled

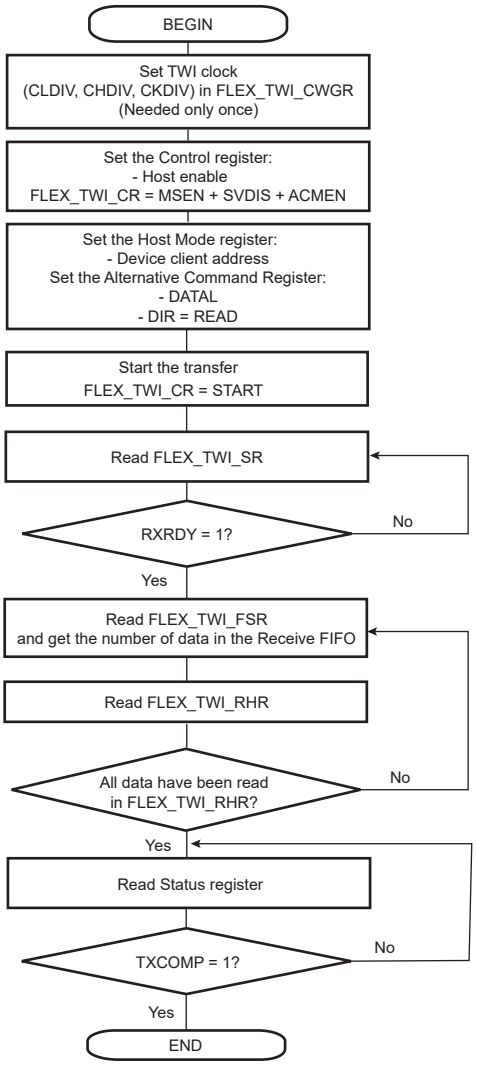
When the Receive FIFO is enabled, FLEX_TWI_RHR access reads the FIFO.

When data are present in the Receive FIFO (RXRDY flag set to '1'), the exact number of bytes can be checked with FLEX_TWI_FLR.RXFL. All the bytes can be read successively in FLEX_TWI_RHR without checking the FLEX_TWI_SR.RXRDY flag between each access.

See figures Receiving Data with FIFO Enabled in Host Mode and Sending/Receiving Data with FIFO Enabled in Client Mode.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-133. Receiving Data with FIFO Enabled in Host Mode

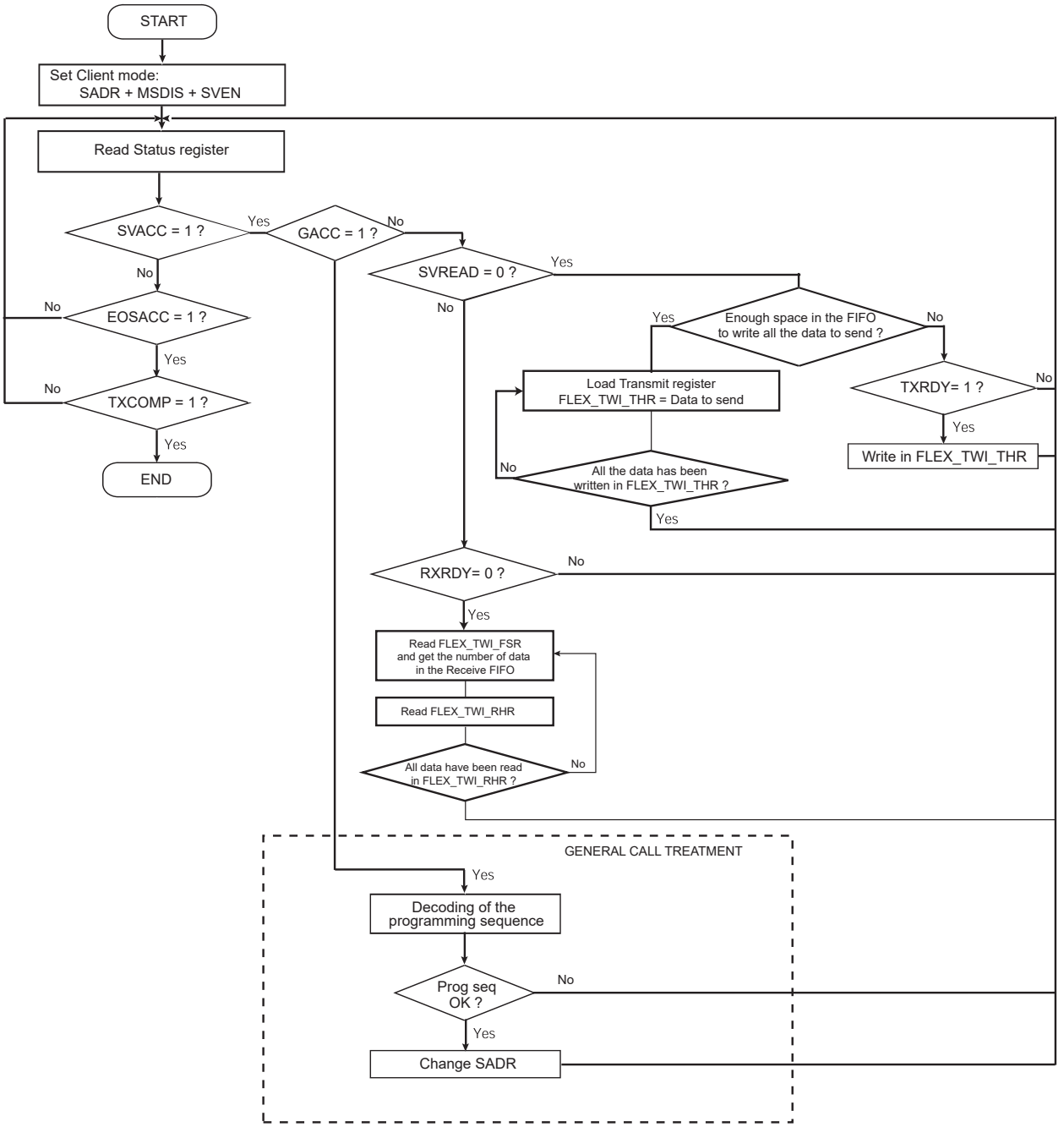


33.9.6.4. Sending/Receiving with FIFO Enabled in Client Mode

See Sending Data with FIFO Enabled and Receiving Data with FIFO Enabled for details.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-134. Sending/Receiving Data with FIFO Enabled in Client Mode



33.9.6.5. Clearing/Flushing FIFOs

Each FIFO can be cleared/flushed using FLEX_TWI_CR.TXFCLR/RXFCLR.

33.9.6.6. TXRDY and RXRDY Behavior

FLEX_TWI_SR.TXRDY/RXRDY flags display a specific behavior when FIFOs are enabled.

TXRDY indicates if a byte can be written in the Transmit FIFO. Thus the TXRDY flag is set as long as the Transmit FIFO can accept new byte. See figure TXRDY Behavior when TXRDYM = 0 in Host Mode.

RXRDY indicates if an unread byte is present in the Receive FIFO. Thus the RXRDY flag is set as soon as one unread byte is in the Receive FIFO. See figure RXRDY Behavior when RXRDYM = 0 in Host and Client Modes.

© 2026 Microchip Technology Inc. and its subsidiaries

TXRDY and RXRDY behavior can be modified using the TXRDYM and RXRDYM fields in the TWI FIFO Mode register (FLEX_TWI_FMR) to reduce the number of accesses to FLEX_TWI_THR/RHR.

As an example, in Host mode, the Transmit FIFO can be loaded with multiple bytes in the same access by configuring TXRDYM>0.

See FLEX_TWI_FMR for the FIFO configuration.

Figure 33-135. TXRDY Behavior when TXRDYM = 0 in Host Mode

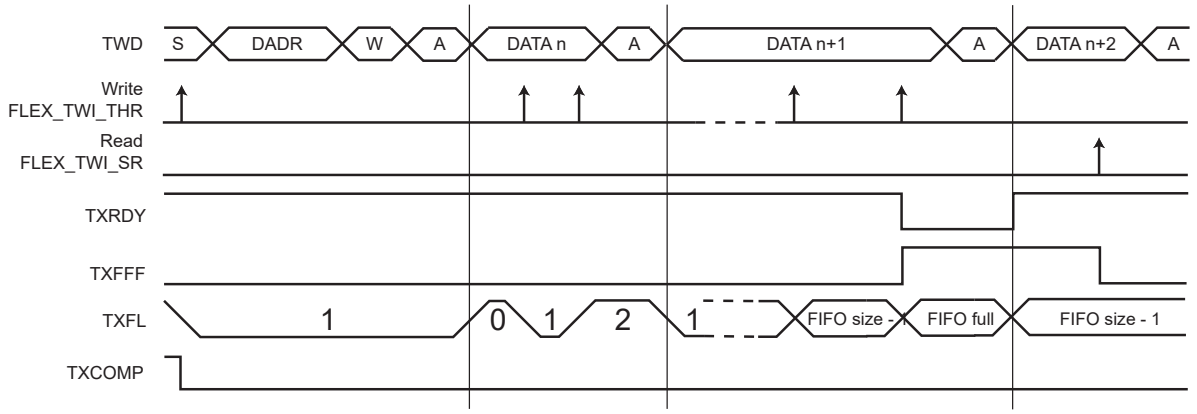


Figure 33-136. RXRDY Behavior when RXRDYM = 0 in Host and Client Modes

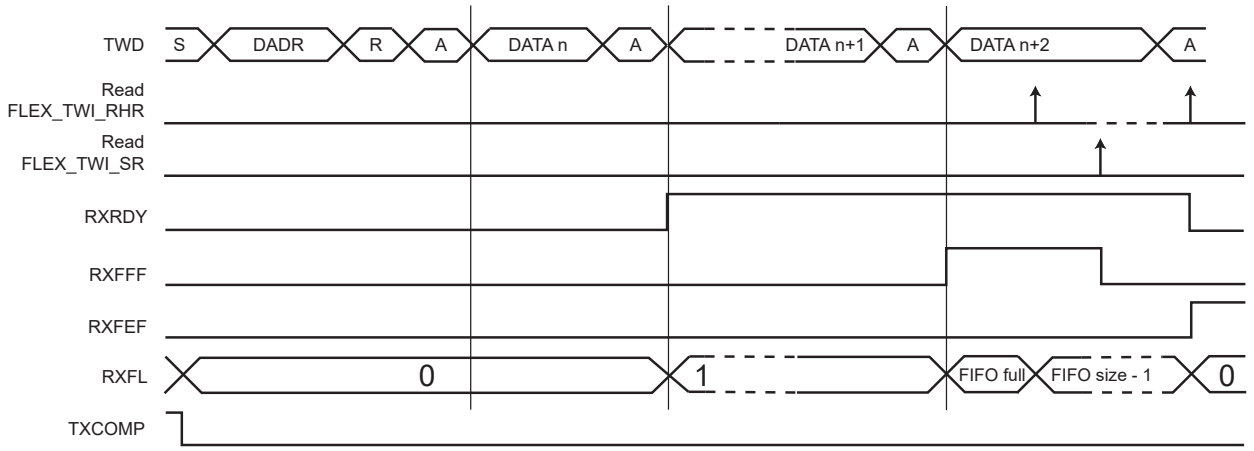
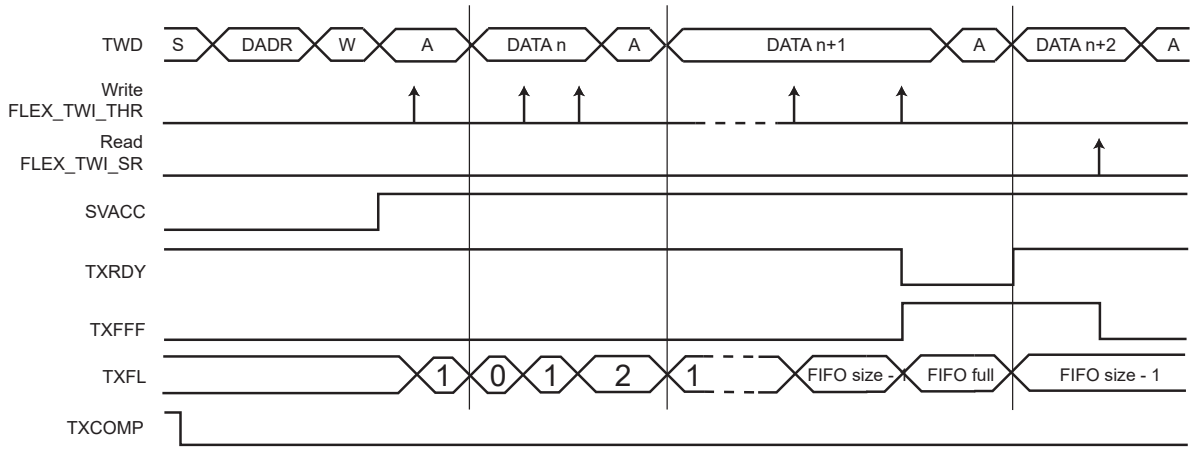


Figure 33-137. TXRDY Behavior when TXRDYM = 0 in Client Mode



© 2026 Microchip Technology Inc. and its subsidiaries

33.9.6.7. TWI Single Data Access

When FIFO is enabled and a byte access is performed in FLEX_TWI_THR, one byte is written in the FIFO. The same behavior applies for FLEX_TWI_RHR.

See TWI Transmit Holding Register and TWI Receive Holding Register.

However, it is possible to write/read multiple data each time FLEX_THR_THR/FLEX_US_RHR is accessed. See TWI Multiple Data Access.

33.9.6.8. TWI Multiple Data Access

It is possible to reduce the number of accesses to/from FLEX_TWI_THR/FLEX_US_RHR required to transfer an amount of data, by concatenating multiple bytes.

Up to four data can be written/read in one FLEX_TWI_THR/FLEX_TWI_RHR access when the FIFO is enabled (FLEX_TWI_CR.FIFOEN=1) and Sniffer mode is disabled (FLEX_TWI_SMR.SNIFF=0).

When the FIFO is enabled, the number of bytes to write/read is defined by the type of access in the holding register. If the access is a byte, only one byte is written/read (single data access), if the access is a halfword or a word a multiple data access is performed. If the access is a halfword, then two bytes are written/read and if the access is a word, four bytes are written/read.

Written/Read data are always right-aligned, as described in sections TWI Receive Holding Register (FIFO Enabled) and TWI Transmit Holding Register (FIFO Enabled).

As an example, if the Transmit FIFO is empty and there are six bytes to send, either of the following write accesses may be performed:

- Six FLEX_TWI_THR-byte write accesses
- Three FLEX_TWI_THR-halfword write accesses
- One FLEX_TWI_THR-word write access and one FLEX_TWI_THR halfword write access

With a Receive FIFO containing six bytes, any of the following read accesses may be performed:

- Six FLEX_TWI_RHR-byte read accesses
- Three FLEX_TWI_RHR-halfword read accesses
- One FLEX_TWI_RHR-word read access and one FLEX_TWI_RHR-halfword read access

33.9.6.8.1. TXRDY and RXRDY Configuration

It is possible to write one or more bytes in the same FLEX_TWI_THR/FLEX_TWI_RHR access. The TXRDY flag indicates if one or more bytes can be written in the FIFO depending on the configuration of FLEX_TWI_FMR.TXRDYM/RXRDYM.

When two bytes are written for each FLEX_TWI_THR access, the TXRDYM field can be configured so that the TXRDY flag is at '1' only when at least two bytes can be written in the Transmit FIFO.

When four bytes are read for each FLEX_TWI_RHR access, the RXRDYM field can be configured so that the RXRDY flag is at '1' only when at least four unread bytes are in the Receive FIFO.

33.9.6.8.2. PDC

The PDC transfer type must be configured according to the FLEX_TWI_FMR.TXRDYM/RXRDYM settings.

As example, FLEX_TWI_FMR.TXRDYM/RXRDYM=0 is not compatible with PDC transfers in word (32-bit).

33.9.6.9. Transmit FIFO Lock

If a frame is terminated early due to a not-acknowledge error (NACK flag), SMBus timeout error (TOUT flag) or host code acknowledge error (MACK flag), a lock is set on the Transmit FIFO preventing any new frame from being sent until it is cleared. This allows clearing the FIFO if needed, resetting PDC channels, etc., without any risk.

FLEX_TWI_SR.LOCK is used to check the state of the Transmit FIFO lock.

© 2026 Microchip Technology Inc. and its subsidiaries

The Transmit FIFO lock can be cleared by setting FLEX_TWI_CR.TXFLCLR to '1'.

33.9.6.10. FIFO Pointer Error

A FIFO overflow is reported in FLEX_TWI_FSR.

If the Transmit FIFO is full and a write access is performed on FLEX_TWI_THR, it generates a Transmit FIFO pointer error and sets FLEX_TWI_FSR.TXFPTEF.

If the number of data written in FLEX_TWI_THR (according to the register access size) is greater than the free space in the Transmit FIFO, a Transmit FIFO pointer error is generated and FLEX_TWI_FSR.TXFPTEF is set.

A FIFO underflow is reported in FLEX_TWI_FSR.

If the number of bytes read in FLEX_TWI_RHR (according to the register access size) is greater than the number of unread bytes in the Receive FIFO, a Receive FIFO pointer error is generated and FLEX_TWI_FSR.RXFPTEF is set.

No pointer error occurs if the FIFO state/level is checked before writing/reading in FLEX_TWI_THR/ FLEX_TWI_RHR. The FIFO state/level can be checked either with TXRDY, RXRDY, TXFL or RXFL. When a pointer error occurs, other FIFO flags may not behave as expected; their states should be ignored.

If a Transmit or Receive pointer error occurs, a software reset must be performed using FLEX_TWI_CR.SWRST. Note that issuing a software reset during transmission may leave a client in an unknown state holding the TWD line. In this case, a Bus Clear command may instruct the client to release the TWD line (the first frame sent afterwards may not be received properly by the client). See Bus Clear Command to initiate the Bus Clear command.

33.9.6.11. FIFO Thresholds

Each Transmit and Receive FIFO includes a threshold feature used to set a flag and an interrupt when a FIFO threshold is crossed. Thresholds are defined as a number of bytes in the FIFO, and the FIFO state (TXFL or RXFL) represents the number of bytes currently in the FIFO.

The Transmit FIFO threshold can be set using the field FLEX_TWI_FMR.TXFTHRES. Each time the Transmit FIFO level goes from 'above threshold' to 'equal to or below threshold', the flag FLEX_TWI_FESR.TXFTHF is set. The application is warned that the Transmit FIFO has reached the defined threshold and that it can be reloaded.

The Receive FIFO threshold can be set using the field FLEX_TWI_FMR.RXFTHRES. Each time the Receive FIFO level goes from 'below threshold' to 'equal to or above threshold', the flag FLEX_TWI_FESR.RXFTHF is set. The application is warned that the Receive FIFO has reached the defined threshold and that it can be read to prevent an underflow.

The TXFTHF and RXFTHF flags can be configured to generate an interrupt using FLEX_TWI_FIER and FLEX_TWI_FIDR.

33.9.6.12. FIFO Flags

FIFOs come with a set of flags which can be configured to generate interrupts through FLEX_TWI_FIER and FLEX_TWI_FIDR.

FIFO flags state can be read in FLEX_TWI_FSR. They are cleared when FLEX_TWI_FSR is read.

33.9.7. TWI Comparison Function on Received Character

The comparison function differs if the asynchronous partial wakeup is enabled or not.

If asynchronous partial wakeup is disabled, the TWI has the capability to extend the address matching on up to three client addresses. The FLEX_TWI_SMR.SADR1EN/SADR2EN/SADR3EN bits enable address matching on additional addresses which can be configured through the FLEX_TWI_SWMR.SADR1/SADR2/SADR3 fields. The DATAMEN bit has no effect.

The SVACC bit is set when there is a comparison match with the received client address.

© 2026 Microchip Technology Inc. and its subsidiaries

33.9.8. Sniffer Mode

The Client Sniffer mode of a TWI can be enabled to ease the analysis/debug of a TWI bus activity. The TWI bus to be analyzed can be monitored by one TWI host embedded in the product or by an I2C host outside the product.

In this mode, the TWI reports all (or part of) the TWI bus activity without impacting the TWI bus (no bus drive is performed). Depending on the MASK field value, only some specific transfers can be logged instead of the whole activity.

The peripheral TWIn can be configured to analyze the peripheral in a full transparent mode via predefined internal connections between TWI instances (n is the index of the TWI instance).

The predefined internal connections provide the capability to use the TWD and TWCK pins for alternate functions while the TWI peripheral is configured in Sniffer Client mode and the selected TWI bus to analyze is carried on these internal links.

The following fields must be programmed before entering Client mode:

1. FLEX_TWI_SMR.SADR: Use the client device address to indicate which frame(s) to log.
2. FLEX_TWI_SMR.MASK: Indicate which SADR bits should be masked and thus which transfers should be logged (set to 0x7F to log the whole TWI bus activity; all SADR bits are masked in this case). General Call accesses will always match.
3. FLEX_TWI_SMR.BSEL: Select the TWI bus to analyze (see figure Sniffer Mode Application Overview).
4. FLEX_TWI_SMR.SNIFF: Set to '1' to enable Client Sniffer mode.
5. FLEX_TWI_CR.MSDIS: Disable Host mode.
6. FLEX_TWI_CR.SVEN: Enable Client mode.

As the device receives the clock, values written in FLEX_TWI_CWGR are not relevant.

Once configured in Client Sniffer mode, the FLEX_TWI_SR.RXRDY bit indicates when a transfer has been logged in FLEX_TWI_RHR. An interrupt can be generated if configured. The FLEX_TWI_SR.OVRE flag indicates if an overrun error occurred if the application is not fast enough to read FLEX_TWI_RHR.

In Client Sniffer mode, FLEX_TWI_RHR logs data as follows:

- The RXDATA field reports the sniffed 8-bit data field.
- The SSTATE field indicates if a START condition has been detected before the 8-bit data field.
- The PSTATE field indicates if a STOP condition has been detected after the previously sniffed 8-bit data field.
- The ASTATE field indicates which acknowledge condition has been detected after the previously sniffed 8-bit data field.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 33-138. Sniffer Mode Application Overview

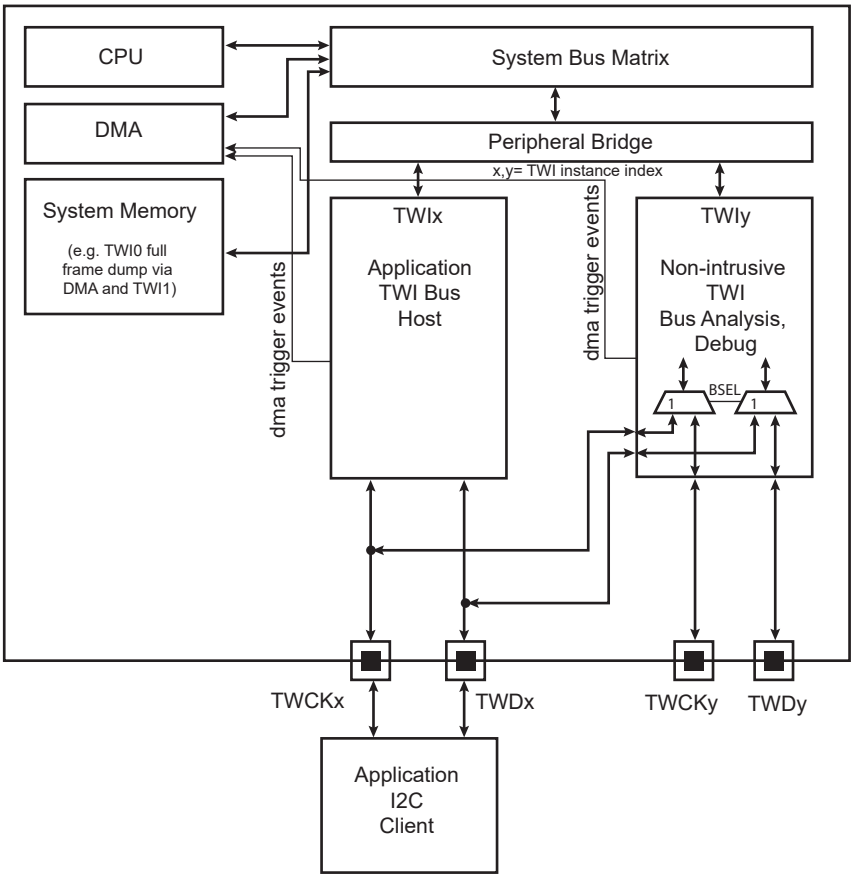
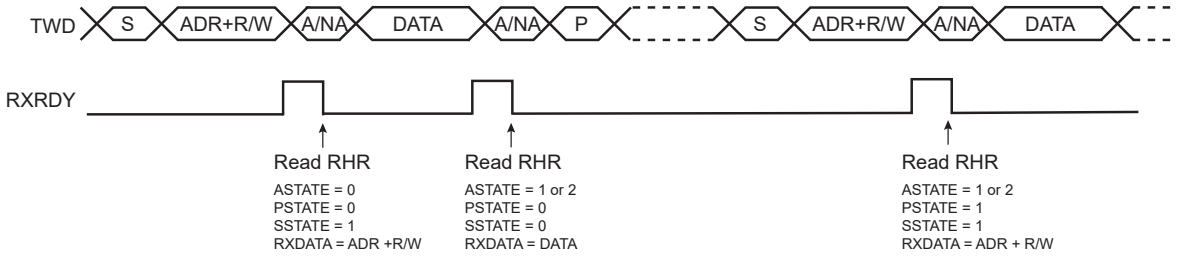


Figure 33-139. Client Sniffer Mode Log



33.9.9. TWI Register Write Protection

The FLEXCOM operating mode (FLEX_MR.OPMODE) must be set to FLEX_MR_OPMODE_TWI to enable access to the write protection registers.

To prevent any single software error from corrupting TWI behavior, certain registers in the address space can be write-protected by setting the WPEN (Write Protection Enable), WPITEN (Write Protection Interrupt Enable), and/or WPCREN (Write Protection Control Enable) bits in the TWI Write Protection Mode Register (FLEX_TWI_WPMR).

If a write access to a write-protected register is detected, the Write Protection Violation Status (WPVS) flag in the TWI Write Protection Status Register (FLEX_TWI_WPSR) is set and the Write Protection Violation Source (WPVSRC) field indicates the register in which the write access has been attempted.

© 2026 Microchip Technology Inc. and its subsidiaries

The WPVS bit is automatically cleared after reading FLEX_TWI_WPSR.

The following register(s) can be write-protected when WPEN is set:

- TWI Client Mode Register
- TWI Clock Waveform Generator Register
- TWI SMBus Timing Register
- TWI Matching Register
- TWI FIFO Mode Register

The following register(s) can be write-protected when WPITEN is set:

- TWI Interrupt Enable Register
- TWI Interrupt Disable Register

The following register(s) can be write-protected when WPCREN is set:

- TWI Control Register

© 2026 Microchip Technology Inc. and its subsidiaries

33.10. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	FLEX_MR	31:24								
		23:16								
		15:8								
		7:0							OPMODE[1:0]	
0x04 ... 0x0F	Reserved									
0x10	FLEX_RHR	31:24								
		23:16								
		15:8	RXDATA[15:8]							
		7:0	RXDATA[7:0]							
0x14 ... 0x1F	Reserved									
0x20	FLEX_THR	31:24								
		23:16								
		15:8	TXDATA[15:8]							
		7:0	TXDATA[7:0]							
0x24 ... 0x01FF	Reserved									
0x0200	FLEX_US_CR	31:24	FIFODIS	FIFOEN		REQCLR		TXFLCLR	RXFCLR	TXFCLR
		23:16			LINWKUP	LINABT	RTSDIS	RTSEN		
		15:8	RETTO	RSTNACK	RSTIT	SENDA	STTTO	STPBRK	STTBRK	RSTSTA
		7:0	TXDIS	TXEN	RXDIS	RXEN	RSTTX	RSTRX		
0x0204	FLEX_US_MR	31:24	ONEBIT	MODSYNC	MAN	FILTER		MAX_ITERATION[2:0]		
		23:16	INVDATA	VAR_SYNC	DSNACK	INACK	OVER	CLKO	MODE9	MSBF
		15:8	CHMODE[1:0]		NBSTOP[1:0]		PAR[2:0]			SYNC
		7:0	CHRL[1:0]		USCLKS[1:0]		USART_MODE[3:0]			
0x0204	FLEX_US_MR (OOK)	31:24				FILTER				
		23:16			OOKEN	OOKRXD	OVER	CLKO	MODE9	MSBF
		15:8	CHMODE[1:0]		NBSTOP[1:0]		PAR[2:0]			SYNC
		7:0	CHRL[1:0]		USCLKS[1:0]		USART_MODE[3:0]			
0x0208	FLEX_US_IER	31:24								MANE
		23:16		CMP			CTSIC			
		15:8			NACK	RXBUFF	TXBUFE	ITER	TXEMPTY	TIMEOUT
		7:0	PARE	FRAME	OVRE	ENDTX	ENDRX	RXBRK	TXRDY	RXRDY
0x0208	FLEX_US_IER (LIN_MODE)	31:24	LINHTE	LINSTE	LINSNRE	LINCE	LINIPE	LINISFE	LINBE	
		23:16								
		15:8	LINTC	LINID	LINBK	RXBUFF	TXBUFE		TXEMPTY	TIMEOUT
		7:0	PARE	FRAME	OVRE	ENDTX	ENDRX		TXRDY	RXRDY
0x020C	FLEX_US_IDR	31:24								MANE
		23:16		CMP			CTSIC			
		15:8			NACK	RXBUFF	TXBUFE	ITER	TXEMPTY	TIMEOUT
		7:0	PARE	FRAME	OVRE	ENDTX	ENDRX	RXBRK	TXRDY	RXRDY
0x020C	FLEX_US_IDR (LIN_MODE)	31:24	LINHTE	LINSTE	LINSNRE	LINCE	LINIPE	LINISFE	LINBE	
		23:16								
		15:8	LINTC	LINID	LINBK	RXBUFF	TXBUFE		TXEMPTY	TIMEOUT
		7:0	PARE	FRAME	OVRE	ENDTX	ENDRX		TXRDY	RXRDY
0x0210	FLEX_US_IMR	31:24								MANE
		23:16		CMP			CTSIC			
		15:8			NACK	RXBUFF	TXBUFE	ITER	TXEMPTY	TIMEOUT
		7:0	PARE	FRAME	OVRE	ENDTX	ENDRX	RXBRK	TXRDY	RXRDY
0x0210	FLEX_US_IMR (LIN_MODE)	31:24	LINHTE	LINSTE	LINSNRE	LINCE	LINIPE	LINISFE	LINBE	
		23:16								
		15:8	LINTC	LINID	LINBK	RXBUFF	TXBUFE		TXEMPTY	TIMEOUT
		7:0	PARE	FRAME	OVRE	ENDTX	ENDRX		TXRDY	RXRDY

© 2026 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0214	FLEX_US_CSR	31:24								MANE
		23:16	CTS	CMP			CTSIC			
		15:8			NACK	RXBUFF	TXBUFE	ITER	TXEMPTY	TIMEOUT
		7:0	PARE	FRAME	OVRE	ENDTX	ENDRX	RXBRK	TXRDY	RXRDY
0x0214	FLEX_US_CSR (LIN_MODE)	31:24	LINHTE	LINSTE	LINSNRE	LINCE	LINIPE	LINISFE	LINBE	
		23:16	LINBLS							
		15:8	LINTC	LINID	LINBK	RXBUFF	TXBUFE		TXEMPTY	TIMEOUT
		7:0	PARE	FRAME	OVRE	ENDTX	ENDRX		TXRDY	RXRDY
0x0218	FLEX_US_RHR	31:24								
		23:16								
		15:8	RXSYNH							RXCHR[8]
		7:0	RXCHR[7:0]							
0x0218	FLEX_US_RHR (FIFO_MULTI_DATA)	31:24	RXCHR3[7:0]							
		23:16	RXCHR2[7:0]							
		15:8	RXCHR1[7:0]							
		7:0	RXCHR0[7:0]							
0x021C	FLEX_US_THR	31:24								
		23:16								
		15:8	TXSYNH							TXCHR[8]
		7:0	TXCHR[7:0]							
0x021C	FLEX_US_THR (FIFO_MULTI_DATA)	31:24	TXCHR3[7:0]							
		23:16	TXCHR2[7:0]							
		15:8	TXCHR1[7:0]							
		7:0	TXCHR0[7:0]							
0x0220	FLEX_US_BRGR	31:24								
		23:16						FP[2:0]		
		15:8	CD[15:8]							
		7:0	CD[7:0]							
0x0224	FLEX_US_RTOR	31:24								
		23:16								TO[16]
		15:8	TO[15:8]							
		7:0	TO[7:0]							
0x0228	FLEX_US_TTGR	31:24								
		23:16								
		15:8								
		7:0	TG[7:0]							
0x022C ... 0x023F	Reserved									
0x0240	FLEX_US_FIDI	31:24								
		23:16								
		15:8	FI_DI_RATIO[15:8]							
		7:0	FI_DI_RATIO[7:0]							
0x0244	FLEX_US_NER	31:24								
		23:16								
		15:8								
		7:0	NB_ERRORS[7:0]							
0x0248 ... 0x024B	Reserved									
0x024C	FLEX_US_IF	31:24								
		23:16								
		15:8								
		7:0	IRDA_FILTER[7:0]							
0x0250	FLEX_US_MAN	31:24	RXIDLEV	DRIFT	ONE	RX_MPOL			RX_PP[1:0]	
		23:16					RX_PL[3:0]			
		15:8				TX_MPOL			TX_PP[1:0]	
		7:0					TX_PL[3:0]			

© 2026 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0254	FLEX_US_LINMR	31:24								
		23:16							SYNCDIS	PDCM
		15:8	DLC[7:0]							
		7:0	WKUPTYP	FSDIS	DLM	CHKTYP	CHKDIS	PARDIS	NACT[1:0]	
0x0258	FLEX_US_LINIR	31:24								
		23:16								
		15:8								
		7:0	IDCHR[7:0]							
0x025C	FLEX_US_LINBRR	31:24								
		23:16						LINFP[2:0]		
		15:8	LINCD[15:8]							
		7:0	LINCD[7:0]							
0x0260 ... 0x028F	Reserved									
0x0290	FLEX_US_CMPR	31:24								VAL2[8]
		23:16	VAL2[7:0]							
		15:8		CMPPAR	CMPMODE[1:0]					VAL1[8]
		7:0	VAL1[7:0]							
0x0294 ... 0x029F	Reserved									
0x02A0	FLEX_US_FMR	31:24			RXFTHRES2[5:0]					
		23:16			RXFTHRES[5:0]					
		15:8			TXFTHRES[5:0]					
		7:0	FRTSC		RXRDYM[1:0]				TXRDYM[1:0]	
0x02A4	FLEX_US_FLR	31:24								
		23:16			RXFL[5:0]					
		15:8								
		7:0			TXFL[5:0]					
0x02A8	FLEX_US_FIER	31:24								
		23:16								
		15:8							RXFTHF2	
		7:0	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
0x02AC	FLEX_US_FIDR	31:24								
		23:16								
		15:8							RXFTHF2	
		7:0	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
0x02B0	FLEX_US_FIMR	31:24								
		23:16								
		15:8							RXFTHF2	
		7:0	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
0x02B4	FLEX_US_FESR	31:24								
		23:16								
		15:8							RXFTHF2	TXFLOCK
		7:0	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
0x02B8 ... 0x02E3	Reserved									
0x02E4	FLEX_US_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0						WPCREN	WPITEN	WPEN
0x02E8	FLEX_US_WPSR	31:24								
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0								WPVS
0x02EC ... 0x03FF	Reserved									

© 2026 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0400	FLEX_SPI_CR	31:24	FIFODIS	FIFOEN						LASTXFER
		23:16							RXFCLR	TXFCLR
		15:8				REQCLR				
		7:0	SWRST						SPIDIS	SPIEN
0x0404	FLEX_SPI_MR	31:24	DLYBCS[7:0]							
		23:16							PCS[1:0]	
		15:8				CMPMODE				
		7:0	LLB		WDRBT	MODFDIS	BRSRCCLK	PCSDEC	PS	MSTR
0x0408	FLEX_SPI_RDR	31:24								
		23:16					PCS[3:0]			
		15:8	RD[15:8]							
		7:0	RD[7:0]							
0x0408	FLEX_SPI_RDR (FIFO_MULTI_DATA_8)	31:24	RD3[7:0]							
		23:16	RD2[7:0]							
		15:8	RD1[7:0]							
		7:0	RD0[7:0]							
0x0408	FLEX_SPI_RDR (FIFO_MULTI_DATA_16)	31:24	RD1[15:8]							
		23:16	RD1[7:0]							
		15:8	RD0[15:8]							
		7:0	RD0[7:0]							
0x040C	FLEX_SPI_TDR	31:24								LASTXFER
		23:16					PCS[3:0]			
		15:8	TD[15:8]							
		7:0	TD[7:0]							
0x040C	FLEX_SPI_TDR (FIFO_MULTI_DATA)	31:24	TD1[15:8]							
		23:16	TD1[7:0]							
		15:8	TD0[15:8]							
		7:0	TD0[7:0]							
0x0410	FLEX_SPI_SR	31:24	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
		23:16								SPIENS
		15:8				SFERR	CMP	UNDES	TXEMPTY	NSSR
		7:0	TXBUFE	RXBUFF	ENDTX	ENDRX	OVRES	MODF	TDRE	RDRF
0x0414	FLEX_SPI_IER	31:24	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
		23:16								
		15:8				SFERR	CMP	UNDES	TXEMPTY	NSSR
		7:0	TXBUFE	RXBUFF	ENDTX	ENDRX	OVRES	MODF	TDRE	RDRF
0x0418	FLEX_SPI_IDR	31:24	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
		23:16								
		15:8				SFERR	CMP	UNDES	TXEMPTY	NSSR
		7:0	TXBUFE	RXBUFF	ENDTX	ENDRX	OVRES	MODF	TDRE	RDRF
0x041C	FLEX_SPI_IMR	31:24	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
		23:16								
		15:8				SFERR	CMP	UNDES	TXEMPTY	NSSR
		7:0	TXBUFE	RXBUFF	ENDTX	ENDRX	OVRES	MODF	TDRE	RDRF
0x0420 ... 0x042F	Reserved									
0x0430	FLEX_SPI_CSR0	31:24	DLYBCT[7:0]							
		23:16	DLYBS[7:0]							
		15:8	SCBR[7:0]							
		7:0	BITS[3:0]				CSAAT	CSNAAT	NCPHA	CPOL
0x0434	FLEX_SPI_CSR1	31:24	DLYBCT[7:0]							
		23:16	DLYBS[7:0]							
		15:8	SCBR[7:0]							
		7:0	BITS[3:0]				CSAAT	CSNAAT	NCPHA	CPOL
0x0438 ... 0x043F	Reserved									

© 2026 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0440	FLEX_SPI_FMR	31:24			RXFTHRES[5:0]					
		23:16			TXFTHRES[5:0]					
		15:8								
		7:0			RXRDYM[1:0]				TXRDYM[1:0]	
0x0444	FLEX_SPI_FLR	31:24								
		23:16			RXFL[5:0]					
		15:8								
		7:0			TXFL[5:0]					
0x0448	FLEX_SPI_CMPR	31:24	VAL2[15:8]							
		23:16	VAL2[7:0]							
		15:8	VAL1[15:8]							
		7:0	VAL1[7:0]							
0x044C ... 0x04E3	Reserved									
0x04E4	FLEX_SPI_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0						WPCREN	WPITEN	WPEN
0x04E8	FLEX_SPI_WPSR	31:24								
		23:16								
		15:8	WPVSRC[7:0]							
		7:0								WPVS
0x04EC ... 0x05FF	Reserved									
0x0600	FLEX_TWI_CR	31:24			FIFODIS	FIFOEN		LOCKCLR		THRCLR
		23:16							ACMDIS	ACMEN
		15:8	CLEAR	PECRQ	PECDIS	PECEN	SMBDIS	SMBEN	HSDIS	HSEN
		7:0	SWRST	QUICK	SVDIS	SVEN	MSDIS	MSEN	STOP	START
0x0600	FLEX_TWI_CR (FIFO_ENABLED)	31:24			FIFODIS	FIFOEN		TXFLCLR	RXFCLR	TXFCLR
		23:16							ACMDIS	ACMEN
		15:8	CLEAR	PECRQ	PECDIS	PECEN	SMBDIS	SMBEN	HSDIS	HSEN
		7:0	SWRST	QUICK	SVDIS	SVEN	MSDIS	MSEN	STOP	START
0x0604	FLEX_TWI_MMR	31:24								NOAP
		23:16		DADR[6:0]						
		15:8				MREAD			IADRSZ[1:0]	
		7:0								
0x0608	FLEX_TWI_SMR	31:24	DATAMEN	SADR3EN	SADR2EN	SADR1EN				
		23:16		SADR[6:0]						
		15:8		MASK[6:0]						
		7:0	SNIFF	SCLWSDIS	BSEL	SADAT	SMHH	SMDA		NACKEN
0x060C	FLEX_TWI_IADR	31:24								
		23:16	IADR[23:16]							
		15:8	IADR[15:8]							
		7:0	IADR[7:0]							
0x0610	FLEX_TWI_CWGR	31:24			HOLD[5:0]					
		23:16				BRSRCCLK		CKDIV[2:0]		
		15:8	CHDIV[7:0]							
		7:0	CLDIV[7:0]							
0x0614 ... 0x061F	Reserved									
0x0620	FLEX_TWI_SR	31:24						SR	SDA	SCL
		23:16	LOCK		SMBHHM	SMBDAM	PECERR	TOUT	SMBAF	MCACK
		15:8	TXBUFE	RXBUFF	ENDTX	ENDRX	EOSACC	SCLWS	ARBLST	NACK
		7:0	UNRE	OVRE	GACC	SVACC	SVREAD	TXRDY	RXRDY	TXCOMP

© 2026 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0620	FLEX_TWI_SR (FIFO_ENABLED)	31:24						SR	SDA	SCL
		23:16	TXFLOCK		SMBHHM	SMBDAM	PECERR	TOUT	SMBAF	MCACK
		15:8	TXBUFE	RXBUFF	ENDTX	ENDRX	EOSACC	SCLWS	ARBLST	NACK
		7:0	UNRE	OVRE	GACC	SVACC	SVREAD	TXRDY	RXRDY	TXCOMP
0x0624	FLEX_TWI_IER	31:24								
		23:16			SMBHHM	SMBDAM	PECERR	TOUT		MCACK
		15:8	TXBUFE	RXBUFF	ENDTX	ENDRX	EOSACC	SCL_WS	ARBLST	NACK
		7:0	UNRE	OVRE	GACC	SVACC		TXRDY	RXRDY	TXCOMP
0x0628	FLEX_TWI_IDR	31:24								
		23:16			SMBHHM	SMBDAM	PECERR	TOUT		MCACK
		15:8	TXBUFE	RXBUFF	ENDTX	ENDRX	EOSACC	SCL_WS	ARBLST	NACK
		7:0	UNRE	OVRE	GACC	SVACC		TXRDY	RXRDY	TXCOMP
0x062C	FLEX_TWI_IMR	31:24								
		23:16			SMBHHM	SMBDAM	PECERR	TOUT		MCACK
		15:8	TXBUFE	RXBUFF	ENDTX	ENDRX	EOSACC	SCL_WS	ARBLST	NACK
		7:0	UNRE	OVRE	GACC	SVACC		TXRDY	RXRDY	TXCOMP
0x0630	FLEX_TWI_RHR	31:24								
		23:16								
		15:8				ASTATE[1:0]		PSTATE	SSTATE[1:0]	
		7:0	RXDATA[7:0]							
0x0630	FLEX_TWI_RHR (FIFO_ENABLED)	31:24	RXDATA3[7:0]							
		23:16	RXDATA2[7:0]							
		15:8	RXDATA1[7:0]							
		7:0	RXDATA0[7:0]							
0x0634	FLEX_TWI_THR	31:24								
		23:16								
		15:8								
		7:0	TXDATA[7:0]							
0x0634	FLEX_TWI_THR (FIFO_ENABLED)	31:24	TXDATA3[7:0]							
		23:16	TXDATA2[7:0]							
		15:8	TXDATA1[7:0]							
		7:0	TXDATA0[7:0]							
0x0638	FLEX_TWI_SMBTR	31:24	THMAX[7:0]							
		23:16	TLOWM[7:0]							
		15:8	TLOWS[7:0]							
		7:0					PRESC[3:0]			
0x063C ... 0x063F	Reserved									
0x0640	FLEX_TWI_ACR	31:24							NPEC	NDIR
		23:16	NDATAL[7:0]							
		15:8							PEC	DIR
		7:0	DATAL[7:0]							
0x0644	FLEX_TWI_FILTR	31:24								
		23:16								
		15:8						THRES[2:0]		
		7:0							PADFEN	FILT
0x0648 ... 0x064B	Reserved									
0x064C	FLEX_TWI_SWMR	31:24	DATAM[7:0]							
		23:16		SADR3[6:0]						
		15:8		SADR2[6:0]						
		7:0		SADR1[6:0]						
0x0650	FLEX_TWI_FMR	31:24			RXFTHRES[5:0]					
		23:16			TXFTHRES[5:0]					
		15:8								
		7:0			RXRDYM[1:0]				TXRDYM[1:0]	

© 2026 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0654	FLEX_TWI_FLR	31:24								
		23:16			RXFL[5:0]					
		15:8								
		7:0			TXFL[5:0]					
0x0658 ... 0x065F	Reserved									
0x0660	FLEX_TWI_FSR	31:24								
		23:16								
		15:8								
		7:0	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
0x0664	FLEX_TWI_FIER	31:24								
		23:16								
		15:8								
		7:0	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
0x0668	FLEX_TWI_FIDR	31:24								
		23:16								
		15:8								
		7:0	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
0x066C	FLEX_TWI_FIMR	31:24								
		23:16								
		15:8								
		7:0	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
0x0670 ... 0x06E3	Reserved									
0x06E4	FLEX_TWI_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0						WPCREN	WPITEN	WPEN
0x06E8	FLEX_TWI_WPSR	31:24	WPVSRC[23:16]							
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0								WPVS

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.1. FLEXCOM Mode Register

Name: FLEX_MR
Offset: 0x000
Reset: 0x00000001
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							OPMODE[1:0]	
Access							R/W	R/W
Reset							0	1

Bits 1:0 – OPMODE[1:0] FLEXCOM Operating Mode

Value	Name	Description
0	NO_COM	No communication
1	USART	All UART-related protocols are selected (RS232, RS485, IrDA, ISO7816, LIN,) SPI/TWI-related registers are not accessible and have no impact on IOs.
2	SPI	SPI operating mode is selected. USART/TWI related registers are not accessible and have no impact on IOs.
3	TWI	All TWI-related protocols are selected (TWI, SMBus). USART/SPI-related registers are not accessible and have no impact on IOs.

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.2. FLEXCOM Receive Holding Register

Name: FLEX_RHR
Offset: 0x010
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	RXDATA[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RXDATA[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – RXDATA[15:0] Receive Data
This register is a mirror of:
- USART Receive Holding Register (FLEX_US_RHR) if FLEX_MR.OPMODE field equals 1
- SPI Receive Data Register (FLEX_SPI_RDR) if FLEX_MR.OPMODE field equals 2
- TWI Transmit Holding Register (FLEX_TWI_RHR) if FLEX_MR.OPMODE field equals 3

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.3. FLEXCOM Transmit Holding Register

Name: FLEX_THR
Offset: 0x020
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	TXDATA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TXDATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – TXDATA[15:0] Transmit Data
This register is a mirror of:
- USART Transmit Holding Register (FLEX_US_THR) if FLEX_MR.OPMODE field equals 1
- SPI Transmit Data Register (FLEX_SPI_TDR) if FLEX_MR.OPMODE field equals 2
- TWI Transmit Holding Register (FLEX_TWI_THR) if FLEX_MR.OPMODE field equals 3

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.4. USART Control Register

Name: FLEX_US_CR
Offset: 0x200
Reset: –
Property: Write-only

This register can only be written if the WPCREN bit is cleared in the USART Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	FIFODIS	FIFOEN		REQCLR		TXFLCLR	RXFCLR	TXFCLR
Access	W	W		W		W	W	W
Reset	–	–		–		–	–	–

Bit	23	22	21	20	19	18	17	16
			LINWKUP	LINABT	RTSDIS	RTSEN		
Access			W	W	W	W		
Reset			–	–	–	–		

Bit	15	14	13	12	11	10	9	8
	RETTO	RSTNACK	RSTIT	SENDA	STTTO	STPBRK	STTBRK	RSTSTA
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	TXDIS	TXEN	RXDIS	RXEN	RSTTX	RSTRX		
Access	W	W	W	W	W	W		
Reset	–	–	–	–	–	–		

Bit 31 – FIFODIS FIFO Disable

Value	Description
0	No effect.
1	Disables the Transmit and Receive FIFOs.

Bit 30 – FIFOEN FIFO Enable

Value	Description
0	No effect.
1	Enables the Transmit and Receive FIFOs.

Bit 28 – REQCLR Request to Clear the Comparison Trigger
Asynchronous partial wakeup enabled:
0: No effect.
1: Clears the potential clock request currently issued by USART, thus the potential system wakeup is cancelled.
Asynchronous partial wakeup disabled:
0: No effect.
1: Restarts the comparison trigger to enable FLEX_US_RHR loading.

Bit 26 – TXFLCLR Transmit FIFO Lock CLEAR

Value	Description
0	No effect.
1	Clears the Transmit FIFO Lock.

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 25 – RXFCLR Receive FIFO Clear

Value	Description
0	No effect.
1	Empties the Receive FIFO.

Bit 24 – TXFCLR Transmit FIFO Clear

Value	Description
0	No effect.
1	Empties the Transmit FIFO.

Bit 21 – LINWKUP Send LIN Wakeup Signal

Value	Description
0	No effect:
1	Sends a wakeup signal on the LIN bus.

Bit 20 – LINABT Abort LIN Transmission

Value	Description
0	No effect.
1	Aborts the current LIN transmission.

Bit 19 – RTSDIS Request to Send Disable

Value	Description
0	No effect.
1	Drives the RTS pin to 0 if FLEX_US_MR.USART_MODE field = 2, else drives the RTS pin to 1 if FLEX_US_MR.USART_MODE field = 0.

Bit 18 – RTSEN Request to Send Enable

Value	Description
0	No effect.
1	Drives the RTS pin to 1 if FLEX_US_MR.USART_MODE field = 2, else drives the RTS pin to 0 if FLEX_US_MR.USART_MODE field = 0.

Bit 15 – RETTO Start Timeout Immediately

Value	Description
0	No effect
1	Immediately restarts timeout period.

Bit 14 – RSTNACK Reset Non Acknowledge

Value	Description
0	No effect
1	Resets FLEX_US_CSR.NACK.

Bit 13 – RSTIT Reset Iterations

Value	Description
0	No effect.
1	Resets FLEX_US_CSR.ITER. No effect if the ISO7816 is not enabled.

Bit 12 – SENDA Send Address

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	No effect.
1	In Multidrop mode only, the next character written to FLEX_US_THR is sent with the address bit set.

Bit 11 – STTTO Clear TIMEOUT Flag and Start Timeout After Next Character Received

Value	Description
0	No effect.
1	Starts waiting for a character before clocking the timeout counter. Immediately disables a timeout period in progress. Resets the FLEX_US_CSR.TIMEOUT status bit.

Bit 10 – STPBRK Stop Break

Value	Description
0	No effect.
1	Stops transmission of the break after a minimum of one character length and transmits a high level during 12-bit periods. No effect if no break is being transmitted.

Bit 9 – STTBRK Start Break

Value	Description
0	No effect.
1	Starts transmission of a break after the characters present in FLEX_US_THR and the Transmit Shift Register have been transmitted. No effect if a break is already being transmitted.

Bit 8 – RSTSTA Reset Status Bits

Value	Description
0	No effect.
1	Resets the PARE, FRAME, OVRE, MANE, LINBE, LINISFE, LINIPE, LINCE, LINSNRE, LINSTE, LINHTE, LINID, LINTC, LINBK, CMP and RXBRK in FLEX_US_CSR status bits, as well as the TXFEF, TXFFF, TXFTHF, RXFEF, RXFFF, RXFTHF, TXFPTEF, RXFPTEF in FLEX_US_FESR status bits.

Bit 7 – TXDIS Transmitter Disable

Value	Description
0	No effect.
1	Disables the transmitter.

Bit 6 – TXEN Transmitter Enable

Value	Description
0	No effect.
1	Enables the transmitter if TXDIS is 0.

Bit 5 – RXDIS Receiver Disable

Value	Description
0	No effect.
1	Disables the receiver.

Bit 4 – RXEN Receiver Enable

Value	Description
0	No effect.
1	Enables the receiver, if RXDIS is 0.

Bit 3 – RSTTX Reset Transmitter

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	No effect.
1	Resets the transmitter.

Bit 2 – RSTRX Reset Receiver

Value	Description
0	No effect.
1	Resets the receiver.

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.5. USART Mode Register

Name: FLEX_US_MR
Offset: 0x204
Reset: 0xC0000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the USART Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	ONEBIT	MODSYNC	MAN	FILTER		MAX_ITERATION[2:0]		
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	–	–	–	–		–	–	–

Bit	23	22	21	20	19	18	17	16
	INVDATA	VAR_SYNC	DSNACK	INACK	OVER	CLKO	MODE9	MSBF
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	CHMODE[1:0]		NBSTOP[1:0]		PAR[2:0]			SYNC
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	CHRL[1:0]		USCLKS[1:0]		USART_MODE[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	–	–	–	–	–	–	–	–

Bit 31 – ONEBIT Start Frame Delimiter Selector

Value	Description
0	Start frame delimiter is COMMAND or DATA SYNC.
1	Start frame delimiter is one bit.

Bit 30 – MODSYNC Manchester Synchronization Mode

Value	Description
0	The Manchester start bit is a 0 to 1 transition
1	The Manchester start bit is a 1 to 0 transition.

Bit 29 – MAN Manchester Encoder/Decoder Enable

Value	Description
0	Manchester encoder/decoder are disabled.
1	Manchester encoder/decoder are enabled.

Bit 28 – FILTER Receive Line Filter

Value	Description
0	The USART does not filter the receive line.
1	The USART filters the receive line using a three-sample filter (1/16-bit clock) (2 over 3 majority).

Bits 26:24 – MAX_ITERATION[2:0] Maximum Number of Automatic Iterations

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0-7	Defines the maximum number of iterations in mode ISO7816, protocol T = 0.

Bit 23 – INVDATA Inverted Data

Value	Description
0	The data field transmitted on TXD line is the same as the one written in FLEX_US_THR or the content read in FLEX_US_RHR is the same as RXD line. Normal mode of operation.
1	The data field transmitted on TXD line is inverted (voltage polarity only) compared to the value written in FLEX_US_THR or the content read in FLEX_US_RHR is inverted compared to what is received on RXD line (or ISO7816 IO line). Inverted mode of operation, useful for contactless card application. To be used with configuration bit MSBF.

Bit 22 – VAR_SYNC Variable Synchronization of Command/Data Sync Start Frame Delimiter

Value	Description
0	User defined configuration of command or data sync field depending on MODSYNC value.
1	The sync field is updated when a character is written into FLEX_US_THR.

Bit 21 – DSNACK Disable Successive NACK

The MAX_ITERATION field must be cleared if DSNACK is cleared.

Value	Description
0	NACK is sent on the ISO line as soon as a parity error occurs in the received character (unless INACK is set).
1	Successive parity errors are counted up to the value specified in the MAX_ITERATION field. These parity errors generate a NACK on the ISO line. As soon as this value is reached, no additional NACK is sent on the ISO line. The flag ITER is asserted.

Bit 20 – INACK Inhibit Non Acknowledge

Value	Description
0	The NACK is generated.
1	The NACK is not generated.

Bit 19 – OVER Oversampling Mode

Value	Description
0	16x Oversampling.
1	8x Oversampling.

Bit 18 – CLKO Clock Output Select

Value	Description
0	The USART does not drive the SCK pin (Synchronous Client mode or Asynchronous mode with external baud rate clock source).
1	The USART drives the SCK pin if USCLKS does not select the external clock SCK (USART Synchronous Host mode).

Bit 17 – MODE9 9-bit Character Length

Value	Description
0	CHRL defines character length.
1	9-bit character length.

Bit 16 – MSBF Bit Order

Value	Description
0	Least significant bit is sent/received first.
1	Most significant bit is sent/received first.

© 2026 Microchip Technology Inc. and its subsidiaries

Bits 15:14 – CHMODE[1:0] Channel Mode

Value	Name	Description
0	NORMAL	Normal mode
1	AUTOMATIC	Automatic Echo. Receiver input is connected to the TXD pin.
2	LOCAL_LOOPBACK	Local Loopback. Transmitter output is connected to the Receiver Input.
3	REMOTE_LOOPBACK	Remote Loopback. RXD pin is internally connected to the TXD pin.

Bits 13:12 – NBSTOP[1:0] Number of Stop Bits

Value	Name	Description
0	1_BIT	1 stop bit
1	1_5_BIT	1.5 stop bit (SYNC = 0) or reserved (SYNC = 1)
2	2_BIT	2 stop bits

Bits 11:9 – PAR[2:0] Parity Type

Value	Name	Description
0	EVEN	Even parity
1	ODD	Odd parity
2	SPACE	Parity forced to 0 (Space)
3	MARK	Parity forced to 1 (Mark)
4	NO	No parity
6	MULTIDROP	Multidrop mode

Bit 8 – SYNC Synchronous Mode Select

Value	Description
0	USART operates in Asynchronous mode (UART).
1	USART operates in Synchronous mode.

Bits 7:6 – CHRL[1:0] Character Length

Value	Name	Description
0	5_BIT	Character length is 5 bits
1	6_BIT	Character length is 6 bits
2	7_BIT	Character length is 7 bits
3	8_BIT	Character length is 8 bits

Bits 5:4 – USCLKS[1:0] Clock Selection

Value	Name	Description
0	MCK	Peripheral clock is selected
1	DIV	Peripheral clock divided (DIV = 8) is selected
2	GCLK	PMC generic clock is selected. If the SCK pin is driven (CLKO = 1), the CD field must be greater than 1.
3	SCK	External pin SCK is selected

Bits 3:0 – USART_MODE[3:0] USART Mode of Operation
PDC transfers are supported in all USART modes of operation.
Values not listed in the table below should be considered 'reserved'.

Value	Name	Description
0x0	NORMAL	Normal mode
0x1	RS485	RS485
0x2	HW_HANDSHAKING	Hardware handshaking
0x4	IS07816_T_0	IS07816 Protocol: T = 0
0x6	IS07816_T_1	IS07816 Protocol: T = 1
0x8	IRDA	IrDA

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Name	Description
0xA	LIN_MASTER	LIN Host mode
0xB	LIN_SLAVE	LIN Client mode
0xC	DATA16BIT_MASTER	16-bit data host
0xD	DATA16BIT_SLAVE	16-bit data client

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.6. USART Mode Register (OOK)

Name: FLEX_US_MR (OOK)
Offset: 0x204
Reset: 0xC0000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the USART Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
				FILTER				
Access				R/W				
Reset				–				

Bit	23	22	21	20	19	18	17	16
			OOKEN	OOKRXD	OVER	CLKO	MODE9	MSBF
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	CHMODE[1:0]		NBSTOP[1:0]		PAR[2:0]			SYNC
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	CHRL[1:0]		USCLKS[1:0]		USART_MODE[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	–	–	–	–	–	–	–	–

Bit 28 – FILTER Receive Line Filter

Value	Description
0	The USART does not filter the receive line.
1	The USART filters the receive line using a three-sample filter (1/16-bit clock) (2 over 3 majority).

Bit 21 – OOKEN OOK Modulation/Demodulation Enabled

Value	Description
0	The OOK modulation and demodulation are disabled.
1	The TXD line is OOK modulated and RXD line is OOK demodulated.

Bit 20 – OOKRXD OOK Demodulation Input Selection
OOKRXD is effective only if OOKEN=1.

Value	Description
0	The OOK demodulation uses the output of the analog comparator.
1	The OOK demodulation uses the RXD line.

Bit 19 – OVER Oversampling Mode

Value	Description
0	16x oversampling (recommended).
1	8x oversampling.

Bit 18 – CLKO Clock Output Select

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	The USART does not drive the SCK pin.
1	The USART drives the SCK pin if USCLKS does not select the external clock SCK.

Bit 17 – MODE9 9-bit Character Length

Value	Description
0	CHRL defines character length.
1	9-bit character length.

Bit 16 – MSBF Bit Order

Value	Description
0	Least significant bit is sent/received first.
1	Most significant bit is sent/received first.

Bits 15:14 – CHMODE[1:0] Channel Mode

Value	Name	Description
0	NORMAL	Normal mode
1	AUTOMATIC	Automatic Echo. Receiver input is connected to the TXD pin.
2	LOCAL_LOOPBACK	Local Loopback. Transmitter output is connected to the Receiver Input.
3	REMOTE_LOOPBACK	Remote Loopback. RXD pin is internally connected to the TXD pin.

Bits 13:12 – NBSTOP[1:0] Number of Stop Bits

Value	Name	Description
0	1_BIT	1 stop bit
1	1_5_BIT	1.5 stop bit (SYNC = 0) or reserved (SYNC = 1)
2	2_BIT	2 stop bits

Bits 11:9 – PAR[2:0] Parity Type

Value	Name	Description
0	EVEN	Even parity
1	ODD	Odd parity
2	SPACE	Parity forced to 0 (Space)
3	MARK	Parity forced to 1 (Mark)
4	NO	No parity
6	MULTIDROP	Multidrop mode

Bit 8 – SYNC Synchronous Mode Select

Value	Description
0	USART operates in Asynchronous mode.
1	USART operates in Synchronous mode.

Bits 7:6 – CHRL[1:0] Character Length

Value	Name	Description
0	5_BIT	Character length is 5 bits
1	6_BIT	Character length is 6 bits
2	7_BIT	Character length is 7 bits
3	8_BIT	Character length is 8 bits

Bits 5:4 – USCLKS[1:0] Clock Selection

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Name	Description
0	MCK	Peripheral clock is selected
1	DIV	Peripheral clock divided (DIV = 8) is selected
2	PCK	PMC programmable clock (PCK) is selected. If the SCK pin is driven (CLKO = 1), the CD field must be greater than 1.
3	SCK	Serial clock (SCK) is selected.

Bits 3:0 – USART_MODE[3:0] USART Mode of Operation
PDC transfers are supported in all USART modes of operation.

Value	Name	Description
0x0	NORMAL	Normal mode

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.7. USART Interrupt Enable Register

Name: FLEX_US_IER
Offset: 0x208
Reset: –
Property: Write-only

For LIN-specific configurations, see USART Interrupt Enable Register (LIN_MODE).

This register can only be written if the WPITEN bit is cleared in the USART Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
								MANE
Access								W
Reset								–

Bit	23	22	21	20	19	18	17	16
		CMP			CTSIC			
Access		W			W			
Reset		–			–			

Bit	15	14	13	12	11	10	9	8
			NACK	RXBUFF	TXBUFE	ITER	TXEMPTY	TIMEOUT
Access			W	W	W	W	W	W
Reset			–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	PARE	FRAME	OVRE	ENDTX	ENDRX	RXBRK	TXRDY	RXRDY
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 24 – MANE Manchester Error Interrupt Enable

Bit 22 – CMP Comparison Interrupt Enable

Bit 19 – CTSIC Clear to Send Input Change Interrupt Enable

Bit 13 – NACK Non Acknowledge Interrupt Enable

Bit 12 – RXBUFF Buffer Full Interrupt Enable (available in all USART modes of operation)

Bit 11 – TXBUFE Buffer Empty Interrupt Enable (available in all USART modes of operation)

Bit 10 – ITER Max number of Repetitions Reached Interrupt Enable

Bit 9 – TXEMPTY TXEMPTY Interrupt Enable

Bit 8 – TIMEOUT Timeout Interrupt Enable

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 7 – PARE Parity Error Interrupt Enable

Bit 6 – FRAME Framing Error Interrupt Enable

Bit 5 – OVRE Overrun Error Interrupt Enable

Bit 4 – ENDTX End of Transmit Interrupt Enable (available in all USART modes of operation)

Bit 3 – ENDRX End of Receive Transfer Interrupt Enable (available in all USART modes of operation)

Bit 2 – RXBRK Receiver Break Interrupt Enable

Bit 1 – TXRDY TXRDY Interrupt Enable

Bit 0 – RXRDY RXRDY Interrupt Enable

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.8. USART Interrupt Enable Register (LIN_MODE)

Name: FLEX_US_IER (LIN_MODE)
Offset: 0x208
Reset: –
Property: Write-only

This configuration is relevant only if USART_MODE = 0xA or 0xB in the USART Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
	LINHTE	LINSTE	LINSNRE	LINCE	LINIPE	LINISFE	LINBE	
Access	W	W	W	W	W	W	W	
Reset	–	–	–	–	–	–	–	

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	LINTC	LINID	LINBK	RXBUFF	TXBUFE		TXEMPTY	TIMEOUT
Access	W	W	W	W	W		W	W
Reset	–	–	–	–	–		–	–

Bit	7	6	5	4	3	2	1	0
	PARE	FRAME	OVRE	ENDTX	ENDRX		TXRDY	RXRDY
Access	W	W	W	W	W		W	W
Reset	–	–	–	–	–		–	–

Bit 31 – LINHTE LIN Header Timeout Error Interrupt Enable

Bit 30 – LINSTE LIN Synch Tolerance Error Interrupt Enable

Bit 29 – LINSNRE LIN Client Not Responding Error Interrupt Enable

Bit 28 – LINCE LIN Checksum Error Interrupt Enable

Bit 27 – LINIPE LIN Identifier Parity Interrupt Enable

Bit 26 – LINISFE LIN Inconsistent Synch Field Error Interrupt Enable

Bit 25 – LINBE LIN Bus Error Interrupt Enable

Bit 15 – LINTC LIN Transfer Completed Interrupt Enable

Bit 14 – LINID LIN Identifier Sent or LIN Identifier Received Interrupt Enable

Bit 13 – LINBK LIN Break Sent or LIN Break Received Interrupt Enable

Bit 12 – RXBUFF Buffer Full Interrupt Enable

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 11 – TXBUFE Buffer Empty Interrupt Enable

Bit 9 – TXEMPTY TXEMPTY Interrupt Enable

Bit 8 – TIMEOUT Timeout Interrupt Enable

Bit 7 – PARE Parity Error Interrupt Enable

Bit 6 – FRAME Framing Error Interrupt Enable

Bit 5 – OVRE Overrun Error Interrupt Enable

Bit 4 – ENDTX End of Transmit Interrupt Enable

Bit 3 – ENDRX End of Receive Transfer Interrupt Enable

Bit 1 – TXRDY TXRDY Interrupt Enable

Bit 0 – RXRDY RXRDY Interrupt Enable

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.9. USART Interrupt Disable Register

Name: FLEX_US_IDR
Offset: 0x20C
Reset: –
Property: Write-only

For LIN-specific configurations, see USART Interrupt Disable Register (LIN_MODE).

This register can only be written if the WPITEN bit is cleared in the USART Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
								MANE
Access								W
Reset								–

Bit	23	22	21	20	19	18	17	16
		CMP			CTSIC			
Access		W			W			
Reset		–			–			

Bit	15	14	13	12	11	10	9	8
			NACK	RXBUFF	TXBUFE	ITER	TXEMPTY	TIMEOUT
Access			W	W	W	W	W	W
Reset			–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	PARE	FRAME	OVRE	ENDTX	ENDRX	RXBRK	TXRDY	RXRDY
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 24 – MANE Manchester Error Interrupt Disable

Bit 22 – CMP Comparison Interrupt Disable

Bit 19 – CTSIC Clear to Send Input Change Interrupt Disable

Bit 13 – NACK Non Acknowledge Interrupt Disable

Bit 12 – RXBUFF Buffer Full Interrupt Disable (available in all USART modes of operation)

Bit 11 – TXBUFE Buffer Empty Interrupt Disable (available in all USART modes of operation)

Bit 10 – ITER Max Number of Repetitions Reached Interrupt Disable

Bit 9 – TXEMPTY TXEMPTY Interrupt Disable

Bit 8 – TIMEOUT Timeout Interrupt Disable

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 7 – PARE Parity Error Interrupt Disable

Bit 6 – FRAME Framing Error Interrupt Disable

Bit 5 – OVRE Overrun Error Interrupt Disable

Bit 4 – ENDTX End of Transmit Interrupt Disable (available in all USART modes of operation)

Bit 3 – ENDRX End of Receive Transfer Interrupt Disable (available in all USART modes of operation)

Bit 2 – RXBRK Receiver Break Interrupt Disable

Bit 1 – TXRDY TXRDY Interrupt Disable

Bit 0 – RXRDY RXRDY Interrupt Disable

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.10. USART Interrupt Disable Register (LIN_MODE)

Name: FLEX_US_IDR (LIN_MODE)
Offset: 0x20C
Reset: –
Property: Write-only

This configuration is relevant only if USART_MODE = 0xA or 0xB in the USART Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
	LINHTE	LINSTE	LINSNRE	LINCE	LINIPE	LINISFE	LINBE	
Access	W	W	W	W	W	W	W	
Reset	–	–	–	–	–	–	–	

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	LINTC	LINID	LINBK	RXBUFF	TXBUFE		TXEMPTY	TIMEOUT
Access	W	W	W	W	W		W	W
Reset	–	–	–	–	–		–	–

Bit	7	6	5	4	3	2	1	0
	PARE	FRAME	OVRE	ENDTX	ENDRX		TXRDY	RXRDY
Access	W	W	W	W	W		W	W
Reset	–	–	–	–	–		–	–

Bit 31 – LINHTE LIN Header Timeout Error Interrupt Disable

Bit 30 – LINSTE LIN Synch Tolerance Error Interrupt Disable

Bit 29 – LINSNRE LIN Client Not Responding Error Interrupt Disable

Bit 28 – LINCE LIN Checksum Error Interrupt Disable

Bit 27 – LINIPE LIN Identifier Parity Interrupt Disable

Bit 26 – LINISFE LIN Inconsistent Synch Field Error Interrupt Disable

Bit 25 – LINBE LIN Bus Error Interrupt Disable

Bit 15 – LINTC LIN Transfer Completed Interrupt Disable

Bit 14 – LINID LIN Identifier Sent or LIN Identifier Received Interrupt Disable

Bit 13 – LINBK LIN Break Sent or LIN Break Received Interrupt Disable

Bit 12 – RXBUFF Buffer Full Interrupt Disable

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 11 – TXBUFE Buffer Empty Interrupt Disable

Bit 9 – TXEMPTY TXEMPTY Interrupt Disable

Bit 8 – TIMEOUT Timeout Interrupt Disable

Bit 7 – PARE Parity Error Interrupt Disable

Bit 6 – FRAME Framing Error Interrupt Disable

Bit 5 – OVRE Overrun Error Interrupt Disable

Bit 4 – ENDTX End of Transmit Interrupt Disable

Bit 3 – ENDRX End of Receive Transfer Interrupt Disable

Bit 1 – TXRDY TXRDY Interrupt Disable

Bit 0 – RXRDY RXRDY Interrupt Disable

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.11. USART Interrupt Mask Register

Name: FLEX_US_IMR
Offset: 0x210
Reset: 0x00000000
Property: Read-only

For LIN-specific configurations, see USART Interrupt Mask Register (LIN_MODE).

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
								MANE
Access								R
Reset								0

Bit	23	22	21	20	19	18	17	16
		CMP			CTSIC			
Access		R			R			
Reset		0			0			

Bit	15	14	13	12	11	10	9	8
			NACK	RXBUFF	TXBUFE	ITER	TXEMPTY	TIMEOUT
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PARE	FRAME	OVRE	ENDTX	ENDRX	RXBRK	TXRDY	RXRDY
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 24 – MANE Manchester Error Interrupt Mask

Bit 22 – CMP Comparison Interrupt Mask

Bit 19 – CTSIC Clear to Send Input Change Interrupt Mask

Bit 13 – NACK Non Acknowledge Interrupt Mask

Bit 12 – RXBUFF Buffer Full Interrupt Mask (available in all USART modes of operation)

Bit 11 – TXBUFE Buffer Empty Interrupt Mask (available in all USART modes of operation)

Bit 10 – ITER Max Number of Repetitions Reached Interrupt Mask

Bit 9 – TXEMPTY TXEMPTY Interrupt Mask

Bit 8 – TIMEOUT Timeout Interrupt Mask

Bit 7 – PARE Parity Error Interrupt Mask

Bit 6 – FRAME Framing Error Interrupt Mask

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 5 – OVRE Overrun Error Interrupt Mask

Bit 4 – ENDTX End of Transmit Interrupt Mask (available in all USART modes of operation)

Bit 3 – ENDRX End of Receive Transfer Interrupt Mask (available in all USART modes of operation)

Bit 2 – RXBRK Receiver Break Interrupt Mask

Bit 1 – TXRDY TXRDY Interrupt Mask

Bit 0 – RXRDY RXRDY Interrupt Mask

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.12. USART Interrupt Mask Register (LIN_MODE)

Name: FLEX_US_IMR (LIN_MODE)
Offset: 0x210
Reset: 0x00000000
Property: Read-only

This configuration is relevant only if USART_MODE = 0xA or 0xB in the USART Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
	LINHTE	LINSTE	LINSNRE	LINCE	LINIPE	LINISFE	LINBE	
Access	R	R	R	R	R	R	R	
Reset	0	0	0	0	0	0	0	

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	LINTC	LINID	LINBK	RXBUFF	TXBUFE		TXEMPTY	TIMEOUT
Access	R	R	R	R	R		R	R
Reset	0	0	0	0	0		0	0

Bit	7	6	5	4	3	2	1	0
	PARE	FRAME	OVRE	ENDTX	ENDRX		TXRDY	RXRDY
Access	R	R	R	R	R		R	R
Reset	0	0	0	0	0		0	0

Bit 31 – LINHTE LIN Header Timeout Error Interrupt Mask

Bit 30 – LINSTE LIN Synch Tolerance Error Interrupt Mask

Bit 29 – LINSNRE LIN Client Not Responding Error Interrupt Mask

Bit 28 – LINCE LIN Checksum Error Interrupt Mask

Bit 27 – LINIPE LIN Identifier Parity Interrupt Mask

Bit 26 – LINISFE LIN Inconsistent Synch Field Error Interrupt Mask

Bit 25 – LINBE LIN Bus Error Interrupt Mask

Bit 15 – LINTC LIN Transfer Completed Interrupt Mask

Bit 14 – LINID LIN Identifier Sent or LIN Identifier Received Interrupt Mask

Bit 13 – LINBK LIN Break Sent or LIN Break Received Interrupt Mask

Bit 12 – RXBUFF Buffer Full Interrupt Mask

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 11 – TXBUFE Buffer Empty Interrupt Mask

Bit 9 – TXEMPTY TXEMPTY Interrupt Mask

Bit 8 – TIMEOUT Timeout Interrupt Mask

Bit 7 – PARE Parity Error Interrupt Mask

Bit 6 – FRAME Framing Error Interrupt Mask

Bit 5 – OVRE Overrun Error Interrupt Mask

Bit 4 – ENDTX End of Transmit Interrupt Mask

Bit 3 – ENDRX End of Receive Transfer Interrupt Mask

Bit 1 – TXRDY TXRDY Interrupt Mask

Bit 0 – RXRDY RXRDY Interrupt Mask

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.13. USART Channel Status Register

Name: FLEX_US_CSR
Offset: 0x214
Reset: 0x00001818
Property: Read-only

For LIN-specific configurations, see USART Channel Status Register (LIN_MODE).

Bit	31	30	29	28	27	26	25	24
								MANE
Access								R
Reset								–

Bit	23	22	21	20	19	18	17	16
	CTS	CMP			CTSIC			
Access	R	R			R			
Reset	–	–			–			

Bit	15	14	13	12	11	10	9	8
			NACK	RXBUFF	TXBUFE	ITER	TXEMPTY	TIMEOUT
Access			R	R	R	R	R	R
Reset			–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	PARE	FRAME	OVRE	ENDTX	ENDRX	RXBRK	TXRDY	RXRDY
Access	R	R	R	R	R	R	R	R
Reset	–	–	–	–	–	–	–	–

Bit 24 – MANE Manchester Error

Value	Description
0	No Manchester error has been detected since the last RSTSTA command was issued.
1	At least one Manchester error has been detected since the last RSTSTA command was issued.

Bit 23 – CTS Image of CTS Input

Value	Description
0	CTS input is driven low.
1	CTS input is driven high.

Bit 22 – CMP Comparison Status

Value	Description
0	No received character matched the comparison criteria programmed in VAL1, VAL2 fields and CMPPAR bit in since the last RSTSTA command was issued.
1	A received character matched the comparison criteria since the last RSTSTA command was issued.

Bit 19 – CTSIC Clear to Send Input Change Flag

Value	Description
0	No input change has been detected on the CTS pin since the last read of FLEX_US_CSR.
1	At least one input change has been detected on the CTS pin since the last read of FLEX_US_CSR.

Bit 13 – NACK Non Acknowledge Interrupt

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	Non acknowledge has not been detected since the last RSTNACK.
1	At least one non acknowledge has been detected since the last RSTNACK.

Bit 12 – RXBUFF RX Buffer Full (cleared by writing FLEX_US_RCR or FLEX_US_RNCR)
FLEX_US_RCR and FLEX_US_RNCR are PDC registers.

Value	Description
0	FLEX_US_RCR or FLEX_US_RNCR has a value other than 0.
1	Both FLEX_US_RCR and FLEX_US_RNCR have a value of 0.

Bit 11 – TXBUFE TX Buffer Empty (cleared by writing FLEX_US_TCR or FLEX_US_TNCR)
FLEX_US_TCR and FLEX_US_TNCR are PDC registers.

Value	Description
0	FLEX_US_TCR or FLEX_US_TNCR has a value other than 0.
1	Both FLEX_US_TCR and FLEX_US_TNCR have a value of 0

Bit 10 – ITER Max Number of Repetitions Reached

Value	Description
0	Maximum number of repetitions has not been reached since the last RSTIT command was issued.
1	Maximum number of repetitions has been reached since the last RSTIT command was issued.

Bit 9 – TXEMPTY Transmitter Empty (cleared by writing FLEX_US_THR)

Value	Description
0	There are characters in either FLEX_US_THR or the Transmit Shift Register, or the transmitter is disabled.
1	There are no characters in FLEX_US_THR, nor in the Transmit Shift Register.

Bit 8 – TIMEOUT Receiver Timeout

Value	Description
0	There has not been a timeout since the last Start Timeout command (FLEX_US_CR.STTTO) or the Timeout Register is 0.
1	There has been a timeout since the last Start Timeout command (FLEX_US_CR.STTTO).

Bit 7 – PARE Parity Error

Value	Description
0	No parity error has been detected since the last RSTSTA command was issued.
1	At least one parity error has been detected since the last RSTSTA command was issued.

Bit 6 – FRAME Framing Error

Value	Description
0	No stop bit has been detected low since the last RSTSTA command was issued.
1	At least one stop bit has been detected low since the last RSTSTA command was issued.

Bit 5 – OVRE Overrun Error

Value	Description
0	No overrun error has occurred since the last RSTSTA command was issued.
1	At least one overrun error has occurred since the last RSTSTA command was issued.

Bit 4 – ENDTX End of TX Buffer (cleared by writing FLEX_US_TCR or FLEX_US_TNCR)
FLEX_US_TCR and FLEX_US_TNCR are PDC registers.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	The Transmit Counter Register has not reached 0 since the last write in FLEX_US_TCR or FLEX_US_TNCR.
1	The Transmit Counter Register has reached 0 since the last write in FLEX_US_TCR or FLEX_US_TNCR.

Bit 3 – ENDRX End of RX Buffer (cleared by writing FLEX_US_RCR or FLEX_US_RNCR)
FLEX_US_RCR and FLEX_US_RNCR are PDC registers.

Value	Description
0	The Receive Counter Register has not reached 0 since the last write in FLEX_US_RCR or FLEX_US_RNCR.
1	The Receive Counter Register has not reached 0 since the last write in FLEX_US_RCR or FLEX_US_RNCR.

Bit 2 – RXBRK Break Received/End of Break

Value	Description
0	No break received or end of break detected since the last RSTSTA command was issued.
1	Break received or end of break detected since the last RSTSTA command was issued.

Bit 1 – TXRDY Transmitter Ready (cleared by writing FLEX_US_THR)
When FIFOs are disabled:
0: A character in FLEX_US_THR is waiting to be transferred to the Transmit Shift Register, or an STTBRK command has been requested, or the transmitter is disabled. As soon as the transmitter is enabled, TXRDY becomes 1.
1: There is no character in FLEX_US_THR.
When FIFOs are enabled:
0: Transmit FIFO is full and cannot accept more data.
1: Transmit FIFO is not full; one or more data can be written according to TXRDYM field configuration.
TXRDY behavior with FIFO enabled is illustrated in TXEMPTY, TXRDY and RXRDY Behavior.

Bit 0 – RXRDY Receiver Ready (cleared by reading FLEX_US_RHR)
When FIFOs are disabled:
0: No complete character has been received since the last read of FLEX_US_RHR or the receiver is disabled. If characters were received when the receiver was disabled, RXRDY changes to 1 when the receiver is enabled.
1: At least one complete character has been received and FLEX_US_RHR has not yet been read.
When FIFOs are enabled:
0: Receive FIFO is empty; no data to read
1: At least one unread data is in the Receive FIFO
RXRDY behavior with FIFO enabled is illustrated in TXEMPTY, TXRDY and RXRDY Behavior.

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.14. USART Channel Status Register (LIN_MODE)

Name: FLEX_US_CSR (LIN_MODE)
Offset: 0x214
Reset: –
Property: Read-only

This configuration is relevant only if USART_MODE = 0xA or 0xB in the USART Mode Register.

Bit	31	30	29	28	27	26	25	24
	LINHTE	LINSTE	LINSNRE	LINCE	LINIPE	LINISFE	LINBE	
Access	R	R	R	R	R	R	R	
Reset	–	–	–	–	–	–	–	

Bit	23	22	21	20	19	18	17	16
	LINBLS							
Access	R							
Reset	–							

Bit	15	14	13	12	11	10	9	8
	LINTC	LINID	LINBK	RXBUFF	TXBUFE		TXEMPTY	TIMEOUT
Access	R	R	R	R	R		R	R
Reset	–	–	–	–	–		–	–

Bit	7	6	5	4	3	2	1	0
	PARE	FRAME	OVRE	ENDTX	ENDRX		TXRDY	RXRDY
Access	R	R	R	R	R		R	R
Reset	–	–	–	–	–		–	–

Bit 31 – LINHTE LIN Header Timeout Error

Value	Description
0	No LIN header timeout error has been detected since the last RSTSTA command was issued.
1	A LIN header timeout error has been detected since the last RSTSTA command was issued.

Bit 30 – LINSTE LIN Synch Tolerance Error

Value	Description
0	No LIN synch tolerance error has been detected since the last RSTSTA command was issued.
1	A LIN synch tolerance error has been detected since the last RSTSTA command was issued.

Bit 29 – LINSNRE LIN Client Not Responding Error

Value	Description
0	No LIN client not responding error has been detected since the last RSTSTA command was issued.
1	A LIN client not responding error has been detected since the last RSTSTA command was issued.

Bit 28 – LINCE LIN Checksum Error

Value	Description
0	No LIN checksum error has been detected since the last RSTSTA command was issued.
1	A LIN checksum error has been detected since the last RSTSTA command was issued.

Bit 27 – LINIPE LIN Identifier Parity Error

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	No LIN identifier parity error has been detected since the last RSTSTA command was issued.
1	A LIN identifier parity error has been detected since the last RSTSTA command was issued.

Bit 26 – LINISFE LIN Inconsistent Synch Field Error

Value	Description
0	No LIN inconsistent synch field error has been detected since the last RSTSTA
1	The USART is configured as a client node and a LIN Inconsistent synch field error has been detected since the last RSTSTA command was issued.

Bit 25 – LINBE LIN Bit Error

Value	Description
0	No bit error has been detected since the last RSTSTA command was issued.
1	A bit error has been detected since the last RSTSTA command was issued.

Bit 23 – LINBLS LIN Bus Line Status

Value	Description
0	LIN bus line is set to 0.
1	LIN bus line is set to 1.

Bit 15 – LINTC LIN Transfer Completed

Value	Description
0	The USART is idle or a LIN transfer is ongoing.
1	A LIN transfer has been completed since the last RSTSTA command was issued.

Bit 14 – LINID LIN Identifier Sent or LIN Identifier Received
If USART operates in LIN Host mode (USART_MODE = 0xA):
0: No LIN identifier has been sent since the last RSTSTA command was issued.
1: At least one LIN identifier has been sent since the last RSTSTA command was issued.
If USART operates in LIN Client mode (USART_MODE = 0xB):
0: No LIN identifier has been received since the last RSTSTA command was issued.
1: At least one LIN identifier has been received since the last RSTSTA.

Bit 13 – LINBK LIN Break Sent or LIN Break Received
Applicable if USART operates in LIN Host mode (USART_MODE = 0xA):
0: No LIN break has been sent since the last RSTSTA command was issued.
1: At least one LIN break has been sent since the last RSTSTA.
If USART operates in LIN Client mode (USART_MODE = 0xB):
0: No LIN break has received sent since the last RSTSTA command was issued.
1: At least one LIN break has been received since the last RSTSTA command was issued.

Bit 12 – RXBUFF RX Buffer Full (cleared by writing FLEX_US_RCR or FLEX_US_RNCR)
FLEX_US_RCR and FLEX_US_RNCR are PDC registers.

Value	Description
0	FLEX_US_TCR or US_TNCR has a value other than 0.
1	Both US_TCR and US_TNCR have a value of 0.

Bit 11 – TXBUFE TX Buffer Empty (cleared by writing FLEX_US_TCR or FLEX_US_TNCR)
FLEX_US_TCR and FLEX_US_TNCR are PDC registers.

Value	Description
0	FLEX_US_TCR or FLEX_US_TNCR has a value other than 0.
1	Both FLEX_US_TCR and FLEX_US_TNCR have a value of 0.

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 9 – TXEMPTY Transmitter Empty (cleared by writing FLEX_US_THR)

Value	Description
0	There are characters in either FLEX_US_THR or the Transmit Shift Register, or the transmitter is disabled.
1	There are no characters in FLEX_US_THR, nor in the Transmit Shift Register.

Bit 8 – TIMEOUT Receiver Timeout

Value	Description
0	There has not been a timeout since the last start timeout command (FLEX_US_CR.STTTO) or the Timeout Register is 0.
1	There has been a timeout since the last start timeout command (FLEX_US_CR.STTTO).

Bit 7 – PARE Parity Error

Value	Description
0	No parity error has been detected since the last RSTSTA command was issued.
1	At least one parity error has been detected since the last RSTSTA command was issued.

Bit 6 – FRAME Framing Error

Value	Description
0	No stop bit has been detected low since the last RSTSTA command was issued.
1	At least one stop bit has been detected low since the last RSTSTA command was issued.

Bit 5 – OVRE Overrun Error

Value	Description
0	No overrun error has occurred since the last RSTSTA command was issued.
1	At least one overrun error has occurred since the last RSTSTA command was issued.

Bit 4 – ENDTX End of TX Buffer (cleared by writing FLEX_US_TCR or FLEX_US_TNCR)
FLEX_US_TCR and FLEX_US_TNCR are PDC registers.

Value	Description
0	The Transmit Counter Register has not reached 0 since the last write in FLEX_US_TCR or FLEX_US_TNCR.
1	The Transmit Counter Register has reached 0 since the last write in FLEX_US_TCR or FLEX_US_TNCR.

Bit 3 – ENDRX End of RX Buffer (cleared by writing FLEX_US_RCR or FLEX_US_RNCR)
FLEX_US_RCR and FLEX_US_RNCR are PDC registers.

Value	Description
0	The Receive Counter Register has not reached 0 since the last write in FLEX_US_RCR or FLEX_US_RNCR.
1	The Receive Counter Register has not reached 0 since the last write in FLEX_US_RCR or FLEX_US_RNCR.

Bit 1 – TXRDY Transmitter Ready (cleared by writing FLEX_US_THR)

Value	Description
0	A character in FLEX_US_THR is waiting to be transferred to the Transmit Shift Register, or the transmitter is disabled. As soon as the transmitter is enabled, TXRDY becomes 1.
1	There is no character in FLEX_US_THR.

Bit 0 – RXRDY Receiver Ready (cleared by reading FLEX_US_RHR)

Value	Description
0	No complete character has been received since the last read of FLEX_US_RHR or the receiver is disabled. If characters were being received when the receiver was disabled, RXRDY changes to 1 when the receiver is enabled.
1	At least one complete character has been received and FLEX_US_RHR has not yet been read.

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.15. USART Receive Holding Register

Name: FLEX_US_RHR
Offset: 0x218
Reset: 0x00000000
Property: Read-only

If FIFO is enabled (FLEX_US_CR.FIFOEN=1), a byte access on FLEX_SPI_TDR reads one byte (FLEX_US_RHR.MODE9=0), see FIFO Single Data Access for details.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	RXSYNH							RXCHR[8]
Access	R							R
Reset	0							0

Bit	7	6	5	4	3	2	1	0
	RXCHR[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 15 – RXSYNH Received Sync

Value	Description
0	Last character received is a data.
1	Last character received is a command.

Bits 8:0 – RXCHR[8:0] Received Character
Last character received if RXRDY is set.

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.16. USART Receive Holding Register (FIFO Multi Data)

Name: FLEX_US_RHR (FIFO_MULTI_DATA)
Offset: 0x218
Reset: 0x00000000
Property: Read-only

To read multi-data in a single access, the FIFO must be enabled (FLEX_US_CR.FIFOEN=1) and FLEX_US_MR.MODE9=0. The access type (byte, halfword or word) determines the number of data written in a single access (1, 2, 4), see FIFO Multiple Data Access for details.

Bit	31	30	29	28	27	26	25	24
	RXCHR3[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RXCHR2[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RXCHR1[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RXCHR0[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0:7, 8:15, 16:23, 24:31 – RXCHRx Received Character
First unread character in the Receive FIFO if RXRDY is set.

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.17. USART Transmit Holding Register

Name: FLEX_US_THR
Offset: 0x21C
Reset: –
Property: Write-only

If FIFO is enabled (FLEX_US_CR.FIFOEN=1), a byte access on FLEX_SPI_TDR writes one byte (FLEX_US_MR.MODE9=0), see FIFO Single Data Access for details.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	TXSYNH							TXCHR[8]
Access	W							W
Reset	–							–

Bit	7	6	5	4	3	2	1	0
	TXCHR[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 15 – TXSYNH Sync Field to be Transmitted

Value	Description
0	The next character sent is encoded as a data. Start frame delimiter is DATA SYNC.
1	The next character sent is encoded as a command. Start frame delimiter is COMMAND SYNC.

Bits 8:0 – TXCHR[8:0] Character to be Transmitted
The next character to be transmitted after the current character if TXRDY is not set.

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.18. USART Transmit Holding Register (FIFO Multi Data)

Name: FLEX_US_THR (FIFO_MULTI_DATA)
Offset: 0x21C
Reset: –
Property: Write-only

To write multi-data in a single access, the FIFO must be enabled (FLEX_US_CR.FIFOEN=1) and FLEX_US_MR.MODE9=0. The access type (byte, halfword or word) determines the number of data written in a single access (1, 2 or 4), see FIFO Multiple Data Access for details.

Bit	31	30	29	28	27	26	25	24
	TXCHR3[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	TXCHR2[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	TXCHR1[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	TXCHR0[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0:7, 8:15, 16:23, 24:31 – TXCHRx Character to be Transmitted
Next character to be transmitted.

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.19. USART Baud Rate Generator Register

Name: FLEX_US_BRGR
Offset: 0x220
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the USART Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						FP[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
	CD[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 18:16 – FP[2:0] Fractional Part

> **WARNING** When the value of field FP is greater than 0, the SCK (oversampling clock) generates nonconstant duty cycles. The SCK high duration is increased by "selected clock" period from time to time. The duty cycle depends on the value of the CD field.

Value	Description
0	Fractional divider is disabled.
1–7	Baud rate resolution, defined by FP × 1/8.

Bits 15:0 – CD[15:0] Clock Divider

CD	USART_MODE ≠ ISO7816			USART_MODE = ISO7816
	SYNC = 0		SYNC = 1	
	OVER = 0	OVER = 1		
0	Baud Rate Clock disabled			
1 to 65535	CD = Selected Clock / (16 × Baud Rate)	CD = Selected Clock / (8 × Baud Rate)	CD = Selected Clock / Baud Rate	CD = Selected Clock / (FI_DI_RATIO × Baud Rate)

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.20. USART Receiver Timeout Register

Name: FLEX_US_RTOR
Offset: 0x224
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the USART Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								TO[16]
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
	TO[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TO[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 16:0 – TO[16:0] Timeout Value
The TO field size is limited to 8 bits if the ISO7816 logic is not implemented on some instances of FLEXCOM. The ISO7816 logic is implemented if it is possible to write FLEX_US_MR.MAX_ITERATIONS=1 (a read operation must be performed after the write operation to check that MAX_ITERATIONS equals 1).

Value	Description
0	The receiver timeout is disabled.
1–131071	The receiver timeout is enabled and the timeout delay is TO × bit period.

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.21. USART Transmitter Timeguard Register

Name: FLEX_US_TTGR
Offset: 0x228
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the USART Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	TG[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – TG[7:0] Timeguard Value

Value	Description
0	The transmitter timeguard is disabled.
1–255	The transmitter timeguard is enabled and TG is timeguard delay / bit period.

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.22. USART FI DI RATIO Register

Name: FLEX_US_FIDI
Offset: 0x240
Reset: 0x174
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the USART Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	FI_DI_RATIO[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	1

Bit	7	6	5	4	3	2	1	0
	FI_DI_RATIO[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	1	1	0	1	0	0

Bits 15:0 – FI_DI_RATIO[15:0] FI Over DI Ratio Value
If US_MR.OOKEN=1, FI_DI_RATIO defines the carrier frequency for modulation of logical 0 in transmitted data stream. See OOK Modulation and Demodulation.

Value	Description
0	If ISO7816 mode is selected, the baud rate generator generates no signal.
1–2	Do not use.
3–2047	If ISO7816 mode is selected, the baud rate is the clock provided on SCK divided by FI_DI_RATIO.

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.23. USART Number of Errors Register

Name: FLEX_US_NER
Offset: 0x244
Reset: 0x00000000
Property: Read-only

This register is relevant only if USART_MODE = 0x4 or 0x6 in the USART Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	NB_ERRORS[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	–	–	–	–	–	–	–	–

Bits 7:0 – NB_ERRORS[7:0] Number of Errors
Total number of errors that occurred during an ISO7816 transfer. This register automatically clears when read.

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.24. USART IrDA FILTER Register

Name: FLEX_US_IF
Offset: 0x24C
Reset: 0x00000000
Property: Read/Write

This register is relevant only if USART_MODE = 0x8 in the USART Mode Register.

This register can only be written if the WPEN bit is cleared in the USART Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	IRDA_FILTER[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – IRDA_FILTER[7:0] IrDA Filter

The IRDA_FILTER value must be defined to meet the following criteria:

$t_{peripheral\ clock} \times (IRDA\_FILTER + 3) < 1.41\ \mu s$

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.25. USART Manchester Configuration Register

Name: FLEX_US_MAN
Offset: 0x250
Reset: 0xB0011004
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the USART Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	RXIDLEV	DRIFT	ONE	RX_MPOL			RX_PP[1:0]	
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	1	0	1	1			0	0

Bit	23	22	21	20	19	18	17	16
					RX_PL[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	1

Bit	15	14	13	12	11	10	9	8
				TX_MPOL			TX_PP[1:0]	
Access				R/W			R/W	R/W
Reset				1			0	0

Bit	7	6	5	4	3	2	1	0
					TX_PL[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	1	0	0

Bit 31 – RXIDLEV Receiver Idle Value

Value	Description
0	Receiver line idle value is 0.
1	Receiver line idle value is 1.

Bit 30 – DRIFT Drift Compensation

Value	Description
0	The USART cannot recover from an important clock drift.
1	The USART can recover from clock drift. The 16X Clock mode must be enabled.

Bit 29 – ONE Must Be Set to 1
Bit 29 must always be set to 1 when programming the FLEX_US_MAN register.

Bit 28 – RX_MPOL Receiver Manchester Polarity

Value	Description
0	Logic zero is coded as a zero-to-one transition, Logic one is coded as a one-to-zero transition.
1	Logic zero is coded as a one-to-zero transition, Logic one is coded as a zero-to-one transition.

Bits 25:24 – RX_PP[1:0] Receiver Preamble Pattern detected
The following values assume that RX_MPOL field is not set:

Value	Name	Description
0	ALL_ONE	The preamble is composed of '1's.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Name	Description
1	ALL_ZERO	The preamble is composed of '0's.
2	ZERO_ONE	The preamble is composed of '01's.
3	ONE_ZERO	The preamble is composed of '10's.

Bits 19:16 – RX_PL[3:0] Receiver Preamble Length

Value	Description
0	The receiver preamble pattern detection is disabled.
1–15	The detected preamble length is RX_PL × Bit Period.

Bit 12 – TX_MPOL Transmitter Manchester Polarity

Value	Description
0	Logic zero is coded as a zero-to-one transition, Logic one is coded as a one-to-zero transition.
1	Logic zero is coded as a one-to-zero transition, Logic one is coded as a zero-to-one transition.

Bits 9:8 – TX_PP[1:0] Transmitter Preamble Pattern
The following values assume that TX_MPOL field is not set:

Value	Name	Description
0	ALL_ONE	The preamble is composed of '1's.
1	ALL_ZERO	The preamble is composed of '0's.
2	ZERO_ONE	The preamble is composed of '01's.
3	ONE_ZERO	The preamble is composed of '10's.

Bits 3:0 – TX_PL[3:0] Transmitter Preamble Length

Value	Description
0	The transmitter preamble pattern generation is disabled.
1–15	The preamble length is TX_PL × Bit Period.

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.26. USART LIN Mode Register

Name: FLEX_US_LINMR
Offset: 0x254
Reset: 0x00000000
Property: Read/Write

This register is relevant only if USART_MODE = 0xA or 0xB in the USART Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							SYNCDIS	PDCM
Access							R/W	R/W
Reset							0	0

Bit	15	14	13	12	11	10	9	8
	DLC[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	WKUPTYP	FSDIS	DLM	CHKTYP	CHKDIS	PARDIS	NACT[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 17 – SYNCDIS Synchronization Disable

Value	Description
0	The synchronization procedure is performed in LIN client node configuration.
1	The synchronization procedure is not performed in LIN client node configuration.

Bit 16 – PDCM PDC Mode

Value	Description
0	The LIN mode register FLEX_US_LINMR is not written by the PDC.
1	The LIN mode register FLEX_US_LINMR (excepting that flag) is written by the PDC.

Bits 15:8 – DLC[7:0] Data Length Control

Value	Description
0–255	Defines the response data length if DLM = 0, in that case the response data length is equal to DLC+1 bytes.

Bit 7 – WKUPTYP Wake-up Signal Type

Value	Description
0	Setting the LINWKUP bit in the control register sends a LIN 2.0 wake-up signal.
1	Setting the LINWKUP bit in the control register sends a LIN 1.3 wake-up signal.

Bit 6 – FSDIS Frame Slot Mode Disable

Value	Description
0	The Frame Slot mode is enabled.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
1	The Frame Slot mode is disabled.

Bit 5 – DLM Data Length Mode

Value	Description
0	The response data length is defined by the DLC field of this register.
1	The response data length is defined by the bits 5 and 6 of the identifier (FLEX_US_LINIR.IDCHR).

Bit 4 – CHKTYP Checksum Type

Value	Description
0	LIN 2.0 “enhanced” checksum
1	LIN 1.3 “classic” checksum

Bit 3 – CHKDIS Checksum Disable

Value	Description
0	In host node configuration, the checksum is computed and sent automatically. In client node configuration, the checksum is checked automatically.
1	Whatever the node configuration is, the checksum is not computed/sent and it is not checked.

Bit 2 – PARDIS Parity Disable

Value	Description
0	In host node configuration, the identifier parity is computed and sent automatically. In host node and client node configuration, the parity is checked automatically.
1	Whatever the node configuration is, the Identifier parity is not computed/sent and it is not checked.

Bits 1:0 – NACT[1:0] LIN Node Action
Values which are not listed in the table must be considered as “reserved”.

Value	Name	Description
0	PUBLISH	The USART transmits the response.
1	SUBSCRIBE	The USART receives the response.
2	IGNORE	The USART does not transmit and does not receive the response.

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.27. USART LIN Identifier Register

Name: FLEX_US_LINIR
Offset: 0x258
Reset: 0x00000000
Property: Read/Write

Write is possible only in LIN host node configuration.

This register is relevant only if USART_MODE = 0xA or 0xB in USART Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	IDCHR[7:0]							
Access	R/W-R	R/W-R	R/W-R	R/W-R	R/W-R	R/W-R	R/W-R	R/W-R
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – IDCHR[7:0] Identifier Character

If USART_MODE = 0xA (host node configuration):

- IDCHR is Read/Write and its value is the identifier character to be transmitted.

If USART_MODE = 0xB (client node configuration):

- IDCHR is Read-only and its value is the last identifier character that has been received.

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.28. USART LIN Baud Rate Register

Name: FLEX_US_LINBRR
Offset: 0x25C
Reset: 0x00000000
Property: Read-only

This register is relevant only if USART_MODE = 0xA or 0xB in USART Mode Register.

Returns the baud rate value after the synchronization process completion.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						LINFP[2:0]		
Access						R	R	R
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
	LINCD[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	LINCD[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 18:16 – LINFP[2:0] Fractional Part after Synchronization

Bits 15:0 – LINCD[15:0] Clock Divider after Synchronization

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.29. USART Comparison Register

Name: FLEX_US_CMPR
Offset: 0x290
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the USART Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
								VAL2[8]
Access								R/W
Reset								0

Bit	23	22	21	20	19	18	17	16
	VAL2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
		CMPPAR	CMPMODE[1:0]					VAL1[8]
Access		R/W	R/W	R/W				R/W
Reset		0	0	0				0

Bit	7	6	5	4	3	2	1	0
	VAL1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 24:16 – VAL2[8:0] Second Comparison Value for Received Character

Value	Description
0–511	The received character must be lower than or equal to the value of VAL2 and higher than or equal to VAL1 to set the FLEX_US_CSR.CMP flag. If asynchronous partial wakeup is enabled in PMC_SLPWK_ER, the UART requests a system wakeup if condition is met.

Bit 14 – CMPPAR Compare Parity

Value	Description
0	The parity is not checked and a bad parity cannot prevent from waking up the system.
1	The parity is checked and a matching condition on data can be cancelled by an error on parity bit, so no wakeup is performed.

Bits 13:12 – CMPMODE[1:0] Comparison Mode

Value	Name	Description
0	FLAG_ONLY	Any character is received and comparison function drives CMP flag.
1	START_CONDITION	Comparison condition must be met to start reception.
2	FILTER	Comparison must be met to receive the current data only

Bits 8:0 – VAL1[8:0] First Comparison Value for Received Character

Value	Description
0–511	The received character must be higher than or equal to the value of VAL1 and lower than or equal to VAL2 to set the FLEX_US_CSR.CMP flag. If asynchronous partial wakeup is enabled in PMC_SLPWK_ER, the UART requests a system wakeup if the condition is met.

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.30. USART FIFO Mode Register

Name: FLEX_US_FMR
Offset: 0x2A0
Reset: 0x00000000
Property: Read/Write

This register reads '0' if the FIFO is disabled (see FLEX_US_CR to enable/disable the internal FIFO).

Bit	31	30	29	28	27	26	25	24
			RXFTHRES2[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
			RXFTHRES[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
			TXFTHRES[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FRTSC		RXRDYM[1:0]				TXRDYM[1:0]	
Access	R/W		R/W	R/W			R/W	R/W
Reset	0		0	0			0	0

Bits 29:24 – RXFTHRES2[5:0] Receive FIFO Threshold 2

Value	Description
0–8	Defines the Receive FIFO threshold 2 value (number of bytes). The FLEX_US_FESR.RXFTHF2 flag will be set when Receive FIFO goes from "above" threshold state to "equal to or below" threshold state.

Bits 21:16 – RXFTHRES[5:0] Receive FIFO Threshold

Value	Description
0–8	Defines the Receive FIFO threshold value (number of bytes). The FLEX_US_FESR.RXFTHF flag will be set when Receive FIFO goes from "below" threshold state to "equal to or above" threshold state.

Bits 13:8 – TXFTHRES[5:0] Transmit FIFO Threshold

Value	Description
0–8	Defines the Transmit FIFO threshold value (number of bytes). The FLEX_US_FESR.TXFTHF flag will be set when Transmit FIFO goes from "above" threshold state to "equal to or below" threshold state.

Bit 7 – FRTSC FIFO RTS Pin Control enable (Hardware Handshaking mode only)
See Hardware Handshaking for details.

Value	Description
0	RTS pin is not controlled by Receive FIFO thresholds.
1	RTS pin is controlled by Receive FIFO thresholds.

Bits 5:4 – RXRDYM[1:0] Receiver Ready Mode
If FIFOs are enabled, the FLEX_US_CSR.RXRDY flag behaves as follows.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Name	Description
0	ONE_DATA	RXRDY will be at level '1' when at least one unread data is in the receive FIFO. When DMA is enabled to transfer data and FLEX_US_MR.MODE9=0 (up to 8 bits to transfer on the line), the chunk of 1 byte must be configured in the DMA. If the transfer is performed by software, the access type must be defined as a byte.
1	TWO_DATA	RXRDY will be at level '1' when at least two unread data are in the receive FIFO. To minimize system bus load, when DMA is enabled to transfer data and FLEX_US_MR.MODE9=0 (up to 8 bits to transfer on the line), the chunk of 1 halfword (1 halfword carries 2 bytes) must be configured in the DMA (chunk size=1 and halfword access). If the transfer is performed by software, the access type can be defined as byte (1 byte per access, 2 accesses) or halfword (2 bytes per access, 1 single access).
2	FOUR_DATA	RXRDY will be at level '1' when at least four unread data are in the receive FIFO. To minimize system bus load, when DMA is enabled to transfer data and FLEX_US_MR.MODE9=0 (up to 8 bits to transfer on the line), the chunk of 1 word (1 word carries 4 bytes) must be configured in the DMA. If the transfer is performed by software, the access type can be defined as byte (1 byte per access, 4 accesses), halfword (2 bytes per access, 2 accesses) or word (4 bytes per access, 1 single access).

Bits 1:0 – TXRDYM[1:0] Transmitter Ready Mode
If FIFOs are enabled, the FLEX_US_CSR.TXRDY flag behaves as follows.

Value	Name	Description
0	ONE_DATA	TXRDY will be at level '1' when at least one data can be written in the transmit FIFO. When DMA is enabled to transfer data and FLEX_US_MR.MODE9=0 (up to 8 bits to transfer on the line), the chunk of 1 byte must be configured in the DMA. If the transfer is performed by software, the access type must be defined as a byte.
1	TWO_DATA	TXRDY will be at level '1' when at least two data can be written in the transmit FIFO. To minimize system bus load, when DMA is enabled to transfer data and FLEX_US_MR.MODE9=0 (up to 8 bits to transfer on the line), the chunk of 1 halfword (1 halfword carries 2 bytes) must be configured in the DMA. If the transfer is performed by software, the access type can be defined as byte (1 byte per access, 2 accesses) or halfword (2 bytes per access, 1 single access).
2	FOUR_DATA	TXRDY will be at level '1' when at least four data can be written in the transmit FIFO. To minimize system bus load, when DMA is enabled to transfer data and FLEX_US_MR.MODE9=0 (up to 8 bits to transfer on the line), the chunk of 1 word (1 word carries 4 bytes) must be configured in the DMA (chunk size=1 and word access). If the transfer is performed by software, the access type can be defined as byte (1 byte per access, 4 accesses), halfword (2 bytes per access, 2 accesses) or word (4 bytes per access, 1 single access).

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.31. USART FIFO Level Register

Name: FLEX_US_FLR
Offset: 0x2A4
Reset: 0x00000000
Property: Read-only

This register reads '0' if the FIFO is disabled (see FLEX_US_CR to enable/disable the internal FIFO).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			RXFL[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			TXFL[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bits 21:16 – RXFL[5:0] Receive FIFO Level

Value	Description
0	There is no unread data in the Receive FIFO.
1–8	Indicates the number of unread data in the Receive FIFO.

Bits 5:0 – TXFL[5:0] Transmit FIFO Level

Value	Description
0	There is no data in the Transmit FIFO.
1–8	Indicates the number of data in the Transmit FIFO.

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.32. USART FIFO Interrupt Enable Register

Name: FLEX_US_FIER
Offset: 0x2A8
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							RXFTHF2	
Access							W	
Reset							–	

Bit	7	6	5	4	3	2	1	0
	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 9 – RXFTHF2 RXFTHF2 Interrupt Enable

Bit 7 – RXFPTEF RXFPTEF Interrupt Enable

Bit 6 – TXFPTEF TXFPTEF Interrupt Enable

Bit 5 – RXFTHF RXFTHF Interrupt Enable

Bit 4 – RXFFF RXFFF Interrupt Enable

Bit 3 – RXFEF RXFEF Interrupt Enable

Bit 2 – TXFTHF TXFTHF Interrupt Enable

Bit 1 – TXFFF TXFFF Interrupt Enable

Bit 0 – TXFEF TXFEF Interrupt Enable

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.33. USART FIFO Interrupt Disable Register

Name: FLEX_US_FIDR
Offset: 0x2AC
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							RXFTHF2	
Access							W	
Reset							–	

Bit	7	6	5	4	3	2	1	0
	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 9 – RXFTHF2 RXFTHF2 Interrupt Disable

Bit 7 – RXFPTEF RXFPTEF Interrupt Disable

Bit 6 – TXFPTEF TXFPTEF Interrupt Disable

Bit 5 – RXFTHF RXFTHF Interrupt Disable

Bit 4 – RXFFF RXFFF Interrupt Disable

Bit 3 – RXFEF RXFEF Interrupt Disable

Bit 2 – TXFTHF TXFTHF Interrupt Disable

Bit 1 – TXFFF TXFFF Interrupt Disable

Bit 0 – TXFEF TXFEF Interrupt Disable

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.34. USART FIFO Interrupt Mask Register

Name: FLEX_US_FIMR
Offset: 0x2B0
Reset: 0x00000000
Property: Read-only

This register reads '0' if the FIFO is disabled (see FLEX_US_CR to enable/disable the internal FIFO).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							RXFTHF2	
Access							R	
Reset							0	

Bit	7	6	5	4	3	2	1	0
	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 9 – RXFTHF2 RXFTHF2 Interrupt Mask

Bit 7 – RXFPTEF RXFPTEF Interrupt Mask

Bit 6 – TXFPTEF TXFPTEF Interrupt Mask

Bit 5 – RXFTHF RXFTHF Interrupt Mask

Bit 4 – RXFFF RXFFF Interrupt Mask

Bit 3 – RXFEF RXFEF Interrupt Mask

Bit 2 – TXFTHF TXFTHF Interrupt Mask

Bit 1 – TXFFF TXFFF Interrupt Mask

Bit 0 – TXFEF TXFEF Interrupt Mask

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.35. USART FIFO Event Status Register

Name: FLEX_US_FESR
Offset: 0x2B4
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							RXFTHF2	TXFLOCK
Access							R	R
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 9 – RXFTHF2 Receive FIFO Threshold Flag 2 (cleared by writing the FLEX_US_CR.RSTSTA bit)

Value	Description
0	Number of unread data in Receive FIFO is above RXFTHRES threshold.
1	Number of unread data in Receive FIFO has reached RXFTHRES2 threshold since the last RSTSTA command was issued.

Bit 8 – TXFLOCK Transmit FIFO Lock

Value	Description
0	The Transmit FIFO is not locked.
1	The Transmit FIFO is locked.

Bit 7 – RXFPTEF Receive FIFO Pointer Error Flag
See FIFO Pointer Error for details.

Value	Description
0	No Receive FIFO pointer occurred.
1	Receive FIFO pointer error occurred. Receiver must be reset.

Bit 6 – TXFPTEF Transmit FIFO Pointer Error Flag
See FIFO Pointer Error for details.

Value	Description
0	No Transmit FIFO pointer occurred.
1	Transmit FIFO pointer error occurred. Transceiver must be reset.

Bit 5 – RXFTHF Receive FIFO Threshold Flag (cleared by writing the FLEX_US_CR.RSTSTA bit)

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	Number of unread data in Receive FIFO is below RXFTHRES threshold.
1	Number of unread data in Receive FIFO has reached RXFTHRES threshold since the last RSTSTA command was issued.

Bit 4 – RXFFF Receive FIFO Full Flag (cleared by writing the FLEX_US_CR.RSTSTA bit)

Value	Description
0	Receive FIFO is not empty.
1	Receive FIFO has been filled since the last RSTSTA command was issued.

Bit 3 – RXFEF Receive FIFO Empty Flag (cleared by writing the FLEX_US_CR.RSTSTA bit)

Value	Description
0	Receive FIFO is not empty.
1	Receive FIFO has been emptied since the last RSTSTA command was issued.

Bit 2 – TXFTHF Transmit FIFO Threshold Flag (cleared by writing the FLEX_US_CR.RSTSTA bit)

Value	Description
0	Number of data in Transmit FIFO is above TXFTHRES threshold.
1	Number of data in Transmit FIFO has reached TXFTHRES threshold since the last RSTSTA command was issued.

Bit 1 – TXFFF Transmit FIFO Full Flag (cleared by writing the FLEX_US_CR.RSTSTA bit)

Value	Description
0	Transmit FIFO is not full.
1	Transmit FIFO has been filled since the last RSTSTA command was issued.

Bit 0 – TXFEF Transmit FIFO Empty Flag (cleared by writing the FLEX_US_CR.RSTSTA bit)

Value	Description
0	Transmit FIFO is not empty.
1	Transmit FIFO has been emptied since the last RSTSTA command was issued.

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.36. USART Write Protection Mode Register

Name: FLEX_US_WPMR
Offset: 0x2E4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
						WPCREN	WPITEN	WPEN
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x555341	PASSWD	Writing any other value in this field aborts the write operation of bits WPEN, WPITEN and WPCREN. Always reads as 0.

Bit 2 – WPCREN Write Protection Control Enable
See USART Register Write Protection for the list of registers that can be write-protected.

Value	Description
0	Disables the write protection on control register if WPKEY corresponds to 0x555341 (“USA” in ASCII).
1	Enables the write protection on control register if WPKEY corresponds to 0x555341 (“USA” in ASCII).

Bit 1 – WPITEN Write Protection Interrupt Enable
See USART Register Write Protection for the list of registers that can be write-protected.

Value	Description
0	Disables the write protection on interrupt registers if WPKEY corresponds to 0x555341 (“USA” in ASCII).
1	Enables the write protection on interrupt registers if WPKEY corresponds to 0x555341 (“USA” in ASCII).

Bit 0 – WPEN Write Protection Enable
See USART Register Write Protection for the list of registers that can be write-protected.

Value	Description
0	Disables the write protection on configuration registers if WPKEY corresponds to 0x555341 (“USA” in ASCII).
1	Enables the write protection on configuration registers if WPKEY corresponds to 0x555341 (“USA” in ASCII).

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.37. USART Write Protection Status Register

Name: FLEX_US_WPSR
Offset: 0x2E8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	WPVSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPVS
Access								R
Reset								0

Bits 23:8 – WPVSRC[15:0] Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protection Violation Status

Value	Description
0	No write protection violation has occurred since the last read of FLEX_US_WPSR.
1	A write protection violation has occurred since the last read of FLEX_US_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.38. SPI Control Register

Name: FLEX_SPI_CR
Offset: 0x400
Reset: –
Property: Write-only

This register can only be written if the WPCREN bit is cleared in the SPI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	FIFODIS	FIFOEN						LASTXFER
Access	W	W						W
Reset	–	–						–

Bit	23	22	21	20	19	18	17	16
							RXFCLR	TXFCLR
Access							W	W
Reset							–	–

Bit	15	14	13	12	11	10	9	8
				REQCLR				
Access				W				
Reset				–				

Bit	7	6	5	4	3	2	1	0
	SWRST						SPIDIS	SPIEN
Access	W						W	W
Reset	–						–	–

Bit 31 – FIFODIS FIFO Disable

Value	Description
0	No effect.
1	Disables the Transmit and Receive FIFOs

Bit 30 – FIFOEN FIFO Enable

Value	Description
0	No effect.
1	Enables the Transmit and Receive FIFOs

Bit 24 – LASTXFER Last Transfer
See Peripheral Selection for more details.

Value	Description
0	No effect.
1	The current NPCS will be de-asserted after the character written in TD has been transferred. When CSAAT is set, the communication with the current serial peripheral can be closed by raising the corresponding NPCS line as soon as TD transfer is completed.

Bit 17 – RXFCLR Receive FIFO Clear

Value	Description
0	No effect.
1	Empties the Receive FIFO.

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 16 – TXFCLR Transmit FIFO Clear

Value	Description
0	No effect.
1	Empties the Transmit FIFO.

Bit 12 – REQCLR Request to Clear the Comparison Trigger

Asynchronous partial wakeup enabled:
0: No effect.
1: Clears the potential clock request currently issued by SPI, thus the potential system wakeup is cancelled.
Asynchronous partial wakeup disabled:
0: No effect.
1: Restarts the comparison trigger to enable FLEX_SPI_RDR loading.

Bit 7 – SWRST SPI Software Reset

The SPI is in Client mode after software reset.
PDC channels are not affected by software reset.

Value	Description
0	No effect.
1	Resets the SPI. A software-triggered hardware reset of the SPI interface is performed.

Bit 1 – SPIDIS SPI Disable

If a transfer is in progress when SPIDIS is set, the SPI completes the transmission of the shifter register and does not start any new transfer, even if the FLEX_US_THR is loaded.
All pins are set to Input mode after completion of the transmission in progress, if any.

Value	Description
0	No effect.
1	Disables the SPI.

Bit 0 – SPIEN SPI Enable

Value	Description
0	No effect.
1	Enables the SPI to transfer and receive data.

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.39. SPI Mode Register

Name: FLEX_SPI_MR
Offset: 0x404
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the SPI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	DLYBCS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
							PCS[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	15	14	13	12	11	10	9	8
				CMPMODE				
Access				R/W				
Reset				0				

Bit	7	6	5	4	3	2	1	0
	LLB		WDRBT	MODFDIS	BRSRCCLK	PCSDEC	PS	MSTR
Access	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset	0		0	0	0	0	0	0

Bits 31:24 – DLYBCS[7:0] Delay Between Chip Selects
This field defines the delay between the inactivation and the activation of NPCS. The DLYBCS time ensures chip selects do not overlap and solves bus contentions in case of peripherals having long data float times.
If DLYBCS is ≤ 6, six peripheral clock periods are inserted by default.
Otherwise, the following equations determine the delay:
If FLEX_SPI_MR.BRSRCCLK = 0: DLYBCS = Delay Between Chip Selects × $f_{peripheral\ clock}$
If FLEX_SPI_MR.BRSRCCLK = 1: DLYBCS = Delay Between Chip Selects × f_{GCLK}

Bits 17:16 – PCS[1:0] Peripheral Chip Select
This field is only used if fixed peripheral select is active (PS = 0).
If PCSDEC = 0:
PCS = x0 NPCS[1:0] = 10
PCS = 01 NPCS[1:0] = 01
PCS = 11 forbidden (no peripheral is selected)
(x = don't care)
If PCSDEC = 1:
NPCS[1:0] output signals = PCS

Bit 12 – CMPMODE Comparison Mode

Value	Name	Description
0	FLAG_ONLY	Any character is received and comparison function drives CMP flag.
1	START_CONDITION	Comparison condition must be met to start reception of all incoming characters until REQCLR is set.

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 7 – LLB Local Loopback Enable

LLB controls the local loopback on the data shift register for testing in Host mode only (MISO is internally connected on MOSI).

Value	Description
0	Local loopback path disabled.
1	Local loopback path enabled.

Bit 5 – WDRBT Wait Data Read Before Transfer

Value	Description
0	No Effect. In Host mode, a transfer can be initiated regardless of the FLEX_SPI_RDR state.
1	In Host mode, a transfer can start only if FLEX_SPI_RDR is empty, i.e., does not contain any unread data. This mode prevents overrun error in reception.

Bit 4 – MODFDIS Mode Fault Detection

Value	Description
0	Mode fault detection is enabled.
1	Mode fault detection is disabled.

Bit 3 – BRSRCCLK Bit Rate Source Clock

If the bit BRSRCCLK = 1, the FLEX_US_CSRx.SCBR field must be programmed with a value greater than 1.

Value	Name	Description
0	PERIPH_CLK	The peripheral clock is the source clock for the bit rate generation.
1	GCLK	GCLK is the source clock for the bit rate generation, thus the bit rate can be independent of the core/peripheral clock.

Bit 2 – PCSDEC Chip Select Decode

When PCSDEC equals one, up to 3 Chip Select signals can be generated with the two NPCS lines using an external 2- to 4-bit decoder. The Chip Select registers define the characteristics of the 3 chip selects, with the following rules:
FLEX_SPI_CSR0 defines peripheral chip select signals 0 to 1.
FLEX_SPI_CSR1 defines peripheral chip select signal 2.

Value	Description
0	The chip selects are directly connected to a peripheral device.
1	The two NPCS chip select lines are connected to a 2- to 4-bit decoder.

Bit 1 – PS Peripheral Select

Value	Description
0	Fixed Peripheral Select
1	Variable Peripheral Select

Bit 0 – MSTR Host/Client Mode

Value	Description
0	SPI is in Client mode.
1	SPI is in Host mode.

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.40. SPI Receive Data Register

Name: FLEX_SPI_RDR
Offset: 0x408
Reset: 0x00000000
Property: Read-only

If FIFO is enabled (FLEX_SPI_CR.FIFOEN) and FLEX_SPI_FMR.RXRDYM = 0, see SPI Single Data Access for details.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					PCS[3:0]			
Access					R	R	R	R
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RD[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RD[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 19:16 – PCS[3:0] Peripheral Chip Select
In Host mode only, these bits indicate the value on the NPCS pins at the end of a transfer. Otherwise, these bits are read as zero.
Note: When using Variable Peripheral Select mode (FLEX_SPI_MR.PS = 1), it is mandatory to set the FLEX_SPI_MR.WDRBT bit to 1 if the PCS field must be processed in FLEX_SPI_RDR.

Bits 15:0 – RD[15:0] Receive Data
Data received by the SPI Interface is stored in this register in a right-justified format. Unused bits are read as zero.

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.41. SPI Receive Data Register (FIFO Multiple Data, 8-bit)

Name: FLEX_SPI_RDR (FIFO_MULTI_DATA_8)
Offset: 0x408
Reset: 0x00000000
Property: Read-only

To read multi-data, the FIFO must be enabled (FLEX_SPI_CR.FIFOEN=1) and FLEX_SPI_MR.PS=0. The access type (byte, halfword or word) determines the number of data written in a single access (1, 2 or 4), see SPI Multiple Data Access for details.

Bit	31	30	29	28	27	26	25	24
	RD3[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RD2[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RD1[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RD0[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0:7, 8:15, 16:23, 24:31 – RDx Receive Data
First unread data in the Receive FIFO. Data received by the SPI Interface is stored in this register in a right-justified format. Unused bits are read as zero.

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.42. SPI Receive Data Register (FIFO Multiple Data, 16-bit)

Name: FLEX_SPI_RDR (FIFO_MULTI_DATA_16)
Offset: 0x408
Reset: 0x00000000
Property: Read-only

To read multi-data, the FIFO must be enabled (FLEX_SPI_CR.FIFOEN=1) and FLEX_SPI_MR.PS=0. The access type (byte, halfword or word) determines the number of data written in a single access (1 or 2), see SPI Multiple Data Access for details.

Bit	31	30	29	28	27	26	25	24
	RD1[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RD1[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RD0[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RD0[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0:15, 16:31 – RDx Receive Data
First unread data in the Receive FIFO. Data received by the SPI Interface is stored in this register in a right-justified format. Unused bits are read as zero.

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.43. SPI Transmit Data Register

Name: FLEX_SPI_TDR
Offset: 0x40C
Reset: –
Property: Write-only

If FIFO is enabled (FLEX_SPI_CR.FIFOEN=1), a byte/halfword access on FLEX_SPI_TDR writes one byte/halfword, see SPI Single Data Access for details.

Bit	31	30	29	28	27	26	25	24
								LASTXFER
Access								W
Reset								–

Bit	23	22	21	20	19	18	17	16
					PCS[3:0]			
Access					W	W	W	W
Reset					–	–	–	–

Bit	15	14	13	12	11	10	9	8
	TD[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	TD[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 24 – LASTXFER Last Transfer
This field is only used if variable peripheral select is active (FLEX_SPI_MR.PS = 1).

Value	Description
0	No effect.
1	The current NPCS is de-asserted after the transfer of the character written in TD. When FLEX_SPI_CSRx.CSAAT is set, the communication with the current serial peripheral can be closed by raising the corresponding NPCS line as soon as TD transfer is completed.

Bits 19:16 – PCS[3:0] Peripheral Chip Select
This field is only used if variable peripheral select is active (FLEX_SPI_MR.PS = 1).
If FLEX_SPI_MR.PCSDEC = 0:
PCS = x0 NPCS[1:0] = 10
PCS = 01 NPCS[1:0] = 01
PCS = 11 forbidden (no peripheral is selected)
(x = don't care)
If FLEX_SPI_MR.PCSDEC = 1:
NPCS[1:0] output signals = PCS

Bits 15:0 – TD[15:0] Transmit Data
Data to be transmitted by the SPI Interface is stored in this register. Information to be transmitted must be written to the transmit data register in a right-justified format.

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.44. SPI Transmit Data Register (FIFO Multiple Data, 8- to 16-bit)

Name: FLEX_SPI_TDR (FIFO_MULTI_DATA)
Offset: 0x40C
Reset: –
Property: Write-only

To write multi-data, the FIFO must be enabled (FLEX_SPI_CR.FIFOEN=1) and FLEX_SPI_MR.PS=0. The access type (byte, halfword or word) determines the number of data written in a single access (1 or 2), see SPI Single Data Access for details.

Bit	31	30	29	28	27	26	25	24
	TD1[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	TD1[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	TD0[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	TD0[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0:15, 16:31 – TDx Transmit Data
Next data to write in the Transmit FIFO. Information to be transmitted must be written to this register in a right-justified format.

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.45. SPI Status Register

Name: FLEX_SPI_SR
Offset: 0x410
Reset: 0x000000F0
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
								SPIENS
Access								R
Reset								0

Bit	15	14	13	12	11	10	9	8
				SFERR	CMP	UNDES	TXEMPTY	NSSR
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TXBUFE	RXBUFF	ENDTX	ENDRX	OVRES	MODF	TDRE	RDRF
Access	R	R	R	R	R	R	R	R
Reset	1	1	1	1	0	0	0	0

Bit 31 – RXFPTEF Receive FIFO Pointer Error Flag
See FIFO Pointer Error for details.
This bit reads '0' if the FIFO is disabled (see FLEX_SPI_CR to enable/disable the internal FIFO).

Value	Description
0	No Receive FIFO pointer occurred.
1	Receive FIFO pointer error occurred. The receiver must be reset.

Bit 30 – TXFPTEF Transmit FIFO Pointer Error Flag
See FIFO Pointer Error for details.
This bit reads '0' if the FIFO is disabled (see FLEX_SPI_CR to enable/disable the internal FIFO).

Value	Description
0	No Transmit FIFO pointer occurred.
1	Transmit FIFO pointer error occurred. The transceiver must be reset.

Bit 29 – RXFTHF Receive FIFO Threshold Flag
This bit reads '0' if the FIFO is disabled (see FLEX_SPI_CR to enable/disable the internal FIFO).

Value	Description
0	Number of unread data in Receive FIFO is below RXFTHRES threshold or RXFTH flag has been cleared.
1	Number of unread data in Receive FIFO has reached RXFTHRES threshold (changing states from "below threshold" to "equal to or above threshold").

Bit 28 – RXFFF Receive FIFO Full Flag
This bit reads '0' if the FIFO is disabled (see FLEX_SPI_CR to enable/disable the internal FIFO).

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	Receive FIFO is not empty or RXFE flag has been cleared.
1	Receive FIFO has been filled (changing states from “not full” to “full”).

Bit 27 – RXFEF Receive FIFO Empty Flag
This bit reads ‘0’ if the FIFO is disabled (see FLEX_SPI_CR to enable/disable the internal FIFO).

Value	Description
0	Receive FIFO is not empty or RXFE flag has been cleared.
1	Receive FIFO has been emptied (changing states from “not empty” to “empty”).

Bit 26 – TXFTHF Transmit FIFO Threshold Flag (cleared on read)
This bit reads ‘0’ if the FIFO is disabled (see FLEX_SPI_CR to enable/disable the internal FIFO).

Value	Description
0	Number of data in Transmit FIFO is above TXFTHRES threshold.
1	Number of data in Transmit FIFO has reached TXFTHRES threshold since the last read of FLEX_SPI_SR.

Bit 25 – TXFFF Transmit FIFO Full Flag (cleared on read)
This bit reads ‘0’ if the FIFO is disabled (see FLEX_SPI_CR to enable/disable the internal FIFO).

Value	Description
0	Transmit FIFO is not full or TXFF flag has been cleared.
1	Transmit FIFO has been filled since the last read of FLEX_SPI_SR.

Bit 24 – TXFEF Transmit FIFO Empty Flag (cleared on read)
This bit reads ‘0’ if the FIFO is disabled (see FLEX_SPI_CR to enable/disable the internal FIFO).

Value	Description
0	Transmit FIFO is not empty.
1	Transmit FIFO has been emptied since the last read of FLEX_SPI_SR.

Bit 16 – SPIENS SPI Enable Status

Value	Description
0	SPI is disabled.
1	SPI is enabled.

Bit 12 – SFERR Client Mode Frame Error (cleared on read)

Value	Description
0	No frame error has been detected for a client access since the last read of FLEX_SPI_SR.
1	In Client mode, the chip select raised while the character defined in FLEX_SPI_CSR0.BITS was not complete.

Bit 11 – CMP Comparison Status (cleared on read)

Value	Description
0	No received character matched the comparison criteria programmed in VAL1 and VAL2 fields in FLEX_SPI_CMPR since the last read of FLEX_SPI_SR.
1	A received character matched the comparison criteria since the last read of FLEX_SPI_SR.

Bit 10 – UNDES Underrun Error Status (Client mode only) (cleared on read)

Value	Description
0	No underrun has been detected since the last read of FLEX_SPI_SR.
1	A transfer starts whereas no data has been loaded in FLEX_SPI_TDR, cleared when FLEX_SPI_SR is read.

Bit 9 – TXEMPTY Transmission Registers Empty (cleared by writing FLEX_SPI_TDR)

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	As soon as data is written in FLEX_SPI_TDR.
1	FLEX_SPI_TDR and internal shift register are empty. If a transfer delay has been defined, TXEMPTY is set after the end of this delay.

Bit 8 – NSSR NSS Rising (cleared on read)

Value	Description
0	No rising edge detected on NSS pin since the last read of FLEX_SPI_SR.
1	A rising edge occurred on NSS pin since the last read of FLEX_SPI_SR.

Bit 7 – TXBUFE TX Buffer Empty (cleared by writing FLEX_SPI_TCR or FLEX_SPI_TNCR)
FLEX_SPI_TCR and FLEX_SPI_TNCR are PDC registers.

Value	Description
0	FLEX_SPI_TCR or FLEX_SPI_TNCR has a value other than 0.
1	Both FLEX_SPI_TCR and FLEX_SPI_TNCR have a value of 0.

Bit 6 – RXBUFF RX Buffer Full (cleared by writing FLEX_SPI_RCR or FLEX_SPI_RNCR)
FLEX_SPI_RCR and FLEX_SPI_RNCR are PDC registers.

Value	Description
0	FLEX_SPI_RCR or FLEX_SPI_RNCR has a value other than 0.
1	Both FLEX_SPI_RCR and FLEX_SPI_RNCR have a value of 0.

Bit 5 – ENDTX End of TX Buffer (cleared by writing FLEX_SPI_TCR or FLEX_SPI_TNCR)
FLEX_SPI_TCR and FLEX_SPI_TNCR are PDC registers.

Value	Description
0	The Transmit Counter register has not reached 0 since the last write in FLEX_SPI_TCR or FLEX_SPI_TNCR.
1	The Transmit Counter register has reached 0 since the last write in FLEX_SPI_TCR or FLEX_SPI_TNCR.

Bit 4 – ENDRX End of RX Buffer (cleared by writing FLEX_SPI_RCR or FLEX_SPI_RNCR)
FLEX_SPI_RCR and FLEX_SPI_RNCR are PDC registers.

Value	Description
0	The Receive Counter register has not reached 0 since the last write in FLEX_SPI_RCR or FLEX_SPI_RNCR.
1	The Receive Counter register has reached 0 since the last write in FLEX_SPI_RCR or FLEX_SPI_RNCR.

Bit 3 – OVRES Overrun Error Status (cleared on read)
An overrun occurs when FLEX_SPI_RDR is loaded at least twice from the shift register since the last read of FLEX_SPI_RDR.

Value	Description
0	No overrun has been detected since the last read of FLEX_SPI_SR.
1	An overrun has occurred since the last read of FLEX_SPI_SR.

Bit 2 – MODF Mode Fault Error (cleared on read)

Value	Description
0	No mode fault has been detected since the last read of FLEX_SPI_SR.
1	A mode fault occurred since the last read of FLEX_SPI_SR.

Bit 1 – TDRE Transmit Data Register Empty (cleared by writing FLEX_SPI_TDR)
When FIFOs are disabled:
0: Data has been written to FLEX_SPI_TDR and not yet transferred to the internal shift register.
1: The last data written to FLEX_SPI_TDR has been transferred to the internal shift register.
TDRE is cleared when the SPI is disabled or at reset. Enabling the SPI sets the TDRE flag.

© 2026 Microchip Technology Inc. and its subsidiaries

When FIFOs are enabled:
0: Transmit FIFO cannot accept more data.
1: Transmit FIFO can accept data; one or more data can be written according to TXRDYM field configuration.
TDRE behavior with FIFOs enabled is illustrated in TXEMPTY, TDRE and RDRF Behavior.

Bit 0 – RDRF Receive Data Register Full (cleared by reading FLEX_SPI_RDR)
When FIFOs are disabled:
0: No data has been received since the last read of FLEX_SPI_RDR.
1: Data has been received and the received data has been transferred from the internal shift register to FLEX_SPI_RDR since the last read of FLEX_SPI_RDR.
When FIFOs are enabled:
0: Receive FIFO is empty; no data to read.
1: At least one unread data is in the Receive FIFO.
RDRF behavior with FIFOs enabled is illustrated in TXEMPTY, TDRE and RDRF Behavior.

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.46. SPI Interrupt Enable Register

Name: FLEX_SPI_IER
Offset: 0x414
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the SPI Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				SFERR	CMP	UNDES	TXEMPTY	NSSR
Access				W	W	W	W	W
Reset				–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	TXBUFE	RXBUFF	ENDTX	ENDRX	OVRES	MODF	TDRE	RDRF
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 31 – RXFPTEF RXFPTEF Interrupt Enable

Bit 30 – TXFPTEF TXFPTEF Interrupt Enable

Bit 29 – RXFTHF RXFTHF Interrupt Enable

Bit 28 – RXFFF RXFFF Interrupt Enable

Bit 27 – RXFEF RXFEF Interrupt Enable

Bit 26 – TXFTHF TXFTHF Interrupt Enable

Bit 25 – TXFFF TXFFF Interrupt Enable

Bit 24 – TXFEF TXFEF Interrupt Enable

Bit 12 – SFERR Client Mode Frame Error Interrupt Enable

Bit 11 – CMP Comparison Interrupt Enable

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 10 – UNDES Underrun Error Interrupt Enable

Bit 9 – TXEMPTY Transmission Registers Empty Enable

Bit 8 – NSSR NSS Rising Interrupt Enable

Bit 7 – TXBUFE Transmit Buffer Empty Interrupt Enable

Bit 6 – RXBUFF Receive Buffer Full Interrupt Enable

Bit 5 – ENDTX End of Transmit Buffer Interrupt Enable

Bit 4 – ENDRX End of Receive Buffer Interrupt Enable

Bit 3 – OVRES Overrun Error Interrupt Enable

Bit 2 – MODF Mode Fault Error Interrupt Enable

Bit 1 – TDRE SPI Transmit Data Register Empty Interrupt Enable

Bit 0 – RDRF Receive Data Register Full Interrupt Enable

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.47. SPI Interrupt Disable Register

Name: FLEX_SPI_IDR
Offset: 0x418
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the SPI Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				SFERR	CMP	UNDES	TXEMPTY	NSSR
Access				W	W	W	W	W
Reset				–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	TXBUFE	RXBUFF	ENDTX	ENDRX	OVRES	MODF	TDRE	RDRF
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 31 – RXFPTEF RXFPTEF Interrupt Disable

Bit 30 – TXFPTEF TXFPTEF Interrupt Disable

Bit 29 – RXFTHF RXFTHF Interrupt Disable

Bit 28 – RXFFF RXFFF Interrupt Disable

Bit 27 – RXFEF RXFEF Interrupt Disable

Bit 26 – TXFTHF TXFTHF Interrupt Disable

Bit 25 – TXFFF TXFFF Interrupt Disable

Bit 24 – TXFEF TXFEF Interrupt Disable

Bit 12 – SFERR Client Mode Frame Error Interrupt Disable

Bit 11 – CMP Comparison Interrupt Disable

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 10 – UNDES Underrun Error Interrupt Disable

Bit 9 – TXEMPTY Transmission Registers Empty Disable

Bit 8 – NSSR NSS Rising Interrupt Disable

Bit 7 – TXBUFE Transmit Buffer Empty Interrupt Disable

Bit 6 – RXBUFF Receive Buffer Full Interrupt Disable

Bit 5 – ENDTX End of Transmit Buffer Interrupt Disable

Bit 4 – ENDRX End of Receive Buffer Interrupt Disable

Bit 3 – OVRES Overrun Error Interrupt Disable

Bit 2 – MODF Mode Fault Error Interrupt Disable

Bit 1 – TDRE SPI Transmit Data Register Empty Interrupt Disable

Bit 0 – RDRF Receive Data Register Full Interrupt Disable

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.48. SPI Interrupt Mask Register

Name: FLEX_SPI_IMR
Offset: 0x41C
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				SFERR	CMP	UNDES	TXEMPTY	NSSR
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TXBUFE	RXBUFF	ENDTX	ENDRX	OVRES	MODF	TDRE	RDRF
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 31 – RXFPTEF RXFPTEF Interrupt Mask

Bit 30 – TXFPTEF TXFPTEF Interrupt Mask

Bit 29 – RXFTHF RXFTHF Interrupt Mask

Bit 28 – RXFFF RXFFF Interrupt Mask

Bit 27 – RXFEF RXFEF Interrupt Mask

Bit 26 – TXFTHF TXFTHF Interrupt Mask

Bit 25 – TXFFF TXFFF Interrupt Mask

Bit 24 – TXFEF TXFEF Interrupt Mask

Bit 12 – SFERR Client Mode Frame Error Interrupt Mask

Bit 11 – CMP Comparison Interrupt Mask

Bit 10 – UNDES Underrun Error Interrupt Mask

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 9 – TXEMPTY Transmission Registers Empty Mask

Bit 8 – NSSR NSS Rising Interrupt Mask

Bit 7 – TXBUFE Transmit Buffer Empty Interrupt Mask

Bit 6 – RXBUFF Receive Buffer Full Interrupt Mask

Bit 5 – ENDTX End of Transmit Buffer Interrupt Mask

Bit 4 – ENDRX End of Receive Buffer Interrupt Mask

Bit 3 – OVRES Overrun Error Interrupt Mask

Bit 2 – MODF Mode Fault Error Interrupt Mask

Bit 1 – TDRE SPI Transmit Data Register Empty Interrupt Mask

Bit 0 – RDRF Receive Data Register Full Interrupt Mask

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.49. SPI Chip Select Register

Name: FLEX_SPI_CSRx
Offset: 0x0430 + x*0x04 [x=0..1]
Reset: 0x00000000
Property: R/W

This register can only be written if the WPEN bit is cleared in the SPI Write Protection Mode Register.

FLEX_SPI_CSRx must be written even if the user wants to use the default reset values. The BITS field is not updated with the translated value unless the register is written.

Bit	31	30	29	28	27	26	25	24
	DLYBCT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DLYBS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SCBR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BITS[3:0]				CSAAT	CSNAAT	NCPHA	CPOL
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – DLYBCT[7:0] Delay Between Consecutive Transfers
This field defines the delay between two consecutive transfers with the same peripheral without removing the chip select. The delay is always inserted after each transfer and before removing the chip select if needed.
When DLYBCT = 0, no delay between consecutive transfers is inserted and the clock keeps its duty cycle over the character transfers.
Otherwise, the following equations determine the delay:
If FLEX_SPI_MR.BRSRCCLK = 0: DLYBCT = Delay Between Consecutive Transfers × $f_{peripheral\ clock}$ / 32
If FLEX_SPI_MR.BRSRCCLK = 1: DLYBCT = Delay Between Consecutive Transfers × f_{GCLK} / 32

Bits 23:16 – DLYBS[7:0] Delay Before SPCK
This field defines the delay from NPCS falling edge (activation) to the first valid SPCK transition.
When DLYBS = 0, the delay is half the SPCK clock period.
Otherwise, the following equations determine the delay:
If FLEX_SPI_MR.BRSRCCLK = 0: DLYBS = Delay Before SPCK × $f_{peripheral\ clock}$
If FLEX_SPI_MR.BRSRCCLK = 1: DLYBS = Delay Before SPCK × f_{GCLK}

Bits 15:8 – SCBR[7:0] Serial Clock Bit Rate
In Host mode, the SPI Interface uses a modulus counter to derive the SPCK bit rate from the clock defined by the bit BRSRCCLK. The bit rate is selected by writing a value from 1 to 255 in the SCBR field. The following equations determine the SPCK bit rate:
If FLEX_SPI_MR.BRSRCCLK = 0: SCBR = $f_{peripheral\ clock}$ / SPCK Bit Rate
If FLEX_SPI_MR.BRSRCCLK = 1: SCBR = f_{GCLK} / SPCK Bit Rate

© 2026 Microchip Technology Inc. and its subsidiaries

Programming the SCBR field to 0 is forbidden. Triggering a transfer while SCBR is at 0 can lead to unpredictable results.
If BRSRCCLK = 1 in FLEX_SPI_MR, SCBR must be programmed with a value greater than 1.
At reset, SCBR is 0 and the user has to program it at a valid value before performing the first transfer.
Note: If one of the FLEX_SPI_CSRx.SCBR fields is set to 1, the other FLEX_SPI_CSRx.SCBR fields must be set to 1 as well, if they are used to process transfers. If they are not used to transfer data, they can be set at any value.

Bits 7:4 – BITS[3:0] Bits Per Transfer
See Note.
The BITS field determines the number of data bits transferred. Reserved values should not be used.

Value	Name	Description
0	8_BIT	8 bits for transfer
1	9_BIT	9 bits for transfer
2	10_BIT	10 bits for transfer
3	11_BIT	11 bits for transfer
4	12_BIT	12 bits for transfer
5	13_BIT	13 bits for transfer
6	14_BIT	14 bits for transfer
7	15_BIT	15 bits for transfer
8	16_BIT	16 bits for transfer
9–15	Reserved	

Bit 3 – CSAAT Chip Select Active After Transfer

Value	Description
0	The Peripheral Chip Select Line rises as soon as the last transfer is achieved.
1	The Peripheral Chip Select does not rise after the last transfer is achieved. It remains active until a new transfer is requested on a different chip select.

Bit 2 – CSNAAT Chip Select Not Active After Transfer (Ignored if CSAAT = 1)
If FLEX_SPI_MR.BRSRCCLK = 0: $\frac{\text{DLYBCS}}{\text{f}_{\text{peripheral clock}}}$(if DLYBCS ≠ 0)
If FLEX_SPI_MR.BRSRCCLK = 1: $\frac{\text{DLYBCS}}{\text{f}_{GCLK}}$
If DLYBCS < 6, a minimum of six periods is introduced.

Value	Description
0	The Peripheral Chip Select does not rise between two transfers if the FLEX_SPI_TDR is reloaded before the end of the first transfer and if the two transfers occur on the same Chip Select.
1	The Peripheral Chip Select rises systematically after each transfer performed on the same client. It remains inactive after the end of transfer for a minimal duration of:

Bit 1 – NCPHA Clock Phase
NCPHA determines which edge of SPCK causes data to change and which edge causes data to be captured. NCPHA is used with CPOL to produce the required clock/data relationship between host and client devices.

Value	Description
0	Data are changed on the leading edge of SPCK and captured on the following edge of SPCK.
1	Data are captured on the leading edge of SPCK and changed on the following edge of SPCK.

Bit 0 – CPOL Clock Polarity
CPOL is used to determine the inactive state value of the serial clock (SPCK). It is used with NCPHA to produce the required clock/data relationship between host and client devices.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	The inactive state value of SPCK is logic level zero.
1	The inactive state value of SPCK is logic level one.

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.50. SPI FIFO Mode Register

Name: FLEX_SPI_FMR
Offset: 0x440
Reset: 0x00000000
Property: Read/Write

This register reads '0' if the FIFO is disabled (see FLEX_SPI_CR to enable/disable the internal FIFO)

Bit	31	30	29	28	27	26	25	24
			RXFTHRES[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
			TXFTHRES[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			RXRDYM[1:0]				TXRDYM[1:0]	
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bits 29:24 – RXFTHRES[5:0] Receive FIFO Threshold

Value	Description
0–8	Defines the Receive FIFO threshold value (number of data). The FLEX_SPI_SR.RXFTH flag will be set when Receive FIFO goes from "below" threshold state to "equal to or above" threshold state.

Bits 21:16 – TXFTHRES[5:0] Transmit FIFO Threshold

Value	Description
0–8	Defines the Transmit FIFO threshold value (number of data). The FLEX_SPI_SR.TXFTH flag will be set when Transmit FIFO goes from "above" threshold state to "equal to or below" threshold state.

Bits 5:4 – RXRDYM[1:0] Receive Data Register Full Mode
If FIFOs are enabled, the FLEX_SPI_SR.RDRF flag behaves as follows.

Value	Name	Description
0	ONE_DATA	RDRF will be at level '1' when at least one unread data is in the receive FIFO. When DMA is enabled to transfer data and FLEX_SPI_CSR0.BITS=0 (8 bits transfered on SPI line), the chunk of 1 byte must be configured in the DMA. When FLEX_SPI_CSR0.BITS>0 (9 to 16 bits transfered on SPI line), the chunk of 1 halfword must be configured in the DMA. If the transfer is performed by software, the access type can be defined as byte or halfword depending on FLEX_SPI_CSR0.BITS.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Name	Description
1	TWO_DATA	RDRF will be at level '1' when at least two unread data are in the receive FIFO. To minimize system bus load, when DMA is enabled to transfer data and FLEX_SPI_CSR0.BITS=0 (8 bits transfered on SPI line), the chunk of 1 halfword (1 halfword carries 2 bytes) must be configured in the DMA. When FLEX_SPI_CSR0.BITS>0 (9 to 16 bits transfered on SPI line), the chunk of 1 word (1 word carries 2 halfwords) must be configured in the DMA. If the transfer is performed by software, the access type can be defined as halfword (2 bytes per access, 1 access when FLEX_SPI_CSR0.BITS=0), or word (2 halfwords per access, 2 accesses when FLEX_SPI_CSR0.BITS>0).
2	FOUR_DATA	RDRF will be at level '1' when at least four unread data are in the receive FIFO. To minimize system bus load, when DMA is enabled to transfer data and FLEX_SPI_CSR0.BITS=0 (8 bits transfered on SPI line), the chunk of 1 word (1 halfword carries 4 bytes) must be configured in the DMA. When FLEX_SPI_CSR0.BITS>0 (9 to 16 bits transfered on SPI line), the chunk of 2 words (1 word carries 4 bytes) must be configured in the DMA. If the transfer is performed by software, the access type can be defined as word (4 bytes per access, 1 access when FLEX_SPI_CSR0.BITS=0 or 2 halfwords per access, 2 accesses when FLEX_SPI_CSR0.BITS>0).

Bits 1:0 – TXRDYM[1:0] Transmit Data Register Empty Mode
If FIFOs are enabled, the FLEX_SPI_SR.TDRE flag behaves as follows.

Value	Name	Description
0	ONE_DATA	TDRE will be at level '1' when at least one data can be written in the transmit FIFO. When DMA is enabled to transfer data, the chunk of 1 data (byte or halfword) must be configured in the DMA depending on FLEX_SPI_CSR0.BITS. If the transfer is performed by software, the access type (byte, halfword) must be defined depending on FLEX_SPI_CSR0.BITS.
1	TWO_DATA	TDRE will be at level '1' when at least two data can be written in the transmit FIFO. FIFO. To minimize system bus load, when DMA is enabled to transfer data, the chunk of 1 word (1 word carries 2 data) must be configured in the DMA. If the transfer is performed by software, the access type must be defined as word (2 data per access, 1 access).

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.51. SPI FIFO Level Register

Name: FLEX_SPI_FLR
Offset: 0x444
Reset: 0x00000000
Property: Read-only

This register reads '0' if the FIFO is disabled (see FLEX_SPI_CR to enable/disable the internal FIFO).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			RXFL[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			TXFL[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bits 21:16 – RXFL[5:0] Receive FIFO Level

Value	Description
0	There is no unread data in the Receive FIFO.
1–8	Indicates the number of unread data in the Receive FIFO.

Bits 5:0 – TXFL[5:0] Transmit FIFO Level

Value	Description
0	There is no data in the Transmit FIFO.
1–8	Indicates the number of data in the Transmit FIFO.

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.52. SPI Comparison Register

Name: FLEX_SPI_CMPR
Offset: 0x448
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the SPI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	VAL2[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	VAL2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	VAL1[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	VAL1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – VAL2[15:0] Second Comparison Value for Received Character

Value	Description
0–65535	The received character must be lower than or equal to the value of VAL2 and higher than or equal to VAL1 to set the FLEX_SPI_CSR.CMP flag. If asynchronous partial wakeup is enabled in PMC_SLPWK_ER, the SPI requests a system wakeup if condition is met.

Bits 15:0 – VAL1[15:0] First Comparison Value for Received Character

Value	Description
0–65535	The received character must be higher than or equal to the value of VAL1 and lower than or equal to VAL2 to set the FLEX_SPI_SR.CMP flag. If asynchronous partial wakeup is enabled in PMC_SLPWK_ER, the SPI requests a system wakeup if the condition is met.

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.53. SPI Write Protection Mode Register

Name: FLEX_SPI_WPMR
Offset: 0x4E4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
						WPCREN	WPITEN	WPEN
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x535049	PASSWD	Writing any other value in this field aborts the write operation of bits WPEN, WPITEN and WPCREN. Always reads as 0.

Bit 2 – WPCREN Write Protection Control Enable

Value	Description
0	Disables the write protection on control register if WPKEY corresponds to 0x535049 (“SPI” in ASCII).
1	Enables the write protection on control register if WPKEY corresponds to 0x535049 (“SPI” in ASCII).

Bit 1 – WPITEN Write Protection Interrupt Enable

Value	Description
0	Disables the write protection on interrupt registers if WPKEY corresponds to 0x535049 (“SPI” in ASCII).
1	Enables the write protection on interrupt registers if WPKEY corresponds to 0x535049 (“SPI” in ASCII).

Bit 0 – WPEN Write Protection Enable

See SPI Register Write Protection for the list of registers that can be write-protected.

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x535049 (“SPI” in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x535049 (“SPI” in ASCII).

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.54. SPI Write Protection Status Register

Name: FLEX_SPI_WPSR
Offset: 0x4E8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPVS
Access								R
Reset								0

Bits 15:8 – WPVSRC[7:0] Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protection Violation Status

Value	Description
0	No write protect violation has occurred since the last read of FLEX_SPI_WPSR.
1	A write protect violation has occurred since the last read of FLEX_SPI_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.55. TWI Control Register

Name: FLEX_TWI_CR
Offset: 0x600
Reset: –
Property: Write-only

This register can only be written if the WPCREN bit is cleared in the TWI Write Protection Mode register.

Bit	31	30	29	28	27	26	25	24
			FIFODIS	FIFOEN		LOCKCLR		THRCLR
Access			W	W		W		W
Reset			–	–		–		–

Bit	23	22	21	20	19	18	17	16
							ACMDIS	ACMEN
Access							W	W
Reset							–	–

Bit	15	14	13	12	11	10	9	8
	CLEAR	PECRQ	PECDIS	PECEN	SMBDIS	SMBEN	HSDIS	HSEN
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	SWRST	QUICK	SVDIS	SVEN	MSDIS	MSEN	STOP	START
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 29 – FIFODIS FIFO Disable

Value	Description
0	No effect.
1	Disable the Transmit and Receive FIFOs

Bit 28 – FIFOEN FIFO Enable

Value	Description
0	No effect.
1	Enable the Transmit and Receive FIFOs

Bit 26 – LOCKCLR Lock Clear

Value	Description
0	No effect.
1	Clear the TWI FSM lock.

Bit 24 – THRCLR Transmit Holding Register Clear

Value	Description
0	No effect.
1	Clear the Transmit Holding register and set TXRDY, TXCOMP flags.

Bit 17 – ACMDIS Alternative Command Mode Disable

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	No effect.
1	Alternative Command mode disabled.

Bit 16 – ACMEN Alternative Command Mode Enable

Value	Description
0	No effect.
1	Alternative Command mode enabled.

Bit 15 – CLEAR Bus CLEAR Command

When TWD (SDA)=0, the Bus Clear command must be performed via the PIO. When TWCK=0, no Bus Clear command can be issued.

Value	Description
0	No effect.
1	When Host mode is enabled and TWD (SDA)=1, sends a Bus Clear command.

Bit 14 – PECRQ PEC Request

Value	Description
0	No effect.
1	A PEC check or transmission is requested.

Bit 13 – PECDIS Packet Error Checking Disable

Value	Description
0	No effect.
1	SMBus PEC (CRC) generation and check disabled.

Bit 12 – PECEN Packet Error Checking Enable

Value	Description
0	No effect.
1	SMBus PEC (CRC) generation and check enabled.

Bit 11 – SMBDIS SMBus Mode Disabled

Value	Description
0	No effect.
1	SMBus mode disabled.

Bit 10 – SMBEN SMBus Mode Enabled

Value	Description
0	No effect.
1	If SMBDIS = 0, SMBus mode enabled.

Bit 9 – HSDIS TWI High-Speed Mode Disabled

Value	Description
0	No effect.
1	High-speed mode disabled.

Bit 8 – HSEN TWI High-Speed Mode Enabled

Value	Description
0	No effect.
1	High-speed mode enabled.

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 7 – SWRST Software Reset

Value	Description
0	No effect.
1	Equivalent to a system reset.

Bit 6 – QUICK SMBus Quick Command

Value	Description
0	No effect.
1	If Host mode is enabled, an SMBus Quick Command is sent.

Bit 5 – SVDIS TWI Client Mode Disabled

Value	Description
0	No effect.
1	Client mode is disabled. The shifter and holding characters (if it contains data) are transmitted in the case of a read operation. In a write operation, the character being transferred must be completely received before disabling.

Bit 4 – SVEN TWI Client Mode Enabled

Switching from Host to Client mode is only permitted when TXCOMP = 1.

Value	Description
0	No effect.
1	Enables Client mode (SVDIS must be written to 0).

Bit 3 – MSDIS TWI Host Mode Disabled

Value	Description
0	No effect.
1	Host mode is disabled, all pending data is transmitted. The shifter and holding characters (if it contains data) are transmitted in case of write operation. In read operation, the character being transferred must be completely received before disabling.

Bit 2 – MSEN TWI Host Mode Enabled

Switching from Client to Host mode is only permitted when TXCOMP = 1.

Value	Description
0	No effect.
1	Enables Host mode (MSDIS must be written to 0).

Bit 1 – STOP Send a STOP Condition

Value	Description
0	No effect.
1	STOP condition is sent just after completing the current byte transmission in Host Read mode. – In single data byte host read, both START and STOP must be set. – In multiple data bytes host read, the STOP must be set after the last data received but one. – In Host Read mode, if a NACK bit is received, the STOP is automatically performed. – In host data write operation, a STOP condition will be sent after the transmission of the current data is finished.

Bit 0 – START Send a START Condition

This action is necessary when the TWI peripheral needs to read data from a client. When configured in Host mode with a write operation, a frame is sent as soon as the user writes a character in the Transmit Holding register (FLEX_TWI_THR).

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	No effect.
1	A frame beginning with a START bit is transmitted according to the features defined in the TWI Host Mode register (FLEX_TWI_MMR).

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.56. TWI Control Register (FIFO_ENABLED)

Name: FLEX_TWI_CR (FIFO_ENABLED)
Offset: 0x600
Reset: –
Property: Write-only

If FIFO is enabled (FLEX_US_CR.FIFOEN=1), see TWI Multiple Data Access for details.

Bit	31	30	29	28	27	26	25	24
			FIFODIS	FIFOEN		TXFLCLR	RXFCLR	TXFCLR
Access			W	W		W	W	W
Reset			–	–		–	–	–

Bit	23	22	21	20	19	18	17	16
							ACMDIS	ACMEN
Access							W	W
Reset							–	–

Bit	15	14	13	12	11	10	9	8
	CLEAR	PECRQ	PECDIS	PECEN	SMBDIS	SMBEN	HSDIS	HSEN
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	SWRST	QUICK	SVDIS	SVEN	MSDIS	MSEN	STOP	START
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 29 – FIFODIS FIFO Disable

Value	Description
0	No effect.
1	Disable the Transmit and Receive FIFOs.

Bit 28 – FIFOEN FIFO Enable

Value	Description
0	No effect.
1	Enable the Transmit and Receive FIFOs.

Bit 26 – TXFLCLR Transmit FIFO Lock CLEAR

Value	Description
0	No effect.
1	Clears the Transmit FIFO Lock.

Bit 25 – RXFCLR Receive FIFO Clear

Value	Description
0	No effect.
1	Empties the Receive FIFO.

Bit 24 – TXFCLR Transmit FIFO Clear

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	No effect.
1	Empties the Transmit FIFO.

Bit 17 – ACMDIS Alternative Command Mode Disable

Value	Description
0	No effect.
1	Alternative Command mode disabled.

Bit 16 – ACMEN Alternative Command Mode Enable

Value	Description
0	No effect.
1	Alternative Command mode enabled.

Bit 15 – CLEAR Bus CLEAR Command

Value	Description
0	No effect.
1	If Host mode is enabled, send a bus clear command.

Bit 14 – PECRQ PEC Request

Value	Description
0	No effect.
1	A PEC check or transmission is requested.

Bit 13 – PECDIS Packet Error Checking Disable

Value	Description
0	No effect.
1	SMBus PEC (CRC) generation and check disabled.

Bit 12 – PECEN Packet Error Checking Enable

Value	Description
0	No effect.
1	SMBus PEC (CRC) generation and check enabled.

Bit 11 – SMBDIS SMBus Mode Disabled

Value	Description
0	No effect.
1	SMBus mode disabled.

Bit 10 – SMBEN SMBus Mode Enabled

Value	Description
0	No effect.
1	If SMBDIS = 0, SMBus mode enabled.

Bit 9 – HSDIS TWI High-Speed Mode Disabled

Value	Description
0	No effect.
1	High-speed mode disabled.

Bit 8 – HSEN TWI High-Speed Mode Enabled

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	No effect.
1	High-speed mode enabled.

Bit 7 – SWRST Software Reset

Value	Description
0	No effect.
1	Equivalent to a system reset.

Bit 6 – QUICK SMBus Quick Command

Value	Description
0	No effect.
1	If Host mode is enabled, a SMBus Quick Command is sent.

Bit 5 – SVDIS TWI Client Mode Disabled

Value	Description
0	No effect.
1	Client mode is disabled. The shifter and holding characters (if it contains data) are transmitted in the case of a read operation. In a write operation, the character being transferred must be completely received before disabling.

Bit 4 – SVEN TWI Client Mode Enabled

Switching from Host to Client mode is only permitted when TXCOMP = 1.

Value	Description
0	No effect.
1	Enables Client mode (SVDIS must be written to 0).

Bit 3 – MSDIS TWI Host Mode Disabled

Value	Description
0	No effect.
1	Host mode is disabled, all pending data is transmitted. The shifter and holding characters (if it contains data) are transmitted in the case of a write operation. In a read operation, the character being transferred must be completely received before disabling.

Bit 2 – MSEN TWI Host Mode Enabled

Switching from Client to Host mode is only permitted when TXCOMP = 1.

Value	Description
0	No effect.
1	Enables Host mode (MSDIS must be written to 0).

Bit 1 – STOP Send a STOP Condition

Value	Description
0	No effect.
1	STOP condition is sent just after completing the current byte transmission in Host Read mode. – In single data byte host read, both START and STOP must be set. – In multiple data bytes host read, the STOP must be set after the last data received but one. – In Host Read mode, if a NACK bit is received, the STOP is automatically performed. – In host data write operation, a STOP condition will be sent after the transmission of the current data is finished.

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 0 – START Send a START Condition

This action is necessary when the TWI peripheral needs to read data from a client. When configured in Host mode with a write operation, a frame is sent as soon as the user writes a character in the Transmit Holding register (FLEX_TWI_THR).

Value	Description
0	No effect.
1	A frame beginning with a START bit is transmitted according to the features defined in the TWI Host Mode register (FLEX_TWI_MMR).

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.57. TWI Host Mode Register

Name: FLEX_TWI_MMR
Offset: 0x604
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
								NOAP
Access								R/W
Reset								0

Bit	23	22	21	20	19	18	17	16
		DADR[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
				MREAD			IADRSZ[1:0]	
Access				R/W			R/W	R/W
Reset				0			0	0

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 24 – NOAP No Auto-Stop On NACK Error

Value	Description
0	A stop condition is sent automatically upon Not-Acknowledge error detection.
1	No automatic action is performed upon Not-Acknowledge error detection.

Bits 22:16 – DADR[6:0] Device Address
The device address is used to access client devices in Read or Write mode. Those bits are only used in Host mode.

Bit 12 – MREAD Host Read Direction

Value	Description
0	Host write direction.
1	Host read direction.

Bits 9:8 – IADRSZ[1:0] Internal Device Address Size

Value	Name	Description
0	NONE	No internal device address
1	1_BYTE	One-byte internal device address
2	2_BYTE	Two-byte internal device address
3	3_BYTE	Three-byte internal device address

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.58. TWI Client Mode Register

Name: FLEX_TWI_SMR
Offset: 0x608
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the TWI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	DATAMEN	SADR3EN	SADR2EN	SADR1EN				
Access	R/W	R/W	R/W	R/W				
Reset	0	0	0	0				

Bit	23	22	21	20	19	18	17	16
		SADR[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
		MASK[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SNIFF	SCLWSDIS	BSEL	SADAT	SMHH	SMDA		NACKEN
Access	R/W	R/W	R/W	R/W	R/W	R/W		R/W
Reset	0	0	0	0	0	0		0

Bit 31 – DATAMEN Data Matching Enable

Value	Description
0	Data matching on first received data is disabled.
1	Data matching on first received data is enabled.

Bit 30 – SADR3EN Client Address 3 Enable

Value	Description
0	Client address 3 matching is disabled.
1	Client address 3 matching is enabled.

Bit 29 – SADR2EN Client Address 2 Enable

Value	Description
0	Client address 2 matching is disabled.
1	Client address 2 matching is enabled.

Bit 28 – SADR1EN Client Address 1 Enable

Value	Description
0	Client address 1 matching is disabled.
1	Client address 1 matching is enabled.

© 2026 Microchip Technology Inc. and its subsidiaries

Bits 22:16 – SADR[6:0] Client Address

The client device address is used in Client mode in order to be accessed by host devices in Read or Write mode.
SADR must be programmed before enabling Client mode or after a general call. Writes at other times have no effect.

Bits 14:8 – MASK[6:0] Client Address Mask

A mask can be applied on the client device address in Client mode in order to allow multiple address answer. For each bit of the MASK field set to one, the corresponding SADR bit will be masked.
If the MASK field is set to 0, no mask is applied to the SADR field.

Bit 7 – SNIFF Client Sniffer Mode

Value	Description
0	Client Sniffer mode is disabled.
1	Client Sniffer mode is enabled.

Bit 6 – SCLWSDIS Clock Wait State Disable

Value	Description
0	No effect.
1	Clock stretching disabled in Client mode, OVRE and UNRE will indicate overrun and underrun.

Bit 5 – BSEL TWI Bus Selection

Value	Description
0	TWI analyzes the TWCK and TWD pins from its TWI bus.
1	

Bit 4 – SADAT Client Address Treated as Data

When Client Sniffer Mode is enabled, the client address is always received as data in FLEX_TWI_RHR and SADAT has no effect.

Value	Description
0	Client address is handled normally (will not trig RXRDY flag and will not fill FLEX_TWI_RHR upon reception).
1	Client address is handled as data field, RXRDY will be set and FLEX_TWI_RHR filled upon client address reception.

Bit 3 – SMHH SMBus Host Header

Value	Description
0	Acknowledge of the SMBus Host Header disabled.
1	Acknowledge of the SMBus Host Header enabled.

Bit 2 – SMDA SMBus Default Address

Value	Description
0	Acknowledge of the SMBus Default Address disabled.
1	Acknowledge of the SMBus Default Address enabled.

Bit 0 – NACKEN Client Receiver Data Phase NACK Enable

Value	Description
0	Normal value to be returned in the ACK cycle of the data phase in Client Receiver mode.
1	NACK value to be returned in the ACK cycle of the data phase in Client Receiver mode.

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.59. TWI Internal Address Register

Name: FLEX_TWI_IADR
Offset: 0x60C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	IADR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	IADR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	IADR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – IADR[23:0] Internal Address
0, 1, 2 or 3 bytes depending on IADRSZ.

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.60. TWI Clock Waveform Generator Register

Name: FLEX_TWI_CWGR
Offset: 0x610
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the TWI Write Protection Mode Register.

FLEX_TWI_CWGR is only used in Host mode.

Bit	31	30	29	28	27	26	25	24
			HOLD[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
				BRSRCCLK		CKDIV[2:0]		
Access				R/W		R/W	R/W	R/W
Reset				0		0	0	0

Bit	15	14	13	12	11	10	9	8
	CHDIV[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CLDIV[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 29:24 – HOLD[5:0] TWD Hold Time Versus TWCK Falling
If High-speed mode is selected TWD is internally modified on the TWCK falling edge to meet the I2C specified maximum hold time, else if High-speed mode is not configured TWD is kept unchanged after TWCK falling edge for a period of (HOLD + 3) × $t_{peripheral\ clock}$.

Bit 20 – BRSRCCLK Bit Rate Source Clock

Value	Name	Description
0	PERIPH_CLK	The peripheral clock is the source clock for the bit rate generation.
1	GCLK	GCLK is the source clock for the bit rate generation, thus the bit rate can be independent of the core/peripheral clock.

Bits 18:16 – CKDIV[2:0] Clock Divider
The CKDIV is used to increase both SCL high and low periods.

Bits 15:8 – CHDIV[7:0] Clock High Divider
The SCL high period is defined as follows:
If FLEX_TWI_FILTR.FILT = 0

- If BRSRCCLK = 0: CHDIV = $((t_{high}/t_{peripheral\ clock}) - 3)/2^{CKDIV}$
- If BRSRCCLK = 1: CHDIV = $(t_{high}/t_{ext\_ck})/2^{CKDIV}$

If FLEX_TWI_FILTR.FILT = 1

- If BRSRCCLK = 0: CHDIV = $((t_{high}/t_{peripheral\ clock}) - 3 - (THRES+1))/2^{CKDIV}$

© 2026 Microchip Technology Inc. and its subsidiaries

- If BRSRCCLK = 1: CHDIV = ((t_{high} - (THRES+1) * $t_{peripheral\ clock}$)/t_{ext_ck})/2^{CKDIV}

Bits 7:0 – CLDIV[7:0] Clock Low Divider

The SCL low period is defined as follows:

If FLEX_TWI_FILTR.FILT = 0

- If BRSRCCLK = 0: CLDIV = ((t_{low}/$t_{peripheral\ clock}$) - 3)/2^{CKDIV}
- If BRSRCCLK = 1: CLDIV = (t_{low}/t_{ext_ck})/2^{CKDIV}

If FLEX_TWI_FILTR.FILT = 1

- If BRSRCCLK = 0: CLDIV = ((t_{low}/$t_{peripheral\ clock}$) - 3 - (THRES+1))/2^{CKDIV}
- If BRSRCCLK = 1: CLDIV = ((t_{low} - (THRES+1) * $t_{peripheral\ clock}$)/t_{ext_ck})/2^{CKDIV}

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.61. TWI Status Register

Name: FLEX_TWI_SR
Offset: 0x620
Reset: 0x0300F009
Property: Read-only

Bit	31	30	29	28	27	26	25	24
						SR	SDA	SCL
Access						R	R	R
Reset						0	1	1

Bit	23	22	21	20	19	18	17	16
	LOCK		SMBHHM	SMBDAM	PECERR	TOUT	SMBAF	MCACK
Access	R		R	R	R	R	R	R
Reset	0		0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TXBUFE	RXBUFF	ENDTX	ENDRX	EOSACC	SCLWS	ARBLST	NACK
Access	R	R	R	R	R	R	R	R
Reset	1	1	1	1	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	UNRE	OVRE	GACC	SVACC	SVREAD	TXRDY	RXRDY	TXCOMP
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	1	0	0	1

Bit 26 – SR Start Repeated

Value	Description
0	No repeated start has been detected since last FLEX_TWI_SR read.
1	At least one repeated start has been detected since last FLEX_TWI_SR read.

Bit 25 – SDA SDA Line Value

Value	Description
0	SDA line sampled value is ‘0’.
1	SDA line sampled value is ‘1’.

Bit 24 – SCL SCL Line Value

Value	Description
0	SCL line sampled value is ‘0’.
1	SCL line sampled value is ‘1.’

Bit 23 – LOCK TWI Lock Due to Frame Errors

Value	Description
0	The TWI is not locked.
1	The TWI is locked due to frame errors (see Handling Errors in Alternative Command and TWI FIFOs).

Bit 21 – SMBHHM SMBus Host Header Address Match (cleared on read)

Value	Description
0	No SMBus Host Header Address received.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
1	A SMBus Host Header Address was received.

Bit 20 – SMBDAM SMBus Default Address Match (cleared on read)

Value	Description
0	No SMBus Default Address received.
1	A SMBus Default Address was received.

Bit 19 – PECERR PEC Error (cleared on read)

Value	Description
0	No SMBus PEC error occurred.
1	A SMBus PEC error occurred.

Bit 18 – TOUT Timeout Error (cleared on read)

Value	Description
0	No SMBus timeout occurred.
1	SMBus timeout occurred.

Bit 17 – SMBAF SMBus Alert Flag (cleared on read)

Value	Description
0	No SMBus client drives the SMBALERT line.
1	At least one SMBus client drives the SMBALERT line.

Bit 16 – MCACK Host Code Acknowledge (cleared on read)
MACK used in Client mode:

Value	Description
0	No host code has been received.
1	A host code has been received.

Bit 15 – TXBUFE TX Buffer Empty (cleared by writing FLEX_TWI_TCR or FLEX_TWI_TNCR)
FLEX_TWI_TCR and FLEX_TWI_TNCR are PDC registers.

Value	Description
0	FLEX_TWI_TCR or FLEX_TWI_TNCR has a value other than 0.
1	Both FLEX_TWI_TCR and FLEX_TWI_TNCR have a value of 0.

Bit 14 – RXBUFF RX Buffer Full (cleared by writing FLEX_TWI_RCR or FLEX_TWI_RNCR)
FLEX_TWI_RCR and FLEX_TWI_RNCR are PDC registers.

Value	Description
0	FLEX_TWI_RCR or FLEX_TWI_RNCR has a value other than 0.
1	Both FLEX_TWI_RCR and FLEX_TWI_RNCR have a value of 0.

Bit 13 – ENDTX End of TX Buffer (cleared by writing FLEX_TWI_TCR or FLEX_TWI_TNCR)
FLEX_TWI_TCR and FLEX_TWI_TNCR are PDC registers.

Value	Description
0	The Transmit Counter Register has not reached 0 since the last write in FLEX_TWI_TCR or FLEX_TWI_TNCR.
1	The Transmit Counter Register has reached 0 since the last write in FLEX_TWI_TCR or FLEX_TWI_TNCR.

Bit 12 – ENDRX End of RX Buffer (cleared by writing FLEX_TWI_RCR or FLEX_TWI_RNCR)
FLEX_TWI_RCR and FLEX_TWI_RNCR are PDC registers.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	The Receive Counter Register has not reached 0 since the last write in FLEX_TWI_RCR or FLEX_TWI_RNCR.
1	The Receive Counter Register has reached 0 since the last write in FLEX_TWI_RCR or FLEX_TWI_RNCR.

Bit 11 – EOSACC End Of Client Access (cleared on read)

This bit is only used in Client mode.
EOSACC behavior can be seen in figures Repeated Start and Reversal from Read Mode to Write Mode and Repeated Start and Reversal from Write Mode to Read Mode.

Value	Description
0	A client access is being performed.
1	Client Access is finished. End Of Client Access is automatically set as soon as SVACC is reset.

Bit 10 – SCLWS Clock Wait State

This bit is only used in Client mode.
SCLWS behavior can be seen in figures Clock Stretching in Read Mode and Clock Stretching in Write Mode.

Value	Description
0	The clock is not stretched.
1	The clock is stretched. FLEX_TWI_THR / FLEX_TWI_RHR buffer is not filled / emptied before the transmission / reception of a new character.

Bit 9 – ARBLST Arbitration Lost (cleared on read)

This bit is only used in Host mode.

Value	Description
0	Arbitration won.
1	Arbitration lost. Another host of the TWI bus has won the multi-host arbitration. TXCOMP is set at the same time.

Bit 8 – NACK Not Acknowledged (cleared on read)

NACK used in Host mode:
0: Each data byte has been correctly received by the far-end side TWI client component.
1: A data or address byte has not been acknowledged by the client component. Set at the same time as TXCOMP.
NACK used in Client Read mode:
0: Each data byte has been correctly received by the host.
1: In Read mode, a data byte has not been acknowledged by the host. When NACK is set, the user must not fill FLEX_TWI_THR even if TXRDY is set, because it means that the host will stop the data transfer or reinitiate it.
Note that in Client Write mode, all data are acknowledged by the TWI.

Bit 7 – UNRE Underrun Error (cleared on read)

This bit is only used in Client mode if clock stretching is disabled.

Value	Description
0	FLEX_TWI_THR has been filled on time.
1	FLEX_TWI_THR has not been filled on time.

Bit 6 – OVRE Overrun Error (cleared on read)

This bit is only used in Client mode if clock stretching is disabled.

Value	Description
0	FLEX_TWI_RHR has not been loaded while RXRDY was set.
1	FLEX_TWI_RHR has been loaded while RXRDY was set. Reset by read in FLEX_TWI_SR when TXCOMP is set.

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 5 – GACC General Call Access (cleared on read)

This bit is only used in Client mode.
GACC behavior can be seen in figure Host Performs a General Call.

Value	Description
0	No general call has been detected.
1	A general call has been detected. After the detection of general call, if need be, the user may acknowledge this access and decode the following bytes and respond according to the value of the bytes.

Bit 4 – SVACC Client Access

This bit is only used in Client mode.
SVACC behavior can be seen in figures Read Access Ordered by a Host, Write Access Ordered by a Host, Repeated Start and Reversal from Read Mode to Write Mode and Repeated Start and Reversal from Write Mode to Read Mode.

Value	Description
0	TWI is not addressed. SVACC is automatically cleared after a NACK or a STOP condition is detected.
1	Indicates that the address decoding sequence has matched (a host has sent SADR). SVACC remains high until a NACK or a STOP condition is detected.

Bit 3 – SVREAD Client Read

This bit is only used in Client mode. When SVACC is low (no client access has been detected) SVREAD is irrelevant.
SVREAD behavior can be seen in figures Read Access Ordered by a Host, Write Access Ordered by a Host, Repeated Start and Reversal from Read Mode to Write Mode and Repeated Start and Reversal from Write Mode to Read Mode.

Value	Description
0	Indicates that a write access is performed by a host.
1	Indicates that a read access is performed by a host.

Bit 2 – TXRDY Transmit Holding Register Ready (cleared by writing FLEX_TWI_THR)

TXRDY used in Host mode:
0: The transmit holding register has not been transferred into the internal shifter. Set to 0 when writing into FLEX_TWI_THR.
1: As soon as a data byte is transferred from FLEX_TWI_THR to internal shifter or if a NACK error is detected, TXRDY is set at the same time as TXCOMP and NACK. TXRDY is also set when MSEN is set (enables TWI).
TXRDY behavior in Host mode can be seen in figures Host Write with One Data Byte,Host Write with Multiple Data Bytes and Host Write with One Byte Internal Address and Multiple Data Bytes.
TXRDY used in Client mode:
0: As soon as data is written in FLEX_TWI_THR, until this data has been transmitted and acknowledged (ACK or NACK).
1: Indicates that FLEX_TWI_THR is empty and that data has been transmitted and acknowledged.
If TXRDY is high and if a NACK has been detected, the transmission will be stopped. Thus when TRDY = NACK = 1, the user must not fill FLEX_TWI_THR to avoid losing it.
TXRDY behavior in Client mode can be seen in figures Read Access Ordered by a Host, Clock Stretching in Read Mode, Repeated Start and Reversal from Read Mode to Write Mode and Repeated Start and Reversal from Write Mode to Read Mode.
When FIFOs are enabled:
0: Transmit FIFO is full and cannot accept more data.
1: Transmit FIFO is not full; one or more data can be written according to TXRDYM field configuration.
TXRDY behavior with FIFOs enabled is illustrated in TXRDY and RXRDY Behavior.

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 1 – RXRDY Receive Holding Register Ready (cleared when reading FLEX_TWI_RHR)

When FIFOs are disabled:
0: No character has been received since the last FLEX_TWI_RHR read operation.
1: A byte has been received in FLEX_TWI_RHR since the last read.
RXRDY behavior in Host mode can be seen in figure Host Read with Multiple Data Bytes.
RXRDY behavior in Client mode can be seen in figures Write Access Ordered by a Host, Clock Stretching in Write Mode, Repeated Start and Reversal from Read Mode to Write Mode and Repeated Start and Reversal from Write Mode to Read Mode.
When FIFOs are enabled:
0: Receive FIFO is empty; no data to read.
1: At least one unread data is in the Receive FIFO.
RXRDY behavior with FIFO enabled is illustrated in TXRDY and RXRDY Behavior.

Bit 0 – TXCOMP Transmission Completed (cleared by writing FLEX_TWI_THR)

TXCOMP used in Host mode:
0: During the length of the current frame.
1: When both the holding register and the internal shifter are empty and STOP condition has been sent.
TXCOMP behavior in Host mode can be seen in figures Host Write with One Byte Internal Address and Multiple Data Bytes and Host Read with Multiple Data Bytes.
TXCOMP used in Client mode:
0: As soon as a Start is detected.
1: After a Stop or a Repeated Start + an address different from SADR is detected.
TXCOMP behavior in Client mode can be seen in figures Clock Stretching in Read Mode, Clock Stretching in Write Mode, Repeated Start and Reversal from Read Mode to Write Mode and Repeated Start and Reversal from Write Mode to Read Mode.

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.62. TWI Status Register (FIFO ENABLED)

Name: FLEX_TWI_SR (FIFO_ENABLED)
Offset: 0x620
Reset: 0x0300F009
Property: Read-only

If FIFO is enabled (FLEX_US_CR.FIFOEN bit), see TWI Multiple Data Access for details.

Bit	31	30	29	28	27	26	25	24
						SR	SDA	SCL
Access						R	R	R
Reset						0	1	1

Bit	23	22	21	20	19	18	17	16
	TXFLOCK		SMBHHM	SMBDAM	PECERR	TOUT	SMBAF	MCACK
Access	R		R	R	R	R	R	R
Reset	0		0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TXBUFE	RXBUFF	ENDTX	ENDRX	EOSACC	SCLWS	ARBLST	NACK
Access	R	R	R	R	R	R	R	R
Reset	1	1	1	1	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	UNRE	OVRE	GACC	SVACC	SVREAD	TXRDY	RXRDY	TXCOMP
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	1	0	0	1

Bit 26 – SR Start Repeated

Value	Description
0	No repeated start has been detected since last FLEX_TWI_SR read.
1	At least one repeated start has been detected since last FLEX_TWI_SR read.

Bit 25 – SDA SDA Line Value

Value	Description
0	SDA line sampled value is ‘0’.
1	SDA line sampled value is ‘1’.

Bit 24 – SCL SCL Line Value

Value	Description
0	SCL line sampled value is ‘0’.
1	SCL line sampled value is ‘1.’

Bit 23 – TXFLOCK Transmit FIFO Lock

Value	Description
0	The Transmit FIFO is not locked.
1	The Transmit FIFO is locked.

Bit 21 – SMBHHM SMBus Host Header Address Match (cleared on read)

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	No SMBus Host Header Address received.
1	A SMBus Host Header Address was received.

Bit 20 – SMBDAM SMBus Default Address Match (cleared on read)

Value	Description
0	No SMBus Default Address received.
1	A SMBus Default Address was received.

Bit 19 – PECERR PEC Error (cleared on read)

Value	Description
0	No SMBus PEC error occurred.
1	A SMBus PEC error occurred.

Bit 18 – TOUT Timeout Error (cleared on read)

Value	Description
0	No SMBus timeout occurred.
1	SMBus timeout occurred.

Bit 17 – SMBAF SMBus Alert Flag (cleared on read)

Value	Description
0	No SMBus client drives the SMBALERT line.
1	At least one SMBus client drives the SMBALERT line.

Bit 16 – MCACK Host Code Acknowledge (cleared on read)
MACK used in Client mode:

Value	Description
0	No host code has been received.
1	A host code has been received.

Bit 15 – TXBUFE TX Buffer Empty (cleared by writing FLEX_TWI_TCR or FLEX_TWI_TNCR)
FLEX_TWI_TCR and FLEX_TWI_TNCR are PDC registers.

Value	Description
0	FLEX_TWI_TCR or FLEX_TWI_TNCR has a value other than 0.
1	Both FLEX_TWI_TCR and FLEX_TWI_TNCR have a value of 0.

Bit 14 – RXBUFF RX Buffer Full (cleared by writing FLEX_TWI_RCR or FLEX_TWI_RNCR)
FLEX_TWI_RCR and FLEX_TWI_RNCR are PDC registers.

Value	Description
0	FLEX_TWI_RCR or FLEX_TWI_RNCR has a value other than 0.
1	Both FLEX_TWI_RCR and FLEX_TWI_RNCR have a value of 0.

Bit 13 – ENDTX End of TX Buffer (cleared by writing FLEX_TWI_TCR or FLEX_TWI_TNCR)
FLEX_TWI_TCR and FLEX_TWI_TNCR are PDC registers.

Value	Description
0	The Transmit Counter Register has not reached 0 since the last write in FLEX_TWI_TCR or FLEX_TWI_TNCR.
1	The Transmit Counter Register has reached 0 since the last write in FLEX_TWI_TCR or FLEX_TWI_TNCR.

Bit 12 – ENDRX End of RX Buffer (cleared by writing FLEX_TWI_RCR or FLEX_TWI_RNCR)
FLEX_TWI_RCR and FLEX_TWI_RNCR are PDC registers.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	The Receive Counter Register has not reached 0 since the last write in FLEX_TWI_RCR or FLEX_TWI_RNCR.
1	The Receive Counter Register has reached 0 since the last write in FLEX_TWI_RCR or FLEX_TWI_RNCR.

Bit 11 – EOSACC End Of Client Access (cleared on read)

This bit is only used in Client mode.
EOSACC behavior can be seen in figures Repeated Start and Reversal from Read Mode to Write Mode and Repeated Start and Reversal from Write Mode to Read Mode.

Value	Description
0	A Client Access is being performed.
1	Client Access is finished. End Of Client Access is automatically set as soon as SVACC is reset.

Bit 10 – SCLWS Clock Wait State

This bit is only used in Client mode.
SCLWS behavior can be seen in figures Clock Stretching in Read Mode and Clock Stretching in Write Mode.

Value	Description
0	The clock is not stretched.
1	The clock is stretched. FLEX_TWI_THR / FLEX_TWI_RHR buffer is not filled / emptied before the transmission / reception of a new character.

Bit 9 – ARBLST Arbitration Lost (cleared on read)

This bit is only used in Host mode.

Value	Description
0	Arbitration won.
1	Arbitration lost. Another host of the TWI bus has won the multi-host arbitration. TXCOMP is set at the same time.

Bit 8 – NACK Not Acknowledged (cleared on read)

NACK used in Host mode:
0: Each data byte has been correctly received by the far-end side TWI client component.
1: A data or address byte has not been acknowledged by the client component. Set at the same time as TXCOMP.
NACK used in Client Read mode:
0: Each data byte has been correctly received by the host.
1: In Read mode, a data byte has not been acknowledged by the host. When NACK is set the user must not fill FLEX_TWI_THR even if TXRDY is set, because it means that the host will stop the data transfer or re initiate it.
Note that in Client Write mode all data are acknowledged by the TWI.

Bit 7 – UNRE Underrun Error (cleared on read)

This bit is only used in Client mode if clock stretching is disabled.

Value	Description
0	FLEX_TWI_THR has been filled on time.
1	FLEX_TWI_THR has not been filled on time.

Bit 6 – OVRE Overrun Error (cleared on read)

This bit is only used in Client mode if clock stretching is disabled.

Value	Description
0	FLEX_TWI_RHR has not been loaded while RXRDY was set.
1	FLEX_TWI_RHR has been loaded while RXRDY was set. Reset by read in FLEX_TWI_SR when TXCOMP is set.

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 5 – GACC General Call Access (cleared on read)

This bit is only used in Client mode.
GACC behavior can be seen in figure Host Performs a General Call.

Value	Description
0	No general call has been detected.
1	A general call has been detected. After the detection of general call, if need be, the user may acknowledge this access and decode the following bytes and respond according to the value of the bytes.

Bit 4 – SVACC Client Access

This bit is only used in Client mode.
SVACC behavior can be seen in figures Read Access Ordered by a Host, Write Access Ordered by a Host, Repeated Start and Reversal from Read Mode to Write Mode and Repeated Start and Reversal from Write Mode to Read Mode.

Value	Description
0	TWI is not addressed. SVACC is automatically cleared after a NACK or a STOP condition is detected.
1	Indicates that the address decoding sequence has matched (a host has sent SADR). SVACC remains high until a NACK or a STOP condition is detected.

Bit 3 – SVREAD Client Read

This bit is only used in Client mode. When SVACC is low (no client access has been detected) SVREAD is irrelevant.
SVREAD behavior can be seen in figures Read Access Ordered by a Host, Write Access Ordered by a Host, Repeated Start and Reversal from Read Mode to Write Mode and Repeated Start and Reversal from Write Mode to Read Mode.

Value	Description
0	Indicates that a write access is performed by a host.
1	Indicates that a read access is performed by a host.

Bit 2 – TXRDY Transmit Holding Register Ready (cleared by writing FLEX_TWI_THR)

TXRDY used in Host mode:
0: The transmit holding register has not been transferred into the internal shifter. Set to 0 when writing into FLEX_TWI_THR.
1: As soon as a data byte is transferred from FLEX_TWI_THR to internal shifter or if a NACK error is detected, TXRDY is set at the same time as TXCOMP and NACK. TXRDY is also set when MSEN is set (enables TWI).
TXRDY behavior in Host mode can be seen in figures Host Write with One Data Byte,Host Write with Multiple Data Bytes and Host Write with One Byte Internal Address and Multiple Data Bytes.
TXRDY used in Client mode:
0: As soon as data is written in FLEX_TWI_THR, until this data has been transmitted and acknowledged (ACK or NACK).
1: Indicates that FLEX_TWI_THR is empty and that data has been transmitted and acknowledged.
If TXRDY is high and if a NACK has been detected, the transmission will be stopped. Thus when TRDY = NACK = 1, the user must not fill FLEX_TWI_THR to avoid losing it.
TXRDY behavior in Client mode can be seen in figures Read Access Ordered by a Host, Clock Stretching in Read Mode, Repeated Start and Reversal from Read Mode to Write Mode and Repeated Start and Reversal from Write Mode to Read Mode.
When FIFOs are enabled:
0: Transmit FIFO is full and cannot accept more data.
1: Transmit FIFO is not full; one or more data can be written according to TXRDYM field configuration.
TXRDY behavior with FIFOs enabled is illustrated in TXRDY and RXRDY Behavior.

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 1 – RXRDY Receive Holding Register Ready (cleared when reading FLEX_TWI_RHR)

When FIFOs are disabled:
0: No character has been received since the last FLEX_TWI_RHR read operation.
1: A byte has been received in FLEX_TWI_RHR since the last read.
RXRDY behavior in Host mode can be seen in figure Host Read with Multiple Data Bytes.
RXRDY behavior in Client mode can be seen in figures Write Access Ordered by a Host, Clock Stretching in Write Mode, Repeated Start and Reversal from Read Mode to Write Mode and Repeated Start and Reversal from Write Mode to Read Mode.
When FIFOs are enabled:
0: Receive FIFO is empty; no data to read.
1: At least one unread data is in the Receive FIFO.
RXRDY behavior with FIFO enabled is illustrated in TXRDY and RXRDY Behavior.

Bit 0 – TXCOMP Transmission Completed (cleared by writing FLEX_TWI_THR)

TXCOMP used in Host mode:
0: During the length of the current frame.
1: When both holding register and internal shifter are empty and STOP condition has been sent.
TXCOMP behavior in Host mode can be seen in figures Host Write with One Byte Internal Address and Multiple Data Bytes and Host Read with Multiple Data Bytes.
TXCOMP used in Client mode:
0: As soon as a Start is detected.
1: After a Stop or a Repeated Start + an address different from SADR is detected.
TXCOMP behavior in Client mode can be seen in figures Clock Stretching in Read Mode, Clock Stretching in Write Mode, Repeated Start and Reversal from Read Mode to Write Mode and Repeated Start and Reversal from Write Mode to Read Mode.

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.63. TWI Interrupt Enable Register

Name: FLEX_TWI_IER
Offset: 0x624
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the TWI Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			SMBHHM	SMBDAM	PECERR	TOUT		MCACK
Access			W	W	W	W		W
Reset			–	–	–	–		–

Bit	15	14	13	12	11	10	9	8
	TXBUFE	RXBUFF	ENDTX	ENDRX	EOSACC	SCL_WS	ARBLST	NACK
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	UNRE	OVRE	GACC	SVACC		TXRDY	RXRDY	TXCOMP
Access	W	W	W	W		W	W	W
Reset	–	–	–	–		–	–	–

Bit 21 – SMBHHM SMBus Host Header Address Match Interrupt Enable

Bit 20 – SMBDAM SMBus Default Address Match Interrupt Enable

Bit 19 – PECERR PEC Error Interrupt Enable

Bit 18 – TOUT Timeout Error Interrupt Enable

Bit 16 – MCACK Host Code Acknowledge Interrupt Enable

Bit 15 – TXBUFE Transmit Buffer Empty Interrupt Enable

Bit 14 – RXBUFF Receive Buffer Full Interrupt Enable

Bit 13 – ENDTX End of Transmit Buffer Interrupt Enable

Bit 12 – ENDRX End of Receive Buffer Interrupt Enable

Bit 11 – EOSACC End Of Client Access Interrupt Enable

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 10 – SCL_WS Clock Wait State Interrupt Enable

Bit 9 – ARBLST Arbitration Lost Interrupt Enable

Bit 8 – NACK Not Acknowledge Interrupt Enable

Bit 7 – UNRE Underrun Error Interrupt Enable

Bit 6 – OVRE Overrun Error Interrupt Enable

Bit 5 – GACC General Call Access Interrupt Enable

Bit 4 – SVACC Client Access Interrupt Enable

Bit 2 – TXRDY Transmit Holding Register Ready Interrupt Enable

Bit 1 – RXRDY Receive Holding Register Ready Interrupt Enable

Bit 0 – TXCOMP Transmission Completed Interrupt Enable

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.64. TWI Interrupt Disable Register

Name: FLEX_TWI_IDR
Offset: 0x628
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the TWI Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			SMBHHM	SMBDAM	PECERR	TOUT		MCACK
Access			W	W	W	W		W
Reset			–	–	–	–		–

Bit	15	14	13	12	11	10	9	8
	TXBUFE	RXBUFF	ENDTX	ENDRX	EOSACC	SCL_WS	ARBLST	NACK
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	UNRE	OVRE	GACC	SVACC		TXRDY	RXRDY	TXCOMP
Access	W	W	W	W		W	W	W
Reset	–	–	–	–		–	–	–

Bit 21 – SMBHHM SMBus Host Header Address Match Interrupt Disable

Bit 20 – SMBDAM SMBus Default Address Match Interrupt Disable

Bit 19 – PECERR PEC Error Interrupt Disable

Bit 18 – TOUT Timeout Error Interrupt Disable

Bit 16 – MCACK Host Code Acknowledge Interrupt Disable

Bit 15 – TXBUFE Transmit Buffer Empty Interrupt Disable

Bit 14 – RXBUFF Receive Buffer Full Interrupt Disable

Bit 13 – ENDTX End of Transmit Buffer Interrupt Disable

Bit 12 – ENDRX End of Receive Buffer Interrupt Disable

Bit 11 – EOSACC End Of Client Access Interrupt Disable

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 10 – SCL_WS Clock Wait State Interrupt Disable

Bit 9 – ARBLST Arbitration Lost Interrupt Disable

Bit 8 – NACK Not Acknowledge Interrupt Disable

Bit 7 – UNRE Underrun Error Interrupt Disable

Bit 6 – OVRE Overrun Error Interrupt Disable

Bit 5 – GACC General Call Access Interrupt Disable

Bit 4 – SVACC Client Access Interrupt Disable

Bit 2 – TXRDY Transmit Holding Register Ready Interrupt Disable

Bit 1 – RXRDY Receive Holding Register Ready Interrupt Disable

Bit 0 – TXCOMP Transmission Completed Interrupt Disable

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.65. TWI Interrupt Mask Register

Name: FLEX_TWI_IMR
Offset: 0x62C
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is disabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			SMBHHM	SMBDAM	PECERR	TOUT		MCACK
Access			R	R	R	R		R
Reset			0	0	0	0		0

Bit	15	14	13	12	11	10	9	8
	TXBUFE	RXBUFF	ENDTX	ENDRX	EOSACC	SCL_WS	ARBLST	NACK
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	UNRE	OVRE	GACC	SVACC		TXRDY	RXRDY	TXCOMP
Access	R	R	R	R		R	R	R
Reset	0	0	0	0		0	0	0

Bit 21 – SMBHHM SMBus Host Header Address Match Interrupt Mask

Bit 20 – SMBDAM SMBus Default Address Match Interrupt Mask

Bit 19 – PECERR PEC Error Interrupt Mask

Bit 18 – TOUT Timeout Error Interrupt Mask

Bit 16 – MCACK Host Code Acknowledge Interrupt Mask

Bit 15 – TXBUFE Transmit Buffer Empty Interrupt Mask

Bit 14 – RXBUFF Receive Buffer Full Interrupt Mask

Bit 13 – ENDTX End of Transmit Buffer Interrupt Mask

Bit 12 – ENDRX End of Receive Buffer Interrupt Mask

Bit 11 – EOSACC End Of Client Access Interrupt Mask

Bit 10 – SCL_WS Clock Wait State Interrupt Mask

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 9 – ARBLST Arbitration Lost Interrupt Mask

Bit 8 – NACK Not Acknowledge Interrupt Mask

Bit 7 – UNRE Underrun Error Interrupt Mask

Bit 6 – OVRE Overrun Error Interrupt Mask

Bit 5 – GACC General Call Access Interrupt Mask

Bit 4 – SVACC Client Access Interrupt Mask

Bit 2 – TXRDY Transmit Holding Register Ready Interrupt Mask

Bit 1 – RXRDY Receive Holding Register Ready Interrupt Mask

Bit 0 – TXCOMP Transmission Completed Interrupt Mask

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.66. TWI Receive Holding Register

Name: FLEX_TWI_RHR
Offset: 0x630
Reset: 0x00000000
Property: Read-only

If FIFO is enabled (FLEX_TWI_CR.FIFOEN=1), a byte access on FLEX_TWI_RHR reads one data, see TWI Single Data Access for details.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				ASTATE[1:0]		PSTATE	SSTATE[1:0]	
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RXDATA[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 12:11 – ASTATE[1:0] Acknowledge State (Client Sniffer Mode only)

Value	Name	Description
0	NONE	No Acknowledge or Nacknowledge detected after previously logged data
1	ACK	Acknowledge (A) detected after previously logged data
2	NACK	Nacknowledge (NA) detected after previously logged data
3	UNDEF	Not defined

Bit 10 – PSTATE Stop State (Client Sniffer Mode only)

Value	Description
0	No STOP (P) detected after previous logged data.
1	Stop detected (P) after previous logged data.

Bits 9:8 – SSTATE[1:0] Start State (Client Sniffer Mode only)

Value	Name	Description
0	NOSTART	No START detected with the logged data
1	START	START (S) detected with the logged data
2	RSTART	Repeated START (Sr) detected with the logged data
3	UNDEF	Not defined

Bits 7:0 – RXDATA[7:0] Host or Client Receive Holding Data

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.67. TWI Receive Holding Register (FIFO Enabled)

Name: FLEX_TWI_RHR (FIFO_ENABLED)
Offset: 0x630
Reset: 0x00000000
Property: Read-only

To read multi-data, the FIFO must be enabled (FLEX_TWI_CR.FIFOEN=1) and Sniffer mode disabled (FLEX_TWI_SMR.SNIFF=0). The access type (byte, halfword or word) determines the number of data written in a single access (1, 2 or 4), see TWI Multiple Data Access for details.

Bit	31	30	29	28	27	26	25	24
	RXDATA3[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RXDATA2[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RXDATA1[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RXDATA0[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – RXDATA3[7:0] Host or Client Receive Holding Data 3

Bits 23:16 – RXDATA2[7:0] Host or Client Receive Holding Data 2

Bits 15:8 – RXDATA1[7:0] Host or Client Receive Holding Data 1

Bits 7:0 – RXDATA0[7:0] Host or Client Receive Holding Data 0

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.68. TWI Transmit Holding Register

Name: FLEX_TWI_THR
Offset: 0x634
Reset: –
Property: Write-only

If FIFO is enabled (FLEX_TWI_CR.FIFOEN=1), a byte access on FLEX_TWI_THR reads one data in a single access, see TWI Single Data Access for details.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	TXDATA[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 7:0 – TXDATA[7:0] Host or Client Transmit Holding Data

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.69. TWI Transmit Holding Register (FIFO Enabled)

Name: FLEX_TWI_THR (FIFO_ENABLED)
Offset: 0x634
Reset: –
Property: Write-only

To write multi-data, the FIFO must be enabled (FLEX_TWI_CR.FIFOEN=1) and Sniffer mode disabled (FLEX_TWI_SMR.SNIFF=0). The access type (byte, halfword or word) determines the number of data written in a single access (1, 2 or 4), see TWI Multiple Data Access for details.

Bit	31	30	29	28	27	26	25	24
	TXDATA3[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	TXDATA2[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	TXDATA1[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	TXDATA0[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:24 – TXDATA3[7:0] Host or Client Transmit Holding Data 3

Bits 23:16 – TXDATA2[7:0] Host or Client Transmit Holding Data 2

Bits 15:8 – TXDATA1[7:0] Host or Client Transmit Holding Data 1

Bits 7:0 – TXDATA0[7:0] Host or Client Transmit Holding Data 0

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.70. TWI SMBus Timing Register

Name: FLEX_TWI_SMBTR
Offset: 0x638
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	THMAX[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	TLOWM[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TLOWS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
					PRESC[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 31:24 – THMAX[7:0] Clock High Maximum Cycles
Clock cycles in clock high maximum count. Prescaled by PRESC. Used for bus free detection. Used to time THIGH:MAX.

Bits 23:16 – TLOWM[7:0] Main System Bus Clock Stretch Maximum Cycles

Value	Description
0	TLOW:MEXT timeout check disabled.
1–255	Clock cycles in main system bus maximum clock stretch count. Prescaled by PRESC. Used to time TLOW:MEXT.

Bits 15:8 – TLOWS[7:0] Client Clock Stretch Maximum Cycles

Value	Description
0	TLOW:SEXT timeout check disabled.
1–255	Clock cycles in client maximum clock stretch count. Prescaled by PRESC. Used to time TLOW:SEXT.

Bits 3:0 – PRESC[3:0] SMBus Clock Prescaler
Used to specify how to prescale the TLOWS, TLOWM and THMAX counters in SMBTR. Counters are prescaled according to the following formula: PRESC = Log(fMCK / $f_{Prescaled}$) / Log(2) - 1

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.71. TWI Alternative Command Register

Name: FLEX_TWI_ACR
Offset: 0x640
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
							NPEC	NDIR
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
	NDATAL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
							PEC	DIR
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	DATAL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 25 – NPEC Next PEC Request (SMBus Mode only)

Value	Description
0	The next transfer does not use a PEC byte.
1	The next transfer uses a PEC byte.

Bit 24 – NDIR Next Transfer Direction

Value	Description
0	Write direction.
1	Read direction.

Bits 23:16 – NDATAL[7:0] Next Data Length

Value	Description
0	No data to send (see Alternative Command).
1–255	Number of bytes to send for the next transfer.

Bit 9 – PEC PEC Request (SMBus Mode only)

Value	Description
0	The transfer does not use a PEC byte.
1	The transfer uses a PEC byte.

Bit 8 – DIR Transfer Direction

Value	Description
0	Write direction.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
1	Read direction.

Bits 7:0 – DATAL[7:0] Data Length

Value	Description
0	No data to send (see Alternative Command).
1–255	Number of bytes to send during the transfer.

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.72. TWI Filter Register

Name: FLEX_TWI_FILTR
Offset: 0x644
Reset: 0x00000000
Property: Read/Write

Important:
FILT and THRES are used to configure digital filters on data and clock lines.

In Standard, Fast and Fast Plus modes, the digital filter must be enabled (FILT=1) and a pulse width threshold defined (THRES > 0).

The field THRES must be set according to the peripheral clock to suppress spikes lower than 50 ns. The recommended value is calculated using the formula below:

THRES > 50 ns/$t_{peripheral\_clock\ (ns)}$

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						THRES[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
							PADFEN	FILT
Access							R/W	R/W
Reset							0	0

Bits 10:8 – THRES[2:0] Digital Filter Threshold

Value	Description
0	No filtering applied on TWI inputs.
1–7	Maximum pulse width of spikes which will be suppressed by the input filter, defined in peripheral clock cycles.

Bit 1 – PADFEN PAD Filter Enable

Value	Description
0	PAD analog filter is disabled.
1	PAD analog filter is enabled. (The analog filter must be enabled if High-speed mode is enabled.)

Bit 0 – FILT RX Digital Filter

TWI digital input filtering follows a majority decision based on three samples from SDA/SCL lines at peripheral clock frequency.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	No filtering applied on TWI inputs.
1	TWI input filtering is active. (Only in Standard and Fast modes)

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.73. TWI Matching Register

Name: FLEX_TWI_SWMR
Offset: 0x64C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	DATAM[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
		SADR3[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
		SADR2[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		SADR1[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 31:24 – DATAM[7:0] Data Match
The TWI module will extend the matching process to the first received data, comparing it with DATAM if the DATAMEN bit is enabled.

Bits 22:16 – SADR3[6:0] Client Address 3
Client address 3. The TWI module will match on this additional address if the SADR3EN bit is enabled.

Bits 14:8 – SADR2[6:0] Client Address 2
Client address 2. The TWI module will match on this additional address if the SADR2EN bit is enabled.

Bits 6:0 – SADR1[6:0] Client Address 1
Client address 1. The TWI module will match on this additional address if the SADR1EN bit is enabled.

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.74. TWI FIFO Mode Register

Name: FLEX_TWI_FMR
Offset: 0x650
Reset: 0x00000000
Property: Read/Write

This register reads '0' if the FIFO is disabled (see FLEX_TWI_CR to enable/disable the internal FIFO). This register can only be written if the WPEN bit is cleared in the TWI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
			RXFTHRES[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
			TXFTHRES[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			RXRDYM[1:0]				TXRDYM[1:0]	
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bits 29:24 – RXFTHRES[5:0] Receive FIFO Threshold

Value	Description
0–8	Defines the Receive FIFO threshold value (number of bytes). The FLEX_TWI_FSR.RXFTH flag will be set when Receive FIFO goes from "below" threshold state to "equal to or above" threshold state.

Bits 21:16 – TXFTHRES[5:0] Transmit FIFO Threshold

Value	Description
0–8	Defines the Transmit FIFO threshold value (number of bytes). The FLEX_TWI_FSR.TXFTH flag will be set when Transmit FIFO goes from "above" threshold state to "equal to or below" threshold state.

Bits 5:4 – RXRDYM[1:0] Receiver Ready Mode
If FIFOs are enabled, the FLEX_TWI_SR.RXRDY flag behaves as follows.

Value	Name	Description
0	ONE_DATA	RXRDY will be at level '1' when at least one unread data is in the receive FIFO. When DMA is enabled to transfer data the chunk of 1 byte must be configured in the DMA. If the transfer is performed by software, the access type (byte, halfword) must be defined accordingly.
1	TWO_DATA	RXRDY will be at level '1' when at least two unread data are in the receive FIFO. To minimize system bus load, when DMA is enabled to transfer data, the chunk of 1 halfword (1 halfword carries 2 bytes) must be configured in the DMA. If the transfer is performed by software, the access type can be defined as byte (1 byte per access, 2 accesses) or halfword (2 bytes per access, 1 single access).

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Name	Description
2	FOUR_DATA	RXRDY will be at level '1' when at least four unread data are in the receive FIFO. To minimize system bus load, when DMA is enabled to transfer data, the chunk of 1 word (1 word carries 4 bytes) must be configured in the DMA. If the transfer is performed by software, the access type can be defined as byte (1 byte per access, 4 accesses), halfword (2 bytes per access, 2 accesses) or word (4 bytes per access, 1 single access).

Bits 1:0 – TXRDYM[1:0] Transmitter Ready Mode
If FIFOs are enabled, the FLEX_TWI_SR.TXRDY flag behaves as follows.

Value	Name	Description
0	ONE_DATA	TXRDY will be at level '1' when at least one data can be written in the transmit FIFO. When DMA is enabled to transfer data, the chunk of 1 byte must be configured in the DMA. If the transfer is performed by software, the access type must be defined as byte.
1	TWO_DATA	TXRDY will be at level '1' when at least two data can be written in the transmit FIFO. To minimize system bus load, when DMA is enabled to transfer data, the chunk of 1 halfword (1 halfword carries 2 bytes) must be configured in the DMA. If the transfer is performed by software, the access type can be defined as byte (1 byte per access, 2 accesses) or halfword (2 bytes per access, 1 single access).
2	FOUR_DATA	TXRDY will be at level '1' when at least four data can be written in the transmit FIFO. To minimize system bus load, when DMA is enabled to transfer data, the chunk of 1 word (1 word carries 4 bytes) must be configured in the DMA. If the transfer is performed by software, the access type can be defined as byte (1 byte per access, 4 accesses), halfword (2 bytes per access, 2 accesses) or word (4 bytes per access, 1 single access).

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.75. TWI FIFO Level Register

Name: FLEX_TWI_FLR
Offset: 0x654
Reset: 0x00000000
Property: Read-only

This register reads '0' if the FIFO is disabled (see FLEX_TWI_CR to enable/disable the internal FIFO).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			RXFL[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			TXFL[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bits 21:16 – RXFL[5:0] Receive FIFO Level

Value	Description
0	There is no unread data in the Receive FIFO.
1–8	Indicates the number of unread data in the Receive FIFO.

Bits 5:0 – TXFL[5:0] Transmit FIFO Level

Value	Description
0	There is no data in the Transmit FIFO.
1–8	Indicates the number of data in the Transmit FIFO.

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.76. TWI FIFO Status Register

Name: FLEX_TWI_FSR
Offset: 0x660
Reset: 0x00000000
Property: Read-only

This register reads '0' if the FIFO is disabled (see FLEX_TWI_CR to enable/disable the internal FIFO)

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 7 – RXFPTEF Receive FIFO Pointer Error Flag
See FIFO Pointer Error for details.

Value	Description
0	No Receive FIFO pointer occurred.
1	Receive FIFO pointer error occurred. Receiver must be reset.

Bit 6 – TXFPTEF Transmit FIFO Pointer Error Flag
See FIFO Pointer Error for details.

Value	Description
0	No Transmit FIFO pointer occurred.
1	Transmit FIFO pointer error occurred. Transceiver must be reset.

Bit 5 – RXFTHF Receive FIFO Threshold Flag

Value	Description
0	Number of unread data in Receive FIFO is below RXFTHRES threshold.
1	Number of unread data in Receive FIFO has reached RXFTHRES threshold since the last read of FLEX_TWI_FSR.

Bit 4 – RXFFF Receive FIFO Full Flag

Value	Description
0	Receive FIFO is not empty.
1	Receive FIFO has been filled since the last read of FLEX_TWI_FSR.

Bit 3 – RXFEF Receive FIFO Empty Flag

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	Receive FIFO is not empty.
1	Receive FIFO has been emptied since the last read of FLEX_TWI_FSR.

Bit 2 – TXFTHF Transmit FIFO Threshold Flag (cleared on read)

Value	Description
0	Number of data in Transmit FIFO is above TXFTHRES threshold.
1	Number of data in Transmit FIFO has reached TXFTHRES threshold since the last read of FLEX_TWI_FSR.

Bit 1 – TXFFF Transmit FIFO Full Flag (cleared on read)

Value	Description
0	Transmit FIFO is not full.
1	Transmit FIFO has been filled since the last read of FLEX_TWI_FSR.

Bit 0 – TXFEF Transmit FIFO Empty Flag (cleared on read)

Value	Description
0	Transmit FIFO is not empty.
1	Transmit FIFO has been emptied since the last read of FLEX_TWI_FSR.

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.77. TWI FIFO Interrupt Enable Register

Name: FLEX_TWI_FIER
Offset: 0x664
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the TWI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 7 – RXFPTEF RXFPTEF Interrupt Enable

Bit 6 – TXFPTEF TXFPTEF Interrupt Enable

Bit 5 – RXFTHF RXFTHF Interrupt Enable

Bit 4 – RXFFF RXFFF Interrupt Enable

Bit 3 – RXFEF RXFEF Interrupt Enable

Bit 2 – TXFTHF TXFTHF Interrupt Enable

Bit 1 – TXFFF TXFFF Interrupt Enable

Bit 0 – TXFEF TXFEF Interrupt Enable

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.78. TWI FIFO Interrupt Disable Register

Name: FLEX_TWI_FIDR
Offset: 0x668
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the TWI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 7 – RXFPTEF RXFPTEF Interrupt Disable

Bit 6 – TXFPTEF TXFPTEF Interrupt Disable

Bit 5 – RXFTHF RXFTHF Interrupt Disable

Bit 4 – RXFFF RXFFF Interrupt Disable

Bit 3 – RXFEF RXFEF Interrupt Disable

Bit 2 – TXFTHF TXFTHF Interrupt Disable

Bit 1 – TXFFF TXFFF Interrupt Disable

Bit 0 – TXFEF TXFEF Interrupt Disable

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.79. TWI FIFO Interrupt Mask Register

Name: FLEX_TWI_FIMR
Offset: 0x66C
Reset: 0x00000000
Property: Read-only

This register reads '0' if the FIFO is disabled (see FLEX_TWI_CR to enable/disable the internal FIFO).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 7 – RXFPTEF RXFPTEF Interrupt Mask

Bit 6 – TXFPTEF TXFPTEF Interrupt Mask

Bit 5 – RXFTHF RXFTHF Interrupt Mask

Bit 4 – RXFFF RXFFF Interrupt Mask

Bit 3 – RXFEF RXFEF Interrupt Mask

Bit 2 – TXFTHF TXFTHF Interrupt Mask

Bit 1 – TXFFF TXFFF Interrupt Mask

Bit 0 – TXFEF TXFEF Interrupt Mask

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.80. TWI Write Protection Mode Register

Name: FLEX_TWI_WPMR
Offset: 0x6E4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
						WPCREN	WPITEN	WPEN
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x545749	PASSWD	Writing any other value in this field aborts the write operation of bits WPEN, WPITEN and WPCREN. Always reads as 0.

Bit 2 – WPCREN Write Protection Control Enable

Value	Description
0	Disables the write protection on control register if WPKEY corresponds to 0x545749 (“TWI” in ASCII).
1	Enables the write protection on control register if WPKEY corresponds to 0x545749 (“TWI” in ASCII).

Bit 1 – WPITEN Write Protection Interrupt Enable

Value	Description
0	Disables the write protection on interrupt registers if WPKEY corresponds to 0x545749 (“TWI” in ASCII).
1	Enables the write protection on interrupt registers if WPKEY corresponds to 0x545749 (“TWI” in ASCII).

Bit 0 – WPEN Write Protection Enable

See TWI Register Write Protection for the list of registers that can be write-protected.

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x545749 (“TWI” in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x545749 (“TWI” in ASCII).

© 2026 Microchip Technology Inc. and its subsidiaries

33.10.81. TWI Write Protection Status Register

Name: FLEX_TWI_WPSR
Offset: 0x6E8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	WPVSRC[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPVSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPVS
Access								R
Reset								0

Bits 31:8 – WPVSRC[23:0] Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protect Violation Status

Value	Description
0	No Write Protection Violation has occurred since the last read of FLEX_TWI_WPSR.
1	A Write Protection Violation has occurred since the last read of FLEX_TWI_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

© 2026 Microchip Technology Inc. and its subsidiaries

34. Quad Serial Peripheral Interface (QSPI)

34.1. Description

The Quad Serial Peripheral Interface (QSPI) is a synchronous serial data link that provides communication with external devices in Host mode.

The QSPI can be used in SPI mode to interface to serial peripherals such as ADCs, DACs, LCD controllers, CAN controllers and sensors, or in Serial Memory mode to interface to serial Flash memories.

The QSPI allows the system to execute code directly from a serial Flash memory (XIP) without code shadowing to RAM. The serial Flash memory mapping is seen in the system as other memories such as ROM, SRAM, DRAM, embedded Flash memory, etc.

With the support of the Quad SPI protocol, the QSPI allows the system to use high-performance serial Flash memories which are small and inexpensive, in place of larger and more expensive parallel Flash memories.

Note: Stacked devices with a rollover in the memory address space at each die boundary are not supported.

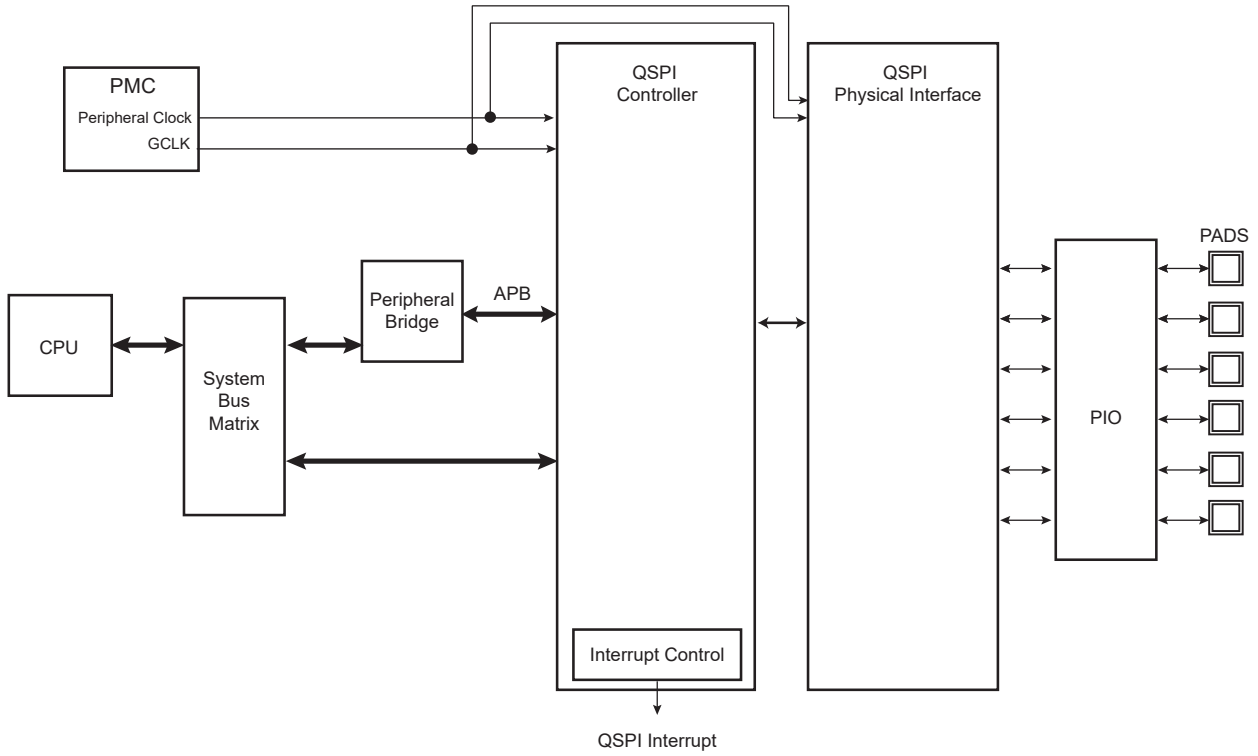
34.2. Embedded Characteristics

- Host SPI Interface
  - Programmable clock phase and clock polarity
  - Programmable transfer delays between consecutive transfers, between clock and data, between deactivation and activation of chip select
- SPI Mode
  - Interface to serial peripherals such as ADCs, DACs, LCD controllers, CAN controllers and sensors
  - 8-bit/16-bit programmable data length
- Serial Memory Mode
  - Interface to serial Flash memories operating in Single-bit SPI, Dual SPI and Quad SPI
  - Interface to serial Flash Memories operating in Single Data Rate or Double Data Rate Modes
  - Supports “Execute In Place” (XIP)— code execution by the system directly from a serial Flash memory
  - Flexible instruction register for compatibility with all serial Flash memories
  - 32-bit address mode (default is 8-bit address) to support serial Flash memories larger than 128 Mbits
  - Continuous Read mode
  - “On-The-Fly” Scrambling/unscrambling
- Connection to PDC Channel Capabilities Optimizes Data Transfers
  - One channel for the receiver, one channel for the transmitter
  - Next buffer support
- Register Write Protection

© 2026 Microchip Technology Inc. and its subsidiaries

34.3. Block Diagram

Figure 34-1. QSPI Block Diagram



34.4. Signal Description

Table 34-1. Signal Description

Pin Name	Pin Description	Type
QSCK	Serial Clock	Output
MOSI (QIO0) (1) (2)	Data Output (Data Input Output 0)	Output (Input/Output)
MISO (QIO1) (1) (2)	Data Input (Data Input Output 1)	Input (Input/Output)
QIO2 (3)	Data Input Output 2	Input/Output
QIO3 (3)	Data Input Output 3	Input/Output
QCS	Peripheral Chip Select	Output

Notes:

1. MOSI and MISO are used for single-bit SPI operation.
2. QIO0–QIO1 are used for Dual SPI operation.
3. QIO0–QIO3 are used for Quad SPI operation.

34.5. Product Dependencies

34.5.1. I/O Lines

The pins used for interfacing the compliant external devices may be multiplexed with PIO lines. The programmer must first program the PIO controllers to assign the QSPI pins to their peripheral functions.

© 2026 Microchip Technology Inc. and its subsidiaries

34.5.2. Power Management

The QSPI may be clocked through the Power Management Controller (PMC), thus the programmer must first configure the PMC to enable the QSPI clocks.

34.5.3. Interrupt Sources

The QSPI has an interrupt line connected to the Interrupt Controller. Handling the QSPI interrupt requires programming the interrupt controller before configuring the QSPI.

34.5.4. Peripheral DMA Controller (PDC)

The QSPI can be used in conjunction with the Peripheral DMA Controller (PDC) in order to reduce processor overhead. For a full description of the PDC, refer to the section "Peripheral DMA Controller (PDC)".

34.6. Functional Description

34.6.1. Register Synchronization

Because the QSPI Controller interface and the QSPI Controller core use different clocks, some events must be synchronized with the core for them to take effect.

The events that require synchronization with the QSPI Controller core are:

- QSPI_CR.QSPIEN
- QSPI_CR.QSPIDIS
- QSPI_CR.SWRST
- QSPI_CR.UPDCFG
- QSPI_CR.STTFR
- QSPI_CR.RTOUT
- QSPI_CR.LASTXFER
- QSPI_RDR.RD (implies synchronization only when QSPI_MR.SMM is set to ‘1 ‘)
- QSPI_TDR.TD

Before accessing any of these bits/fields, check that QSPI_SR.SYNCBSY is at ‘0’.

When the synchronization process is in progress, QSPI_SR.SYNCBSY is at ‘1’.

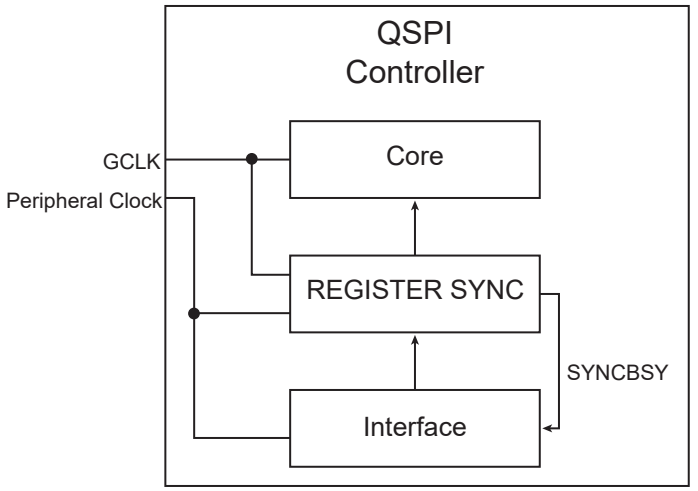
As long as QSPI_SR.SYNCBSY is at ‘1’, no access to the registers requiring synchronization is allowed.

When using QSPI_CR.UPDCFG to update the system configuration, it is mandatory for QSPI_SR.SYNCBSY to be at ‘0’ before and after writing QSPI_CR.UPDCFG. This ensures that the update is completed before going on to the next step.

Note: In some specific cases, it is not necessary to check QSPI_SR.SYNCBSY for QSPI_RDR.RD and QSPI_TDR.TD if the TDRE or RDRF flags are checked. See Instruction Transmission Flow Diagram SMRM = 0 and TFRTYP = 0 (Memory Register Access), Instruction Transmission Flow Diagram SMRM = 1 and TFRTYP = 0 (Memory Write Access) for detailed procedures.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 34-2. Register Synchronization with QSPI Controller Core



34.6.2. Updating the QSPI Configuration

At any time, the QSPI Controller core configuration can be updated by writing the QSPI_CR.UPDCFG bit to '1'. Note that QSPI_SR.SYNCBSY must be '0' before writing QSPI_CR.UPDCFG.

The configuration registers that require synchronization (writing QSPI_CR.UPDCFG to '1') with the QSPI Controller core are:

- QSPI_MR
- QSPI_SCR
- QSPI_IAR
- QSPI_WICR
- QSPI_IFR
- QSPI_RICR
- QSPI_SMR
- QSPI_SKR
- QSPI_WRACNT

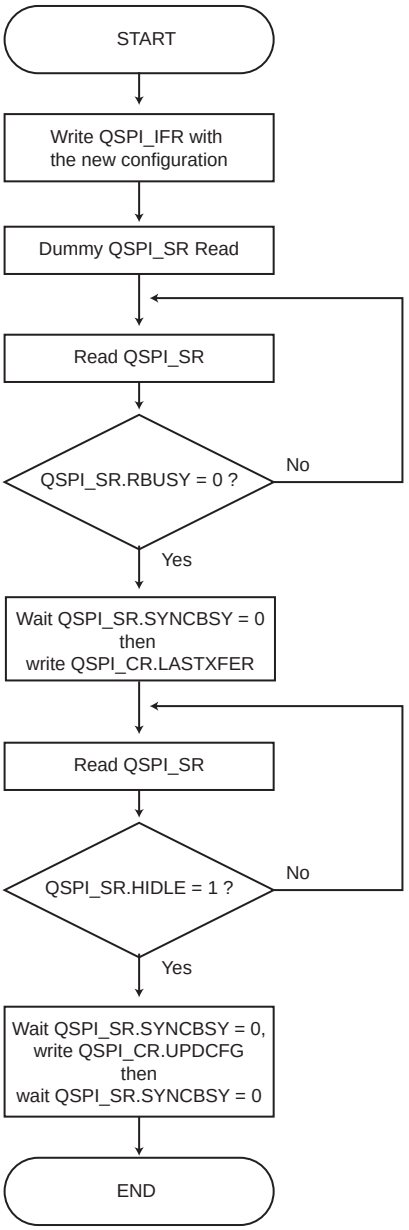
34.6.2.1. Changing QSPI_IFR Configuration (QSPI_MR.SMM = 1)

Accesses in Serial Memory mode (QSPI_MR.SMM=1) are triggered by QSPI register accesses or by performing accesses in the QSPI memory space depending on the configuration of QSPI_IFR.SMRM and QSPI_IFR.TFRTYP. For details, see Instruction Frame and QSPI Access Triggering.

If QSPI_IFR is to be modified after the initial configuration, the procedure illustrated in the figure below must be followed to ensure no frame loss. For details, see QSPI Access Triggering.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 34-3. Changing QSPI_IFR



34.6.3. Serial Clock Phase and Polarity

Four combinations of polarity and phase are available for data transfers. The clock polarity is programmed with the CPOL bit in the QSPI Serial Clock register (QSPI_SCR). The CPHA bit in the QSPI_SCR programs the clock phase. These two parameters determine the edges of the clock signal on which data is driven and sampled. Each of the two parameters has two possible states, resulting in four possible combinations that are incompatible with one another. Thus, the interfaced client must use the same parameter values to communicate.

Only Mode 0 is supported with QSPI_MR.SMM = 1.

The table below shows the four modes and corresponding parameter settings.

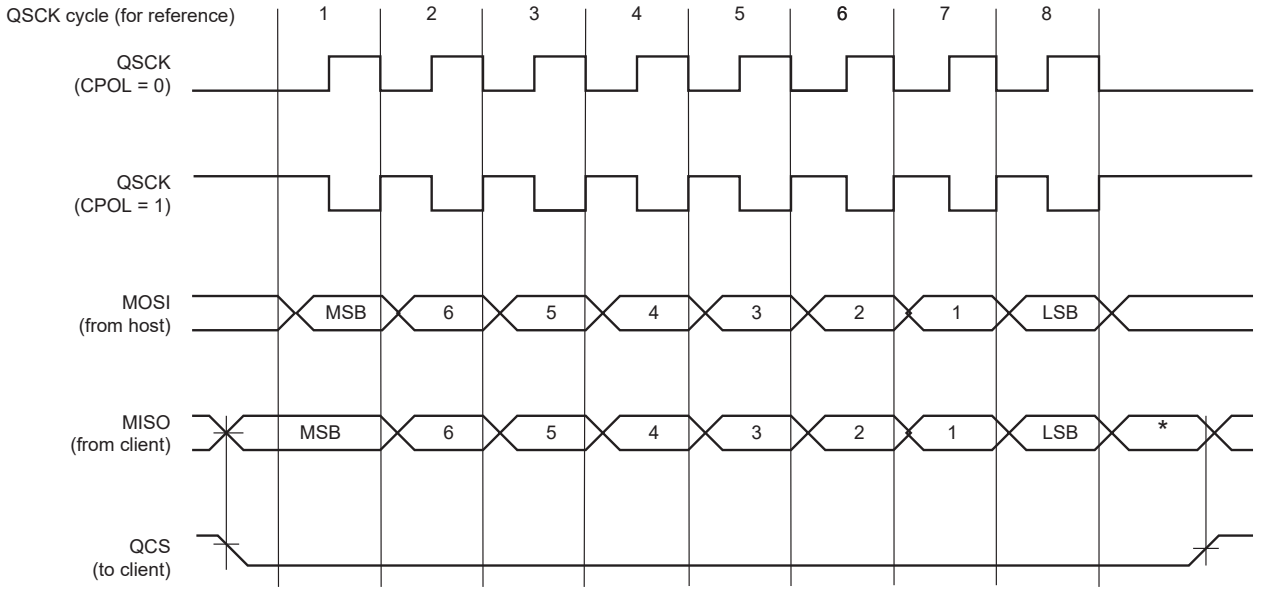
© 2026 Microchip Technology Inc. and its subsidiaries

Table 34-2. QSPI Bus Clock Modes

QSPI Clock Mode	QSPI_SCR.CPOL	QSPI_SCR.CPHA	Shift QSCK Edge	Capture QSCK Edge	QSCK Inactive Level
0	0	0	Falling	Falling	Low
1	0	1	Rising	Rising	Low
2	1	0	Rising	Rising	High
3	1	1	Falling	Falling	High

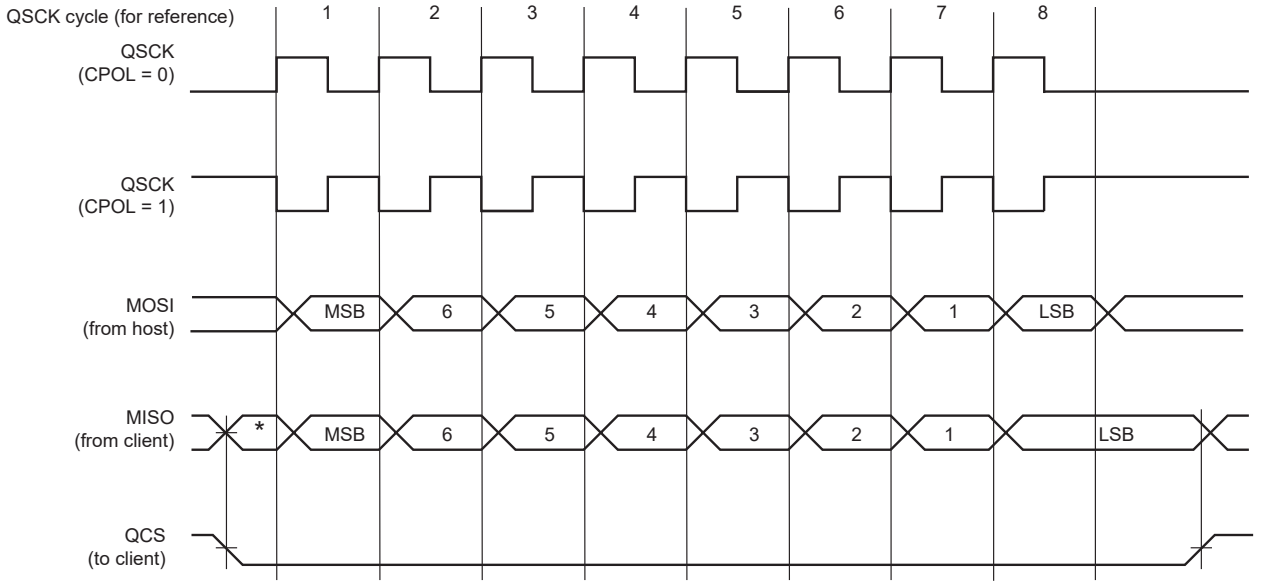
The figures below show examples of data transfers.

Figure 34-4. QSPI Transfer Format (QSPI_SCR.CPHA = 0, 8 bits per transfer)



* Not defined, but normally MSB of previous character received.

Figure 34-5. QSPI Transfer Format (QSPI_SCR.CPHA = 1, 8 bits per transfer)



* Not defined but normally LSB of previous character transmitted.

© 2026 Microchip Technology Inc. and its subsidiaries

34.6.4. Transfer Delays

The figure below shows several consecutive transfers while the chip select is active. Three delays can be programmed to modify the transfer waveforms:

- The delay between the deactivation and the activation of QCS, programmed by writing QSPI_MR.DLYCS. Allows to adjust the minimum time of QCS at high level.
- The delay before QSCK, programmed by writing QSPI_SR.DLYBS. Allows the start of QSCK to be delayed after the chip select has been asserted.
- The delay between consecutive transfers, programmed by writing QSPI_MR.DLYBCT.
  - If QSPI_MR.SMM = 0. Allows insertion of a delay between two consecutive transfers.
  - If QSPI_MR.SMM = 1. Allows insertion of a delay between last QSCK pulse and QCS rise.

These delays allow the QSPI to be adapted to the interfaced peripherals and their speed and bus release time.

Figure 34-6. Programmable Delays (SMM = 0)

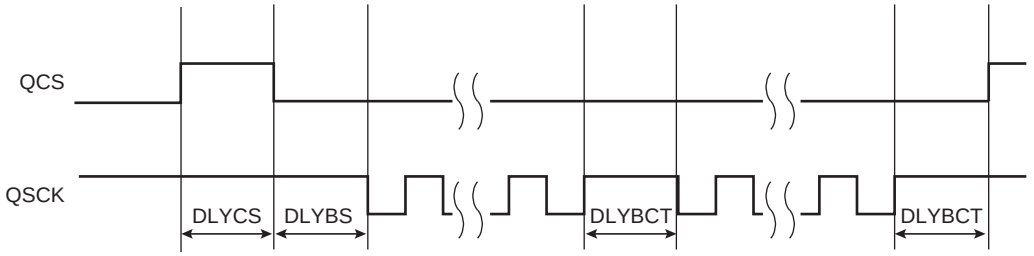
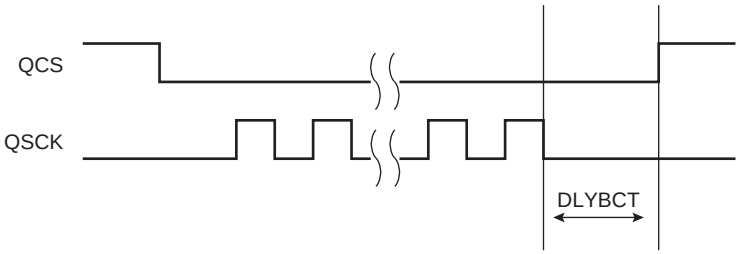


Figure 34-7. DLYBCT with SMM = 1



34.6.5. QSPI SPI Mode

In SPI mode, the QSPI acts as a standard SPI Host.

To activate this mode, QSPI_MR.SMM must be written to '0'.

34.6.5.1. SPI Mode Operations

The QSPI in standard SPI mode operates on the GCLK clock. It fully controls the data transfers to and from the client connected to the SPI bus. The QSPI drives the chip select line to the client (QCS) and the serial clock signal (QSCK).

The QSPI features two holding registers, the Transmit Data register (QSPI_TDR) and the Receive Data register (QSPI_RDR), and a single internal shift register. The holding registers maintain the data flow at a constant rate.

After enabling the QSPI, a data transfer begins when the processor writes to QSPI_TDR. The written data is immediately transferred to the internal shift register and transfer on the SPI bus starts. While the data in the internal shift register is shifted on the MOSI line, the MISO line is sampled and shifted to the internal shift register. Receiving data cannot occur without transmitting data. If receiving mode is not needed, for example when communicating with a client receiver only (such as an LCD), the receive status flags in the Status register (QSPI_ISR) can be discarded.

© 2026 Microchip Technology Inc. and its subsidiaries

If new data is written in QSPI_TDR during the transfer, it is retained there until the current transfer is completed. Then, the received data is transferred from the internal shift register to QSPI_RDR, the data in QSPI_TDR is loaded in the internal shift register and a new transfer starts.

The transfer of a data written in QSPI_TDR in the internal shift register is indicated by the Transmit Data Register Empty (TDRE) bit in QSPI_ISR. When new data is written in QSPI_TDR, this bit is cleared. QSPI_ISR.TDRE is used to trigger the Transmit PDC channel.

The end of transfer is indicated by the TXEMPTY flag in QSPI_ISR. If a transfer delay (DLYBCT) is greater than 0 for the last transfer, QSPI_ISR.TXEMPTY is set after the completion of this delay. The peripheral clock and GCLK clock can be switched off at this time.

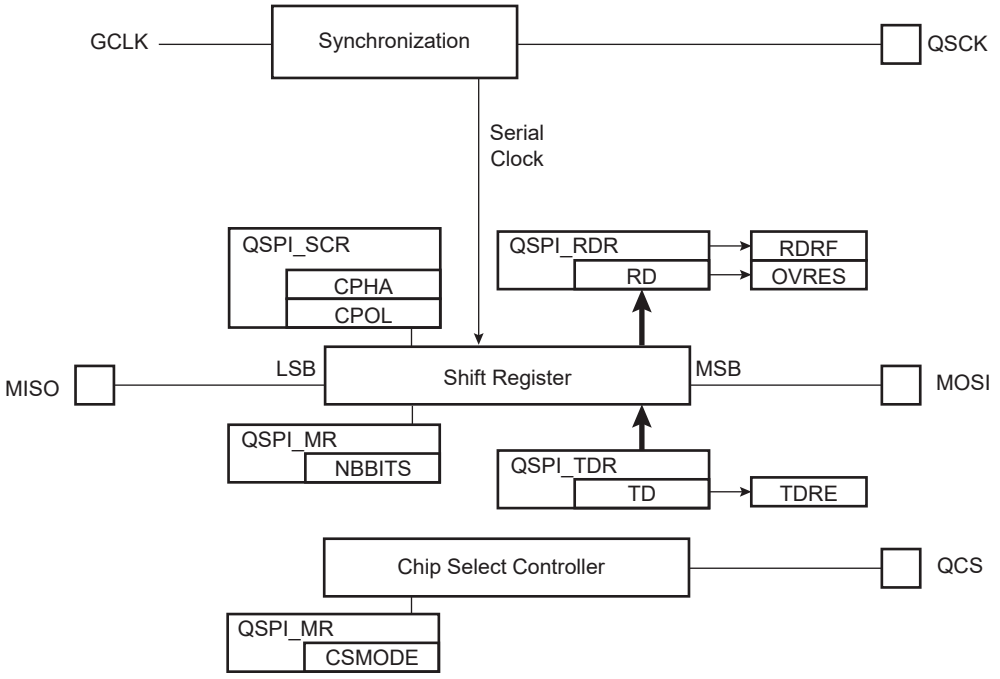
The transfer of received data from the internal shift register in QSPI_RDR is indicated by the Receive Data Register Full (RDRF) bit in QSPI_ISR. When the received data is read, the QSPI_ISR.RDRF bit is cleared.

If QSPI_RDR has not been read before new data is received, the Overrun Error Status (OVRES) bit in QSPI_ISR is set. As long as this flag is set, new data will overwrite the old QSPI_RDR.RD value. The user must read QSPI_ISR to clear the OVRES bit.

SPI Mode Block Diagram shows a block diagram of the SPI when operating in Host mode. SPI Mode Flow Diagram shows a flow chart describing how transfers are handled.

34.6.5.2. SPI Mode Block Diagram

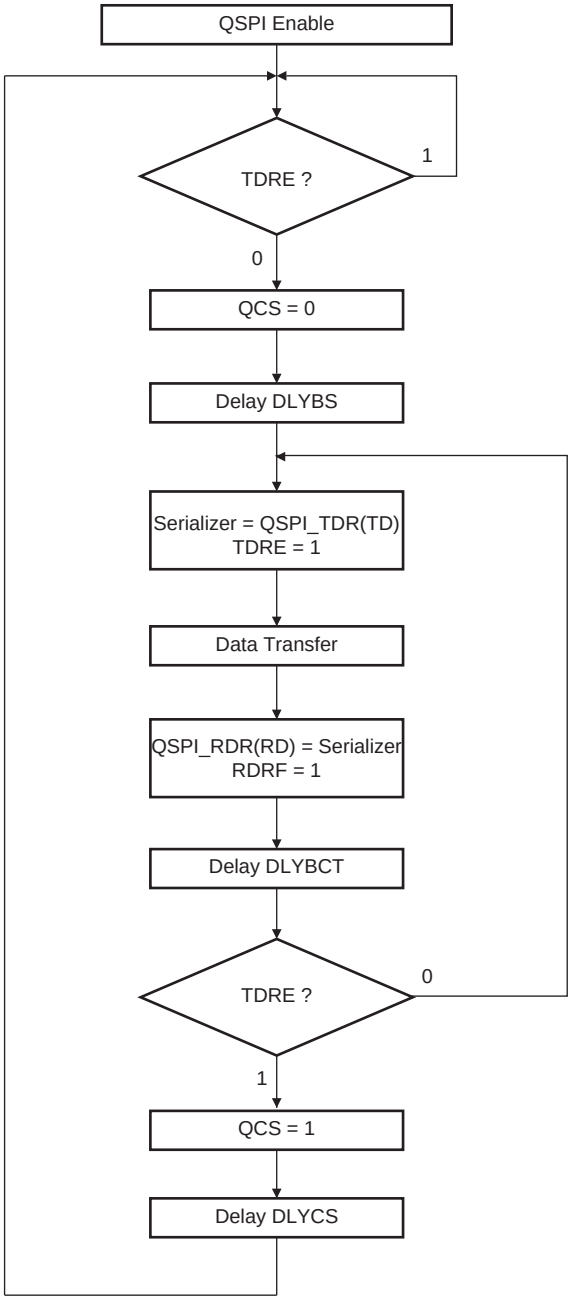
Figure 34-8. SPI Mode Block Diagram



© 2026 Microchip Technology Inc. and its subsidiaries

34.6.5.3. SPI Mode Flow Diagram

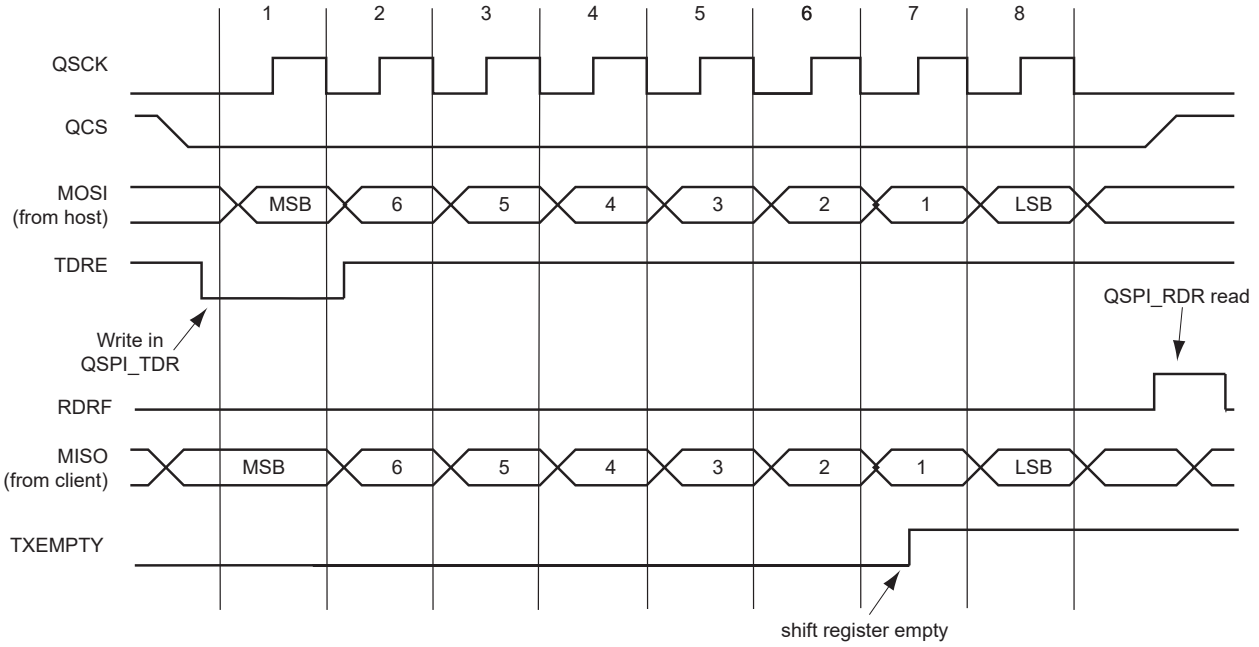
Figure 34-9. SPI Mode Flow Diagram



The figure below shows Transmit Data Register Empty (TDRE), Receive Data Register Full (RDRF) and Transmission Register Empty (TXEMPTY) status flags behavior within the QSPI_SR during an 8-bit data transfer in Fixed mode, without PDC.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 34-10. Status Register Flags Behavior



Notes: Due to the internal architecture (see Block Diagram):

- A latency occurs between the TXEMPTY rise and the end of the frame.
- A delay occurs between the end of the frame and the RDRF flag rise.

34.6.5.4. Peripheral Deselection without PDC

During a transfer of more than one data on a Chip Select without the PDC, the QSPI_TDR is loaded by the processor and the flag TDRE rises as soon as the content of the QSPI_TDR is transferred into the internal shift register. When this flag is detected high, the QSPI_TDR can be reloaded. If this reload by the processor occurs before the end of the current transfer, the Chip Select is not deasserted between the two transfers. Depending on the application software handling the QSPI_SR flags (by interrupt or polling method) or servicing other interrupts or other tasks, the processor may not reload the QSPI_TDR in time to keep the chip select active (low). A null Delay Between Consecutive Transfer (DLYBCT) value in the QSPI_MR gives even less time for the processor to reload the QSPI_TDR. With some SPI client peripherals, requiring the chip select line to remain active (low) during a full set of transfers may lead to communication errors.

To facilitate interfacing with such devices, QSPI_MR.CSMODE may be configured to '1'. This allows the chip select lines to remain in their current state (low = active) until the end of transfer is indicated by the Last Transfer (LASTXFER) bit in the Control register (QSPI_CR). Even if the QSPI_TDR is not reloaded, the chip select remains active. To have the chip select line rise at the end of the last data transfer, QSPI_CR.LASTXFER must be written to '1' at the same time or after writing the last data to transmit into the QSPI_TDR.

34.6.5.5. Peripheral Deselection with PDC

When the PDC is used, the Chip Select line remains low during the transfer since the TDRE flag is managed by the PDC itself. Reloading the QSPI_TDR by the PDC is done as soon as the TDRE flag is set to one. In this case, setting the CSMODE field to 1 might not be needed. However, when other PDC channels connected to other peripherals are also in use, the QSPI PDC could be delayed by another PDC with a higher priority on the bus. Having PDC buffers in slower memories such as Flash memory or SDRAM compared to fast internal SRAM, may lengthen the reload time of the QSPI_TDR by the PDC as well. This means that the QSPI_TDR might not be reloaded in time to keep the chip select line low. In this case, the chip select line may toggle between data transfer and according to

© 2026 Microchip Technology Inc. and its subsidiaries

some SPI client devices, the communication might get lost. It may be necessary to configure the CSMODE field to 1.

When the CSMODE field is configured to 0, the QCS does not rise in all cases between two transfers on the same peripheral. During a transfer on a Chip Select, the flag TDRE rises as soon as the content of the QSPI_TDR is transferred into the internal shifter. When this flag is detected, the QSPI_TDR can be reloaded. If this reload occurs before the end of the current transfer and if the next transfer is performed on the same chip select as the current transfer, the Chip Select is not de-asserted between the two transfers. This might lead to difficulties for interfacing with some serial peripherals requiring the chip select to be de-asserted after each transfer. To facilitate interfacing with such devices, the QSPI_MR can be programmed with the CSMODE field at 2.

34.6.6. QSPI Serial Memory Mode

In Serial Memory mode, the QSPI acts as a serial Flash memory controller. The QSPI can be used to read data from the serial Flash memory allowing the CPU to execute code from it (XIP execute in place). The QSPI can also be used to control the serial Flash memory (Program, Erase, Lock, etc.) by sending specific commands. In this mode, the QSPI is compatible with single-bit SPI, Dual SPI and Quad SPI protocols.

To activate this mode, QSPI_MR.SMM must be written to '1'.

In Serial Memory mode, data is transferred either by QSPI_TDR and QSPI_RDR or by writing or read in the QSPI memory space (0x04000000–0x06000000) depending on QSPI_IFR.TFRTYP and QSPI_IFR.SMRM configuration.

34.6.6.1. Initialization

The QSPI initialization procedure must follow the specific steps below to ensure proper behavior.

1. Configure QSPI_MR, QSPI_SCR, QSPI_SMR, etc.
2. Wait for QSPI_SR.SYNCBSY = 0, then write QSPI_CR.UPDCFG. Wait for QSPI_SR.SYNCBSY = 0.
3. Write QSPI_CR.QSPIEN.
4. Wait for QSPI_SR.QSPIENS = 1.

34.6.6.2. Suspend

The QSPI suspend procedure must follow some specific state to ensure proper behavior.

1. Wait for QSPI_SR.RBUSY = 0 and QSPI_SR.HIDLE = 1.
2. Set QSPI_CR.QSPIDIS and wait for QSPI_SR.QSPIENS = 0.
3. Disable the GCLK clock.
4. Disable the peripheral clock.

34.6.6.3. Instruction Frame

In order to control serial Flash memories, the QSPI is able to send instructions via the SPI bus (READ, PROGRAM, ERASE, LOCK, etc.). Because the instruction set implemented in serial Flash memories is memory-vendor dependent, the QSPI includes a complete Instruction Frame register (QSPI_IFR) to ensure compatibility with all serial Flash memories.

An instruction frame includes:

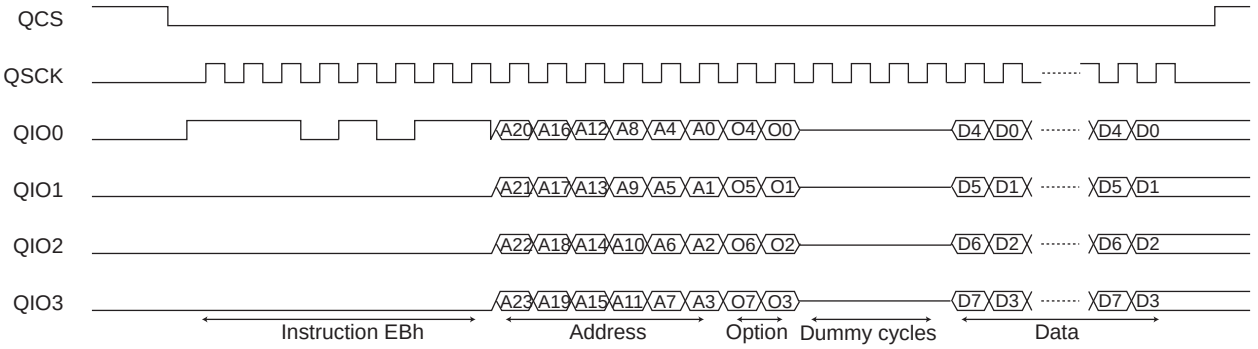
- An instruction code. The instruction is optional in some cases (see Continuous Read Mode).
- An address (size: 8, 16, 24 or 32 bits). The address is optional but is required by instructions such as READ, PROGRAM, ERASE, LOCK. By default, the address is 8 bits long, but can be increased up to 32 bits to support serial Flash memories larger than 128 Mbits (16 Mbytes).
- An option code (size: 1/2/4/8 bits). The option code is used to activate the XIP mode or the Continuous Read mode (see Continuous Read Mode) for READ instructions, in some serial Flash memory devices. These modes improve the data read latency.

© 2026 Microchip Technology Inc. and its subsidiaries

- Dummy cycles. Dummy cycles are optional but required by some READ instructions.
- Data bytes are optional. Data bytes are present for data transfer instructions such as READ or PROGRAM.

The instruction code, the address/option and the data can be sent with Single-bit SPI, Dual SPI or Quad SPI protocols.

Figure 34-11. Instruction Frame



34.6.6.4. Instruction Frame Transmission

If the instruction frame includes the instruction code and/or the option code, the user must configure the instruction code and/or the option code to send by writing the fields WRINST, WROPT, RDINST and RDOPT in the Write Instruction Code register (QSPI_WICR) and the Read Instruction Code register (QSPI_RICR). QSPI_WICR configures instruction code and option code for write accesses, and QSPI_RICR configures instruction code and option code for read accesses. If a frame is without data (QSPI_IFR.DATAEN = 0), then QSPI_WICR is used for instruction code and option code.

Then, the user must write QSPI_IFR to configure the instruction frame depending on which instruction must be sent.

The instruction frame is configured by the following bits and fields of QSPI_IFR:

- WIDTH field—used to configure which data lanes are used to send the instruction code, the address, the option code and to transfer the data.
- INSTEN bit—used to enable the send of an instruction code.
- ADDREN bit—used to enable the send of an address after the instruction code.
- OPTEN bit—used to enable the send of an option code after the address.
- DATAEN bit—used to enable the transfer of data (READ or PROGRAM instruction).
- OPTL field—used to configure the option code length. The value written in OPTL must be consistent with the value written in the field WIDTH. For example: OPTL = 0 (1-bit option code) is not consistent with WIDTH = 6 (option code sent with QuadSPI protocol, thus the minimum length of the option code is 4 bits).
- ADDRL bit—used to configure the address length.
- TFRTYP field—used to define which type of data transfer must be performed.
- CRM bit—used to enable the continuous read mode, see Continuous Read Mode.
- DDREN bit—used to configure the Double Data Rate mode; instruction code is still transmitted in Single Data Rate mode. Instruction code can be transmitted in DDR mode by writing a '1' to QSPI_IFR.DDRCMDEN.
- NBDUM field—used to configure the number of dummy cycles when reading data from the serial Flash memory. Between the address/option and the data, with some instructions, dummy cycles are inserted by the serial Flash memory.

© 2026 Microchip Technology Inc. and its subsidiaries

- SMRM bit—when TFRTYP = '0', defines if instruction frame transmission is triggered by register accesses or QSPI memory space accesses.
- APBTFRTYP bit—used to define the APB register transfer to memory type (read or write) when QSPI_IFR.TFRTYP is written to '0'.
- DQSEN bit—used to define if the targeted memory supplies a DQS signal.
- DDRCMDEN bit—used to define if the instruction code must be sent in DDR mode when QSPI_IFR.DDREN bit is written to '1'.
- PROTTYP bit—used to define the QSPI protocol type.

See QSPI Instruction Frame Register.

For instruction frame transmission, memory array accesses are different from the memory register/command accesses, as well as from accesses using the peripheral bus interface or the system bus interface.

Table 34-3. QSPI Access Triggering

QSPI_IFR Configuration	Accesses Triggered by Performing QSPI Memory Space Access	Accesses Triggered by QSPI Register Accesses
QSPI_IFR.TFRTYP = 0 and QSPI_IFR.SMRM = 1	No	Yes
QSPI_IFR.TFRTYP = 0 and QSPI_IFR.SMRM = 0	Yes	No
QSPI_IFR.TFRTYP = 1 or QSPI_IFR.SMRM = 0	Yes	No

34.6.6.4.1. Memory Registers/Commands Access

To perform memory register/command accesses, QSPI_IFR.TFRTYP must be set to '0'.

If the frame does not contain any data (such as the WRITE ENABLE command) or if QSPI_IFR.SMRM is set to '1', the user must first configure the address to send by writing QSPI_IAR.ADDR (in the case the frame contains an address field).

- SMRM = 1

When QSPI_IFR.SMRM is set to '1', accesses to the memory are triggered and controlled by QSPI registers.

QSPI_IAR.ADDR must be configured if the frame contains an address field.

QSPI_IFR.APBTFRTYP is used to define whether the access is a read access or a write access. Write frames are triggered by writing QSPI_TDR and read frames are triggered by setting QSPI_CR.STTFR. Each time a new transfer trigger is issued, an SPI transfer is performed with a byte size. Another byte is read each time QSPI_RDR is read (flag RDRF shows when a data can be read in QSPI_RDR) or written each time QSPI_TDR is written (flag TDRE shows when a new data can be written). The SPI transfer ends by writing QSPI_CR.LASTXFER. See Instruction Transmission Flow Diagram SMRM = 1 and TFRTYP = 0 (Memory Register Access) for details.

- SMRM = 0

When QSPI_IFR.SMRM is set to '0', accesses to the memory are triggered by performing an access in the QSPI memory space. The address of the instruction frame is defined by the address of the first data access in the QSPI memory space. The addresses of the next accesses are not used by the QSPI.

© 2026 Microchip Technology Inc. and its subsidiaries

QSPI_WRACNT.NBWRA defines the number of bytes to be sent to the memory before QSPI_ISR.LWRA rises, which indicates that the last byte has been transmitted. QSPI_WRACNT.NBWRA is reset and begins counting after each start of the instruction frame. At each system bus access, an SPI transfer is performed with the same size. For example, a halfword system bus access leads to a 16-bit SPI transfer, and a byte system bus access leads to an 8-bit SPI transfer. A special case is an instruction frame with address field and no data. In this case the instruction frame is triggered by setting QSPI_CR.STTFR and QPI_IAR is used for the address field. See Instruction Transmission Flow Diagram SMRM = 0 and TFRTYP = 0 (Memory Register Access) for details.

34.6.6.4.2. Memory Array Access

To access the QSPI memory array, QSPI_IFR.TFRTYP must be set to '1'. In the case of write access to the QSPI memory array, QSPI_IFR.TFRTYP can be set to '0' with QSPI_IFR.SMRM set to '1'.

- TFRTYP = 0 and SMRM = 1

This configuration is allowed for write memory array access only (read access to the memory array is not supported in this configuration). When QSPI_IFR.SMRM is set to '1', accesses to the memory are triggered and controlled by QSPI registers. QSPI_IAR.ADDR must be configured with the address of the first data to write if the frame contains an address field, this field is the one used for the instruction frame address field. The QSPI_IFR.APBTFRTYP must be set to '0'. Write frames are triggered writing the QSPI_TDR register. Each time a new transfer trigger is issued, an SPI transfer is performed with a byte size . Another byte is written each time QSPI_TDR is written (flag TDRE shows when a new data can be written). If a data is not consecutive to the previously sent data, a new frame must be issued. The SPI transfer ends by writing QSPI_CR.LASTXFER. See Instruction Transmission Flow Diagram SMRM = 1 and TFRTYP = 0 (Memory Write Access) for details.

- TFRTYP = 1

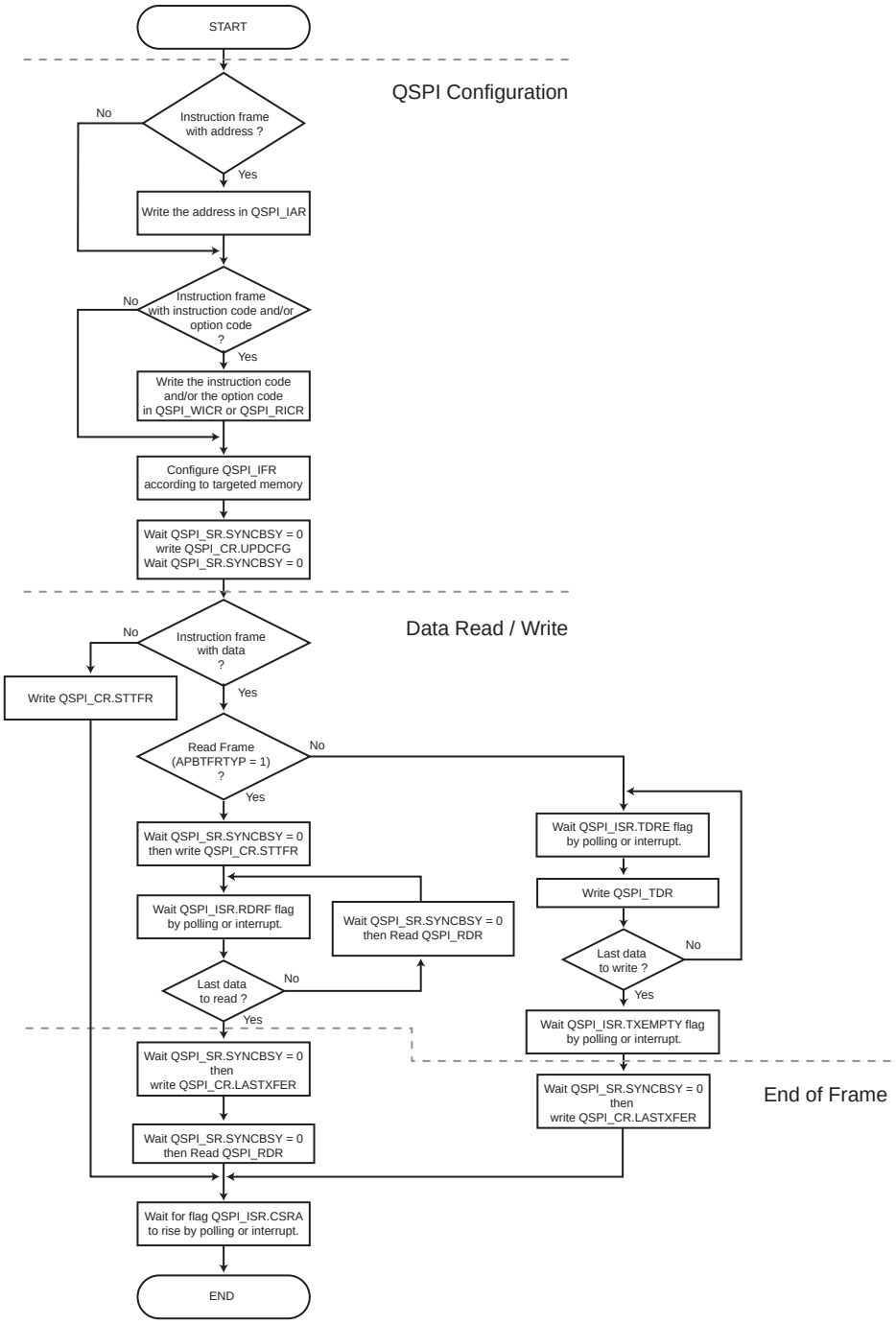
When QSPI_IFR.TFRTYP is set to '1', accesses to the memory are triggered by performing an access in the QSPI memory space. The address of the instruction frame is defined by the address of the first data access in the QSPI memory space. Each time the accesses become nonsequential (addresses are not consecutive), a new instruction frame is sent with the last system bus access address. In this way, the system can read/write data at a random location in the serial memory. The size of the SPI transfers may differ from the size of the system bus access for read accesses. QSPI_WRACNT.NBWRA generates a rising flag when a given number of bytes have been sent to the memory. QSPI_ISR.LWRA indicates the transmission of the last byte. The NBWRA internal counter is reset and begins counting after each start of instruction frame.

In the case of read accesses to the memory array, QSPI_SR.RBUSY must be at '0' before terminating the frame. The last SPI transfer ends by writing QSPI_CR.LASTXFER. See Instruction Transmission Flow Diagram TFRTYP = 1, Memory Write Access and Instruction Transmission Flow Diagram TFRTYP = 1, Memory Read Access for details.

The following figures illustrate instruction transmission management.

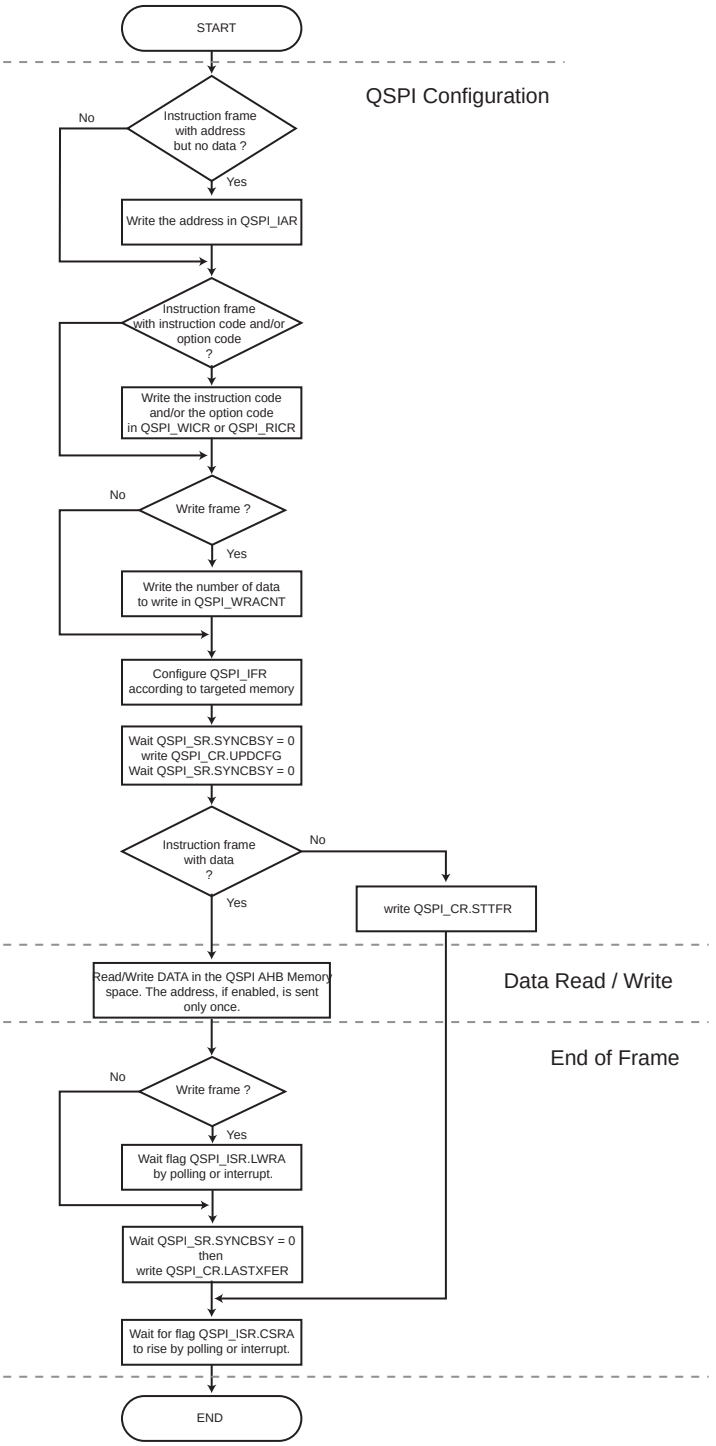
© 2026 Microchip Technology Inc. and its subsidiaries

Figure 34-12. Instruction Transmission Flow Diagram SMRM = 1 and TFRTYP = 0 (Memory Register Access)



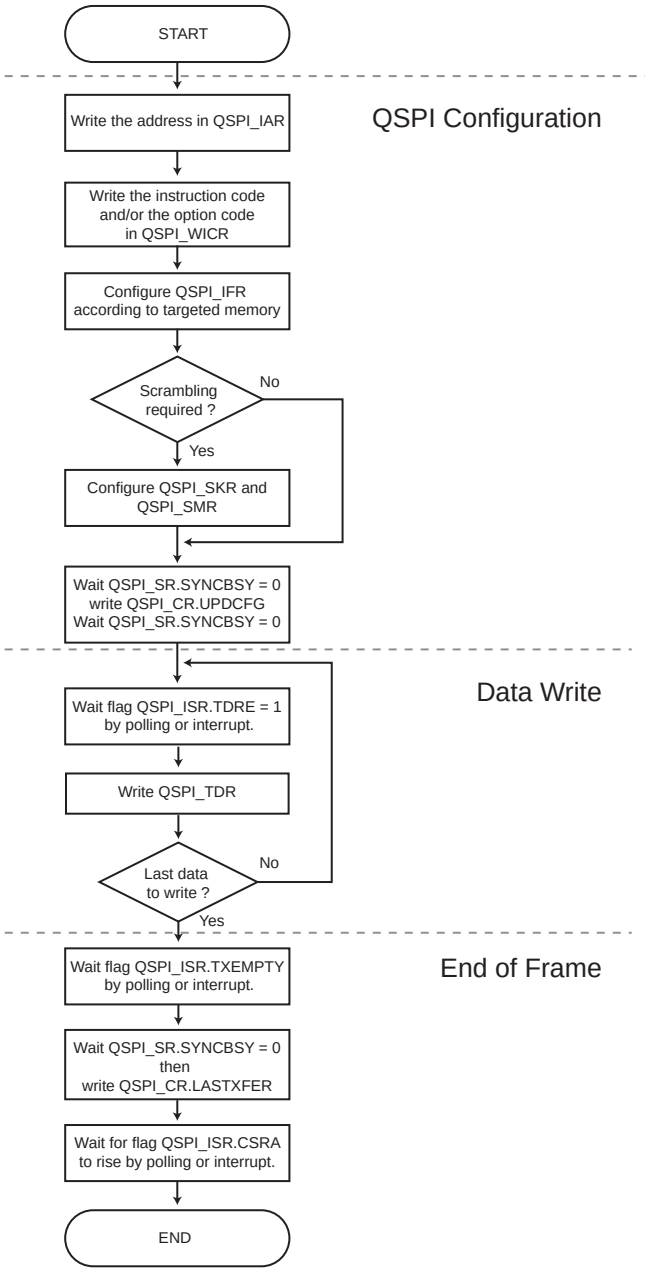
© 2026 Microchip Technology Inc. and its subsidiaries

Figure 34-13. Instruction Transmission Flow Diagram SMRM = 0 and TFRTYP = 0 (Memory Register Access)



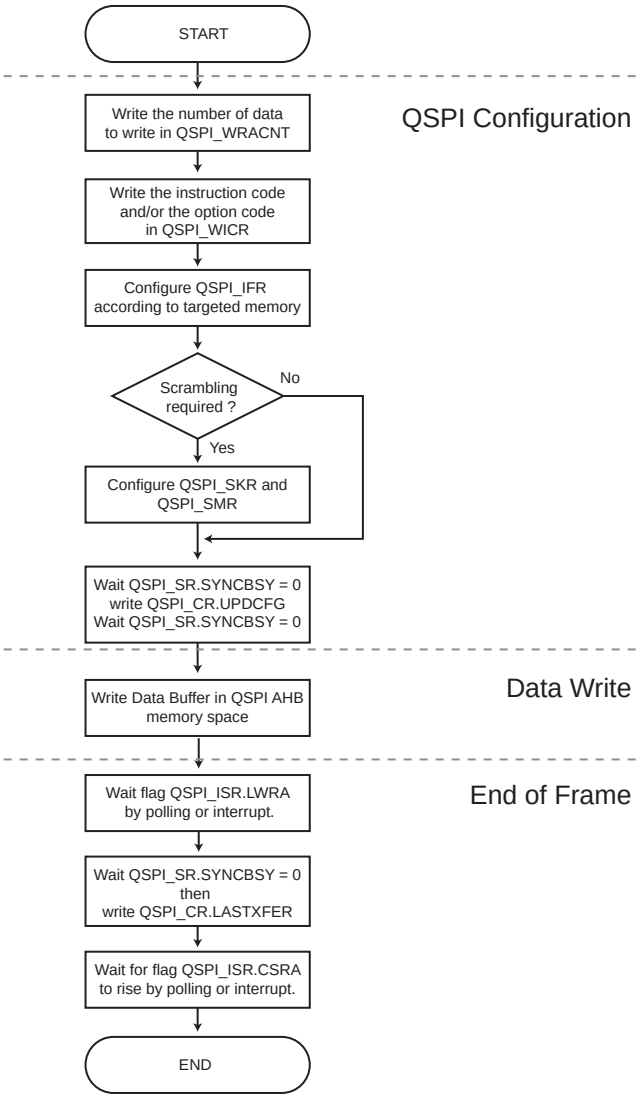
© 2026 Microchip Technology Inc. and its subsidiaries

Figure 34-14. Instruction Transmission Flow Diagram SMRM = 1 and TFRTYP = 0 (Memory Write Access)



© 2026 Microchip Technology Inc. and its subsidiaries

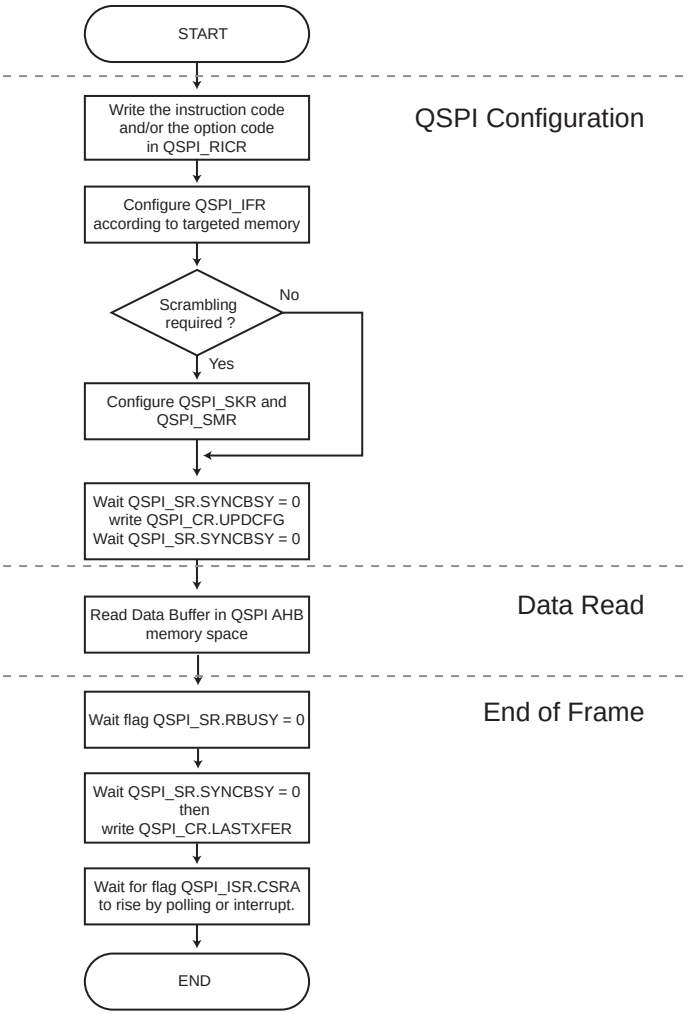
Figure 34-15. Instruction Transmission Flow Diagram TFRTYP = 1, Memory Write Access



Note: For a QSPI Flash memory write, the maximum data buffer size is the Flash page buffer size.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 34-16. Instruction Transmission Flow Diagram TFRTYP = 1, Memory Read Access



34.6.6.5. Write Memory Transfer

Many QSPI memories impose sequential write of the data buffer. When using this type of memory, the device configuration must ensure that the access order is preserved when crossing the different data buses to the QSPI.

34.6.6.6. Read Memory Transfer

The user can access the data of the serial memory by sending an instruction with QSPI_IFR.DATAEN = 1 and QSPI_IFR.TFRTYP = 1.

In this mode, the QSPI is able to read data at random address into the serial Flash memory, allowing the CPU to execute code directly from it (XIP execute-in-place).

In order to fetch data, the user must first configure the instruction frame by writing the QSPI_IFR. Then data can be read at any address in the QSPI address space mapping. The address of the system bus read accesses match the address of the data inside the serial Flash memory.

When Fetch mode is enabled, several instruction frames can be sent before writing QSPI_CR.LASTXFR. Each time the system bus read accesses become nonsequential (addresses are not consecutive), a new instruction frame is sent with the corresponding address.

34.6.6.7. Continuous Read Mode

The QSPI is compatible with the Continuous Read mode which is implemented in some serial Flash memories.

© 2026 Microchip Technology Inc. and its subsidiaries

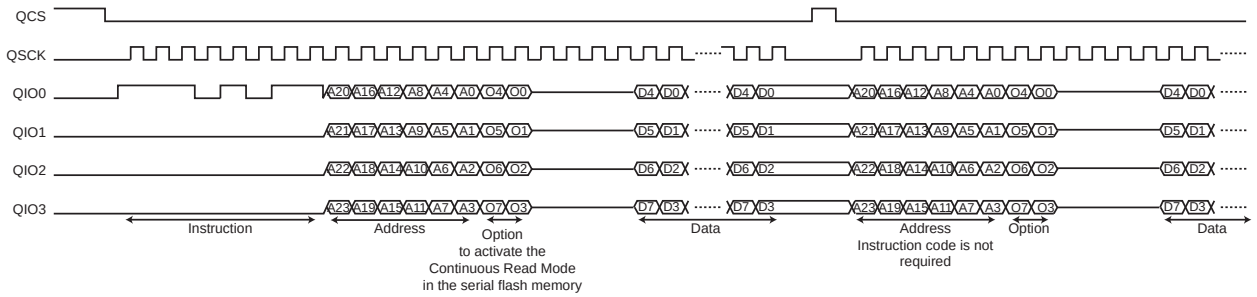
In Continuous Read mode, the instruction overhead is reduced by excluding the instruction code from the instruction frame. When the Continuous Read mode is activated in a serial Flash memory by a specific option code, the instruction code is stored in the memory. For the next instruction frames, the instruction code is not required as the memory uses the stored one.

In the QSPI, Continuous Read mode is used when reading data from the memory (QSPI_IFR.TFRTYP = 1). The addresses of the system bus read accesses are often nonsequential and this leads to many instruction frames that have the same instruction code. By disabling the send of the instruction code, the Continuous Read mode reduces the access time of the data.

To be functional, this mode must be enabled in both the QSPI and the serial Flash memory. The Continuous Read mode is enabled in the QSPI by writing CRM to '1' in the QSPI_IFR (TFRTYP must equal 1). The Continuous Read mode is enabled in the serial Flash memory by sending a specific option code.

CAUTION If the Continuous Read mode is not supported by the serial Flash memory or disabled, CRM bit must not be written to '1', otherwise data read out of the serial Flash memory is unpredictable.

Figure 34-17. Continuous Read Mode



34.6.6.8. Instruction Frame Transmission Examples

All waveforms in the following examples describe SPI transfers in SPI Clock mode 0 (QSPI_SCR.CPOL = 0 and QSPI_SCR.CPHA = 0; see Serial Clock Phase and Polarity).

All system bus accesses described below refer to the system bus address phase. System bus wait cycles and system bus data phases are not shown.

Example 1:

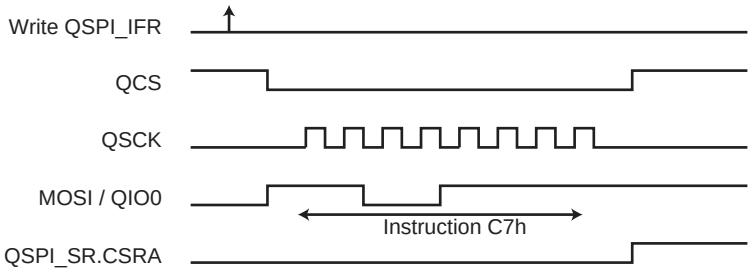
Instruction in Single-bit SPI, without address, without option, without data.

Command: CHIP ERASE (C7h).

- Write 0x0000_00C7 in QSPI_WICR.
- Write 0x0000_0010 in QSPI_IFR.
- Update configuration (See Updating the QSPI Configuration)
- Write QSPI_CR.STTFR to '1'.
- Wait for QSPI_ISR.CSRA to rise.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 34-18. Instruction Transmission Waveform 1



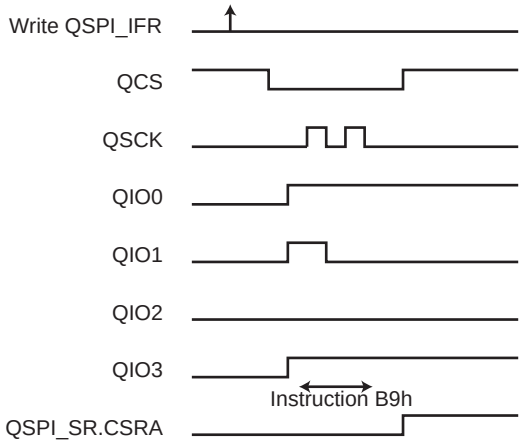
Example 2:

Instruction in Quad SPI, without address, without option, without data.

Command: POWER DOWN (B9h)

- Write 0x0000_00B9 in QSPI_WICR.
- Write 0x0000_0016 in QSPI_IFR.
- Update Configuration (See Updating the QSPI Configuration)
- Write QSPI_CR.STTFR to '1'
- Wait for QSPI_ISR.CSRA to rise.

Figure 34-19. Instruction Transmission Waveform 2



Example 3:

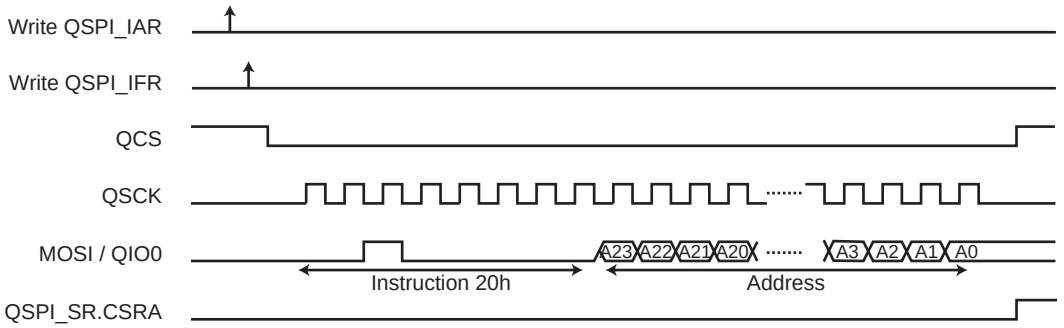
Instruction in Single-bit SPI, with address in Single-bit SPI, without option, without data.

Command: BLOCK ERASE (20h)

- Write the address (of the block to erase) in QSPI_IAR.
- Write 0x0000_0020 in QSPI_WICR.
- Write 0x0000_0030 in QSPI_IFR.
- Update Configuration (See Updating the QSPI Configuration)
- Write QSPI_CR.STTFR to '1'
- Wait for QSPI_ISR.CSRA to rise.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 34-20. Instruction Transmission Waveform 3



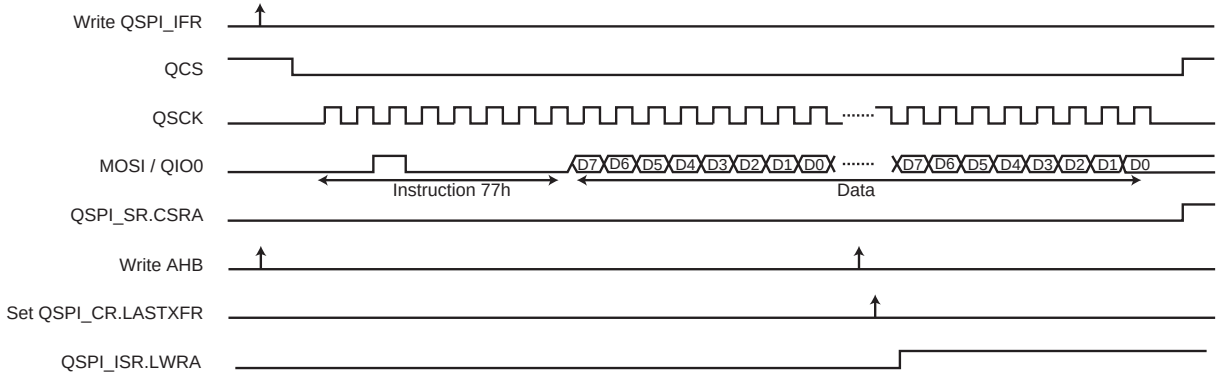
Example 4:

Instruction in Single-bit SPI, without address, without option, with data write in Single-bit SPI.

Command: SET BURST (77h)

- Write 0x0000_0077 in QSPI_WICR.
- Write 0x0000_0090 in QSPI_IFR.
- Write QSPI_WRACNT.NBWRA with the number of bytes to write.
- Update Configuration (See Updating the QSPI Configuration).
- Write data in the system bus memory space (0x04000000–0x06000000).
  The address of system bus write accesses is not used.
- Wait for QSPI_ISR.LWRA to rise.
- Write a '1' to QSPI_CR.LASTXFR.
- Wait for QSPI_ISR.CSRA to rise.

Figure 34-21. Instruction Transmission Waveform 4



Example 5:

Instruction in Single-bit SPI, with address in Dual SPI, without option, with data write in Dual SPI.

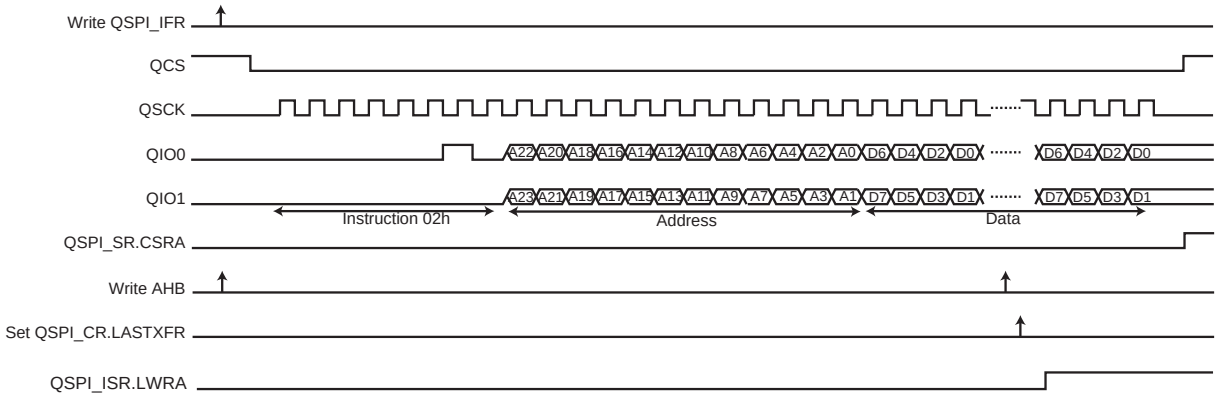
Command: BYTE/PAGE PROGRAM (02h)

- Write 0x0000_0002 in QSPI_WICR.
- Write 0x0000_18B3 in QSPI_IFR.
- Write QSPI_WRACNT.NBWRA with the number of bytes to write.
- Update Configuration (See Updating the QSPI Configuration).

© 2026 Microchip Technology Inc. and its subsidiaries

- Write data in the QSPI system bus memory space (0x04000000–0x06000000).
  The address of the first system bus write access is sent in the instruction frame.
  The address of the next system bus write accesses is not used.
- Wait for QSPI_ISR.LWRA to rise.
- Write a '1' to QSPI_CR.LASTXFR.
- Wait for QSPI_ISR.CSRA to rise.

Figure 34-22. Instruction Transmission Waveform 5



Example 6:

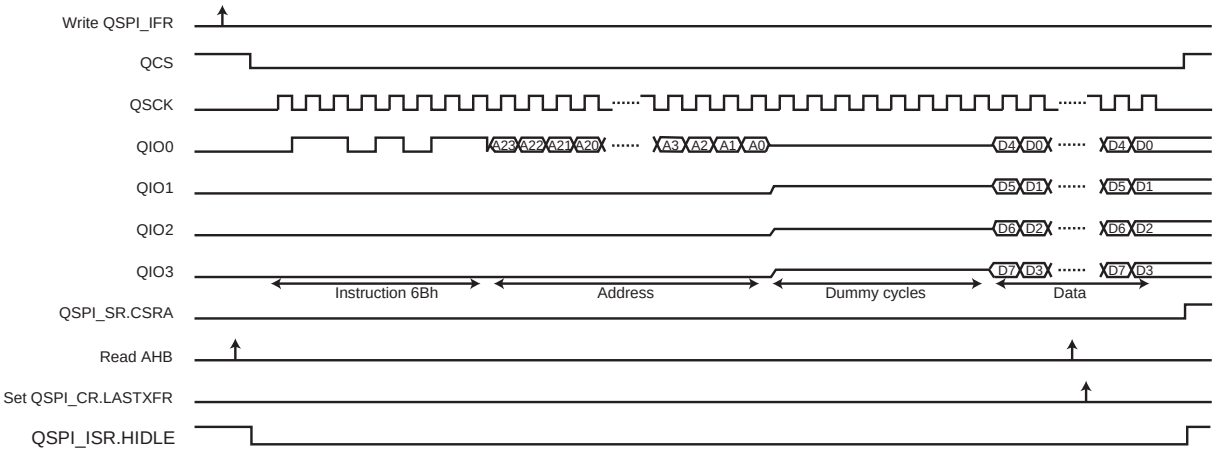
Instruction in Single-bit SPI, with address in Single-bit SPI, without option, with data read in Quad SPI, with eight dummy cycles.

Command: QUAD_OUTPUT READ ARRAY (6Bh)

- Write 0x0000_006B in QSPI_RICR.
- Write 0x0008_18B2 in QSPI_IFR.
- Update Configuration (See Updating the QSPI Configuration).
- Read data in the QSPI system bus memory space (0x04000000–0x06000000).
  The address of the first system bus read access is sent in the instruction frame.
  The address of the next system bus read accesses is not used.
- Wait for QSPI_SR.RBUSY to be at '0'.
- Write a '1' to QSPI_CR.LASTXFR.
- Wait for QSPI_ISR.CSRA to rise.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 34-23. Instruction Transmission Waveform 6



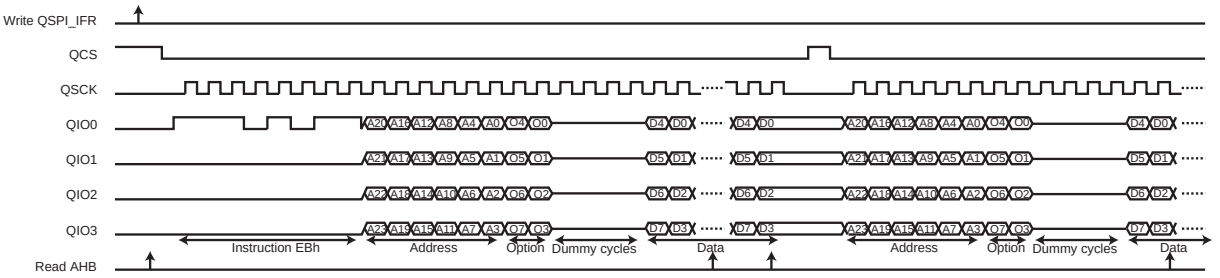
Example 7:

Instruction in Single-bit SPI, with address and option in Quad SPI, with data read in Quad SPI, with four dummy cycles, with fetch and continuous read.

Command: FAST READ QUAD I/O (EBh) - 8-BIT OPTION (0x30h)

- Write 0x0030_00EB in QSPI_RICR.
- Write 0x0004_3BF4 in QSPI_IFR.
- Update Configuration (See Updating the QSPI Configuration).
- Read data in the QSPI system bus memory space (0x04000000–0x06000000).
  Fetch is enabled, the address of the system bus read accesses is always used.
- Wait for QSPI_SR.RBUSY to be at '0'.
- Write a '1' to QSPI_CR.LASTXFR.
- Wait for QSPI_ISR.CSRA to rise.

Figure 34-24. Instruction Transmission Waveform 7



Example 8:

Instruction in Quad SPI, with address in Quad SPI, without option, with data read in Quad SPI, with two dummy cycles, with fetch.

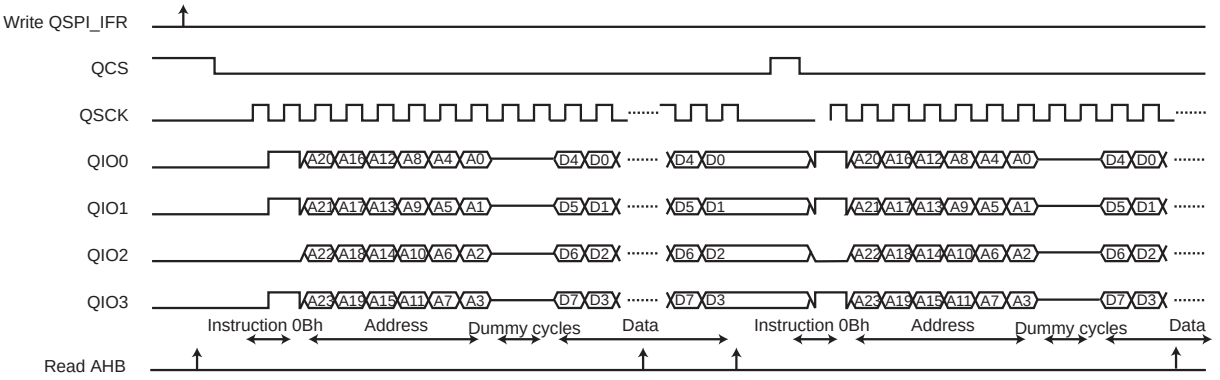
Command: HIGH-SPEED READ (0Bh)

- Write 0x0000_000B in QSPI_RICR.
- Write 0x0002_28B6 in QSPI_IFR.
- Update Configuration (See Updating the QSPI Configuration).
- Read data in the QSPI system bus memory space (0x04000000–0x06000000).
  Fetch is enabled, the address of the system bus read accesses is always used.

© 2026 Microchip Technology Inc. and its subsidiaries

- Wait for QSPI_SR.RBUSY to be at '0'.
- Write a '1' to QSPI_CR.LASTXFR.
- Wait for QSPI_ISR.CSRA to rise.

Figure 34-25. Instruction Transmission Waveform 8



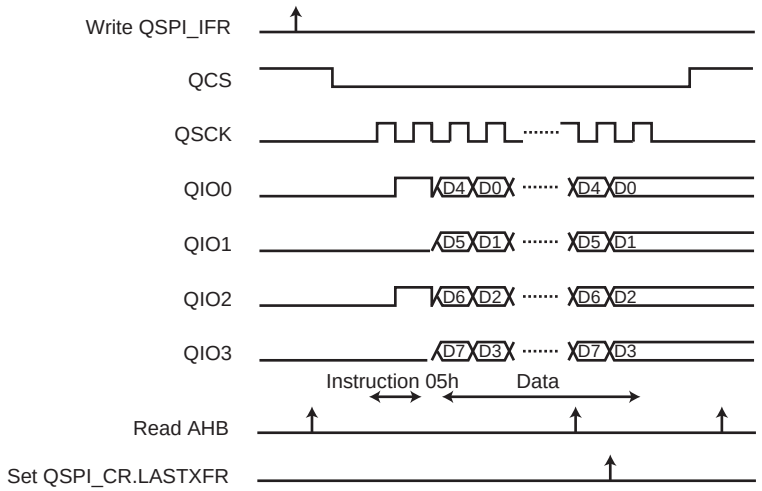
Example 9:

Instruction in Quad SPI, without address, without option, with data read in Quad SPI, without dummy cycles, without fetch.

Command: HIGH-SPEED READ (05h)

- Write 0x0000_0005 in QSPI_RICR.
- Write 0x0000_0096 in QSPI_IFR.
- Update Configuration (See Updating the QSPI Configuration).
- Read data in the QSPI system bus memory space (0x04000000–0x06000000). Fetch is disabled.
- Wait for QSPI_SR.RBUSY to be at '0'.
- Write a '1' to QSPI_CR.LASTXFR.
- Wait for QSPI_ISR.CSRA to rise.

Figure 34-26. Instruction Transmission Waveform 9



Example 10:

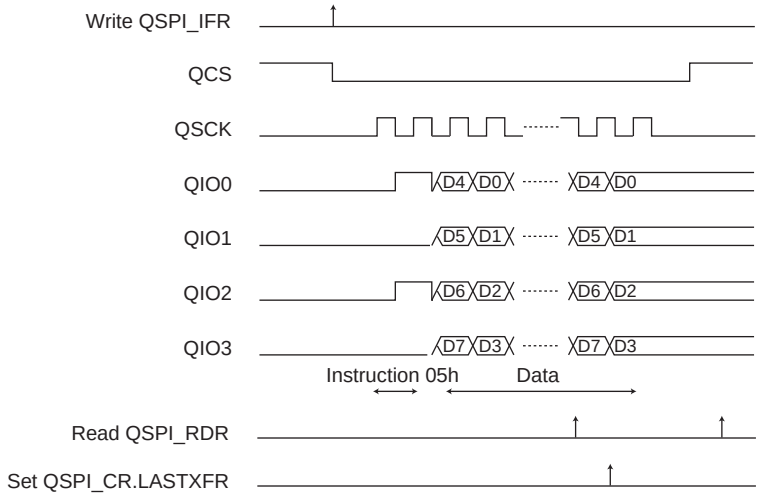
© 2026 Microchip Technology Inc. and its subsidiaries

Instruction in Quad SPI, without address, without option, with data read in Quad SPI, without dummy cycles, without fetch, read launched through APB interface.

Command: HIGH-SPEED READ (05h)

- Set SMRM to '1' and TFRTYP to '0' in QSPI_MR
- Write 0x0000_0005 in QSPI_RICR.
- Write 0x0100_0096 in QSPI_IFR.
- Update Configuration (See Updating the QSPI Configuration).
- Write a '1' to QSPI_CR.STTFR.
- Wait flag RDRF and Read data in the QSPI_RDR register
  Fetch is disabled.
- Write a '1' to QSPI_CR.LASTXFR.
- Wait for QSPI_ISR.CSRA to rise.
- Read data in the QSPI_RDR register.

Figure 34-27. Instruction Transmission Waveform 10



34.6.6.9. Time-out

The QSPI includes a time-out counter. This time-out counter detects any blocking state on the memory side (hardware connection issue, unknown command sent, etc,). If the QSPI is expecting data from the memory, it starts counting and if the counter reaches the value set in QSPI_TOUT.TCNTM, the flag QSPI_ISR.TOUT rises.

When the TOUT flag rises, any access waiting on the system bus interface is released with an error response so that the system bus is released. Any further access to the QSPI system bus interface receives an error response until the TOUT flag is cleared.

After a time-out error, it is mandatory to clear the TOUT flag by writing QSPI_CR.RTOUT to '1'. This also resets the QSPI internal state machines.

If QSPI_TOUT.TCNTM is set to '0', the time-out feature is disabled.

34.6.7. Scrambling/Unscrambling Function

The scrambling/unscrambling function cannot be performed on devices other than memories. Data is scrambled when written to memory and unscrambled when data is read.

© 2026 Microchip Technology Inc. and its subsidiaries

The external data lines can be scrambled in order to prevent intellectual property data located in off-chip memories from being easily recovered by analyzing data at the package pin level of either the microcontroller or the QSPI client device (memory, for example).

The scrambling/unscrambling function can be enabled by writing a '1' to the SCREN bit in the QSPI Scrambling Mode Register (QSPI_SMR).

The scrambling and unscrambling are performed on-the-fly without impacting the throughput.

The scrambling method depends on the user-configurable user scrambling key (field USRK) in the QSPI Scrambling Key Register (QSPI_SKR). QSPI_SKR is only accessible in Write mode.

When QSPI_SMR.SCRKL has been written once to '1', QSPI_SKR.USRK cannot be written again until the next reset.

If QSPI_SMR.RVDIS is written to '0', the scrambling/unscrambling algorithm includes the user scrambling key plus a random value depending on device processing characteristics. Data scrambled by a given microcontroller cannot be unscrambled by another.

If QSPI_SMR.RVDIS is written to '1', the scrambling/unscrambling algorithm includes only the user scrambling key. No random value is part of the key.

The user scrambling key or the seed for key generation must be securely stored in a reliable nonvolatile memory in order to recover data from the off-chip memory. Any data scrambled with a given key cannot be recovered if the key is lost.

34.6.7.1. Clearing Scrambling Keys on a Tamper Event

On a tamper detection event on WKUP0/1 pins (refer to the section "Supply Controller (SUPC)"), an immediate clear of the scrambling key (QSPI_SKR) is performed if QSPI_MR.TAMPCLR is set.

34.6.7.2. Clearing Scrambling Keys On Embedded Flash Erase

The scrambling keys are cleared when a Flash erase has been initiated by the erase pin.

34.6.8. Register Write Protection

To prevent any single software error from corrupting QSPI behavior, certain registers in the address space can be write-protected by setting the WPEN bit in the QSPI Write Protection Mode Register (QSPI_WPMR).

If a write access to a write-protected register is detected, the WPVS flag in the QSPI Write Protection Status Register (QSPI_WPSR) is set and the field WPVSRC indicates the register in which the write access has been attempted.

The WPVS bit is automatically cleared after reading the QSPI_WPSR.

The following registers can be write-protected when WPEN is set in QSPI_WPMR:

- QSPI Mode Register
- QSPI Serial Clock Register
- QSPI Instruction Address Register
- QSPI Write Instruction Code Register
- QSPI Instruction Frame Register
- QSPI Read Instruction Code Register
- QSPI Scrambling Mode Register
- QSPI Scrambling Key Register
- QSPI Write Access Counter Register
- QSPI Timeout Register

The following registers can be write-protected when WPCREN is set in QSPI_WPMR:

© 2026 Microchip Technology Inc. and its subsidiaries

- QSPI Control Register

The following registers can be write-protected when WPITEN is set in QSPI_WPMR:

- QSPI Interrupt Enable Register
- QSPI Interrupt Disable Register

© 2026 Microchip Technology Inc. and its subsidiaries

34.7. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	QSPI_CR	31:24								LASTXFER
		23:16								
		15:8						RTOUT	STTFR	UPDCFG
		7:0	SWRST						QSPIDIS	QSPIEN
0x04	QSPI_MR	31:24	DLYCS[7:0]							
		23:16	DLYBCT[7:0]							
		15:8					NBBITS[3:0]			
		7:0	TAMPCLR		CSMODE[1:0]			WDRBT		SMM
0x08	QSPI_RDR	31:24								
		23:16								
		15:8	RD[15:8]							
		7:0	RD[7:0]							
0x0C	QSPI_TDR	31:24								
		23:16								
		15:8	TD[15:8]							
		7:0	TD[7:0]							
0x10	QSPI_ISR	31:24								
		23:16							TOUT	
		15:8	CSRA	CSFA	QITR	QITF	LWRA	INSTRE		CSR
		7:0	TXBUFE	RXBUFF	ENDTX	ENDRX	OVRES	TXEMPTY	TDRE	RDRF
0x14	QSPI_IER	31:24								
		23:16							TOUT	
		15:8	CSRA	CSFA	QITR	QITF	LWRA	INSTRE		CSR
		7:0	TXBUFE	RXBUFF	ENDTX	ENDRX	OVRES	TXEMPTY	TDRE	RDRF
0x18	QSPI_IDR	31:24								
		23:16							TOUT	
		15:8	CSRA	CSFA	QITR	QITF	LWRA	INSTRE		CSR
		7:0	TXBUFE	RXBUFF	ENDTX	ENDRX	OVRES	TXEMPTY	TDRE	RDRF
0x1C	QSPI_IMR	31:24								
		23:16							TOUT	
		15:8	CSRA	CSFA	QITR	QITF	LWRA	INSTRE		CSR
		7:0	TXBUFE	RXBUFF	ENDTX	ENDRX	OVRES	TXEMPTY	TDRE	RDRF
0x20	QSPI_SCR	31:24								
		23:16	DLYBS[7:0]							
		15:8								
		7:0							CPHA	CPOL
0x24	QSPI_SR	31:24								
		23:16								
		15:8								
		7:0				HIDLE	RBUSY	CSS	QSPIENS	SYNCBSY
0x28 ... 0x2F	Reserved									
0x30	QSPI_IAR	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x34	QSPI_WICR	31:24								
		23:16	WROPT[7:0]							
		15:8								
		7:0	WRINST[7:0]							
0x38	QSPI_IFR	31:24			PROTTYP[1:0]			DDRCMDEN	DQSEN	APBTFRTYP
		23:16	SMRM			NBDUM[4:0]				
		15:8	DDREN	CRM		TFRTYP	ADDRL[1:0]		OPTL[1:0]	
		7:0	DATAEN	OPTEN	ADDREN	INSTEN	WIDTH[3:0]			
0x3C	QSPI_RICR	31:24								
		23:16	RDOPT[7:0]							
		15:8								
		7:0	RDINST[7:0]							

© 2026 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x40	QSPI_SMR	31:24								
		23:16								
		15:8								
		7:0						SCRKL	RVDIS	SCREN
0x44	QSPI_SKR	31:24	USRK[31:24]							
		23:16	USRK[23:16]							
		15:8	USRK[15:8]							
		7:0	USRK[7:0]							
0x48 ... 0x53	Reserved									
0x54	QSPI_WRACNT	31:24	NBWRA[31:24]							
		23:16	NBWRA[23:16]							
		15:8	NBWRA[15:8]							
		7:0	NBWRA[7:0]							
0x58 ... 0x63	Reserved									
0x64	QSPI_TOUT	31:24								
		23:16								
		15:8	TCNTM[15:8]							
		7:0	TCNTM[7:0]							
0x68 ... 0xE3	Reserved									
0xE4	QSPI_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0						WPCREN	WPITEN	WPEN
0xE8	QSPI_WPSR	31:24								
		23:16								
		15:8	WPVSRC[7:0]							
		7:0								WPVS

© 2026 Microchip Technology Inc. and its subsidiaries

34.7.1. QSPI Control Register

Name: QSPI_CR
Offset: 0x00
Reset: –
Property: Write-only

This register can only be written if the WPCREN bit is cleared in the QSPI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
								LASTXFER
Access								W
Reset								–

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						RTOUT	STTFR	UPDCFG
Access						W	W	W
Reset						–	–	–

Bit	7	6	5	4	3	2	1	0
	SWRST						QSPIDIS	QSPIEN
Access	W						W	W
Reset	–						–	–

Bit 24 – LASTXFER Last Transfer

Value	Description
0	No effect.
1	The chip select is deasserted after the end of character transmission.

Bit 10 – RTOUT Reset Time-out

Value	Description
0	No effect.
1	Request a TOUT flag reset.

Bit 9 – STTFR Start Transfer

Value	Description
0	No effect.
1	Starts the transfer when TFRTYP = 0 and SMRM = 1 or when DATAEN = 0.

Bit 8 – UPDCFG Update Configuration

Value	Description
0	No effect.
1	Requests an update of the QSPI Controller core configuration.

Bit 7 – SWRST QSPI Software Reset
PDC channels state are not affected by a software reset.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	No effect.
1	Resets the QSPI. A software-triggered hardware reset of the QSPI interface is performed.

Bit 1 – QSPIDIS QSPI Disable

As soon as QSPIDIS is set, the QSPI finishes its transfer.
All pins are set in Input mode and no data is received or transmitted.
If a transfer is in progress, the transfer is finished before the QSPI is disabled.
If QSPIEN and QSPIDIS are at '1' when QSPI_CR is written, the QSPI is disabled.

Value	Description
0	No effect.
1	Disables the QSPI.

Bit 0 – QSPIEN QSPI Enable

Value	Description
0	No effect.
1	Enables the QSPI to transfer and receive data.

© 2026 Microchip Technology Inc. and its subsidiaries

34.7.2. QSPI Mode Register

Name: QSPI_MR
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

This register can only be written if bit WPEN is cleared in the QSPI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	DLYCS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DLYBCT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
					NBBITS[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TAMPCLR		CSMODE[1:0]			WDRBT		SMM
Access	R/W		R/W	R/W		R/W		R/W
Reset	0		0	0		0		0

Bits 31:24 – DLYCS[7:0] Minimum Inactive QCS Delay
Defines the minimum delay between the deactivation and the activation of QCS. The DLYCS time ensures the client minimum deselect time is respected.
If DLYCS written to '0', one GCLK period is inserted by default.
Otherwise, the following equation determines the delay:
DLYCS = Minimum inactive × f_{GCLK}

Bits 23:16 – DLYBCT[7:0] Delay Between Consecutive Transfers
• SMM = 0
Defines the delay between two consecutive transfers with the same peripheral without removing the chip select. The delay is always inserted after each transfer and before removing the chip select if needed.
The following equation determines the delay:
DLYBCT = (Delay Between Consecutive Transfers × f_{GCLK}) / 32
• SMM = 1
Defines the delay between last QSCK pulse and QCS rise.
When DLYBCT is written to '0', no delay is inserted and the clock keeps its duty cycle over the character transfers.
The following equation determines the delay:
DLYBCT = Delay Between Consecutive Transfers × f_{GCLK}

Bits 11:8 – NBBITS[3:0] Number Of Bits Per Transfer
NBBITS is used only when SMM is set to '0'.

Value	Name	Description
0	8_BIT	8 bits for transfer

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Name	Description
8	16_BIT	16 bits for transfer

Bit 7 – TAMPCLR Tamper Clear Enable

Value	Description
0	A tamper detection event has no effect on QSPI scrambling keys.
1	A tamper detection event immediately clears QSPI scrambling keys.

Bits 5:4 – CSMODE[1:0] Chip Select Mode

The CSMODE field determines how the chip select is deasserted.
This field is forced to LASTXFER when SMM is written to '1'.

Value	Name	Description
0	NOT_RELOADED	The chip select is deasserted if QSPI_TDR.TD has not been reloaded before the end of the current transfer.
1	LASTXFER	The chip select is deasserted when the bit LASTXFER is written to '1' and the character written in QSPI_TDR.TD has been transferred.
2	SYSTEMATICALLY	The chip select is deasserted systematically after each transfer.

Bit 2 – WDRBT Wait Data Read Before Transfer

Value	Name	Description
0	DISABLED	No effect. In SPI mode, a transfer can be initiated whatever the state of QSPI_RDR is.
1	ENABLED	In SPI mode, a transfer can start only if QSPI_RDR is empty, i.e., does not contain any unread data. This mode prevents overrun error in reception.

Bit 0 – SMM Serial Memory Mode

Value	Name	Description
0	SPI	The QSPI is in SPI mode.
1	MEMORY	The QSPI is in Serial Memory mode.

© 2026 Microchip Technology Inc. and its subsidiaries

34.7.3. QSPI Receive Data Register

Name: QSPI_RDR
Offset: 0x08
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	RD[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RD[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – RD[15:0] Receive Data

Data received by the QSPI is stored in this register right-justified. Unused bits read zero.

- QSPI_MR.SMM = 0

RD is defined by QSPI_MR.NBBITS.

- QSPI_MR.SMM = 1

RD is 8 bits.

© 2026 Microchip Technology Inc. and its subsidiaries

34.7.4. QSPI Transmit Data Register

Name: QSPI_TDR
Offset: 0x0C
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	TD[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	TD[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 15:0 – TD[15:0] Transmit Data

Data to be transmitted by the QSPI is stored in this register. Information to be transmitted must be written to the Transmit Data register in a right-justified format.

• QSPI_MR.SMM = 0

TD is defined by QSPI_MR.NBBITS field.

• QSPI_MR.SMM = 1

TD is 8 bits.

© 2026 Microchip Technology Inc. and its subsidiaries

34.7.5. QSPI Interrupt Status Register

Name: QSPI_ISR
Offset: 0x10
Reset: 0x000000F0
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							TOUT	
Access							R	
Reset							0	

Bit	15	14	13	12	11	10	9	8
	CSRA	CSFA	QITR	QITF	LWRA	INSTRE		CSR
Access	R	R	R	R	R	R		R
Reset	0	0	0	0	0	0		0

Bit	7	6	5	4	3	2	1	0
	TXBUFE	RXBUFF	ENDTX	ENDRX	OVRES	TXEMPTY	TDRE	RDRF
Access	R	R	R	R	R	R	R	R
Reset	1	1	1	1	0	0	0	0

Bit 17 – TOUT QSPI Time-out

Value	Description
0	No QSPI time-out occurred since the last write of RTOUT bit on QSPI_CR.
1	At least one QSPI time-out occurred since the last write of RTOUT bit on QSPI_CR.

Bit 15 – CSRA Chip Select Rise Autoclear

Value	Description
0	No chip select rise has been detected since beginning of the last command or the last read of QSPI_ISR.
1	One chip select rise has been detected since the beginning of the last command or the last read of QSPI_ISR.

Bit 14 – CSFA Chip Select Fall Autoclear

Value	Description
0	No chip select fall has been detected since end of the last command or the last read of QSPI_ISR.
1	One chip select fall has been detected since the end of the last command or the last read of QSPI_ISR.

Bit 13 – QITR QSPI Interrupt Rise

Value	Description
0	No rising of the QSPI memory interrupt line has been detected since the last read of QSPI_ISR.
1	At least one QSPI memory interrupt line rising edge occurred since the last read of QSPI_ISR.

Bit 12 – QITF QSPI Interrupt Fall

Value	Description
0	No falling of the QSPI memory interrupt line has been detected since the last read of QSPI_ISR.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
1	At least one QSPI memory interrupt line falling edge occurred since the last read of QSPI_ISR.

Bit 11 – LWRA Last Write Access (cleared on read)

Value	Description
0	Last write access has not been sent since the last read of QSPI_SR or NBWRA = 0.
1	At least one last write access has been sent since the last read of QSPI_SR.

Bit 10 – INSTRE Instruction End Status (cleared on read)

Value	Description
0	No instruction end has been detected since the last read of QSPI_SR.
1	At least one instruction end has been detected since the last read of QSPI_SR.

Bit 8 – CSR Chip Select Rise (cleared on read)

Value	Description
0	No chip select rise has been detected since the last read of QSPI_SR.
1	At least one chip select rise has been detected since the last read of QSPI_SR.

Bit 7 – TXBUFE TX Buffer Empty
QSPI_TCR and QSPI_TNCR are physically located in the PDC.

Value	Description
0	QSPI_TCR or QSPI_TNCR has a value other than 0.
1	Both QSPI_TCR and QSPI_TNCR have a value of 0.

Bit 6 – RXBUFF RX Buffer Full
QSPI_RCR and QSPI_RNCR are physically located in the PDC.

Value	Description
0	QSPI_RCR or QSPI_RNCR has a value other than 0.
1	Both QSPI_RCR and QSPI_RNCR have a value of 0.

Bit 5 – ENDTX End of TX Buffer
QSPI_TCR and QSPI_TNCR are physically located in the PDC.

Value	Description
0	The Receive Counter register has not reached 0 since the last write in QSPI_TCR or QSPI_TNCR.
1	The Receive Counter register has reached 0 since the last write in QSPI_TCR or QSPI_TNCR.

Bit 4 – ENDRX End of RX Buffer
QSPI_RCR and QSPI_RNCR are physically located in the PDC.

Value	Description
0	The Receive Counter register has not reached 0 since the last write in QSPI_RCR or QSPI_RNCR.
1	The Receive Counter register has reached 0 since the last write in QSPI_RCR or QSPI_RNCR.

Bit 3 – OVRES Overrun Error Status (cleared on read)
An overrun occurs when QSPI_RDR is loaded at least twice from the serializer since the last read of the QSPI_RDR.

Value	Description
0	No overrun has been detected since the last read of QSPI_ISR.
1	At least one overrun error has occurred since the last read of QSPI_ISR.

Bit 2 – TXEMPTY Transmission Registers Empty (cleared by writing QSPI_TDR)

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	As soon as data is written in QSPI_TDR.
1	QSPI_TDR and the internal shifter are empty. If a transfer delay has been defined, TXEMPTY is set after the completion of such delay.

Bit 1 – TDRE Transmit Data Register Empty (cleared by writing QSPI_TDR)

TDRE equals zero when the QSPI is disabled or at reset. The QSPI enable command sets this bit to one.

Value	Description
0	Data has been written to QSPI_TDR and not yet transferred to the serializer.
1	The last data written in the QSPI_TDR has been transferred to the serializer.

Bit 0 – RDRF Receive Data Register Full (cleared by reading QSPI_RDR)

Value	Description
0	No data has been received since the last read of QSPI_RDR.
1	Data has been received and the received data has been transferred from the serializer to QSPI_RDR since the last read of QSPI_RDR.

© 2026 Microchip Technology Inc. and its subsidiaries

34.7.6. QSPI Interrupt Enable Register

Name: QSPI_IER
Offset: 0x14
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the QSPI Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							TOUT	
Access							W	
Reset							–	

Bit	15	14	13	12	11	10	9	8
	CSRA	CSFA	QITR	QITF	LWRA	INSTRE		CSR
Access	W	W	W	W	W	W		W
Reset	–	–	–	–	–	–		–

Bit	7	6	5	4	3	2	1	0
	TXBUFE	RXBUFF	ENDTX	ENDRX	OVRES	TXEMPTY	TDRE	RDRF
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 17 – TOUT QSPI Time-out Interrupt Enable

Bit 15 – CSRA Chip Select Rise Autoclear Interrupt Enable

Bit 14 – CSFA Chip Select Fall Autoclear Interrupt Enable

Bit 13 – QITR QSPI Interrupt Rise Interrupt Enable

Bit 12 – QITF QSPI Interrupt Fall Interrupt Enable

Bit 11 – LWRA Last Write Access Interrupt Enable

Bit 10 – INSTRE Instruction End Interrupt Enable

Bit 8 – CSR Chip Select Rise Interrupt Enable

Bit 7 – TXBUFE Transmit Buffer Empty Interrupt Enable

Bit 6 – RXBUFF Receive Buffer Full Interrupt Enable

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 5 – ENDTX End of Transmit Buffer Interrupt Enable

Bit 4 – ENDRX End of Receive Buffer Interrupt Enable

Bit 3 – OVRES Overrun Error Interrupt Enable

Bit 2 – TXEMPTY Transmission Registers Empty Enable

Bit 1 – TDRE Transmit Data Register Empty Interrupt Enable

Bit 0 – RDRF Receive Data Register Full Interrupt Enable

© 2026 Microchip Technology Inc. and its subsidiaries

34.7.7. QSPI Interrupt Disable Register

Name: QSPI_IDR
Offset: 0x18
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the QSPI Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							TOUT	
Access							W	
Reset							–	

Bit	15	14	13	12	11	10	9	8
	CSRA	CSFA	QITR	QITF	LWRA	INSTRE		CSR
Access	W	W	W	W	W	W		W
Reset	–	–	–	–	–	–		–

Bit	7	6	5	4	3	2	1	0
	TXBUFE	RXBUFF	ENDTX	ENDRX	OVRES	TXEMPTY	TDRE	RDRF
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 17 – TOUT QSPI Time-out Interrupt Disable

Bit 15 – CSRA Chip Select Rise Autoclear Interrupt Disable

Bit 14 – CSFA Chip Select Fall Autoclear Interrupt Disable

Bit 13 – QITR QSPI Interrupt Rise Interrupt Disable

Bit 12 – QITF QSPI Interrupt Fall Interrupt Disable

Bit 11 – LWRA Last Write Access Interrupt Disable

Bit 10 – INSTRE Instruction End Interrupt Disable

Bit 8 – CSR Chip Select Rise Interrupt Disable

Bit 7 – TXBUFE Transmit Buffer Empty Interrupt Disable

Bit 6 – RXBUFF Receive Buffer Full Interrupt Disable

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 5 – ENDTX End of Transmit Buffer Interrupt Disable

Bit 4 – ENDRX End of Receive Buffer Interrupt Disable

Bit 3 – OVRES Overrun Error Interrupt Disable

Bit 2 – TXEMPTY Transmission Registers Empty Disable

Bit 1 – TDRE Transmit Data Register Empty Interrupt Disable

Bit 0 – RDRF Receive Data Register Full Interrupt Disable

© 2026 Microchip Technology Inc. and its subsidiaries

34.7.8. QSPI Interrupt Mask Register

Name: QSPI_IMR
Offset: 0x1C
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							TOUT	
Access							R	
Reset							0	

Bit	15	14	13	12	11	10	9	8
	CSRA	CSFA	QITR	QITF	LWRA	INSTRE		CSR
Access	R	R	R	R	R	R		R
Reset	0	0	0	0	0	0		0

Bit	7	6	5	4	3	2	1	0
	TXBUFE	RXBUFF	ENDTX	ENDRX	OVRES	TXEMPTY	TDRE	RDRF
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 17 – TOUT QSPI Time-out Interrupt Mask

Bit 15 – CSRA Chip Select Rise Autoclear Interrupt Mask

Bit 14 – CSFA Chip Select Fall Autoclear Interrupt Mask

Bit 13 – QITR QSPI Interrupt Rise Interrupt Mask

Bit 12 – QITF QSPI Interrupt Fall Interrupt Mask

Bit 11 – LWRA Last Write Access Interrupt Mask

Bit 10 – INSTRE Instruction End Interrupt Mask

Bit 8 – CSR Chip Select Rise Interrupt Mask

Bit 7 – TXBUFE Transmit Buffer Empty Interrupt Mask

Bit 6 – RXBUFF Receive Buffer Full Interrupt Mask

Bit 5 – ENDTX End of Transmit Buffer Interrupt Mask

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 4 – ENDRX End of Receive Buffer Interrupt Mask

Bit 3 – OVRES Overrun Error Interrupt Mask

Bit 2 – TXEMPTY Transmission Registers Empty Mask

Bit 1 – TDRE Transmit Data Register Empty Interrupt Mask

Bit 0 – RDRF Receive Data Register Full Interrupt Mask

© 2026 Microchip Technology Inc. and its subsidiaries

34.7.9. QSPI Serial Clock Register

Name: QSPI_SCR
Offset: 0x20
Reset: 0x00000000
Property: Read/Write

This register can only be written if bit WPEN is cleared in the QSPI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	DLYBS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							CPHA	CPOL
Access							R/W	R/W
Reset							0	0

Bits 23:16 – DLYBS[7:0] Delay Before QSCK
This field defines the delay from QCS valid to the first valid QSCK transition.
When DLYBS equals zero, the QCS valid to QSCK transition is 1/2 the QSCK clock period.
Otherwise, the following equation determines the delay:
DLYBS = Delay Before QSCK × f_{GCLK}

Bit 1 – CPHA Clock Phase
CPHA determines which edge of QSCK causes data to change and which edge causes data to be captured. CPHA is used with CPOL to produce the required clock/data relationship between host and client devices.

Value	Description
0	Data is captured on the leading edge of QSCK and changed on the following edge of QSCK.
1	Data is changed on the leading edge of QSCK and captured on the following edge of QSCK.

Bit 0 – CPOL Clock Polarity
CPOL is used to determine the inactive state value of the serial clock (QSCK). It is used with CPHA to produce the required clock/data relationship between host and client devices.

Value	Description
0	The inactive state value of QSCK is logic level zero.
1	The inactive state value of QSCK is logic level one.

© 2026 Microchip Technology Inc. and its subsidiaries

34.7.10. QSPI Status Register

Name: QSPI_SR
Offset: 0x24
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				HIDLE	RBUSY	CSS	QSPIENS	SYNCBSY
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit 4 – HIDLE QSPI Idle

Value	Description
0	The QSPI is not in Idle state (either transmitting or Chip Select is active).
1	The QSPI is in Idle state (not transmitting and Chip Select is inactive).

Bit 3 – RBUSY Read Busy

Value	Description
0	The system bus interface has no activity.
1	The system bus interface is currently processing accesses.

Bit 2 – CSS Chip Select Status

Value	Description
0	The chip select is asserted.
1	The chip select is not asserted.

Bit 1 – QSPIENS QSPI Enable Status

Value	Description
0	The QSPI is disabled.
1	The QSPI is enabled.

Bit 0 – SYNCBSY Synchronization Busy

Value	Description
0	No event synchronization between the QSPI Controller interface and QSPI Controller core is ongoing. Register accesses requiring synchronization are allowed, see Register Synchronization.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
1	Event synchronization between the QSPI Controller interface and QSPI Controller core is ongoing. Register accesses requiring synchronization are not allowed, see Register Synchronization.

© 2026 Microchip Technology Inc. and its subsidiaries

34.7.11. QSPI Instruction Address Register

Name: QSPI_IAR
Offset: 0x30
Reset: 0x00000000
Property: Read/Write

This register can only be written if bit WPEN is cleared in the QSPI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	ADDR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ADDR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ADDR[31:0] Address
Address to send to the serial Flash memory in the instruction frame.

© 2026 Microchip Technology Inc. and its subsidiaries

34.7.12. QSPI Write Instruction Code Register

Name: QSPI_WICR
Offset: 0x34
Reset: 0x00000000
Property: Read/Write

This register can only be written if bit WPEN is cleared in the QSPI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	WROPT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	WRINST[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – WROPT[7:0] Write Option Code
Option code to send to the serial Flash memory in case of write transfer.

Bits 7:0 – WRINST[7:0] Write Instruction Code
Instruction code to send to the serial Flash memory in case of write transfer.

© 2026 Microchip Technology Inc. and its subsidiaries

34.7.13. QSPI Instruction Frame Register

Name: QSPI_IFR
Offset: 0x38
Reset: 0x00000000
Property: Read/Write

This register can only be written if bit WPEN is cleared in the QSPI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
			PROTTYP[1:0]			DDRCMDEN	DQSEN	APBTFRTYP
Access			R/W	R/W		R/W	R/W	R/W
Reset			0	0		0	0	0

Bit	23	22	21	20	19	18	17	16
	SMRM			NBDUM[4:0]				
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DDREN	CRM		TFRTYP	ADDRL[1:0]		OPTL[1:0]	
Access	R/W	R/W		R/W	R/W	R/W	R/W	R/W
Reset	0	0		0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DATAEN	OPTEN	ADDREN	INSTEN	WIDTH[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 29:28 – PROTTYP[1:0] Protocol Type
Values not listed below are considered 'reserved'.

Value	Name	Description
0	STD_SPI	Standard (Q)SPI Protocol

Bit 26 – DDRCMDEN DDR Mode Command Enable

Value	Name	Description
0	DISABLED	Transfer of instruction field is performed in Single Data Rate mode even if DDREN is written to '1'.
1	ENABLED	Transfer of instruction field is performed in Double Data Rate mode if DDREN bit is written to '1'. If DDREN is written to '0', the instruction field is sent in Single Data Rate mode.

Bit 25 – DQSEN DQS Sampling Enable

Value	Description
0	Data from the memory are not sampled with DQS signal.
1	Data from the memory are sampled with DQS signal.

Bit 24 – APBTFRTYP Peripheral Bus Transfer Type

Value	Description
0	Peripheral bus register transfer to the memory is a write transfer. Used when TRFTYP is written to '0' and SMRM to '1'.
1	Peripheral bus register transfer to the memory is a read transfer. Used when TRFTYP is written to '0' and SMRM to '1'.

Bit 23 – SMRM Serial Memory Register Mode

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	Serial Memory registers are written via system bus access. See Instruction Frame Transmission for details.
1	Serial Memory registers are written via peripheral bus access. See Instruction Frame Transmission for details.

Bits 20:16 – NBDUM[4:0] Number Of Dummy Cycles
Defines the number of dummy cycles required by the serial Flash memory before data transfer.

Bit 15 – DDREN DDR Mode Enable
DDRCMDEN defines how the instruction field is sent when Double Data Rate mode is enabled. If DDRCMDEN is at '0', the instruction field is sent in Single Data Rate mode.

Value	Name	Description
0	DISABLED	Transfers are performed in Single Data Rate mode.
1	ENABLED	Transfers are performed in Double Data Rate mode, whereas the instruction field is still transferred in Single Data Rate mode.

Bit 14 – CRM Continuous Read Mode

Value	Name	Description
0	DISABLED	Continuous Read mode is disabled.
1	ENABLED	Continuous Read mode is enabled.

Bit 12 – TFRTYP Data Transfer Type

Value	Name	Description
0	TRSFR_REGISTER	Read/Write of memory register, write of memory page buffer. This configuration implies the following: • Either the AHB or the APB interface can be used to initiate the transfer (SMRM bit). • Returned data represents the memory register value. • If the APB interface is used, the RDRF and TDRE flags help to control the frame. • Scrambling is possible only if the APB interface is used. • For HyperFlash memories the "target" bit is set to register space in the HyperFlash header.
1	TRSFR_MEMORY	Read/Write accesses to the memory space. This configuration implies the following: • Only the System Bus interface can be used to trigger accesses. • Access to random location is possible. • Address mask is applied and full size accesses only are performed. • Internal optimization algorithm is enabled to minimize latency. • Data are returned in a system bus protocol compliant way. • Seamless scrambling is enabled. • Seamless handling of HyperFlash Write Buffer programming command (one command for each data) • Seamless protocol-specific page boundary handling • Address shift is handled seamlessly (halfword memories, for example).

Bits 11:10 – ADDRL[1:0] Address Length
The ADDRL bit determines the length of the address.

Value	Name	Description
0	8_BIT	8-bit address size
1	16_BIT	16-bit address size
2	24_BIT	24-bit address size
3	32_BIT	32-bit address size

© 2026 Microchip Technology Inc. and its subsidiaries

Bits 9:8 – OPTL[1:0] Option Code Length

Determines the length of the option code. The value written in OPTL must be consistent with the value written in the field WIDTH. For example, OPTL = 0 (1-bit option code) is not consistent with WIDTH = 6 (option code sent with QuadSPI protocol, thus the minimum length of the option code is 4 bits).

Value	Name	Description
0	OPTION_1BIT	The option code is 1 bit long.
1	OPTION_2BIT	The option code is 2 bits long.
2	OPTION_4BIT	The option code is 4 bits long.
3	OPTION_8BIT	The option code is 8 bits long.

Bit 7 – DATAEN Data Enable

Value	Description
0	No data is sent/received to/from the serial Flash memory.
1	Data is sent/received to/from the serial Flash memory.

Bit 6 – OPTEN Option Enable

Value	Description
0	The option is not sent to the serial Flash memory.
1	The option is sent to the serial Flash memory.

Bit 5 – ADDREN Address Enable

Value	Description
0	The transfer address is not sent to the serial Flash memory.
1	The transfer address is sent to the serial Flash memory.

Bit 4 – INSTEN Instruction Enable

Value	Description
0	The instruction is not sent to the serial Flash memory.
1	The instruction is sent to the serial Flash memory.

Bits 3:0 – WIDTH[3:0] Width of Instruction Code, Address, Option Code and Data

Value	Name	Description
0	SINGLE_BIT_SPI	Instruction: Single-bit SPI / Address-Option: Single-bit SPI / Data: Single-bit SPI
1	DUAL_OUTPUT	Instruction: Single-bit SPI / Address-Option: Single-bit SPI / Data: Dual SPI
2	QUAD_OUTPUT	Instruction: Single-bit SPI / Address-Option: Single-bit SPI / Data: Quad SPI
3	DUAL_IO	Instruction: Single-bit SPI / Address-Option: Dual SPI / Data: Dual SPI
4	QUAD_IO	Instruction: Single-bit SPI / Address-Option: Quad SPI / Data: Quad SPI
5	DUAL_CMD	Instruction: Dual SPI / Address-Option: Dual SPI / Data: Dual SPI
6	QUAD_CMD	Instruction: Quad SPI / Address-Option: Quad SPI / Data: Quad SPI

© 2026 Microchip Technology Inc. and its subsidiaries

34.7.14. QSPI Read Instruction Code Register

Name: QSPI_RICR
Offset: 0x3C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	RDOPT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	RDINST[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – RDOPT[7:0] Read Option Code
Option code to send to the serial Flash memory in case of read transfer.

Bits 7:0 – RDINST[7:0] Read Instruction Code
Instruction code to send to the serial Flash memory in case of read transfer.

© 2026 Microchip Technology Inc. and its subsidiaries

34.7.15. QSPI Scrambling Mode Register

Name: QSPI_SMR
Offset: 0x40
Reset: 0x00000000
Property: Read/Write

This register can only be written if bit WPEN is cleared in the QSPI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						SCRKL	RVDIS	SCREN
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – SCRKL Scrambling Key Lock

Value	Description
0	No action.
1	QSPI_SKR.USRK cannot be written until the next VDDCORE reset.

Bit 1 – RVDIS Scrambling/Unscrambling Random Value Disable

Value	Description
0	The scrambling/unscrambling algorithm includes the user scrambling key plus a random value that may differ between devices.
1	The scrambling/unscrambling algorithm includes only the user scrambling key.

Bit 0 – SCREN Scrambling/Unscrambling Enable

Value	Name	Description
0	DISABLED	The scrambling/unscrambling is disabled.
1	ENABLED	The scrambling/unscrambling is enabled.

© 2026 Microchip Technology Inc. and its subsidiaries

34.7.16. QSPI Scrambling Key Register

Name: QSPI_SKR
Offset: 0x44
Reset: –
Property: Write-only

This register can only be written if bit WPEN is cleared in the QSPI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	USRK[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	USRK[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	USRK[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	USRK[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – USRK[31:0] User Scrambling Key

© 2026 Microchip Technology Inc. and its subsidiaries

34.7.17. QSPI Write Access Counter Register

Name: QSPI_WRACNT
Offset: 0x54
Reset: 0x00000000
Property: Read/Write

This register can only be written if bit WPEN is cleared in the QSPI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	NBWRA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	NBWRA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	NBWRA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	NBWRA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – NBWRA[31:0] Number of Write Accesses
Defines the number of write accesses before the rise of QSPI_ISR.LWRA.

© 2026 Microchip Technology Inc. and its subsidiaries

34.7.18. QSPI Timeout Register

Name: QSPI_TOUT
Offset: 0x64
Reset: 0x00000000
Property: Read/Write

This register can only be written if bit WPEN is cleared in the QSPI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	TCNTM[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TCNTM[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – TCNTM[15:0] Time-out Counter Maximum Value
Indicates the time in GCLK clock periods when the connected client does not answer and before the TOUT flag is set.

© 2026 Microchip Technology Inc. and its subsidiaries

34.7.19. QSPI Write Protection Mode Register

Name: QSPI_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
						WPCREN	WPITEN	WPEN
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x515350	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0.

Bit 2 – WPCREN Write Protection Control Register Enable

Value	Description
0	Disables the write protection on the Control register if WPKEY corresponds to 0x515350.
1	Enables the write protection on the Control register if WPKEY corresponds to 0x515350.

Bit 1 – WPITEN Write Protection Interrupt Enable

Value	Description
0	Disables the write protection on Interrupt registers if WPKEY corresponds to 0x515350.
1	Enables the write protection on Interrupt registers if WPKEY corresponds to 0x515350.

Bit 0 – WPEN Write Protection Enable

See Register Write Protection for the list of registers that can be protected.

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x515350 (“QSP” in ASCII)
1	Enables the write protection if WPKEY corresponds to 0x515350 (“QSP” in ASCII)

© 2026 Microchip Technology Inc. and its subsidiaries

34.7.20. QSPI Write Protection Status Register

Name: QSPI_WPSR
Offset: 0xE8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPVS
Access								R
Reset								0

Bits 15:8 – WPVSRC[7:0] Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protection Violation Status

Value	Description
0	No write protection violation has occurred since the last read of the QSPI_WPSR.
1	A write protection violation has occurred since the last read of the QSPI_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

© 2026 Microchip Technology Inc. and its subsidiaries

35. Segment LCD Controller (SLCDC)

35.1. Description

The Segment Liquid Crystal Display Controller (SLCDC) drives a monochrome passive liquid crystal display (LCD) with up to 8 common terminals and up to 32 segment terminals.

An LCD consists of several segments (pixels or complete symbols) which can be visible or invisible. A segment has two electrodes with liquid crystal between them. When a voltage above a threshold voltage is applied across the liquid crystal, the segment becomes visible.

The voltage must alternate to avoid an electrophoresis effect in the liquid crystal, which degrades the display. Hence the waveform across a segment must not have a DC component.

The SLCDC is programmable to support many different requirements such as:

- Adjusting the driving time of the LCD pads in order to save power and increase the controllability of the DC offset
- Driving smaller LCD (down to 1 common by 1 segment)
- Adjusting the SLCDC frequency in order to obtain the best compromise between frequency and consumption and adapt it to the LCD driver
- Assigning the segments in a user defined pattern to simplify the use of the digital functions multiplexed on these pins

Table 35-1. List of Terms

Term	Description
LCD	A passive display panel with terminals leading directly to a segment
Segment	The least viewing element (pixel) which can be on or off
Common(s)	Denotes how many segments are connected to a segment terminal
Duty	1/(Number of common terminals on a current LCD display)
Bias	1/(Number of voltage levels used driving an LCD display -1)
Frame Rate	Number of times the LCD segments are energized per second

35.2. Embedded Characteristics

The SLCDC provides the following capabilities:

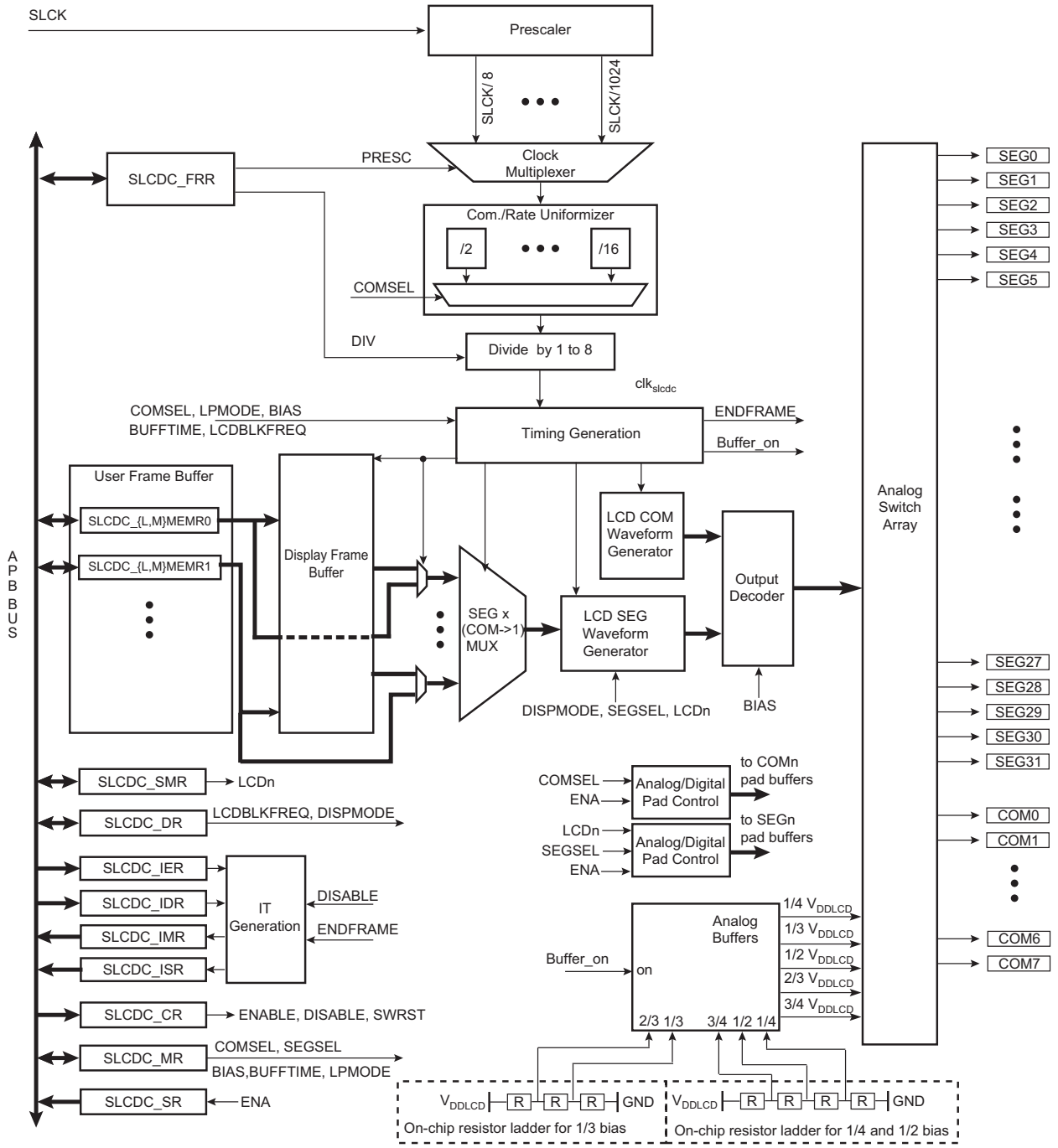
- Display Capacity: Up to 32 Segments and 8 Common Terminals
- Support from Static to 1/8 Duty
- Supports: Static and 1/2, 1/3 and 1/4 Bias
- Two LCD Supply Sources:
  - Internal: 16 Steps, Software Selectable On-chip VDDLCD for Contrast Control
  - External: User-Defined Voltage on VDDLCD Pin for Contrast Control
- Flexible Selection of Frame Frequency
- Two Interrupt Sources: End Of Frame and Disable
- Versatile Display Modes
- Equal Source and Sink Capability to Maximize LCD Lifetime
- Segment and Common Pins Not Needed to Drive the Display Can be Used as Ordinary I/O Pins
- Segments Layout Can be Fully Defined by User to Optimize Usage of Multiplexed Digital Functions
- Latching of Display Data Gives Full Freedom in Register Updates

© 2026 Microchip Technology Inc. and its subsidiaries

- Power-Saving Modes for Extremely Low Power Consumption
- Register Write Protection

35.3. Block Diagram

Figure 35-1. SLCDC Block Diagram



© 2026 Microchip Technology Inc. and its subsidiaries

35.4. I/O Lines Description

Table 35-2. I/O Lines Description

Name	Description	Type
SEG [31:0]	Segments control signals	Output
COM [7:0]	Commons control signals	Output

35.5. Product Dependencies

35.5.1. I/O Lines

The pins used for interfacing the SLCDC may be multiplexed with PIO lines. Refer to the section “Block Diagram”.

If I/O lines of the SLCDC are not used by the application, they can be used for other purposes by the PIO Controller.

By default (SLCDC_SMR0 cleared), the assignment of the segment controls and commons are automatically done depending on COMSEL and SEGSEL in the Mode register (SLCDC_MR). For example, if 10 segments are programmed in the SEGSEL field, they are automatically assigned to SEG[9:0] whereas the remaining SEG pins are automatically selected to be driven by the multiplexed digital functions.

In any case, the user can define a new layout pattern for the segment assignment by programming SLCDC_SMR0 in order to optimize the usage of multiplexed digital function. If at least one bit is set in SLCDC_SMR0, the corresponding I/O line is driven by an LCD segment, whereas any cleared bit of this register selects the corresponding multiplexed digital function.

35.5.2. Power Management

The SLCDC is clocked by the slow clock (SLCK). All the timings are based upon a typical value of 32 kHz for SLCK.

The LCD segment/common pad buffers are supplied by the VDDLCD domain when they are driven by the SLCDC; otherwise they are supplied by VDD3V3. Refer to SLCDC Mode Register and SLCDC Segment Map Register 0.

35.5.3. Interrupt Sources

The SLCDC interrupt line is connected to one of the internal sources of the Interrupt Controller. Using the SLCDC interrupt requires prior programming of the Interrupt Controller.

35.6. Functional Description

The use of the SLCDC comprises three phases of functionality: initialization sequence, display phase, and disable sequence.

- Initialization sequence:

1. Select the LCD supply source in the SUPC:
   - Internal or External
2. Select the clock division (SLCDC_FRR) to use a proper frame rate.
3. Enter the number of common and segments terminals (SLCDC_MR).
4. Select the bias in compliance with the LCD manufacturer datasheet (SLCDC_MR).
5. Enter buffer driving time (SLCDC_MR).
6. Define the segments remapping pattern if required (SLCDC_SMR0).

- During the display phase:

© 2026 Microchip Technology Inc. and its subsidiaries

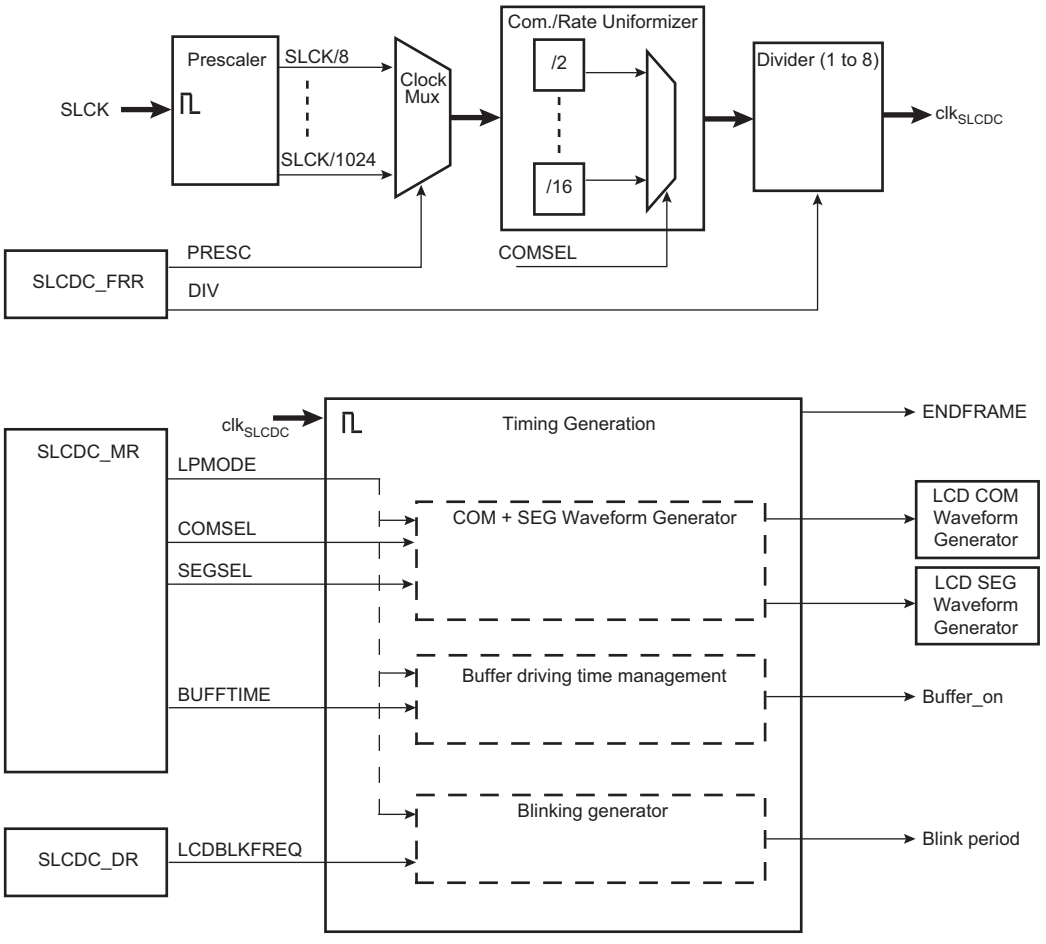
1. Data may be written at any time in the SLCDC memory. The data is automatically latched and displayed at the next LCD frame.
2. It is possible to:
   - Adjust the contrast
     - in the SUPC, only if internal LCD supply is used, or
     - via external VDDLCD pin, otherwise
   - Adjust the frame frequency
   - Adjust buffer driving time
   - Reduce the SLCDC consumption by entering in low-power waveform at any time
   - Use the large set of display features such as blinking, inverted blink, etc.

- Disable sequence: See Disabling the SLCDC.

35.6.1. Clock Generation

35.6.1.1. Block Diagram

Figure 35-2. Clock Generation Block Diagram



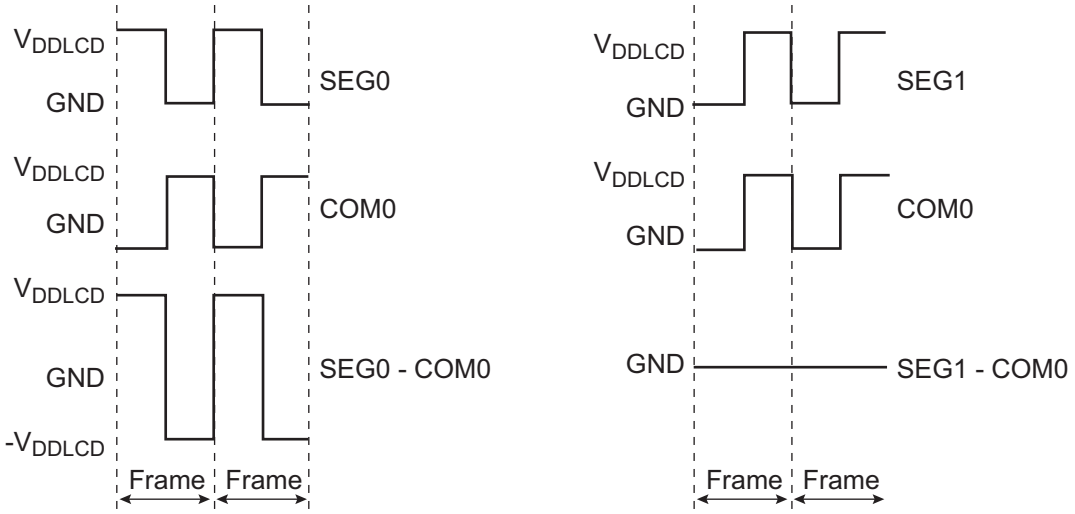
35.6.2. Waveform Generation

35.6.2.1. Static Duty and Bias

This kind of display is driven with the waveform shown in the following figure. SEG0–COM0 is the voltage across a segment that is on, and SEG1–COM0 is the voltage across a segment that is off.

© 2026 Microchip Technology Inc. and its subsidiaries

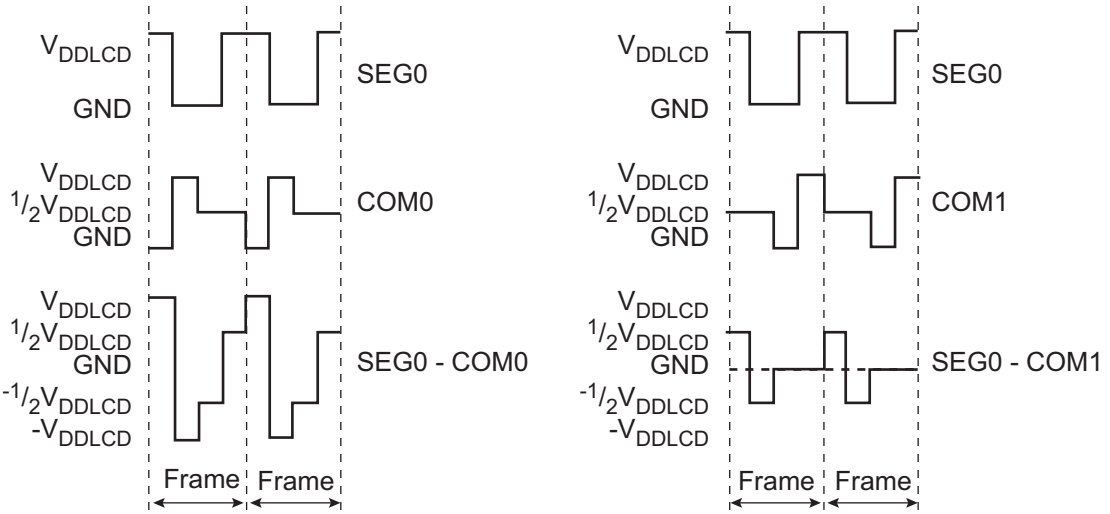
Figure 35-3. Driving an LCD with One Common Terminal



35.6.2.2. 1/2 Duty and 1/2 Bias

For an LCD with two common terminals (1/2 duty), a more complex waveform must be used to control segments individually. Although 1/3 bias can be selected, 1/2 bias is most common for these displays. In the waveform shown in the following figure, SEG0–COM0 is the voltage across a segment that is on, and SEG0–COM1 is the voltage across a segment that is off.

Figure 35-4. Driving an LCD with Two Common Terminals

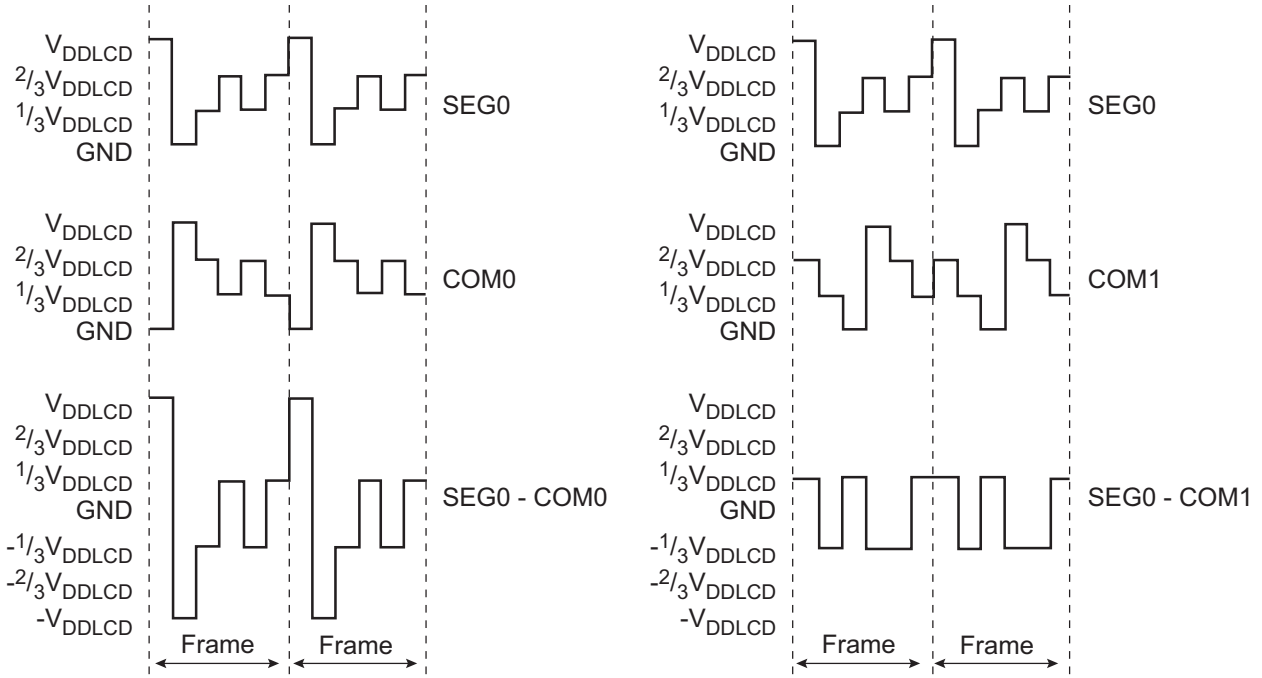


35.6.2.3. 1/3 Duty and 1/3 Bias

1/3 bias is recommended for an LCD with three common terminals (1/3 duty). In the waveform shown in the following figure, SEG0–COM0 is the voltage across a segment that is on and SEG0–COM1 is the voltage across a segment that is off.

© 2026 Microchip Technology Inc. and its subsidiaries

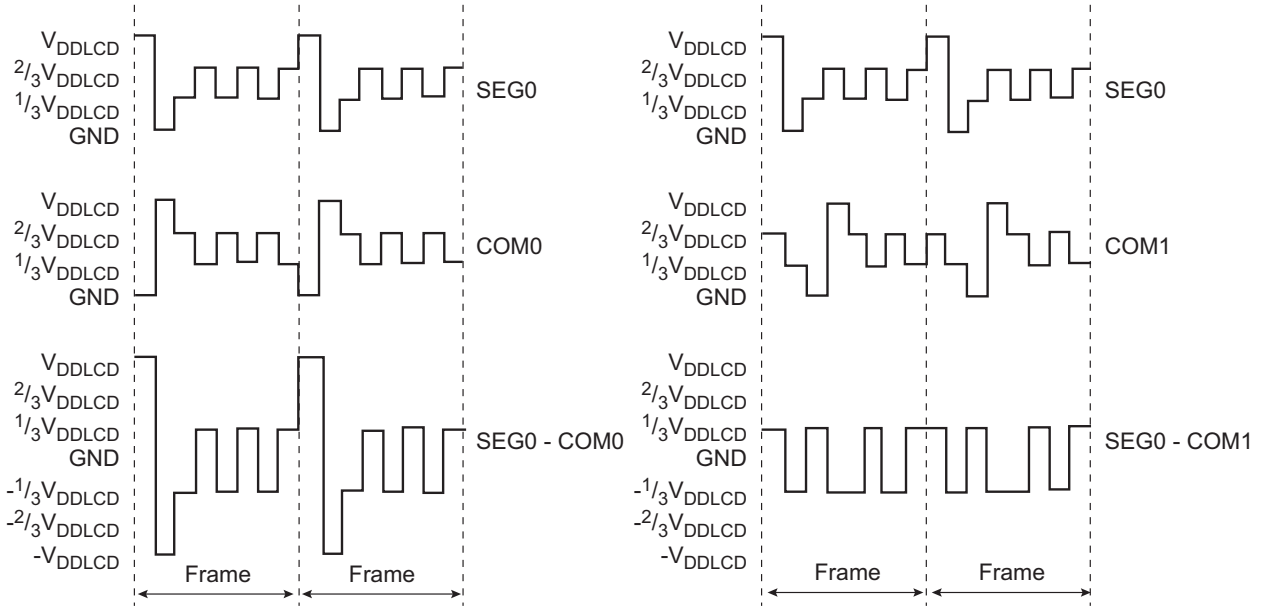
Figure 35-5. Driving an LCD with Three Common Terminals



35.6.2.4. 1/4 Duty and 1/3 Bias

1/3 bias is optimal for LCD displays with four common terminals (1/4 duty). In the waveform shown in the following figure, SEG0–COM0 is the voltage across a segment that is on and SEG0–COM1 is the voltage across a segment that is off.

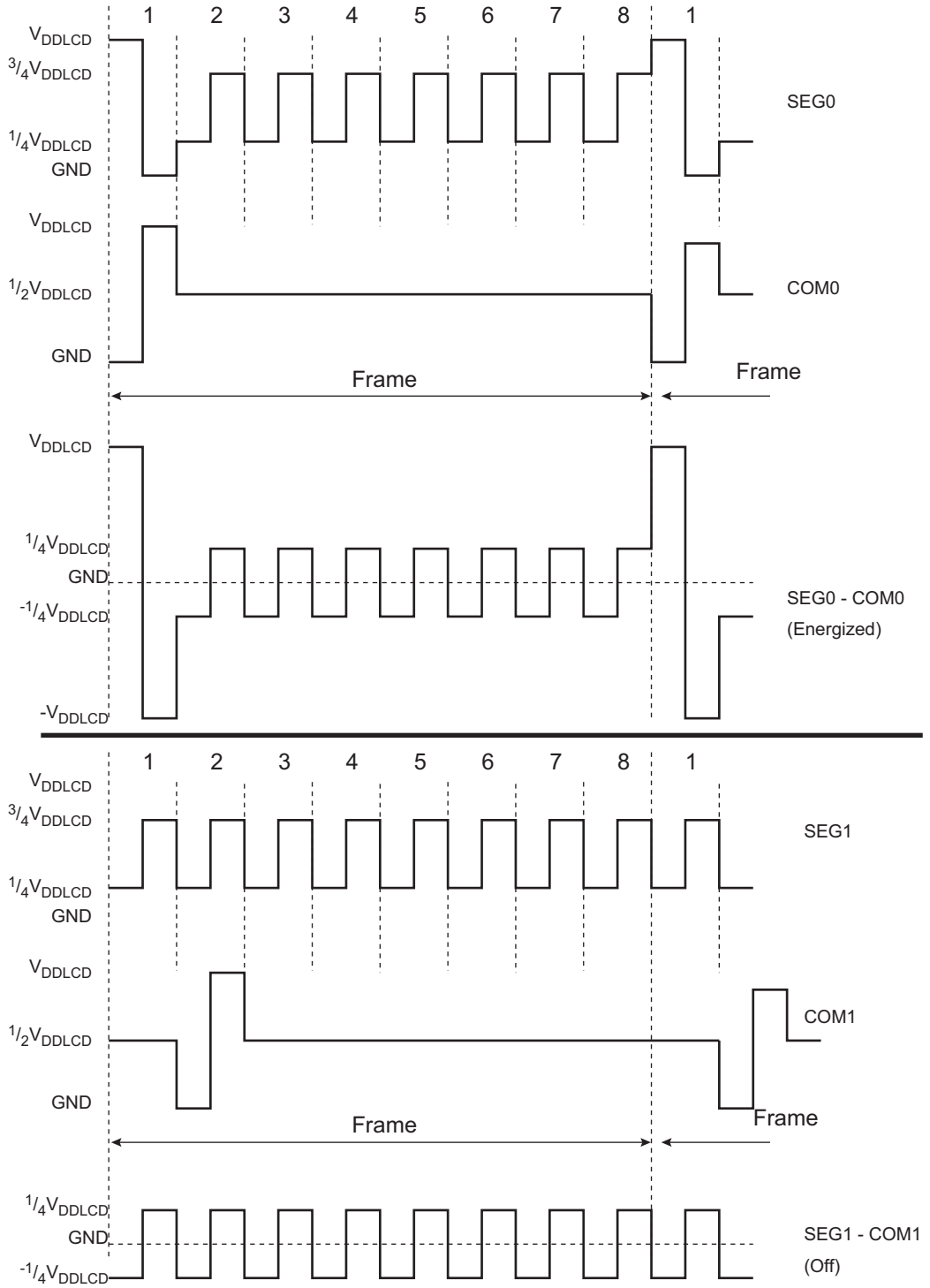
Figure 35-6. Driving an LCD with Four Common Terminals



© 2026 Microchip Technology Inc. and its subsidiaries

35.6.2.5. 1/8 Duty, 1/4 Bias

Figure 35-7. Driving an LCD with Eight Common Terminals

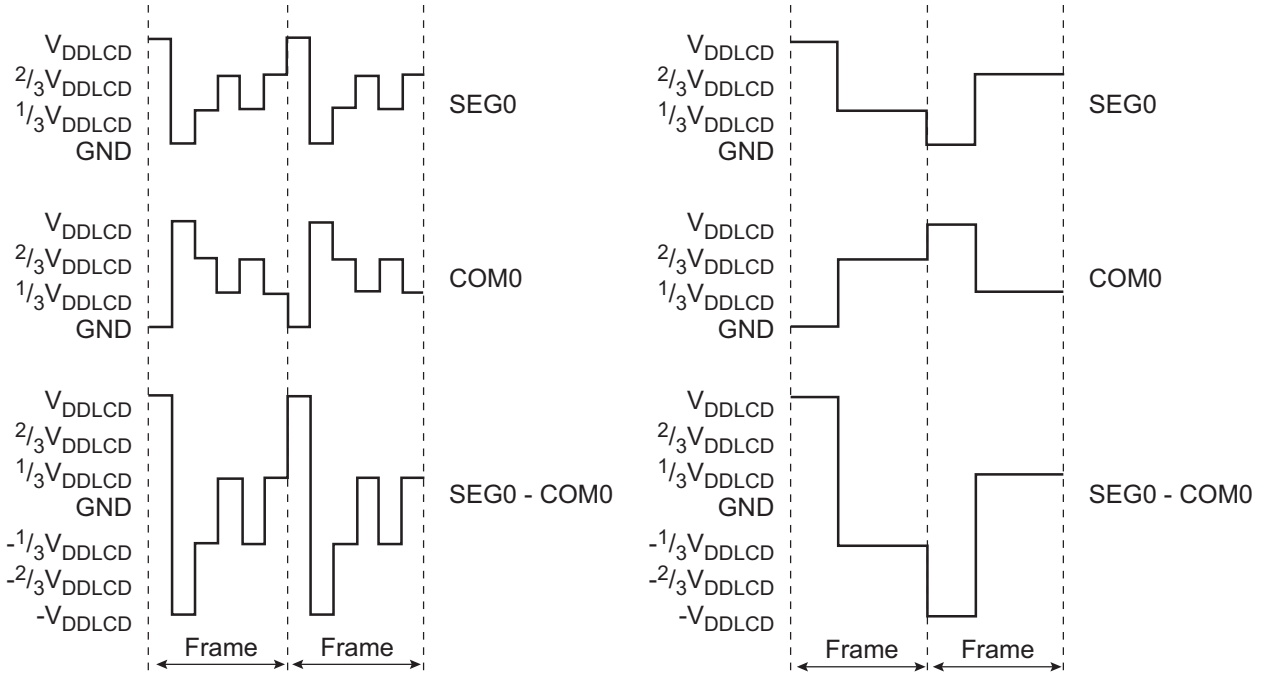


35.6.2.6. Low-Power Waveform

To reduce toggle activity and hence power consumption, a low-power waveform can be selected by writing SLCDC_MR.LPMODE to '1'. The default and low-power waveform is shown in the following figure for 1/3 duty and 1/3 bias. For other selections of duty and bias, the effect is similar.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 35-8. Default and Low-Power Waveform



Note: Refer to the TN-LCD panel electrical specification to verify that low-power waveforms are supported.

35.6.2.7. Frame Rate

The Frame Rate register (SLCDC_FRR) enables the generation of the frequency used by the SLCDC. It is done by a prescaler (division by 8, 16, 32, 64, 128, 256, 512 and 1024) followed by a finer divider (division by 1, 2, 3, 4, 5, 6, 7 or 8).

To calculate the needed frame frequency, the equation below must be used:

$$f_{\text{frame}} = \frac{f_{\text{SLCK}}}{(\text{PRESC} \cdot \text{DIV} \cdot \text{NCOM})}$$

where:

- f_{SLCK} = slow clock frequency
- f_{frame} = frame frequency
- PRESC = prescaler value (8, 16, 32, 64, 128, 256, 512 or 1024)
- DIV = divider value (1, 2, 3, 4, 5, 6, 7, or 8)
- NCOM = depends on the number of commons and is defined in the table below
  Note: NCOM is automatically provided by the SLCDC.

For example, if SLCDC_MR.COMSEL is written to '0' (1 common terminal on display device), the SLCDC introduces a divider by 16 so that NCOM = 16. If COMSEL is written to '3' (3 common terminals on display device), the SLCDC introduces a divider by 5 so that the NCOM remains close to 16 (the frame rate is standardized regardless of the number of driven commons).

Table 35-3. NCOM

Number of Commons	NCOM	Divider
1	16	16
2	16	8
3	15	5

© 2026 Microchip Technology Inc. and its subsidiaries

Table 35-3. NCOM (continued)

Number of Commons	NCOM	Divider
4	16	4
5	15	3
6	18	3
7	14	2
8	16	2

35.6.2.8. Buffer Driving Time

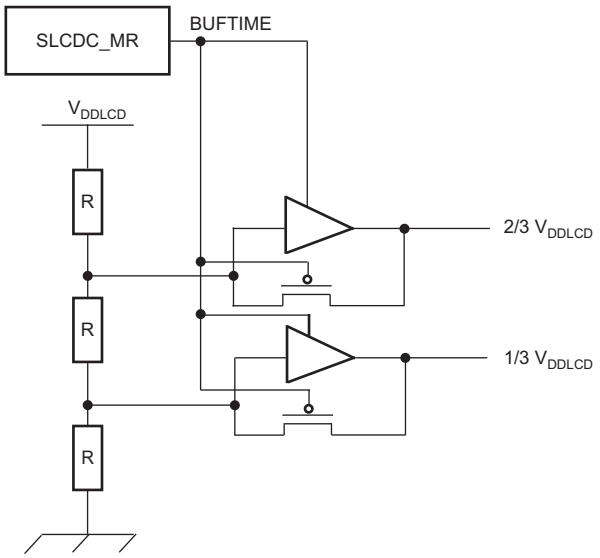
The driving time feature may prevent DC voltage across the panel when there are significant differences in size between pixels and thus different capacitor values associated with each pixel.

When driving time is disabled (SLCDC_MR.BUFTIME=0), the charge and discharge period observed across the common and segments may be significant and may affect the display contrast.

Enabling the driving time feature reduces this period and leads to a reduced impedance of the driver for the configured period. However, this results inan increase in power consumption and thus it is recommended to reduce the period to the minimum acceptable.

In many cases, the driving time can be disabled.

Figure 35-9. Buffer Driving Example for Bias = 1/3



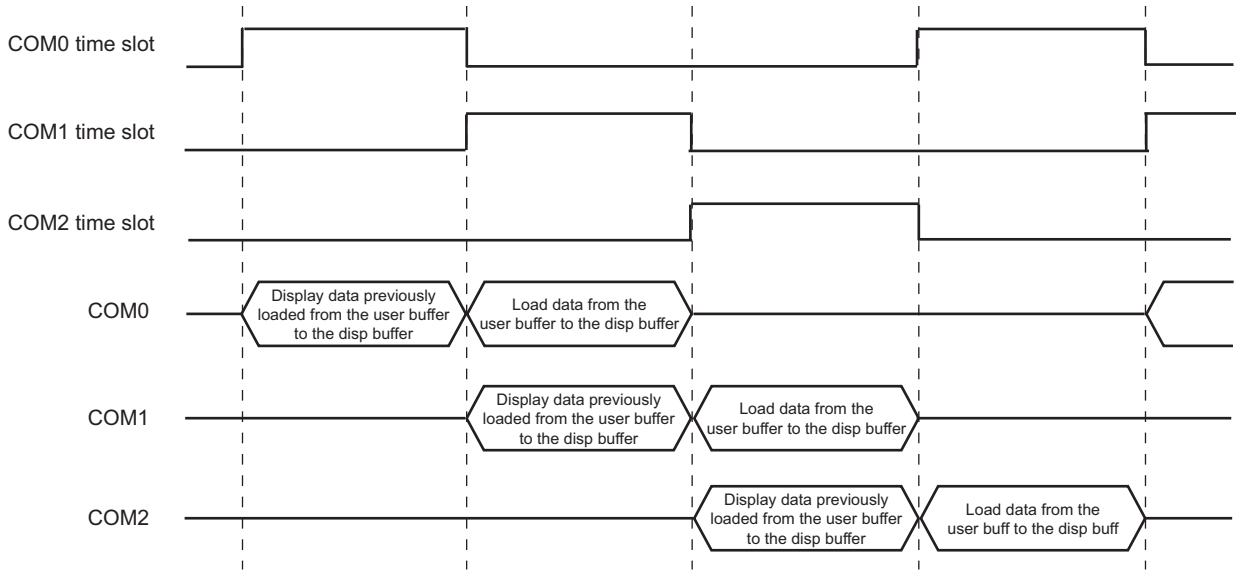
35.6.3. Number of Commons, Segments and Bias

It is important to note that the selection of the number of commons, segments and the bias can be programmed when the SLCDC is disabled.

© 2026 Microchip Technology Inc. and its subsidiaries

35.6.4. SLCDC Memory

Figure 35-10. Memory Management



When a bit in the display memory (SLCDC_LMEMRx) is written to '1', the corresponding segment is energized (on), and non-energized when a bit in the display memory is written to '0'.

At the beginning of each common, the display buffer is updated. The value of the previous common is latched in the display memory (its value is transferred from the user buffer to the frame buffer).

The advantages of this solution are:

- Ability to access the user buffer at any time in the frame, in any display mode and even in low-power waveform
- Ability to change only one pixel without reloading the picture

35.6.5. Display Features

In order to improve the flexibility of SLCDC the following set of display modes are embedded:

- Force Mode off: all pixels are turned off and the memory content is kept.
- Force Mode on: all pixels are turned on and the memory content is kept.
- Inverted mode: all pixels are set in the inverted state as defined in SLCDC memory and the memory content is kept.
- Two blinking modes:
  - Standard Blinking mode: all pixels are alternately turned off to the predefined state in SLCDC memory at LCDBLKFREQ frequency.
  - Inverted Blinking mode: all pixels are alternately turned off to the predefined opposite state in SLCDC memory at LCDBLKFREQ frequency.
- Buffer Swap mode: all pixels are alternatively assigned to the state defined in the user buffer then to the state defined in the display buffer.

35.6.6. Buffer Swap Mode

This mode is used to assign all pixels to two states alternatively without reloading the user buffer at each change.

The means to alternatively display two states is as follows:

© 2026 Microchip Technology Inc. and its subsidiaries

1. Initially, the SLCDC must be in Normal mode or in Standard Blinking mode.
2. Data corresponding to the first pixel state is written in the user buffer (through the SLCDC_MEM registers).
3. Wait two ENDFRAME events (to be sure that the user buffer is entirely transferred in the display buffer).
4. SLCDC_DR must be programmed with DISPMODE = 6 (User Buffer Only Load mode). This mode blocks the automatic transfer from the user buffer to the display buffer.
5. Wait ENDFRAME event. (The display mode is internally updated at the beginning of each frame.)
6. Data corresponding to the second pixel state is written in the user buffer (through the SLCDC_MEM registers). So, now the first pixel state is in the display buffer and the second pixel state is in the user buffer.
7. SLCDC_DR must be programmed with DISPMODE = 7 (Buffer Swap mode) and LCDBLKFREQ must be programmed with the required blinking frequency (if not previously done).

Now, each state is alternatively displayed at LCDBLKFREQ frequency.

Except for the phase dealing with the storage of the two display states, the management of the Buffer Swap mode is the same as the Standard Blinking mode.

35.6.7. Disabling the SLCDC

There are two ways to disable the SLCDC:

- By using SLCDC_CR.LCDDIS (recommended method). In this case, SLCDC configuration and memory content are maintained.
- By using SLCDC_CR.SWRST that acts as a hardware reset for SLCDC only.

Both methods are described in the following sections.

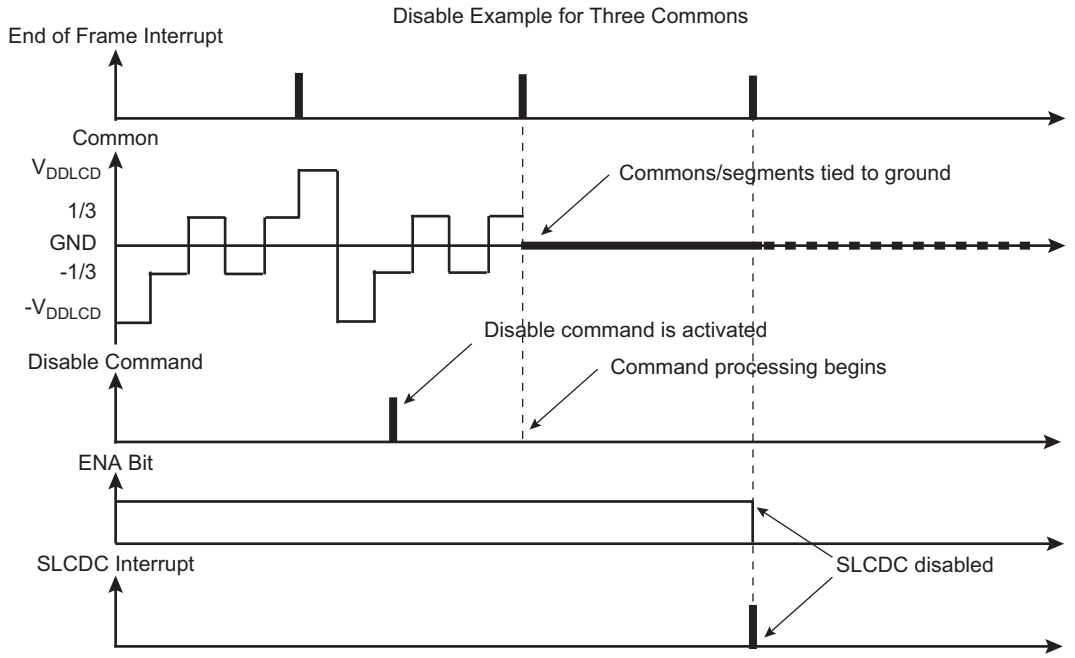
35.6.7.1. Disable Bit

SLCDC_CR.LCDDIS can be set at any time. When the LCD Disable Command is activated during a frame, the SLCDC is not immediately stopped (see the following figure).

The next frame is generated in “All Ground” mode, whereby all commons and segments are tied to ground. At the end of this “All Ground” frame, the disable interrupt is asserted if SLCDC_IMR.DIS is set. The SLCDC is then disabled.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 35-11. Disabling Sequence

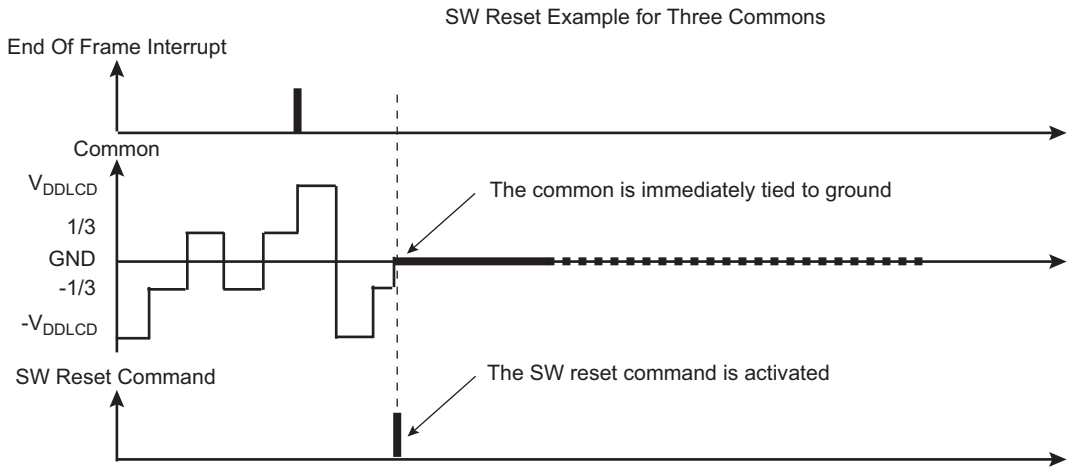


35.6.7.2. Software Reset

When the SLCDC software reset command is activated during a frame, it is immediately processed and all commons and segments are tied to ground.

Note that in the case of a software reset, the disable interrupt is not asserted.

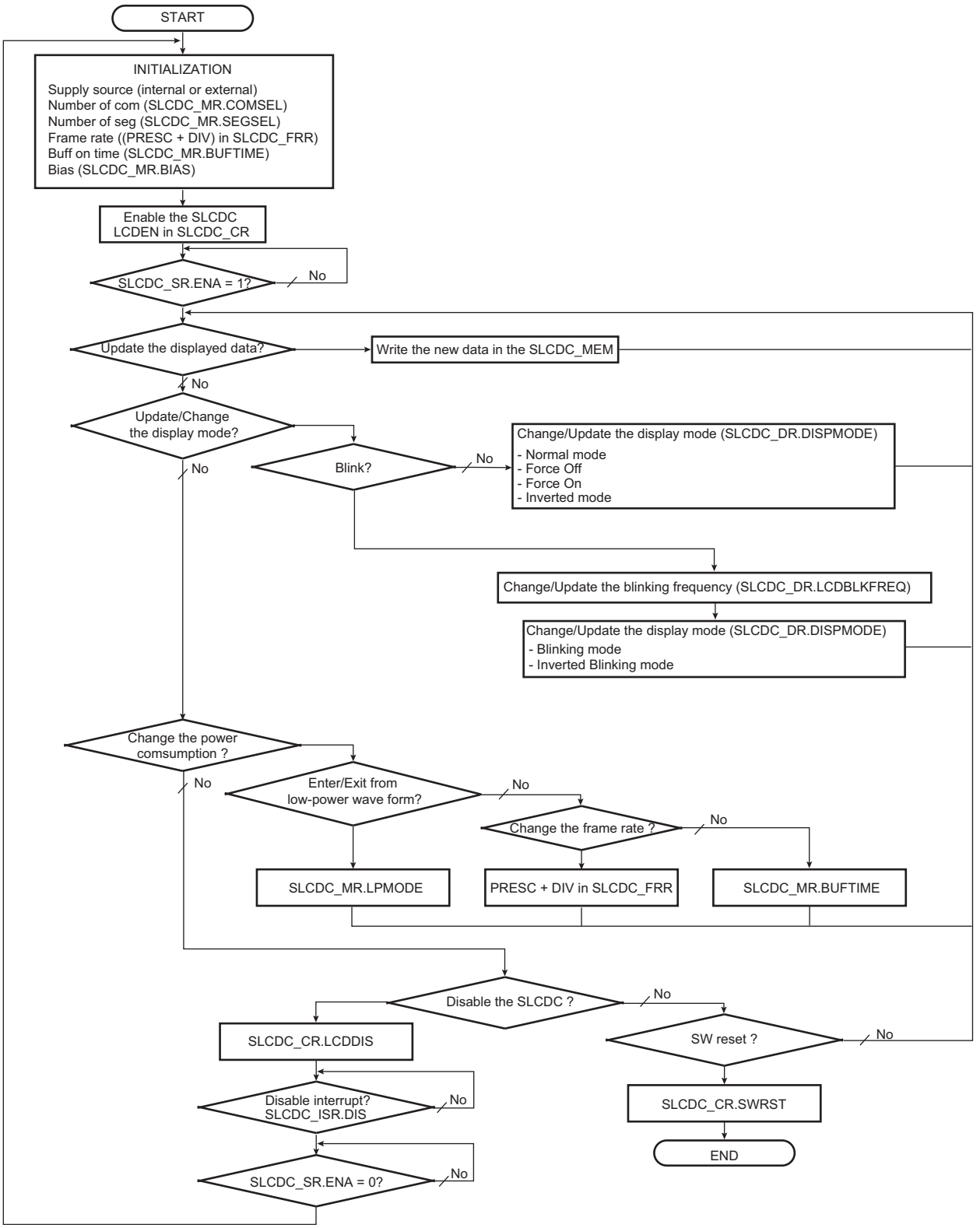
Figure 35-12. Software Reset



© 2026 Microchip Technology Inc. and its subsidiaries

35.6.8. Flowchart

Figure 35-13. SLCDC Flow Chart



© 2026 Microchip Technology Inc. and its subsidiaries

35.6.9. User Buffer Organization

The pixels to be displayed are written into SLCDC_LMEMRx. There are up to two 32-bit registers for each common terminal. The following table provides the address mapping of all commons/ segments to be displayed.

When segment remap is used (SLCDC_SMR0 differs from 0), the unmapped segments must be kept cleared to limit internal signal switching.

Table 35-4. Commons Segments Address Mapping

Register	Common Terminal	SEG31 .. SEG0	Memory Address
SLCDC_LMEMR7	COM7	SLCDC_LMEMR7[31:0]	0x238
SLCDC_LMEMR6	COM6	SLCDC_LMEMR6[31:0]	0x230
SLCDC_LMEMR5	COM5	SLCDC_LMEMR5[31:0]	0x228
SLCDC_LMEMR4	COM4	SLCDC_LMEMR4[31:0]	0x220
SLCDC_LMEMR3	COM3	SLCDC_LMEMR3[31:0]	0x218
SLCDC_LMEMR2	COM2	SLCDC_LMEMR2[31:0]	0x210
SLCDC_LMEMR1	COM1	SLCDC_LMEMR1[31:0]	0x208
SLCDC_LMEMR0	COM0	SLCDC_LMEMR0[31:0]	0x200

35.6.10. Segments Mapping Function

By default the segments pins (SEG0:31) are automatically assigned according to the SEGSEL configuration in the SLCDC_MR. The unused SEG I/O pins are forced to be driven by a digital peripheral or can be used as I/O through the PIO controller.

The automatic assignment is performed if the segment mapping function is not used (SLCDC_SMR0 is cleared). The following table provides such assignments.

Table 35-5. Segment Pin Assignments

SEGSEL	I/O Port in Use as Segment Driver	I/O Port Pin if SLCDC_SMR0 = 0
0	SEG0	SEG1:31
1	SEG0:1	SEG2:31
...	...	...
30	SEG0:30	SEG31
31	SEG0:30	None

Programming is straightforward in this mode but it prevents flexibility of use of the digital peripheral multiplexed on SEG0:31 especially when the number of segments to drive is close to the maximum (32).

For example, if SEGSEL is set to 30, only the digital peripheral associated to SEG31 can be used and none of the other digital peripherals multiplexed on SEG0:30 I/O can be used.

To offer a flexible selection of digital peripherals multiplexed on SEG0:31 the user can manually configure the SEG I/O pins to be driven by the SLCDC.

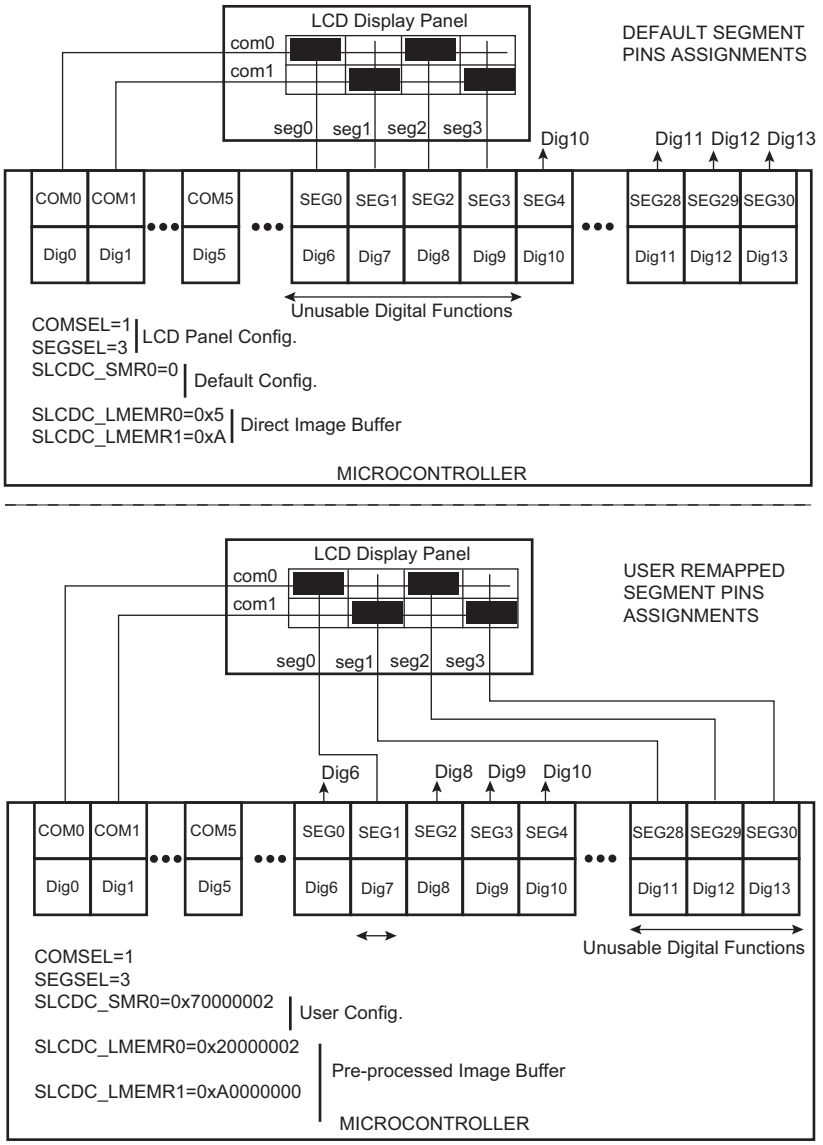
This is done by programming SLCDC_SMR0. As soon as their values differ from 0 the Segment Remapping mode is used.

When configuring a logic 1 at index n (n = 0..31) in SLCDC_SMR0, the SLCDC forces the SEGn I/O pin to be driven by a segment waveform. In this mode, the value of SEGSEL in SLCDC_MR is ignored.

In Remapping mode, the software dispatches the pixels into SLCDC_LMEMRx according to what is programmed in SLCDC_SMR0.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 35-14. Segments Remapping Example



35.6.11. Register Write Protection

To prevent any single software error from corrupting SLCDC behavior, certain registers in the address space can be write-protected by setting the WPEN bit in the SLCDC Write Protection Mode Register (SLCDC_WPMR).

The following registers can be write-protected when WPEN is written to 1:

- SLCDC Mode Register
- SLCDC Frame Rate Register
- SLCDC Display Register
- SLCDC Segment Map Register 0

The following registers can be write-protected when WPITEN is written to 1:

- SLCDC Interrupt Enable Register
- SLCDC Interrupt Disable Register

© 2026 Microchip Technology Inc. and its subsidiaries

The following registers can be write-protected when WPCREN is written to 1:

- SLCDC Control Register

35.7. Waveform Specifications

35.7.1. DC Characteristics

Refer to the section “DC Characteristics”.

35.7.2. LCD Contrast

The peak value (VDDLCD) on the output waveform determines the LCD contrast. VDDLCD is controlled by software if the internal LCD power supply is used, or via the voltage applied on the VDDLCD pin if an external LCD power supply is used.

The contrast is configured by the LCD voltage regulator (refer to the section “Supply Controller (SUPC)”).

© 2026 Microchip Technology Inc. and its subsidiaries

35.8. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	SLCDC_CR	31:24								
		23:16								
		15:8	FRZKEY[7:0]							
		7:0	FRZMAP	FRZDF			SWRST		LCDDIS	LCDEN
0x04	SLCDC_MR	31:24								LPMODE
		23:16			BIAS[1:0]		BUFTIME[3:0]			
		15:8			SEGSEL[5:0]					
		7:0						COMSEL[2:0]		
0x08	SLCDC_FRR	31:24								
		23:16								
		15:8						DIV[2:0]		
		7:0						PRESC[2:0]		
0x0C	SLCDC_DR	31:24								
		23:16								
		15:8	LCDBLKFREQ[7:0]							
		7:0						DISPMODE[2:0]		
0x10	SLCDC_SR	31:24								
		23:16								
		15:8								
		7:0	MAPFRZS	DFFRZS						ENA
0x14 ... 0x1F	Reserved									
0x20	SLCDC_IER	31:24								
		23:16								
		15:8								
		7:0						DIS		ENDFRAME
0x24	SLCDC_IDR	31:24								
		23:16								
		15:8								
		7:0						DIS		ENDFRAME
0x28	SLCDC_IMR	31:24								
		23:16								
		15:8								
		7:0						DIS		ENDFRAME
0x2C	SLCDC_ISR	31:24								
		23:16								
		15:8								
		7:0						DIS		ENDFRAME
0x30	SLCDC_SMR0	31:24	LCD31	LCD30	LCD29	LCD28	LCD27	LCD26	LCD25	LCD24
		23:16	LCD23	LCD22	LCD21	LCD20	LCD19	LCD18	LCD17	LCD16
		15:8	LCD15	LCD14	LCD13	LCD12	LCD11	LCD10	LCD9	LCD8
		7:0	LCD7	LCD6	LCD5	LCD4	LCD3	LCD2	LCD1	LCD0
0x34 ... 0xE3	Reserved									
0xE4	SLCDC_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0						WPCREN	WPITEN	WPEN
0xE8 ... 0x01FF	Reserved									
0x0200	SLCDC_LMEMR0	31:24	LPIXEL[31:24]							
		23:16	LPIXEL[23:16]							
		15:8	LPIXEL[15:8]							
		7:0	LPIXEL[7:0]							

© 2026 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0204 ... 0x0207	Reserved									
0x0208	SLCDC_LMEMR1	31:24	LPIXEL[31:24]							
		23:16	LPIXEL[23:16]							
		15:8	LPIXEL[15:8]							
		7:0	LPIXEL[7:0]							
0x020C ... 0x020F	Reserved									
0x0210	SLCDC_LMEMR2	31:24	LPIXEL[31:24]							
		23:16	LPIXEL[23:16]							
		15:8	LPIXEL[15:8]							
		7:0	LPIXEL[7:0]							
0x0214 ... 0x0217	Reserved									
0x0218	SLCDC_LMEMR3	31:24	LPIXEL[31:24]							
		23:16	LPIXEL[23:16]							
		15:8	LPIXEL[15:8]							
		7:0	LPIXEL[7:0]							
0x021C ... 0x021F	Reserved									
0x0220	SLCDC_LMEMR4	31:24	LPIXEL[31:24]							
		23:16	LPIXEL[23:16]							
		15:8	LPIXEL[15:8]							
		7:0	LPIXEL[7:0]							
0x0224 ... 0x0227	Reserved									
0x0228	SLCDC_LMEMR5	31:24	LPIXEL[31:24]							
		23:16	LPIXEL[23:16]							
		15:8	LPIXEL[15:8]							
		7:0	LPIXEL[7:0]							
0x022C ... 0x022F	Reserved									
0x0230	SLCDC_LMEMR6	31:24	LPIXEL[31:24]							
		23:16	LPIXEL[23:16]							
		15:8	LPIXEL[15:8]							
		7:0	LPIXEL[7:0]							
0x0234 ... 0x0237	Reserved									
0x0238	SLCDC_LMEMR7	31:24	LPIXEL[31:24]							
		23:16	LPIXEL[23:16]							
		15:8	LPIXEL[15:8]							
		7:0	LPIXEL[7:0]							

© 2026 Microchip Technology Inc. and its subsidiaries

35.8.1. SLCDC Control Register

Name: SLCDC_CR
Offset: 0x0
Reset: –
Property: Write-only

This register can only be written if the WPCREN bit is cleared in the SLCDC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	FRZKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	FRZMAP	FRZDF			SWRST		LCDDIS	LCDEN
Access	W	W			W		W	W
Reset	–	–			–		–	–

Bits 15:8 – FRZKEY[7:0] Freeze Key

Value	Name	Description
0x4E	PASSWD	Writing any other value in this field aborts the write operation of the FRZDF bit or FRZMAP bit. Always reads as 0.

Bit 7 – FRZMAP Freeze Remap Configuration

Value	Description
0	No effect.
1	Freezes the configuration of remap registers until a VDDLCD power-up is performed (the SLCDC_CR.SWRST has no effect) if FRZKEY corresponds to 0x4E).

Bit 6 – FRZDF Freeze Display Features Configuration

Value	Description
0	No effect.
1	Freezes the configuration of SLCDC_MR and SLCDC_FRR until a VDDLCD power-up is performed (the SLCDC_CR.SWRST has no effect) if FRZKEY corresponds to 0x4E. For example, the configuration of a number of commons/segments is protected against any software error. The FRZKEY must be written to 0x4E at the same time to enable the freeze of remap registers.

Bit 3 – SWRST Software Reset

Value	Description
0	No effect.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
1	Equivalent to a power-up reset. When this command is performed, the SLCDC immediately ties all segments end commons lines to values corresponding to a “ground voltage”.

Bit 1 – LCDDIS Disable LCDC

LCDDIS is processed at the beginning of the next frame.

Value	Description
0	No effect.
1	The SLCDC is disabled.

Bit 0 – LCDEN Enable the LCDC

Value	Description
0	No effect.
1	The SLCDC is enabled.

© 2026 Microchip Technology Inc. and its subsidiaries

35.8.2. SLCDC Mode Register

Name: SLCDC_MR
Offset: 0x4
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the Write Protection Mode register and the bit DFFRZS is cleared in the Status register.

Bit	31	30	29	28	27	26	25	24
								LPMODE
Access								R/W
Reset								0

Bit	23	22	21	20	19	18	17	16
			BIAS[1:0]		BUFTIME[3:0]			
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
			SEGSEL[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
						COMSEL[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 24 – LPMODE Low-Power Mode
Processed at beginning of next frame.

Value	Description
0	Normal mode.
1	Low-power waveform is enabled.

Bits 21:20 – BIAS[1:0] LCD Display Configuration
For safety reasons, can be configured when SLCDC is disabled.

Value	Name	Description
0	STATIC	Static
1	BIAS_1_2	Bias 1/2
2	BIAS_1_3	Bias 1/3
3	BIAS_1_4	Bias 1/4

Bits 19:16 – BUFTIME[3:0] Buffer On-Time
(Processed at beginning of next frame)

Value	Name	Description
0	OFF	Nominal drive time is 0% of SLCK period
1	X2_SLCK_PERIOD	Nominal drive time is 2 periods of SLCK clock
2	X4_SLCK_PERIOD	Nominal drive time is 4 periods of SLCK clock
3	X8_SLCK_PERIOD	Nominal drive time is 8 periods of SLCK clock
4	X16_SLCK_PERIOD	Nominal drive time is 16 periods of SLCK clock
5	X32_SLCK_PERIOD	Nominal drive time is 32 periods of SLCK clock

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Name	Description
6	X64_SLCK_PERIOD	Nominal drive time is 64 periods of SLCK clock
7	X128_SLCK_PERIOD	Nominal drive time is 128 periods of SLCK clock
8	PERCENT_50	Nominal drive time is 50% of SLCK period
9	PERCENT_100	Nominal drive time is 100% of SLCK period

Bits 13:8 – SEGSEL[5:0] Selection of the Number of Segments
For safety reasons, can be configured when SLCDC is disabled.
SEGSEL must be programmed with the number of segments of the display panel minus 1.
If segment remapping function is not used (i.e., SLCDC_SMR0 equal 0) the SEGn [n = 0..31] I/O pins where n is greater than SEGSEL are forced to be driven by digital function. When segments remapping function is used, SEGn pins are driven by SLCDC only if corresponding PIXELn configuration bit is set in SLCDC_SMR0.

Bits 2:0 – COMSEL[2:0] Selection of the Number of Commons
For safety reasons, can be configured when SLCDC is disabled.

Value	Name	Description
0	COM_0	COM0 is driven by SLCDC, COM1:7 are driven by digital function
1	COM_0TO1	COM0:1 are driven by SLCDC, COM2:7 are driven by digital function
2	COM_0TO2	COM0:2 are driven by SLCDC, COM3:7 are driven by digital function
3	COM_0TO3	COM0:3 are driven by SLCDC, COM4:7 are driven by digital function
4	COM_0TO4	COM0:4 are driven by SLCDC, COM5:7 are driven by digital function
5	COM_0TO5	COM0:5 are driven by SLCDC, COM6:7 are driven by digital function
6	COM_0TO6	COM0:6 are driven by SLCDC, COM7:7 are driven by digital function
7	COM_0TO7	COM0:7 are driven by SLCDC, COM8:7 are driven by digital function

© 2026 Microchip Technology Inc. and its subsidiaries

35.8.3. SLCDC Frame Rate Register

Name: SLCDC_FRR
Offset: 0x8
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the Write Protection Mode register and the bit DFFRZS is cleared in the Status register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						DIV[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
						PRESC[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 10:8 – DIV[2:0] Clock Division
Processed at beginning of next frame.

Value	Name	Description
0	PRESC_CLK_DIV1	Clock output from prescaler is divided by 1
1	PRESC_CLK_DIV2	Clock output from prescaler is divided by 2
2	PRESC_CLK_DIV3	Clock output from prescaler is divided by 3
3	PRESC_CLK_DIV4	Clock output from prescaler is divided by 4
4	PRESC_CLK_DIV5	Clock output from prescaler is divided by 5
5	PRESC_CLK_DIV6	Clock output from prescaler is divided by 6
6	PRESC_CLK_DIV7	Clock output from prescaler is divided by 7
7	PRESC_CLK_DIV8	Clock output from prescaler is divided by 8

Bits 2:0 – PRESC[2:0] Clock Prescaler
Processed at beginning of next frame.

Value	Name	Description
0	SLCK_DIV8	Slow clock is divided by 8
1	SLCK_DIV16	Slow clock is divided by 16
2	SLCK_DIV32	Slow clock is divided by 32
3	SLCK_DIV64	Slow clock is divided by 64
4	SLCK_DIV128	Slow clock is divided by 128
5	SLCK_DIV256	Slow clock is divided by 256
6	SLCK_DIV512	Slow clock is divided by 512
7	SLCK_DIV1024	Slow clock is divided by 1024

© 2026 Microchip Technology Inc. and its subsidiaries

35.8.4. SLCDC Display Register

Name: SLCDC_DR
Offset: 0xC
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the SLCDC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	LCDBLKFREQ[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
						DISPMODE[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 15:8 – LCDBLKFREQ[7:0] LCD Blinking Frequency Selection
Processed at beginning of next frame.
Blinking frequency = Frame Frequency/LCDBLKFREQ[7:0].
0 written in LCDBLKFREQ stops blinking.

Bits 2:0 – DISPMODE[2:0] Display Mode Register
Processed at beginning of next frame.

Value	Name	Description
0	NORMAL	Normal Mode—Latched data are displayed.
1	FORCE_OFF	Force Off Mode—All pixels are invisible. (The SLCDC memory is unchanged.)
2	FORCE_ON	Force On Mode—All pixels are visible. (The SLCDC memory is unchanged.)
3	BLINKING	Blinking Mode—All pixels are alternately turned off to the predefined state in SLCDC memory at LCDBLKFREQ frequency. (The SLCDC memory is unchanged.)
4	INVERTED	Inverted Mode—All pixels are set in the inverted state as defined in SLCDC memory. (The SLCDC memory is unchanged.)
5	INVERTED_BLINK	Inverted Blinking Mode—All pixels are alternately turned off to the predefined opposite state in SLCDC memory at LCDBLKFREQ frequency. (The SLCDC memory is unchanged.)
6	USER_BUFFER_LOAD	User Buffer Only Load Mode—Blocks the automatic transfer from User Buffer to Display Buffer.
7	BUFFERS_SWAP	Buffer Swap Mode—All pixels are alternatively assigned to the state defined in the User Buffer, then to the state defined in the Display Buffer at LCDBLKFREQ frequency.

© 2026 Microchip Technology Inc. and its subsidiaries

35.8.5. SLCDC Status Register

Name: SLCDC_SR
Offset: 0x10
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	MAPFRZS	DFFRZS						ENA
Access	R	R						R
Reset	0	0						0

Bit 7 – MAPFRZS Remapping Configuration Freeze Status

Value	Description
0	SLCDC_SMR can be written if SLCDC_WPMR.WPEN=0.
1	SLCDC_SMR is locked until next VDDLCD power-up.

Bit 6 – DFFRZS Display Panel Features Configuration Freeze Status

Value	Description
0	SLCDC_MR and SLCDC_FRR can be written if SLCDC_WPMR.WPEN=0.
1	SLCDC_MR and SLCDC_FRR are locked until next VDDLCD power-up.

Bit 0 – ENA Enable Status (Automatically Set/Reset)

Value	Description
0	The SLCDC is disabled.
1	The SLCDC is enabled.

© 2026 Microchip Technology Inc. and its subsidiaries

35.8.6. SLCDC Interrupt Enable Register

Name: SLCDC_IER
Offset: 0x20
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the SLCDC Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						DIS		ENDFRAME
Access						W		W
Reset						–		–

Bit 2 – DIS Disable Completion Interrupt Enable

Bit 0 – ENDFRAME End of Frame Interrupt Enable

© 2026 Microchip Technology Inc. and its subsidiaries

35.8.7. SLCDC Interrupt Disable Register

Name: SLCDC_IDR
Offset: 0x24
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the SLCDC Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						DIS		ENDFRAME
Access						W		W
Reset						–		–

Bit 2 – DIS Disable Completion Interrupt Disable

Bit 0 – ENDFRAME End of Frame Interrupt Disable

© 2026 Microchip Technology Inc. and its subsidiaries

35.8.8. SLCDC Interrupt Mask Register

Name: SLCDC_IMR
Offset: 0x28
Reset: –
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						DIS		ENDFRAME
Access						R		R
Reset						–		–

Bit 2 – DIS Disable Completion Interrupt Mask

Bit 0 – ENDFRAME End of Frame Interrupt Mask

© 2026 Microchip Technology Inc. and its subsidiaries

35.8.9. SLCDC Interrupt Status Register

Name: SLCDC_ISR
Offset: 0x2C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						DIS		ENDFRAME
Access						R		R
Reset						0		0

Bit 2 – DIS Disable Completion Interrupt Status

Value	Description
0	The SLCDC is enabled.
1	The SLCDC is disabled.

Bit 0 – ENDFRAME End of Frame Interrupt Status

Value	Description
0	No End of Frame occurred since the last read.
1	End of Frame occurred since the last read.

© 2026 Microchip Technology Inc. and its subsidiaries

35.8.10. SLCDC Segment Map Register 0

Name: SLCDC_SMR0
Offset: 0x30
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the Write Protection Mode register and the bit FRZS is cleared in the Status register.

Bit	31	30	29	28	27	26	25	24
	LCD31	LCD30	LCD29	LCD28	LCD27	LCD26	LCD25	LCD24
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	LCD23	LCD22	LCD21	LCD20	LCD19	LCD18	LCD17	LCD16
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	LCD15	LCD14	LCD13	LCD12	LCD11	LCD10	LCD9	LCD8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	LCD7	LCD6	LCD5	LCD4	LCD3	LCD2	LCD1	LCD0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – LCDx LCD Segment Mapped on SEGx I/O Pin
For safety reasons, can be configured when SLCDC is disabled.

Value	Description
0	The corresponding I/O pin is driven either by SLCDC or digital function, depending on the SEGSEL field configuration in SLCDC_MR.
1	An LCD segment is driven on the corresponding I/O pin.

© 2026 Microchip Technology Inc. and its subsidiaries

35.8.11. SLCDC Write Protection Mode Register

Name: SLCDC_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

See Register Write Protection for the list of registers which can be protected.

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
						WPCREN	WPITEN	WPEN
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x534C43	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0.

Bit 2 – WPCREN Write Protection Control Enable

Value	Description
0	Disables the write protection on the control register if WPKEY corresponds to 0x534C43 (“SLC” in ASCII).
1	Enables the write protection on the control register if WPKEY corresponds to 0x534C43 (“SLC” in ASCII).

Bit 1 – WPITEN Write Protection Interrupt Enable

Value	Description
0	Disables the write protection on interrupt registers if WPKEY corresponds to 0x534C43 (“SLC” in ASCII).
1	Enables the write protection on interrupt registers if WPKEY corresponds to 0x534C43 (“SLC” in ASCII).

Bit 0 – WPEN Write Protection Enable

Value	Description
0	Disables write protection if WPKEY corresponds to 0x534C43 (“SLC” in ASCII).
1	Enables write protection if WPKEY corresponds to 0x534C43 (“SLC” in ASCII).

© 2026 Microchip Technology Inc. and its subsidiaries

35.8.12. SLCDC LSB Memory Register x

Name: SLCDC_LMEMRx
Offset: 0x0200 + x*0x08 [x=0..7]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	LPIXEL[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	LPIXEL[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	LPIXEL[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	LPIXEL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – LPIXEL[31:0] LSB Pixels Pattern Associated to COMx Terminal
LPIXEL[n] (n = 0..31) drives SEGn terminal.

Value	Description
0	The pixel associated to COMx terminal is not visible (if Non-inverted Display mode is used).
1	The pixel associated to COMx terminal is visible (if Non-inverted Display mode is used).

© 2026 Microchip Technology Inc. and its subsidiaries

36. Timer Counter (TC)

36.1. Description

A Timer Counter (TC) module includes three identical TC channels. The number of implemented TC modules is device-specific.

Each TC channel can be independently programmed to perform a wide range of functions including frequency measurement, event counting, interval measurement, pulse generation, delay timing and pulse width modulation.

Each channel has three external clock inputs, five internal clock inputs and two multipurpose input/output signals which can be configured by the user. Each channel drives an internal interrupt signal which can be programmed to generate processor interrupts.

The TC embeds a quadrature decoder (QDEC) connected in front of the timers and driven by TIOA0, TIOB0 and TIOB1 inputs. When enabled, the QDEC performs the input lines filtering, decoding of quadrature signals and connects to the timers/counters in order to read the position and speed of the motor through the user interface.

The TC block has two global registers which act upon all TC channels:

- Block Control register (TC_BCR)—allows channels to be started simultaneously with the same instruction
- Block Mode register (TC_BMR)—defines the external clock inputs for each channel, allowing them to be chained

36.2. Embedded Characteristics

- Total of Twelve Channels
- 32-bit Channel Size
- Wide Range of Functions Including:
  - Frequency measurement
  - Event counting
  - Interval measurement
  - Pulse generation
  - Delay timing
  - Pulse Width Modulation
  - Up/down capabilities
  - Quadrature decoder with real time filtering reports
  - 2-bit Gray up/down count for stepper motor
- Each Channel is User-Configurable and Contains:
  - Three external clock inputs
  - Five internal clock inputs
  - Two multipurpose input/output signals acting as trigger event
  - Trigger/capture events can be directly synchronized by PWM signals
- Read of the Capture Registers by the PDC
- Compare Event Fault Generation for PWM
- Register Write Protection
- Safety/Security Reports

© 2026 Microchip Technology Inc. and its subsidiaries

36.3. Block Diagram

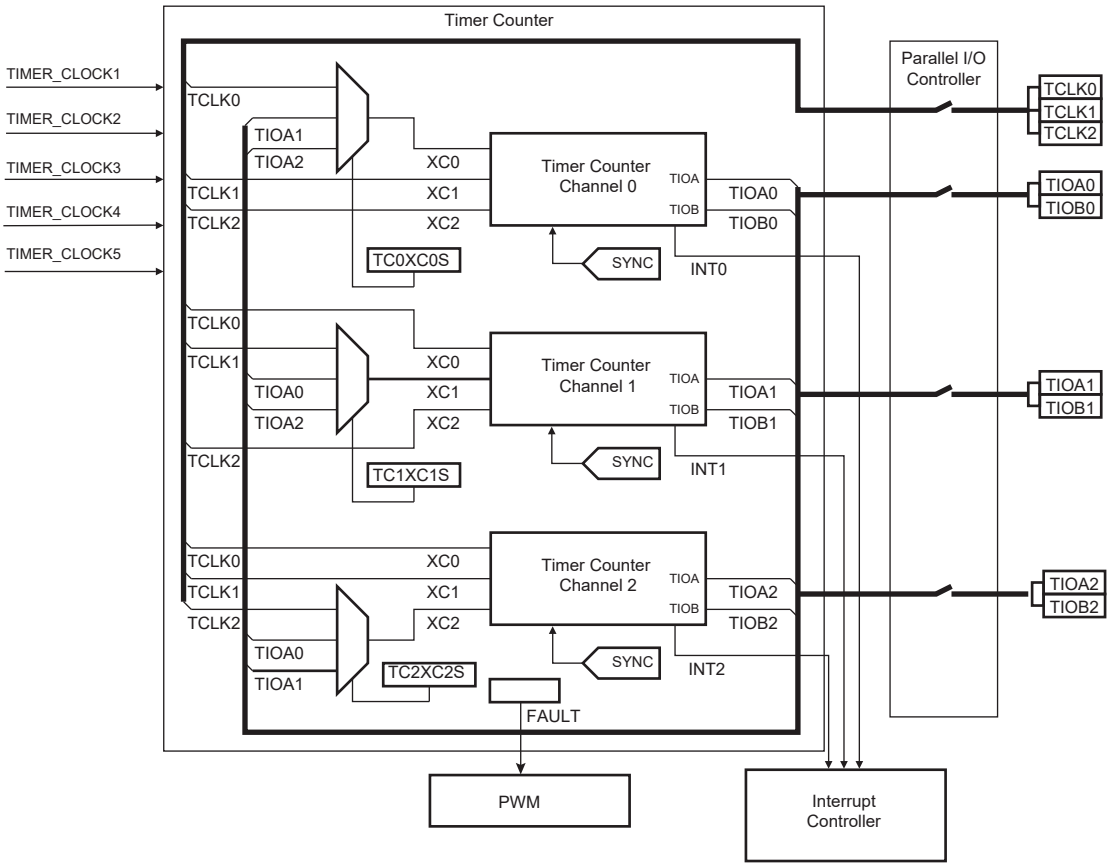
Table 36-1. Timer Counter Clock Assignment

Name	Definition
TIMER_CLOCK1	GCLK
TIMER_CLOCK2	MCK/8
TIMER_CLOCK3	MCK/32
TIMER_CLOCK4	MCK/128
TIMER_CLOCK5 (See Note)	MD_SLCK

Note:

1. The GCLK [TC_ID] frequency must be at least three times lower than peripheral clock frequency.

Figure 36-1. TC Block Diagram



Notes:

1. This figure provides pin names of a first instance of a Timer Counter block (i.e., instance TC0). For any subsequent instances, the signal numbering increments. For example, “TCLK3-TCLK5”, "TIOA3-TIOA5” and "TIOB3-TIOB5” are the external clock input pins of a second Timer Counter block (i.e., instance TC1).
2. The QDEC connections are detailed in Figure 36-17.

© 2026 Microchip Technology Inc. and its subsidiaries

Table 36-2. Channel Signal Description

Signal Name	Description
XC0, XC1, XC2	Channel clock source that can be connected to TIOAx, TIOBx, TCLKx
TIMER_CLOCK1-5	Channel clock source from system clocks
TIOAx	Capture mode: Timer Counter input Waveform mode: Timer Counter output
TIOBx	Capture mode: Timer Counter input Waveform mode: Timer Counter input/output
INT	Interrupt signal output (internal signal)
SYNC	Synchronization input signal (from Configuration register)

36.4. Pin List

Table 36-3. Pin List

Pin Name	Description	Type
TCLK0–TCLK2	External clock input	Input
TIOA0–TIOA2	I/O line A	I/O
TIOB0–TIOB2	I/O line B	I/O

Note: This table provides pin names of a first instance of a Timer Counter block (i.e., instance TC0). For any subsequent instances, the signal numbering increments. For example, “TCLK3-TCLK5”, "TIOA3-TIOA5” and "TIOB3-TIOB5” are the external clock input pins of a second Timer Counter block (i.e., instance TC1).

36.5. Product Dependencies

36.5.1. I/O Lines

The pins used for interfacing the compliant external devices may be multiplexed with PIO lines. The programmer must first program the PIO controllers to assign the TC pins to their peripheral functions.

36.5.2. Power Management

The TC is clocked through the Power Management Controller (PMC), thus the programmer must first configure the PMC to enable the Timer Counter clock of each channel.

36.5.3. Interrupt Sources

36.5.4. Synchronization Inputs from PWM

The TC has trigger/capture inputs internally connected to the PWM. See Synchronization with PWM and refer to the implementation of the Pulse Width Modulation (PWM) in this product.

36.5.5. Fault Output

The TC has the FAULT output internally connected to the fault input of PWM. See Fault Mode and refer to the implementation of the Pulse Width Modulation (PWM) in this product.

36.6. Functional Description

36.6.1. Description

All channels of the Timer Counter are independent and identical in operation except when the QDEC is enabled. The registers for channel programming are listed in Register Summary.

© 2026 Microchip Technology Inc. and its subsidiaries

36.6.2. 32-bit Counter

Each 32-bit channel is organized around a 32-bit counter. The value of the counter is incremented at each positive edge of the selected clock. When the counter has reached the value 2^{32}-1 and passes to zero, an overflow occurs and the COVFS bit in the Interrupt Status register (TC_SR) is set.

The current value of the counter is accessible in real time by reading the Counter Value register (TC_CV). The counter can be reset by a trigger. In this case, the counter value passes to zero on the next valid edge of the selected clock.

36.6.3. Clock Selection

Input clock signals of each channel can be connected either to the external inputs TCLKx, or to the internal I/O signals TIOAx for chaining(1) by programming the Block Mode register (TC_BMR). See the figure Clock Chaining Selection.

Each channel can independently select a source for its counter(2):

- Signals from other channels: XC0, XC1 or XC2
- Signals from the system: GCLK, MCK/8, MCK/32, MCK/128, MD_SLCK

This selection is made by the TCCLKS bits in the Channel Mode register (TC_CMRx).

The selected clock can be inverted with TC_CMRx.CLKI. This allows counting on the opposite edges of the clock.

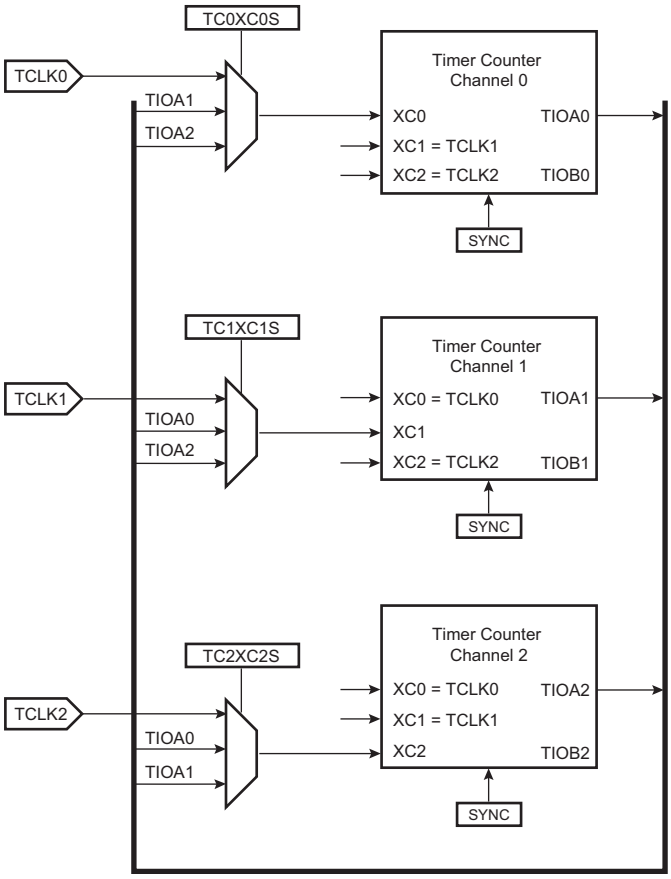
The burst function allows the clock to be validated when an external signal is high. The BURST parameter in the TC_CMRx defines this signal (none, XC0, XC1, XC2). See the figure Clock Selection.

Notes:

1. In Waveform mode, to chain two timers, it is mandatory to initialize some parameters:
   - Configure TIOx outputs to 1 or 0 by writing the required value to TC_CMRx.ASWTRG.
   - Bit TC_BCR.SYNC must be written to 1 to start the channels at the same time.
2. In all cases, if an external clock or asynchronous internal clock GCLK [TC_ID] is used, the duration of each of its levels must be longer than the peripheral clock period, so the clock frequency will be at least 2.5 times lower than the peripheral clock.

© 2026 Microchip Technology Inc. and its subsidiaries

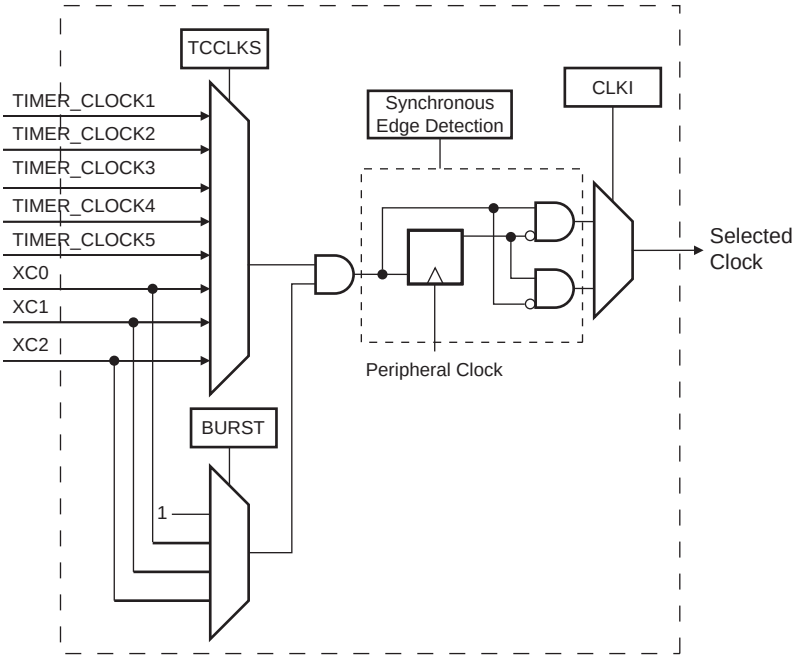
Figure 36-2. Clock Chaining Selection



Note: This figure provides pin names of a first instance of a Timer Counter block (i.e., instance TC0). For any subsequent instances, the signal numbering increments. For example, “TCLK3-TCLK5”, "TIOA3-TIOA5” and "TIOB3-TIOB5” are the external clock input pins of a second Timer Counter block (i.e., instance TC1).

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 36-3. Clock Selection



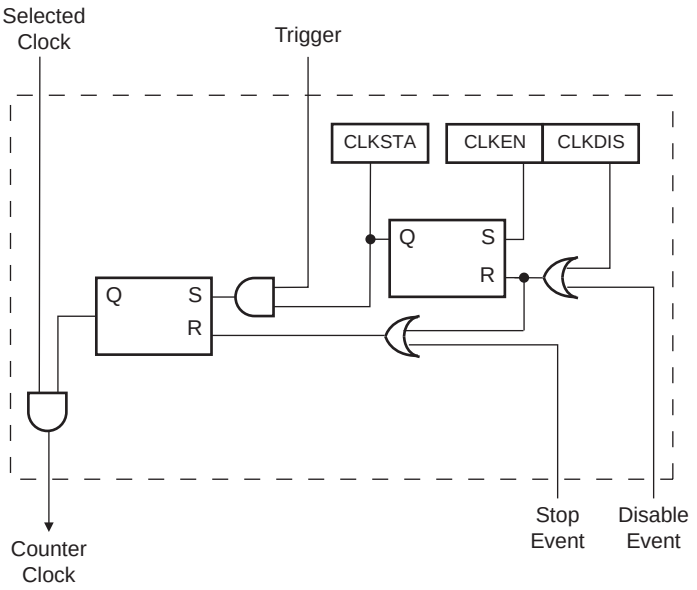
36.6.4. Clock Control

The clock of each counter can be controlled in two different ways: it can be enabled/disabled and started/stopped, as shown in the following figure.

- The clock can be enabled or disabled by the user with the CLKEN and the CLKDIS commands in the Channel Control register (TC_CCR). In Capture mode it can be disabled by an RB load event if TC_CMRx.LDBDIS is set to ‘1’. In Waveform mode, it can be disabled by an RC Compare event if TC_CMRx.CPCDIS is set to ‘1’. When disabled, the start or the stop actions have no effect: only a CLKEN command in the TC_CCR can reenable the clock. When the clock is enabled, TC_SR.CLKSTA is set.
- The clock can also be started or stopped: a trigger (software, synchro, external or compare) always starts the clock. The clock can be stopped by an RB load event in Capture mode (TC_CMRx.LDBSTOP = 1) or an RC compare event in Waveform mode (TC_CMRx.CPCSTOP = 1). The start and the stop commands are effective only if the clock is enabled.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 36-4. Clock Control



36.6.5. Operating Modes

Each channel can operate independently in two different modes:

- Capture mode provides measurement on signals.
- Waveform mode provides wave generation.

The TC operating mode is programmed with TC_CMRx.WAVE.

In Capture mode, TIOAx and TIOBx are configured as inputs.

In Waveform mode, TIOAx is always configured to be an output and TIOBx is an output if it is not selected to be the external trigger.

36.6.6. Trigger Events

An input trigger event resets the internal counter value and starts the counter clock. Three types of trigger are common to both modes, and an external trigger is available to each mode (Capture or Waveform).

Regardless of the input trigger event used, it will be taken into account at the following active edge of the selected clock. This means that the counter value can be read differently from zero just after a trigger event, especially when a low frequency signal is selected as the clock.

The following input trigger events are common to both modes:

- Software trigger: Each channel has a software trigger, available by setting TC_CCR.SWTRG.
- SYNC: Each channel has a synchronization signal SYNC. When asserted, this signal has the same effect as a software trigger. The SYNC signals of all channels are asserted simultaneously by writing TC_BCR with SYNC set.
- Compare RC trigger: RC is implemented in each channel and can provide a trigger when the counter value matches the RC value if TC_CMRx.CPCTRG is set.

The timer channel can also be configured to be triggered by an external event.

In Capture mode, the external trigger signal can be selected between TIOAx and TIOBx.

In Waveform mode, an external event can be programmed on one of the following signals: TIOBx, XC0, XC1 or XC2. This external event can then be programmed to perform a trigger by setting TC_CMRx.ENETRG.

© 2026 Microchip Technology Inc. and its subsidiaries

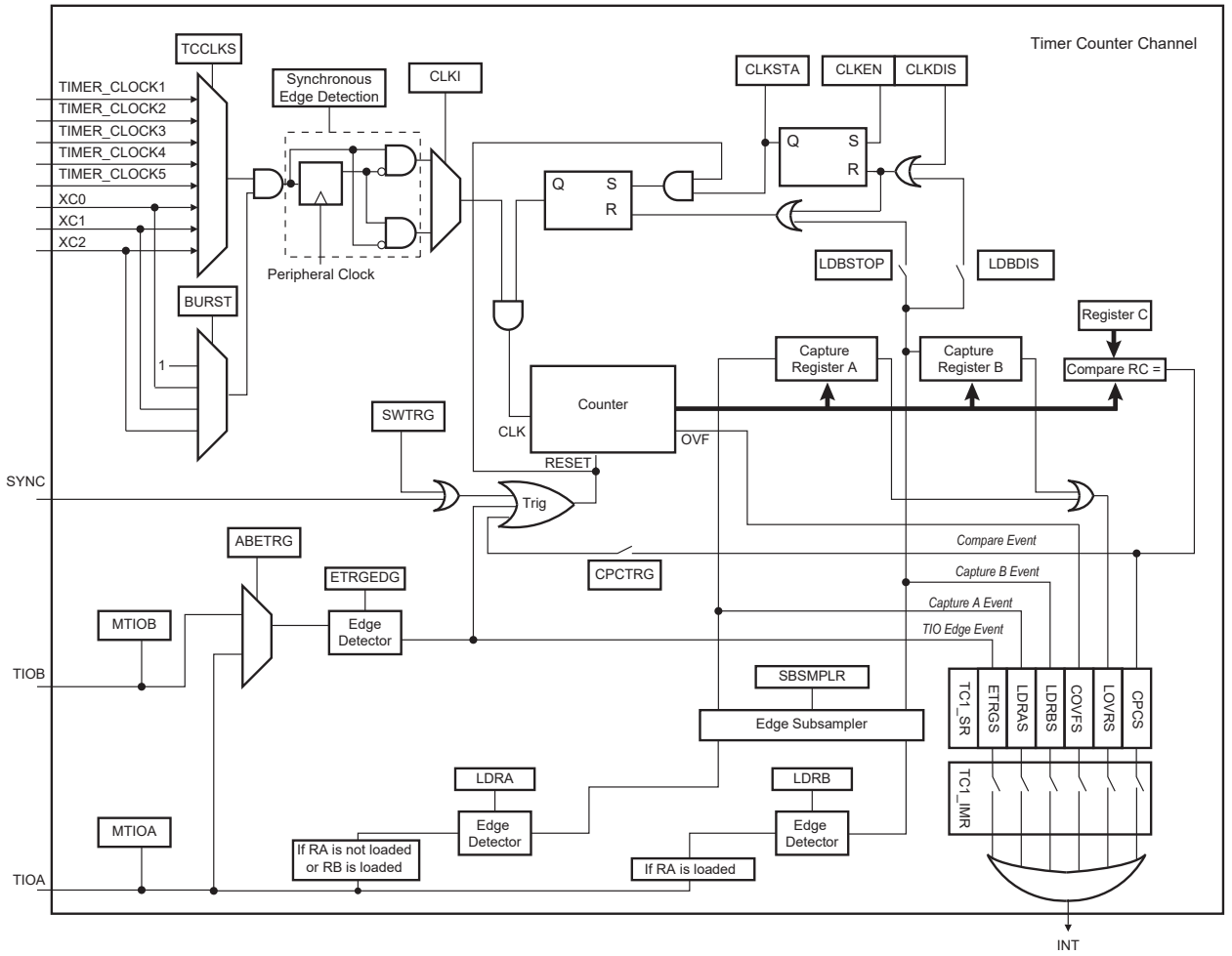
If an external trigger event is used, the duration of the pulses must be longer than the peripheral clock period to be detected.

36.6.7. Trigger Conditions

In addition to the SYNC signal, the software trigger and the RC compare trigger, an external trigger can be defined.

The ABETRG bit in the TC_CMR selects TIOAx or TIOBx input signal as an external trigger or the trigger signal from the output comparator of the PWM module. The External Trigger Edge Selection parameter (ETRGEDG field in TC_CMR) defines the edge (rising, falling, or both) detected to generate an external trigger. If ETRGEDG = 0 (none), the external trigger is disabled.

Figure 36-5. Capture Mode



36.6.8. Capture Mode

Capture mode is entered by clearing TC_CMRx.WAVE.

Capture mode allows the TC channel to perform measurements such as pulse timing, frequency, period, duty cycle and phase on TIOAx and TIOBx signals which are considered as inputs.

The figure Capture Mode shows the configuration of the TC channel when programmed in Capture mode.

© 2026 Microchip Technology Inc. and its subsidiaries

36.6.9. Capture Registers A and B

Registers A and B (TC_RA and TC_RB) are used as capture registers. They can be loaded with the counter value when a programmable event occurs on the signal TIOAx.

TC_CMRx.LDRA defines the TIOAx selected edge for the loading of TC_RA, and TC_CMRx.LDRB defines the TIOAx selected edge for the loading of TC_RB.

The subsampling ratio defined by TC_CMRx.SBSMPLR is applied to these selected edges, so that the loading of Register A and Register B occurs once every 1, 2, 4, 8 or 16 selected edges.

TC_RA is loaded only if it has not been loaded since the last trigger or if TC_RB has been loaded since the last loading of TC_RA.

TC_RB is loaded only if TC_RA has been loaded since the last trigger or the last loading of TC_RB.

Loading TC_RA or TC_RB before the read of the last value loaded sets TC_SR.LOVRS. In this case, the old value is overwritten.

When DMA is used, the Register AB (TC_RAB) address must be configured as source address of the transfer. TC_RAB provides the next unread value from TC_RA and TC_RB. It may be read by the DMA after a request has been triggered upon loading TC_RA or TC_RB.

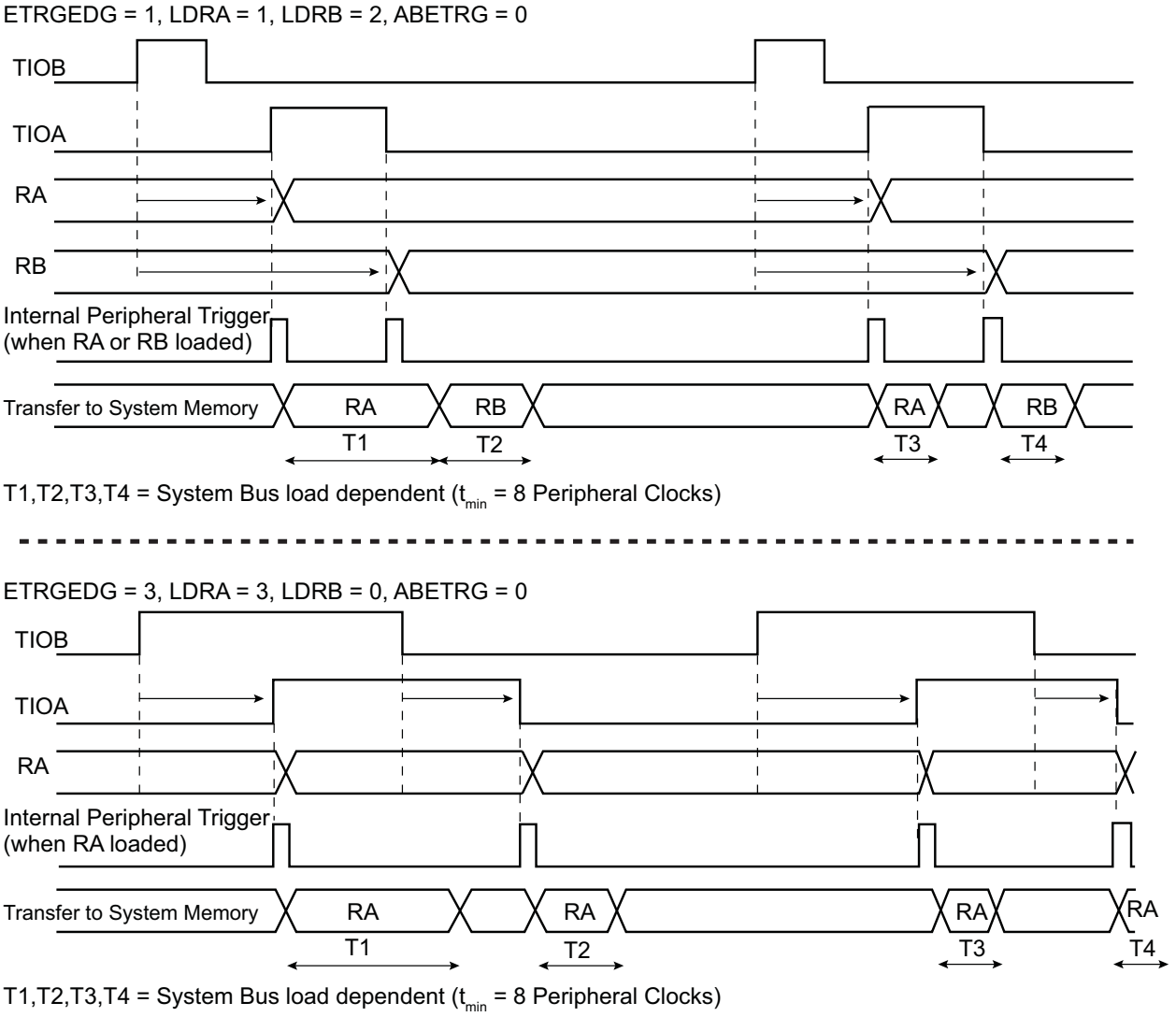
36.6.10. Transferring Timer Values with PDC in Capture Mode

The PDC can perform access from the TC to system memory in Capture mode only.

The following figure illustrates how TC_RA and TC_RB can be loaded in the system memory without processor intervention.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 36-6. Example of Transfer with PDC in Capture Mode



36.6.11. Waveform Mode

Waveform mode is entered by setting the TC_CMRx.WAVE bit.

In Waveform mode, the TC channel generates one or two PWM signals with the same frequency and independently programmable duty cycles, or generates different types of one-shot or repetitive pulses.

In this mode, TIOAx is configured as an output and TIOBx is defined as an output if it is not used as an external event (EEVT parameter in TC_CMR).

The figure Waveform Mode shows the configuration of the TC channel when programmed in Waveform operating mode.

36.6.12. Waveform Selection

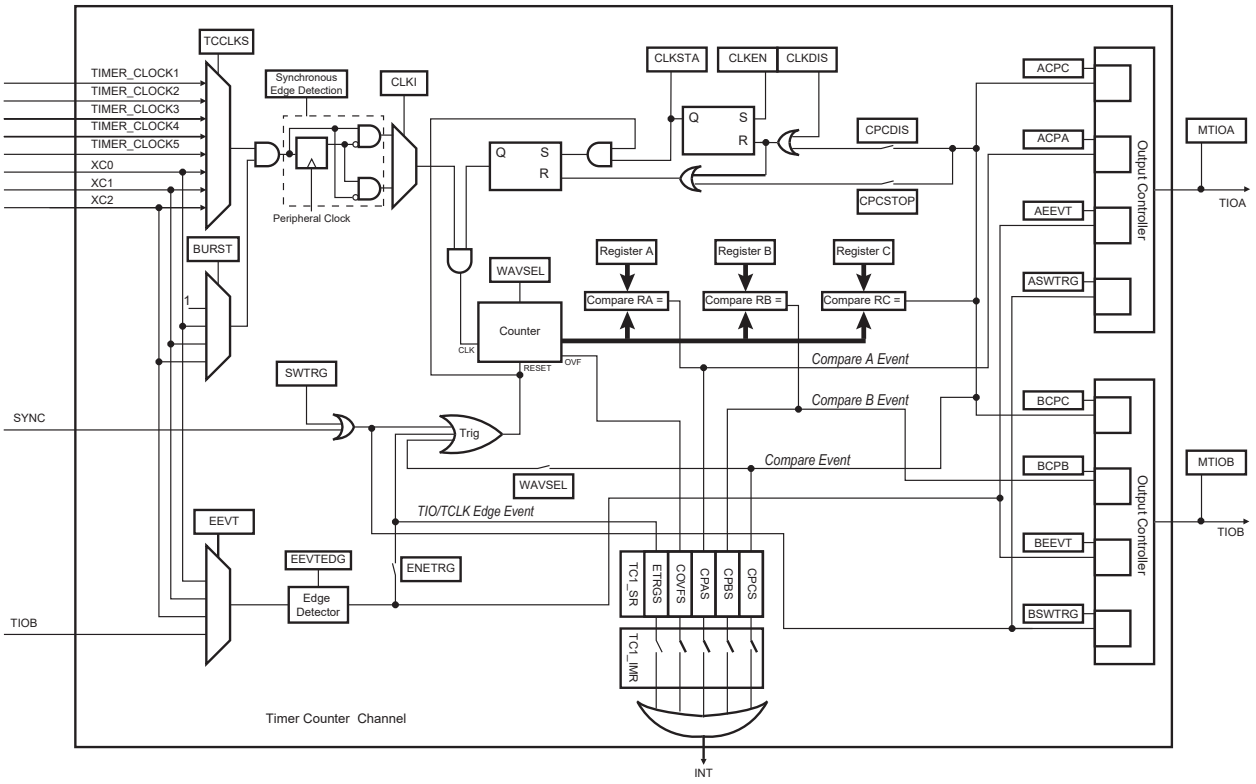
Depending on the WAVSEL parameter in TC_CMR, the behavior of TC_CV varies.

With any selection, TC_RA, TC_RB and TC_RC can all be used as compare registers.

RA Compare is used to control the TIOAx output, RB Compare is used to control the TIOBx output (if correctly configured) and RC Compare is used to control TIOAx and/or TIOBx outputs.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 36-7. Waveform Mode



36.6.12.1. WAVSEL = 00

When WAVSEL = 00, the value of TC_CV is incremented from 0 to 2^{32}-1. Once 2^{32}-1 has been reached, the value of TC_CV is reset. Incrementation of TC_CV starts again and the cycle continues.

An external event trigger or a software trigger can reset the value of TC_CV. It is important to note that the trigger may occur at any time.

See the following figures.

RC Compare cannot be programmed to generate a trigger in this configuration. At the same time, RC Compare can stop the counter clock (CPCSTOP = 1 in TC_CMR) and/or disable the counter clock (CPCDIS = 1 in TC_CMR).

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 36-8. WAVSEL = 00 without Trigger

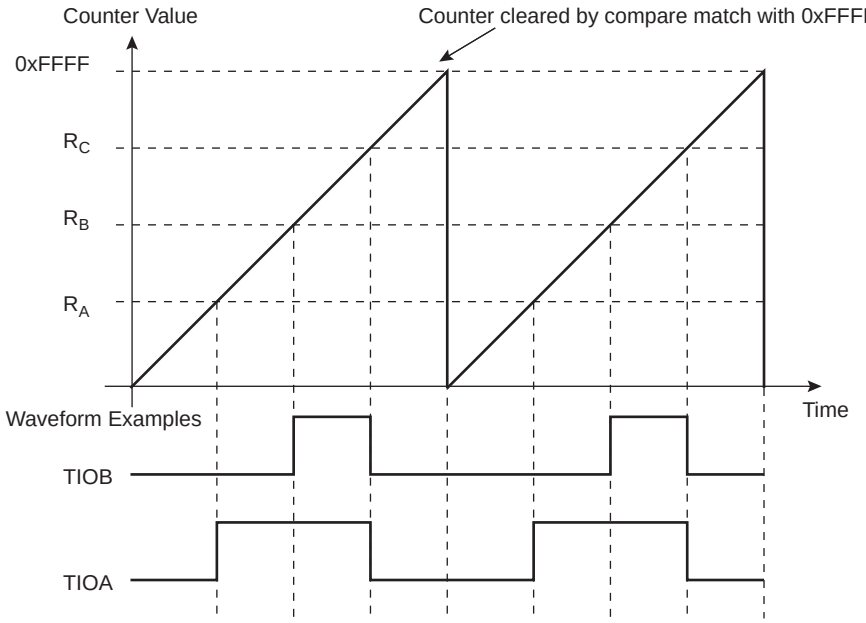
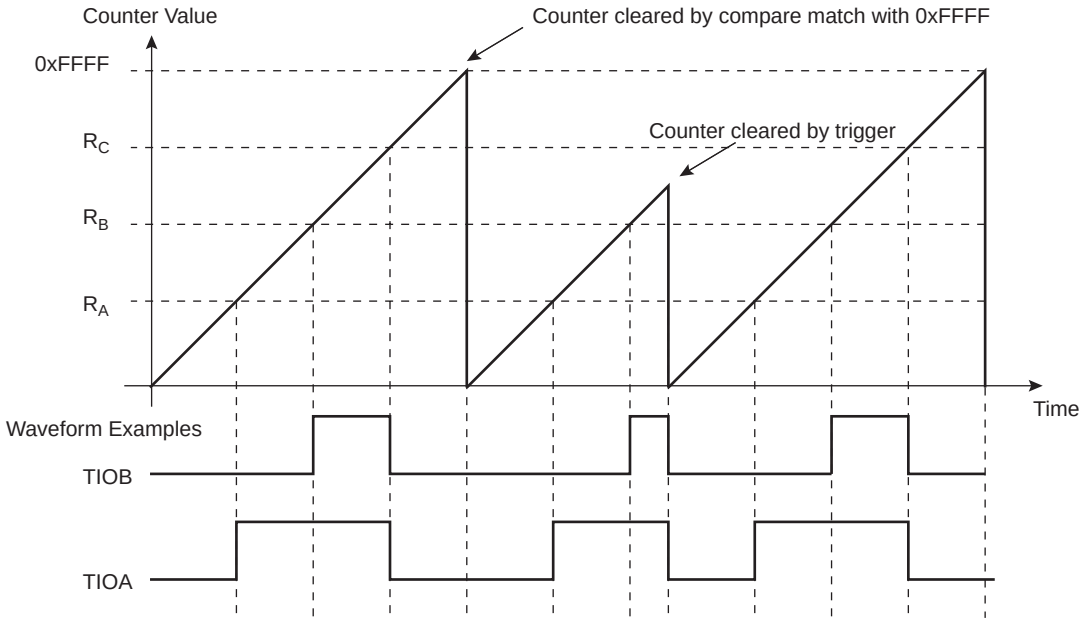


Figure 36-9. WAVSEL = 00 with Trigger



36.6.12.2. WAVSEL = 10

When WAVSEL = 10, the value of TC_CV is incremented from 0 to the value of RC, then automatically reset on a RC Compare. Once the value of TC_CV has been reset, it is then incremented and so on.

It is important to note that TC_CV can be reset at any time by an external event or a software trigger if both are programmed correctly.

See the following figures.

In addition, RC Compare can stop the counter clock (CPCSTOP = 1 in TC_CMR) and/or disable the counter clock (CPCDIS = 1 in TC_CMR).

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 36-10. WAVSEL = 10 without Trigger

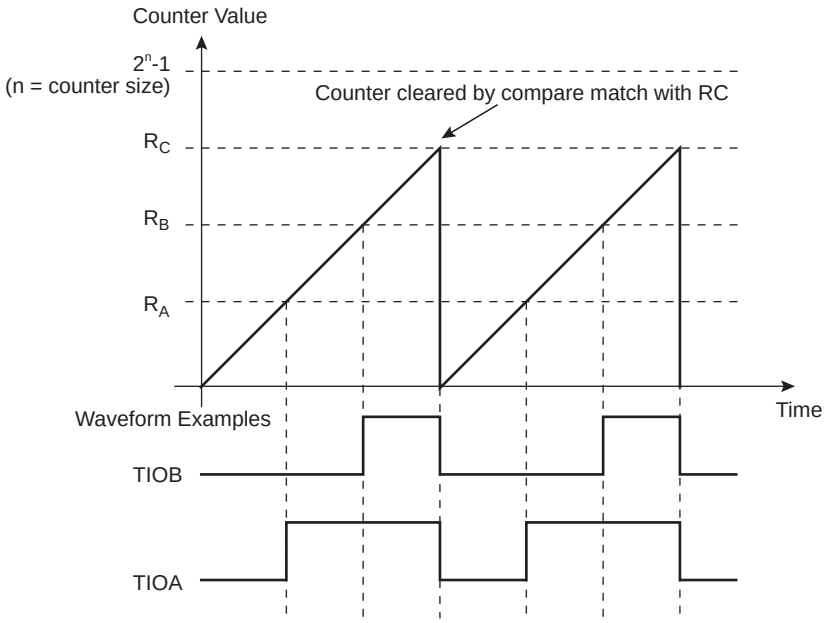
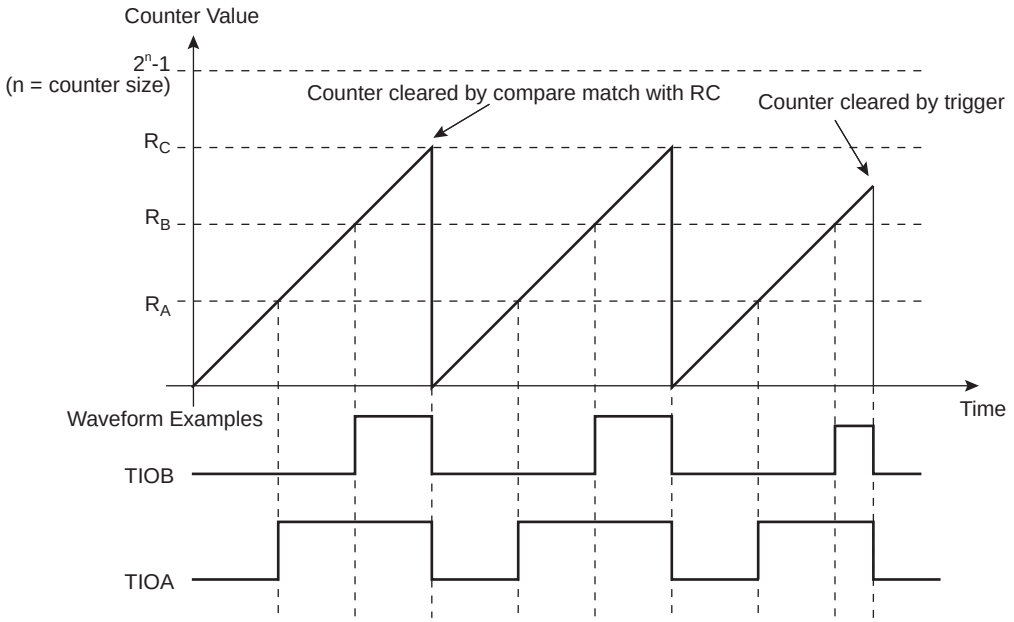


Figure 36-11. WAVSEL = 10 with Trigger



36.6.12.3. WAVSEL = 01

When WAVSEL = 01, the value of TC_CV is incremented from 0 to 2^{32}-1 . Once 2^{32}-1 is reached, the value of TC_CV is decremented to 0, then reincremented to 2^{32}-1 and so on.

A trigger such as an external event or a software trigger can modify TC_CV at any time. If a trigger occurs while TC_CV is incrementing, TC_CV then decrements. If a trigger is received while TC_CV is decrementing, TC_CV then increments.

See the following figures.

RC Compare cannot be programmed to generate a trigger in this configuration.

© 2026 Microchip Technology Inc. and its subsidiaries

At the same time, RC Compare can stop the counter clock (CPCSTOP = 1) and/or disable the counter clock (CPCDIS = 1).

Figure 36-12. WAVSEL = 01 without Trigger

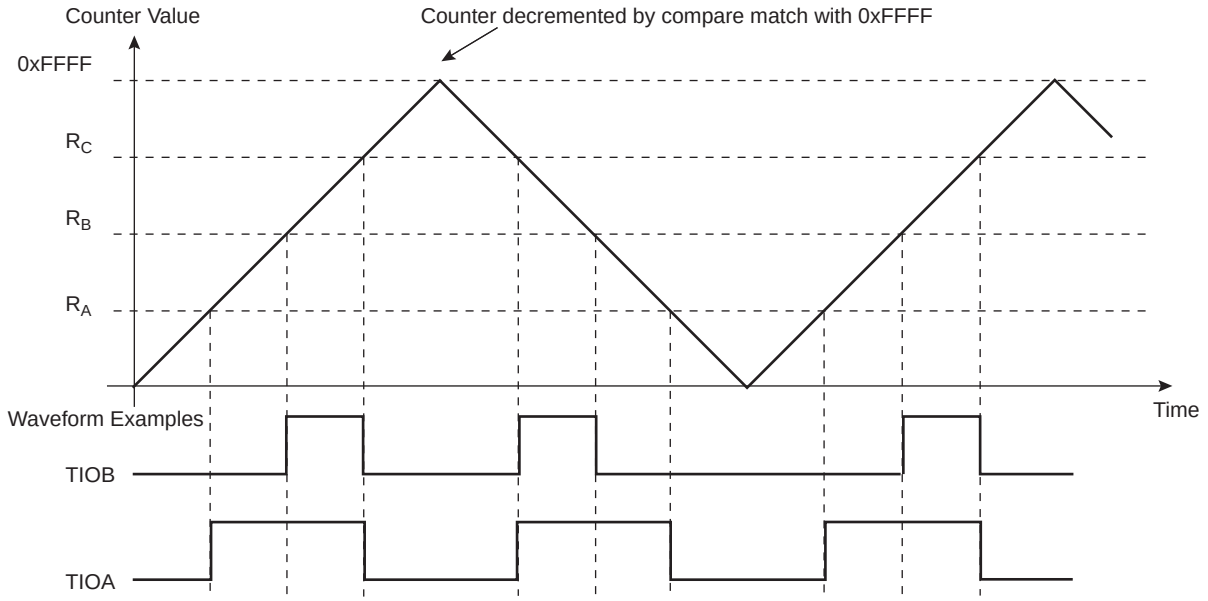
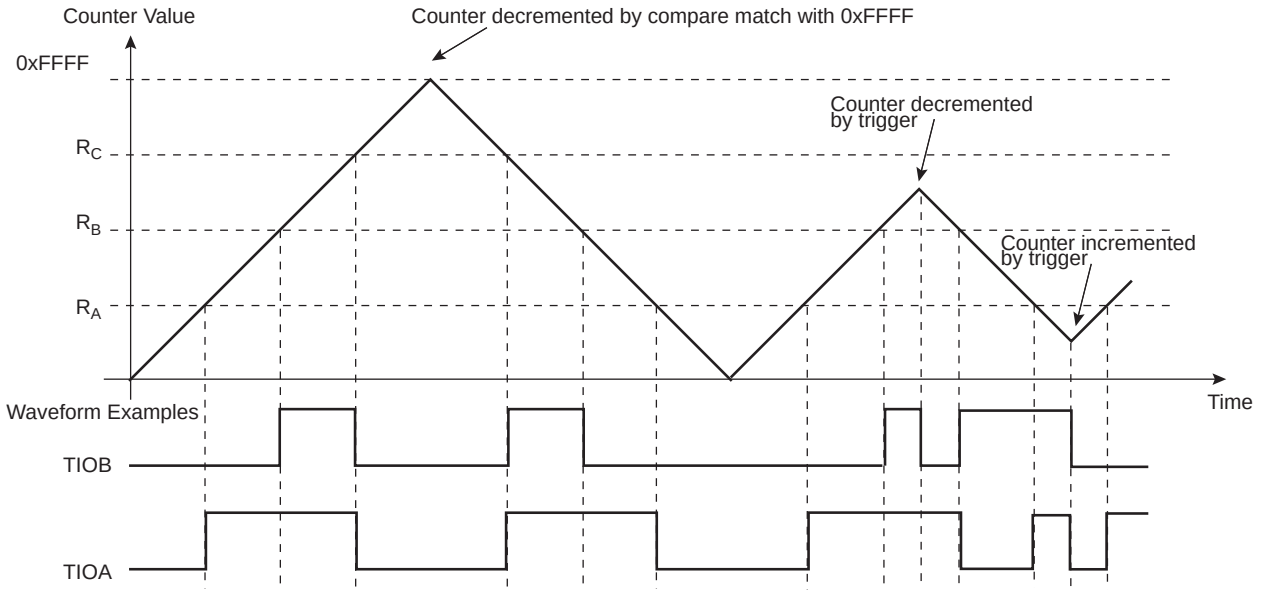


Figure 36-13. WAVSEL = 01 with Trigger



36.6.12.4. WAVSEL = 11

When WAVSEL = 11, the value of TC_CV is incremented from 0 to RC. Once RC is reached, the value of TC_CV is decremented to 0, then reincremented to RC and so on.

A trigger such as an external event or a software trigger can modify TC_CV at any time. If a trigger occurs while TC_CV is incrementing, TC_CV then decrements. If a trigger is received while TC_CV is decrementing, TC_CV then increments.

See the following figures.

RC Compare can stop the counter clock (CPCSTOP = 1) and/or disable the counter clock (CPCDIS = 1).

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 36-14. WAVSEL = 11 without Trigger

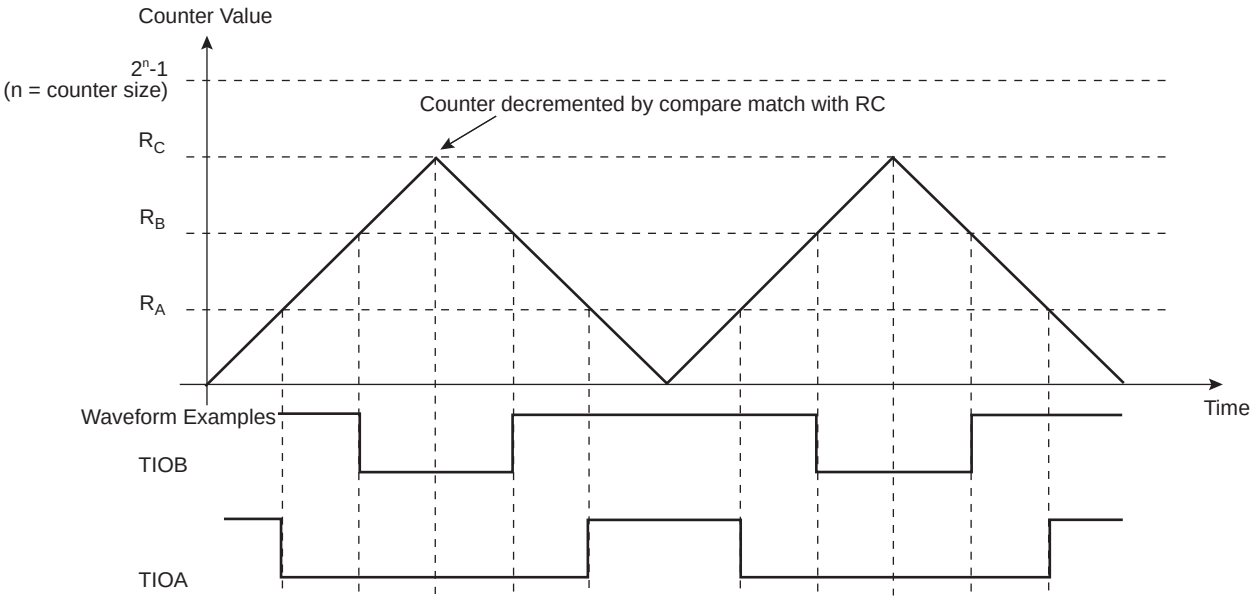
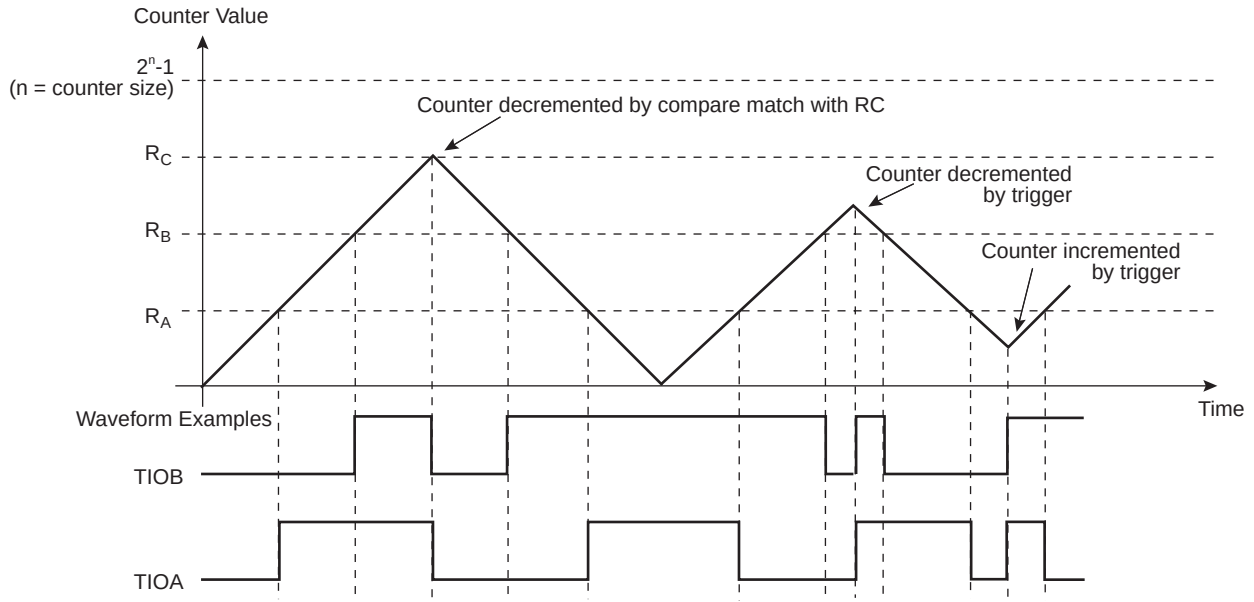


Figure 36-15. WAVSEL = 11 with Trigger



36.6.13. External Event/Trigger Conditions

An external event can be programmed to be detected on one of the clock sources (XC0, XC1, XC2) or TIOBx. The external event selected can then be used as a trigger.

The event trigger is selected using TC_CMR.EEVT. The trigger edge (rising, falling or both) for each of the possible external triggers is defined in TC_CMR.EEVTEDG. If EEVTEDG is cleared (none), no external event is defined.

If TIOBx is defined as an external event signal (EEVT = 0), TIOBx is no longer used as an output and the compare register B is not used to generate waveforms and subsequently no IRQs. In this case, the TC channel can only generate a waveform on TIOAx.

When an external event is defined, it can be used as a trigger by setting TC_CMR.ENETRG.

© 2026 Microchip Technology Inc. and its subsidiaries

As in Capture mode, the SYNC signal and the software trigger are also available as triggers. RC Compare can also be used as a trigger depending on the parameter WAVSEL.

36.6.14. Synchronization with PWM

The inputs TIOAx/TIOBx can be bypassed, and thus channel trigger/capture events can be directly driven by the independent PWM module.

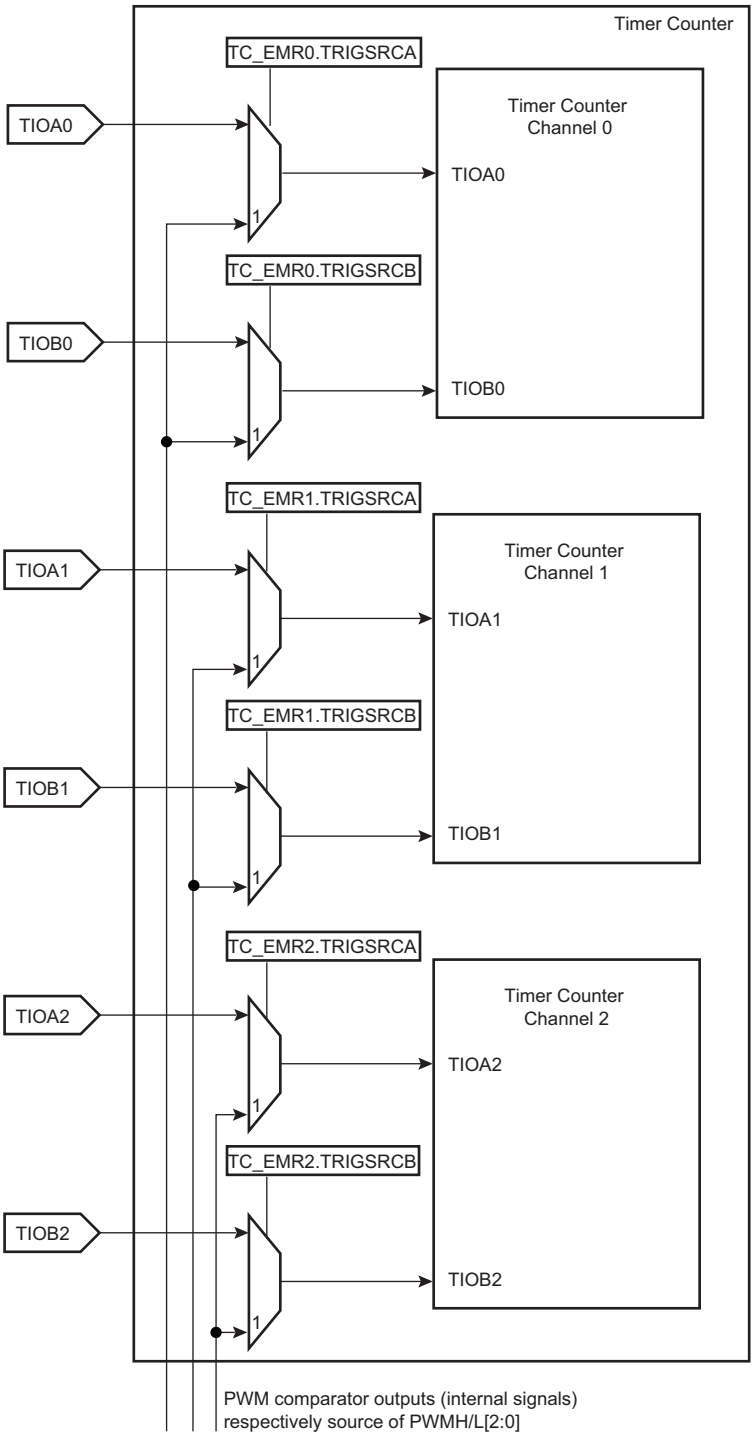
PWM comparator outputs (internal signals without dead-time insertion - OCx), respectively source of the PWMH/L[2:0] outputs, are routed to the internal TC inputs. These specific TC inputs are multiplexed with TIOA/B input signal to drive the internal trigger/capture events.

The selection is made in the Extended Mode register (TC_EMR) fields TRIGSRCA and TRIGSRCB (see TC_EMRx).

Each channel of the TC module can be synchronized by a different PWM channel as described in the following figure.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 36-16. Synchronization with PWM



Note: This figure provides pin names of the first instance of a Timer Counter block (i.e., instance TC0). For any subsequent instances, the signal numbering increments. For example, "TIOA3-TIOA5" and "TIOB3-TIOB5" are the external IO pins of a second Timer Counter block (i.e., instance TC1).

36.6.15. Output Controller

The output controller defines the output level changes on TIOAx and TIOBx following an event. TIOBx control is used only if TIOBx is defined as output (not as an external event).

© 2026 Microchip Technology Inc. and its subsidiaries

The following events control TIOAx and TIOBx:

- Software trigger
- External event
- RC compare

RA Compare controls TIOAx, and RB Compare controls TIOBx. Each of these events can be programmed to set, clear or toggle the output as defined in the corresponding parameter in TC_CMR.

36.6.16. Quadrature Decoder

36.6.16.1. Description

The quadrature decoder (QDEC) is driven by TIOA0, TIOB0 and TIOB1 input pins and drives the timer counter of channel 0 and 1. Channel 2 can be used as a time base in case of speed measurement requirements (see the following figure).

When writing a '0' to TC_BMR.QDEN, the QDEC is bypassed and the IO pins are directly routed to the timer counter function.

TIOA0 and TIOB0 are to be driven by the two dedicated quadrature signals from a rotary sensor mounted on the shaft of the off-chip motor.

A third signal from the rotary sensor can be processed through pin TIOB1 and is typically dedicated to be driven by an index signal if it is provided by the sensor. This signal is not required to decode the quadrature signals PHA, PHB.

TC_CMRx.TCCLKS must be configured to select XC0 input (i.e., 0x101). Field TC0XC0S has no effect as soon as the QDEC is enabled.

Either speed or position/revolution can be measured. Position channel 0 accumulates the edges of PHA, PHB input signals giving a high accuracy on motor position whereas channel 1 accumulates the index pulses of the sensor, therefore the number of rotations. Concatenation of both values provides a high level of precision on motion system position.

In Speed mode, position cannot be measured but revolution can be measured.

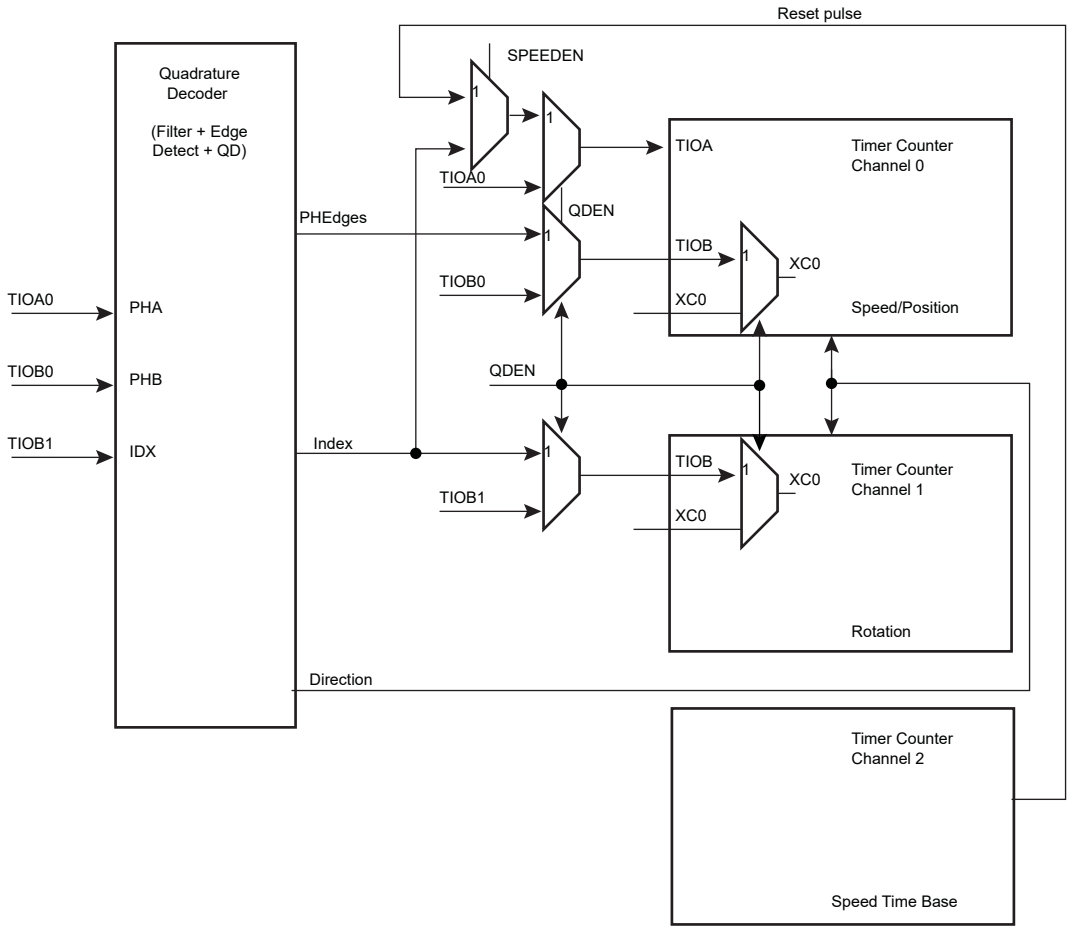
Inputs from the rotary sensor can be filtered prior to downstream processing. Accommodation of input polarity, phase definition and other factors are configurable.

Interruptions can be generated on different events.

A compare function (using TC_RC) is available on channel 0 (speed/position) or channel 1 (rotation) and can generate an interrupt by means of TC_SRx.CPCS.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 36-17. Predefined Connection of the Quadrature Decoder with Timer Counters



Note: This figure provides pin names of the first instance of a Timer Counter block (i.e., instance TC0). For any subsequent instances, the signal numbering increments. For example, "TIOA3-TIOA5" and "TIOB3-TIOB5" are the external IO pins of a second Timer Counter block (i.e., instance TC1).

36.6.16.2. Input Preprocessing

Input preprocessing consists of capabilities to take into account rotary sensor factors such as polarities and phase definition followed by configurable digital filtering.

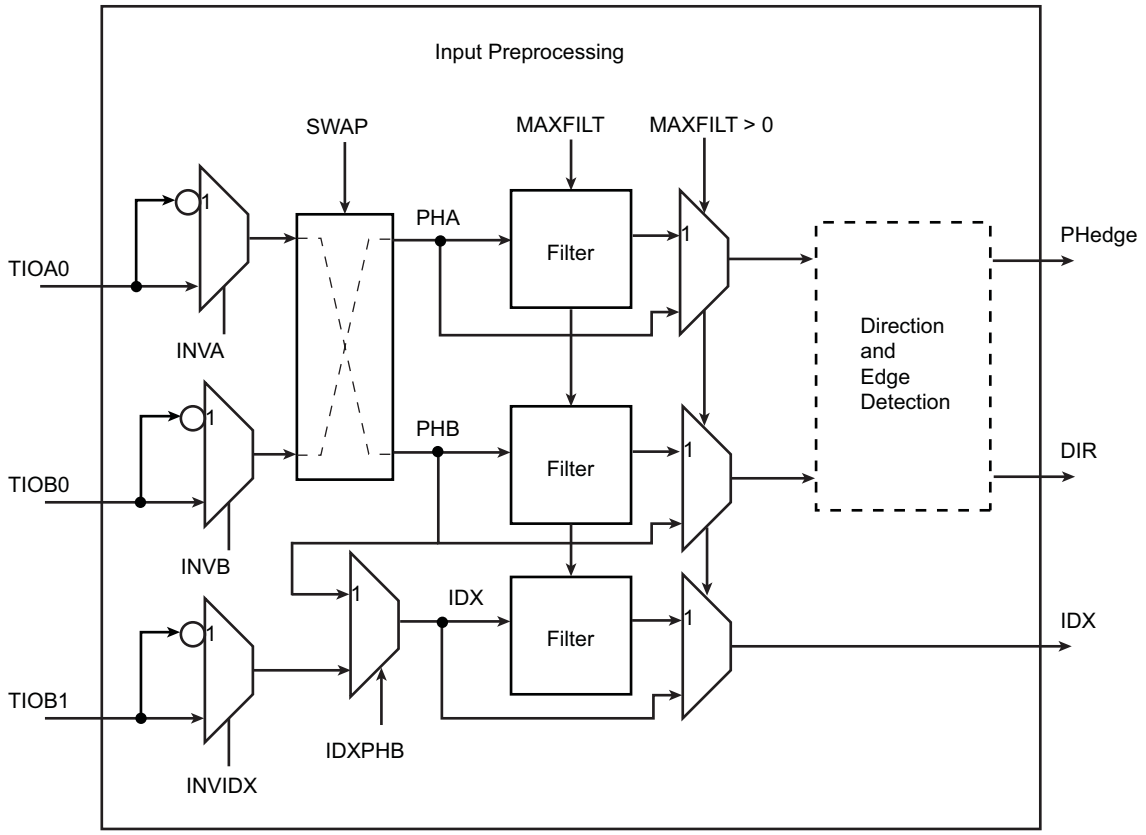
Each input can be negated and swapping PHA, PHB is also configurable.

TC_BMR. MAXFILT is used to configure a minimum duration for which the pulse is stated as valid. When the filter is active, pulses with a duration lower than (MAXFILT +1) × $t_{peripheral\ clock}$ are not passed to downstream logic.

The value of (MAXFILT +1) × $t_{peripheral\ clock}$ must not be greater than 10% of the minimum pulse on PHA, PHB or index when the rotary encoder speed is at its maximum. This speed depends on the application.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 36-18. Input Stage

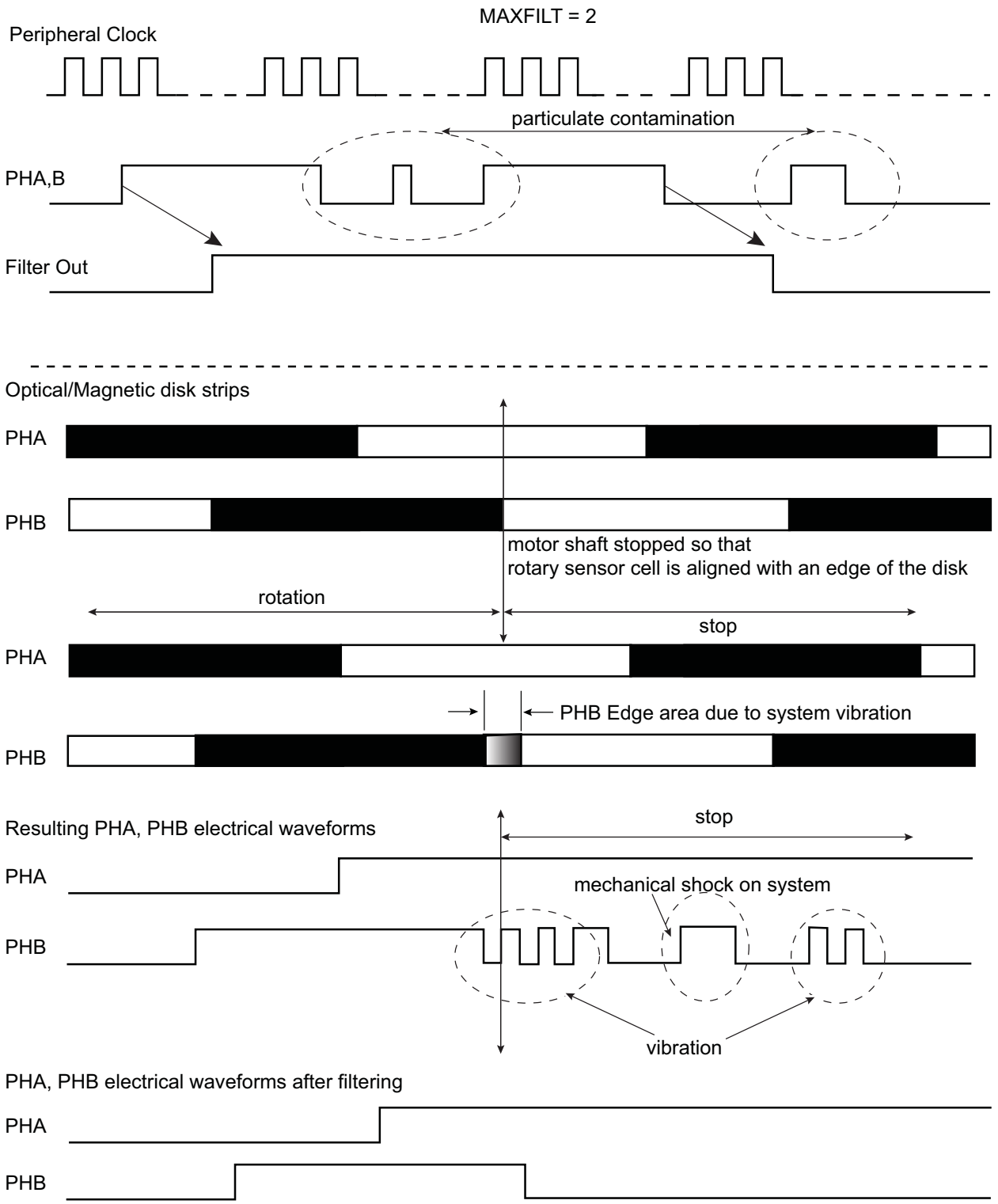


Input filtering can efficiently remove spurious pulses that might be generated by the presence of particulate contamination on the optical or magnetic disk of the rotary sensor.

Spurious pulses can also occur in environments with high levels of electromagnetic interference. Or, simply if vibration occurs even when rotation is fully stopped and the shaft of the motor is in such a position that the beginning of one of the reflective or magnetic bars on the rotary sensor disk is aligned with the light or magnetic (Hall) receiver cell of the rotary sensor. Any vibration can make the PHA, PHB signals toggle for a short duration.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 36-19. Filtering Examples



36.6.16.3. Direction Status and Change Detection

After filtering, the quadrature signals are analyzed to extract the rotation direction and edges of the two quadrature signals detected in order to be counted by TC logic downstream.

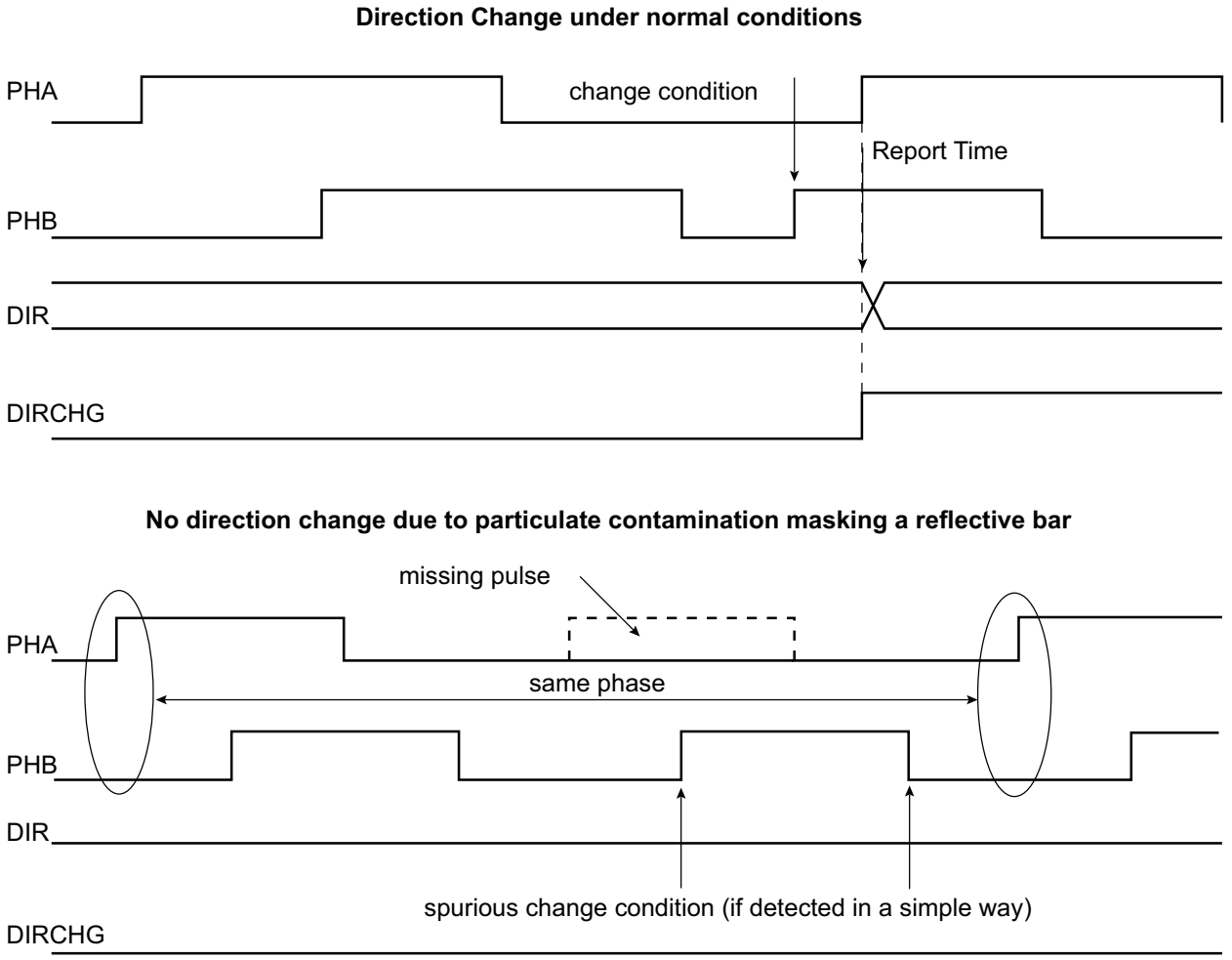
© 2026 Microchip Technology Inc. and its subsidiaries

The direction status can be directly read at anytime in the TC_QISR. The polarity of the direction flag status depends on the configuration written in TC_BMR. INVA, INVB, INVIDX, SWAP modify the polarity of DIR flag.

Any change in rotation direction is reported in the TC_QISR and can generate an interrupt.

The direction change condition is reported as soon as two consecutive edges on a phase signal have sampled the same value on the other phase signal and there is an edge on the other signal. The two consecutive edges of one phase signal sampling the same value on other phase signal is not sufficient to declare a direction change, as particulate contamination may mask one or more reflective bars on the optical or magnetic disk of the sensor. See the following figure for waveforms.

Figure 36-20. Rotation Change Detection

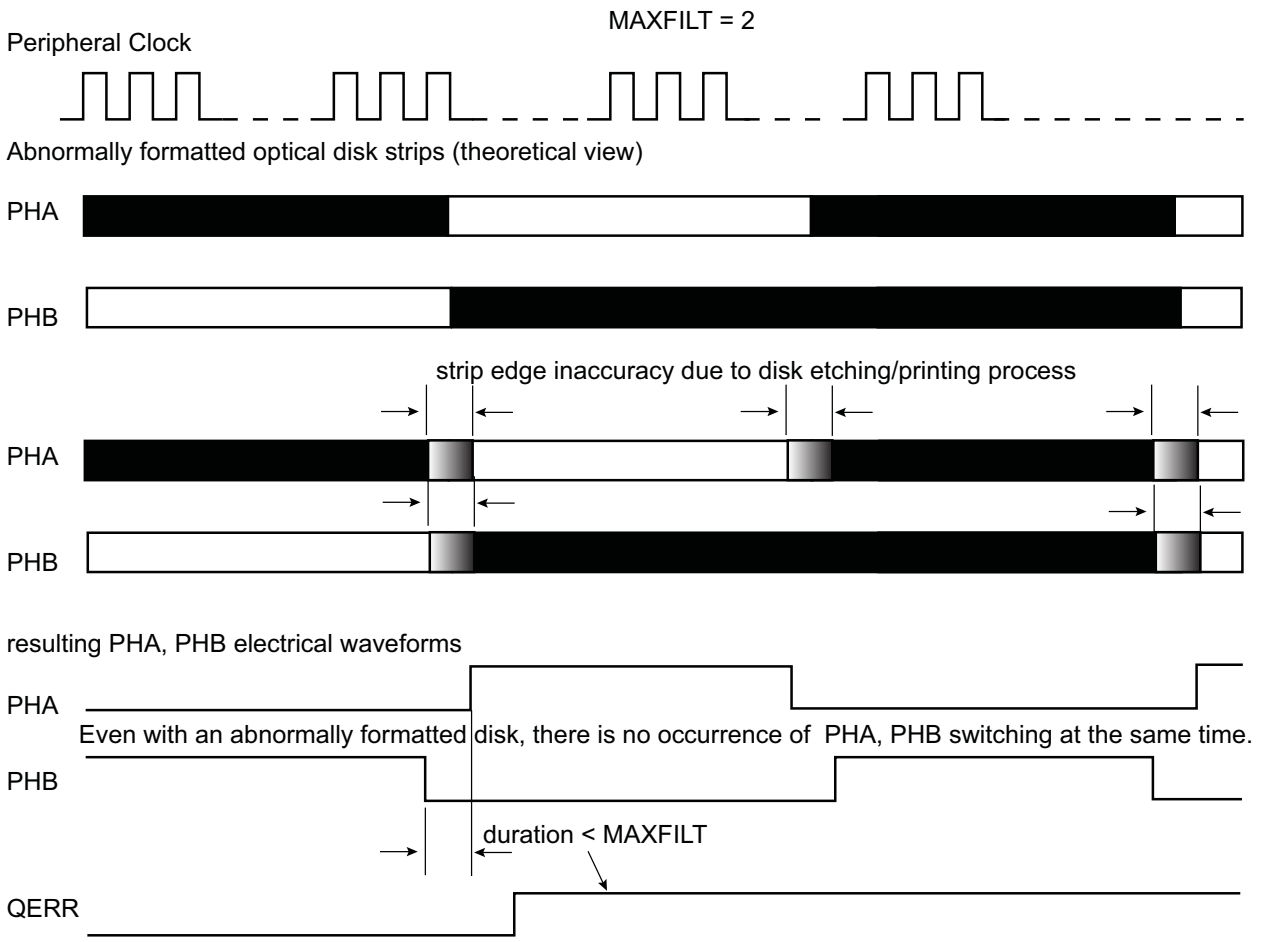


The direction change detection is disabled when TC_BMR.QDTRANS is set. In this case, the DIR flag report must not be used.

A quadrature error is also reported by the QDEC via TC_QISR.QERR. This error is reported if the time difference between two edges on PHA, PHB is lower than a predefined value. This predefined value is configurable and corresponds to (TC_BMR.MAXFILT + 1) × $t_{peripheral\ clock}$ ns. After being filtered, there is no reason to have two edges closer than (TC_BMR.MAXFILT + 1) × $t_{peripheral\ clock}$ ns under normal mode of operation.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 36-21. Quadrature Error Detection



MAXFILT must be tuned according to several factors such as the peripheral clock frequency, type of rotary sensor and rotation speed to be achieved.

36.6.16.4. Position and Rotation Measurement

When TC_BMR.POSEN is set, the motor axis position is processed on channel 0 (by means of the PHA, PHB edge detections) and the number of motor revolutions are recorded on channel 1 if the IDX signal is provided on the TIOB1 input. If no IDX signal is available, the internal counter can be cleared for each revolution if the number of counts per revolution is configured in TC_RC0.RC and the TC_CMR.CPCTRG bit is written to '1'. The position measurement can be read in the TC_CV0 register and the rotation measurement can be read in the TC_CV1 register.

Channel 0 and 1 must be configured in Capture mode (TC_CMR0.WAVE = 0). 'Rising edge' must be selected as the External Trigger Edge (TC_CMR.ETRGEDG = 0x01) and 'TIOAx' must be selected as the External Trigger (TC_CMR.ABETRG = 0x1). The process must be started by configuring TC_CCR.CLKEN and TC_CCR.SWTRG.

In parallel, the number of edges are accumulated on TC channel 0 and can be read on the TC_CV0 register.

Therefore, the accurate position can be read on both TC_CV registers and concatenated to form a 32-bit word.

The TC channel 0 is cleared for each increment of IDX count value.

Depending on the quadrature signals, the direction is decoded and allows to count up or down in TC channels 0 and 1. The direction status is reported on TC_QISR.

© 2026 Microchip Technology Inc. and its subsidiaries

36.6.16.5. Speed Measurement

When TC_BMR.SPEEDEN is set, the speed measure is enabled on channel 0.

A time base must be defined on channel 2 by writing the TC_RC2 period register. Channel 2 must be configured in Waveform mode (WAVE bit set) in TC_CMR2. The WAVSEL field must be defined with 0x10 to clear the counter by comparison and matching with TC_RC value. Field ACPC must be defined at 0x11 to toggle TIOAx output.

This time base is automatically fed back to TIOAx of channel 0 when QDEN and SPEEDEN are set.

Channel 0 must be configured in Capture mode (WAVE = 0 in TC_CMR0). TC_CMR0.ABETRG must be configured at 1 to select TIOAx as a trigger for this channel.

EDGTRG must be set to 0x01, to clear the counter on a rising edge of the TIOAx signal and field LDRA must be set accordingly to 0x01, to load TC_RA0 at the same time as the counter is cleared (LDRB must be set to 0x01). As a consequence, at the end of each time base period the differentiation required for the speed calculation is performed.

The process must be started by configuring bits CLKEN and SWTRG in the TC_CCR.

The speed can be read on field RA in TC_RA0.

Channel 1 can still be used to count the number of revolutions of the motor.

36.6.16.6. Detecting a Missing Index Pulse

To detect a missing index pulse due contamination, dust, etc., the TC_SR0.CPCS flag can be used. It is also possible to assert the interrupt line if the TC_SR0.CPCS flag is enabled as a source of the interrupt by writing a '1' to TC_IER0.CPCS.

The TC_RC0.RC field must be written with the nominal number of counts per revolution provided by the rotary encoder, plus a margin to eliminate potential noise (ex: if the nominal count per revolution is 1024, then TC_RC0.RC=1026).

If the index pulse is missing, the timer value is not cleared and the nominal value is exceeded, then the comparator on the RC triggers an event, TC_SR0.CPCS=1, and the interrupt line is asserted if TC_IER0.CPCS=1.

The missing index pulse detection is only valid if the bit TC_QISR.DIRCHG=0.

36.6.16.7. Detecting a Badly Located Index

The digital filter reduces effects of dust, scratches or contamination on the index line. If the contamination creates a pulse surpassing the filter capacity (in particular at low speed), this can be interpreted as an index pulse, even if it is badly placed.

An on-the-fly detection of a badly-placed index is enabled when TC_BMR.BIDXCE=1. TC_RC0.RC must be written with the nominal number of counts per revolution provided by the rotary encoder + 2. For example, if the nominal count per revolution is 1024, then TC_RC0.RC=1026. This margin of 2 eliminates potential noise.

If the pulse is processed while the current value of the counter (TC_CV0) is outside the value programmed in TC_RC0.RC ±2, the TC_SR0.LDRAS flag is written to '1' and the location of the pulse is written in TC_RA0.

This detection circuitry does not prevent a rotary encoder integrity check before use.

36.6.16.8. Detecting Contamination/Dust at Rotary Encoder Low Speed

The contamination/dust that can be filtered when the rotary encoder speed is high may not be filtered at low speed, thus creating unsollicited direction change, etc.

At low speed, even a minor contamination may appear as a long pulse, and thus not filtered and processed as a standard quadrature encoder pulse.

This contamination can be detected by using the similar method as the missing index detection.

© 2026 Microchip Technology Inc. and its subsidiaries

A contamination exists on a phase line if TC_SR.CPCS = 1 and TC_QISR.DIRCHG = 1 when there is no sollicited change of direction.

36.6.16.9. Missing Pulse Detection and Autocorrection

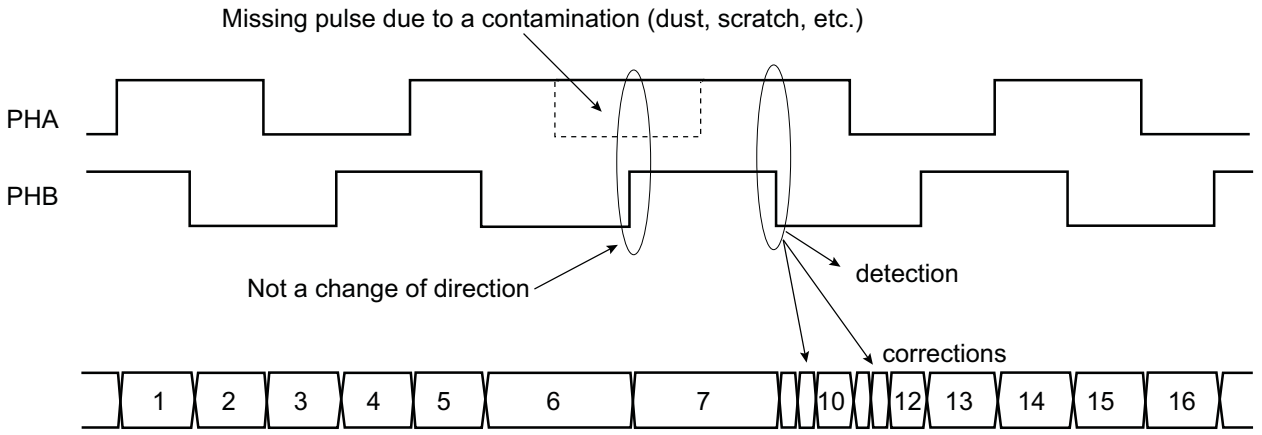
The QDEC is equipped with a circuitry which detects and corrects some errors that may result from contamination on optical disks or other materials producing the quadrature phase signals.

The detection and autocorrection only works if the Count mode is configured for both phases (EDGPHA = 1 in TC_BMR) and is enabled (AUTOC = 1 in TC_BMR).

If a pulse is missing on a phase signal, it is automatically detected and the pulse count reported in the CV field of the TC_CV0/1 is automatically corrected.

There is no detection if both phase signals are affected at the same location on the device providing the quadrature signals because the detection requires a valid phase signal to detect the contamination on the other phase signal.

Figure 36-22. Detection and Autocorrection of Missing Pulses



If a quadrature device is undamaged, the number of pulses counted for a predefined period of time must be the same with or without detection and autocorrection feature.

Therefore, if the measurement results differ, a contamination exists on the device producing the quadrature signals.

This does not substitute the measurements of the number of pulses between two index pulses (if available) but provides a complementary method to detect damaged quadrature devices.

When the device providing quadrature signals is severely damaged, potentially leading to a number of consecutive missing pulses greater than 1, the downstream processing may be affected. It is possible to define the maximum admissible number of consecutive missing pulses before issuing a Missing Pulse Error flag (MPE in TC_QISR). The threshold triggering an MPE flag report can be configured in TC_BMR.MAXCMP. If the field MAXCMP is cleared, MPE never rises. The flag MAXCMP can trigger an interrupt while the QDEC is operating, thus providing a real time report of a potential problem on the quadrature device.

36.6.16.10. Report of Filtered Pulses due to Contamination/Dust

When the digital filter removes pulses created by contamination/dust, a report is provided in the TC QDEC Interrupt Status Register. A separate flag is provided for each line (PHA, PHB, Index) and one flag is provided when a missing pulse is corrected (a '1' must be written to TC_BMR.AUTOC). If TC_QISR.FPHA=1, a pulse has been filtered on PHA line. If TC_QISR.FPHB=1, a pulse has been filtered on PHB line. If TC_QISR.FIDX=1, a pulse has been filtered on Index line. If TC_QISR.FMP=1, a missing pulse has been corrected by the missing pulse detection logic.

© 2026 Microchip Technology Inc. and its subsidiaries

The on-the-fly detection and associated flags can be used to anticipate further effects of contamination/dust, in particular if the flag is written to '1' when the rotary encoder speed is high. This may cause abnormal behavior when the rotary encoder slows down if the abnormal pulse is no longer filtered (surpassing digital filter capabilities).

This detection circuitry does not prevent rotary encoder integrity check before use.

36.6.17. 2-bit Gray Up/Down Counter for Stepper Motor

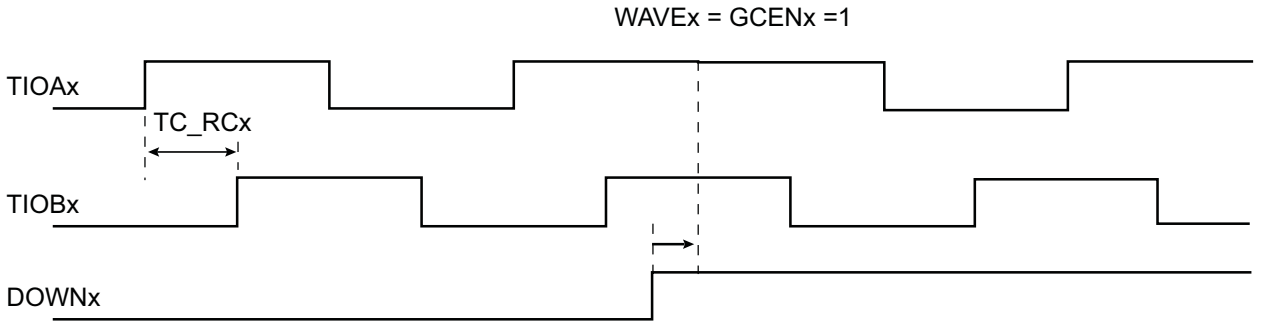
Each channel can be independently configured to generate a 2-bit Gray count waveform on corresponding TIOAx, TIOBx outputs by means of TC_SMMRx.GCEN.

Up or Down count can be defined by writing TC_SMMRx.DOWN.

It is mandatory to configure the channel in Waveform mode in the TC_CMR.

The period of the counters can be programmed in TC_RCx.

Figure 36-23. 2-bit Gray Up/Down Counter



36.6.18. Fault Mode

At any time, the TC_RCx registers can be used to perform a comparison on the respective current channel counter value (TC_CVx) with the value of TC_RCx register.

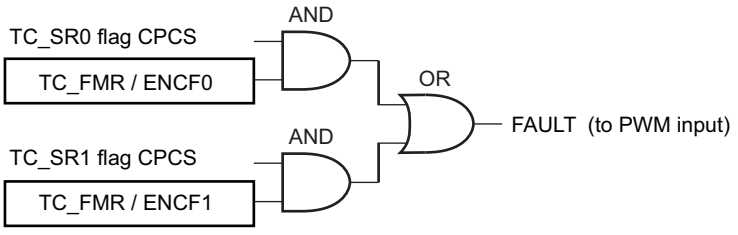
The CPCSx flags can be set accordingly and an interrupt can be generated.

This interrupt is processed but requires an unpredictable amount of time to be achieve the required action.

It is possible to trigger the FAULT output of the TIMER1 with CPCS from TC_SR0 and/or CPCS from TC_SR1. Each source can be independently enabled/disabled in the TC_FMR.

This can be useful to detect an overflow on speed and/or position when QDEC is processed and to act immediately by using the FAULT output.

Figure 36-24. Fault Output Generation



36.6.19. Register Write Protection

To prevent any single software error from corrupting TC behavior, certain registers in the address space can be write-protected by setting the WPEN bit , WPITEN (Write Protection Interrupt Enable),

© 2026 Microchip Technology Inc. and its subsidiaries

and/or WPCREN (Write Protection Control Enable) in the TC Write Protection Mode Register (TC_WPMR).

If a write access to the protected registers is detected, the WPVS flag in the "TC Safety Status Register" (TC_SSRx) is set and the field WPVSRC indicates the register in which the write access has been attempted.

The Timer Counter clock of the first channel must be enabled to access TC_WPMR.

The following registers can be write-protected when WPEN is set:

- TC Block Mode Register
- TC Channel Mode Register Capture Mode
- TC Channel Mode Register Waveform Mode
- TC Fault Mode Register
- TC Stepper Motor Mode Register
- TC Register A
- TC Register B
- TC Register C
- TC Extended Mode Register

The following registers can be write-protected when WPITEN is set:

- TC Interrupt Enable Register
- TC Interrupt Disable Register
- TC QDEC Interrupt Enable Register
- TC QDEC Interrupt Disable Register

The following register can be write-protected when WPCREN is set:

- TC Channel Control Register
- TC Block Control Register

36.6.20. Security and Safety Analysis and Reports

Several type of checks are performed when a TC channel is enabled.

The peripheral clock of the TC is monitored by a specific circuitry to detect abnormal waveforms on the internal clock net that may affect the behavior of the TDES. Corruption on the triggering edge of the clock or a pulse with a minimum duration may be identified. If the flag TC_SSRx.CGD is set, an abnormal condition occurred on the internal clock net. This flag is not set under normal operating conditions.

The internal counter of a TC channel is also monitored and if an abnormal state is detected, the flag TC_SSRx.SEQE is set. This flag cannot be set under normal operating conditions.

The software accesses to the TC are monitored and if an incorrect access is performed, the flag TC_SSRx.SWE is set. The type of incorrect/abnormal software access is reported in TC_SSRx.SWETYP (see TC_CSRx for details). As an example, when the TC channel is configured in Capture mode, reading TC_RAx when TC_SRx.LDRAS=0 asserts the SWE flags. TC_SSR.ECLASS is an indicator reporting the criticality of the SWETYP report.

The flags CGD, SEQE, SWE, WPVS are automatically cleared when TC_SSRx is read.

If one of these flags is set, the flag TC_SRx.SECE is set and triggers an interrupt if the TC_IMRx.SECE bit is ‘1’. SECE is cleared by reading TC_SRx.

© 2026 Microchip Technology Inc. and its subsidiaries

36.7. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	TC_CCR0	31:24								
		23:16								
		15:8								
		7:0						SWTRG	CLKDIS	CLKEN
0x04	TC_CMR0 (CAPTURE MODE)	31:24								
		23:16		SBSMPLR[2:0]			LDRB[1:0]		LDRA[1:0]	
		15:8	WAVE	CPCTRG				ABETRG	ETRGEDG[1:0]	
		7:0	LDBDIS	LDBSTOP	BURST[1:0]		CLKI		TCCLKS[2:0]	
0x04	TC_CMR0 (WAVEFORM MODE)	31:24	BSWTRG[1:0]		BEEVT[1:0]		BCPC[1:0]		BCPB[1:0]	
		23:16	ASWTRG[1:0]		AEEVT[1:0]		ACPC[1:0]		ACPA[1:0]	
		15:8	WAVE	WAVSEL[1:0]		ENETRG	EEVT[1:0]		EEVTEDG[1:0]	
		7:0	CPCDIS	CPCSTOP	BURST[1:0]		CLKI		TCCLKS[2:0]	
0x08	TC_SMMR0	31:24								
		23:16								
		15:8								
		7:0							DOWN	GCEN
0x0C	TC_RAB0	31:24	RAB[31:24]							
		23:16	RAB[23:16]							
		15:8	RAB[15:8]							
		7:0	RAB[7:0]							
0x10	TC_CV0	31:24	CV[31:24]							
		23:16	CV[23:16]							
		15:8	CV[15:8]							
		7:0	CV[7:0]							
0x14	TC_RA0	31:24	RA[31:24]							
		23:16	RA[23:16]							
		15:8	RA[15:8]							
		7:0	RA[7:0]							
0x18	TC_RB0	31:24	RB[31:24]							
		23:16	RB[23:16]							
		15:8	RB[15:8]							
		7:0	RB[7:0]							
0x1C	TC_RC0	31:24	RC[31:24]							
		23:16	RC[23:16]							
		15:8	RC[15:8]							
		7:0	RC[7:0]							
0x20	TC_SR0	31:24								
		23:16						MTIOB	MTIOA	CLKSTA
		15:8						SECE	RXBUFF	ENDRX
		7:0	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS
0x24	TC_IER0	31:24								
		23:16								
		15:8						SECE	RXBUFF	ENDRX
		7:0	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS
0x28	TC_IDR0	31:24								
		23:16								
		15:8						SECE	RXBUFF	ENDRX
		7:0	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS
0x2C	TC_IMR0	31:24								
		23:16								
		15:8						SECE	RXBUFF	ENDRX
		7:0	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS
0x30	TC_EMR0	31:24								
		23:16								
		15:8								NODIVCLK
		7:0			TRIGSRCB[1:0]				TRIGSRCA[1:0]	

© 2026 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x34	TC_CSR0	31:24								
		23:16						MTIOB	MTIOA	CLKSTA
		15:8								
		7:0								
0x38	TC_SSR0	31:24	ECLASS				SWETYP[3:0]			
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0					SWE	SEQE	CGD	WPVS
0x3C ... 0x3F	Reserved									
0x40	TC_CCR1	31:24								
		23:16								
		15:8								
		7:0						SWTRG	CLKDIS	CLKEN
0x44	TC_CMR1 (CAPTURE MODE)	31:24								
		23:16		SBSMPLR[2:0]			LDRB[1:0]		LDRA[1:0]	
		15:8	WAVE	CPCTRG				ABETRG	ETRGEDG[1:0]	
		7:0	LDBDIS	LDBSTOP	BURST[1:0]		CLKI	TCCLKS[2:0]		
0x44	TC_CMR1 (WAVEFORM MODE)	31:24	BSWTRG[1:0]		BEEVT[1:0]		BCPC[1:0]		BCPB[1:0]	
		23:16	ASWTRG[1:0]		AEEVT[1:0]		ACPC[1:0]		ACPA[1:0]	
		15:8	WAVE	WAVSEL[1:0]		ENETRG	EEVT[1:0]		EEVTEDG[1:0]	
		7:0	CPCDIS	CPCSTOP	BURST[1:0]		CLKI	TCCLKS[2:0]		
0x48	TC_SMMR1	31:24								
		23:16								
		15:8								
		7:0							DOWN	GCEN
0x4C	TC_RAB1	31:24	RAB[31:24]							
		23:16	RAB[23:16]							
		15:8	RAB[15:8]							
		7:0	RAB[7:0]							
0x50	TC_CV1	31:24	CV[31:24]							
		23:16	CV[23:16]							
		15:8	CV[15:8]							
		7:0	CV[7:0]							
0x54	TC_RA1	31:24	RA[31:24]							
		23:16	RA[23:16]							
		15:8	RA[15:8]							
		7:0	RA[7:0]							
0x58	TC_RB1	31:24	RB[31:24]							
		23:16	RB[23:16]							
		15:8	RB[15:8]							
		7:0	RB[7:0]							
0x5C	TC_RC1	31:24	RC[31:24]							
		23:16	RC[23:16]							
		15:8	RC[15:8]							
		7:0	RC[7:0]							
0x60	TC_SR1	31:24								
		23:16						MTIOB	MTIOA	CLKSTA
		15:8						SECE	RXBUFF	ENDRX
		7:0	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS
0x64	TC_IER1	31:24								
		23:16								
		15:8						SECE	RXBUFF	ENDRX
		7:0	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS
0x68	TC_IDR1	31:24								
		23:16								
		15:8						SECE	RXBUFF	ENDRX
		7:0	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS

© 2026 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x6C	TC_IMR1	31:24								
		23:16								
		15:8						SECE	RXBUFF	ENDRX
		7:0	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS
0x70	TC_EMR1	31:24								
		23:16								
		15:8								NODIVCLK
		7:0			TRIGSRCB[1:0]				TRIGSRCA[1:0]	
0x74	TC_CSR1	31:24								
		23:16						MTIOB	MTIOA	CLKSTA
		15:8								
		7:0								
0x78	TC_SSR1	31:24	ECLASS				SWETYP[3:0]			
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0					SWE	SEQE	CGD	WPVS
0x7C ... 0x7F	Reserved									
0x80	TC_CCR2	31:24								
		23:16								
		15:8								
		7:0						SWTRG	CLKDIS	CLKEN
0x84	TC_CMR2 (CAPTURE MODE)	31:24								
		23:16		SBSMPLR[2:0]			LDRB[1:0]		LDRA[1:0]	
		15:8	WAVE	CPCTRG				ABETRG	ETRGEDG[1:0]	
		7:0	LDBDIS	LDBSTOP	BURST[1:0]		CLKI	TCCLKS[2:0]		
0x84	TC_CMR2 (WAVEFORM MODE)	31:24	BSWTRG[1:0]		BEEVT[1:0]		BCPC[1:0]		BCPB[1:0]	
		23:16	ASWTRG[1:0]		AEEVT[1:0]		ACPC[1:0]		ACPA[1:0]	
		15:8	WAVE	WAVSEL[1:0]		ENETRG	EEVT[1:0]		EEVTEDG[1:0]	
		7:0	CPCDIS	CPCSTOP	BURST[1:0]		CLKI	TCCLKS[2:0]		
0x88	TC_SMMR2	31:24								
		23:16								
		15:8								
		7:0							DOWN	GCEN
0x8C	TC_RAB2	31:24	RAB[31:24]							
		23:16	RAB[23:16]							
		15:8	RAB[15:8]							
		7:0	RAB[7:0]							
0x90	TC_CV2	31:24	CV[31:24]							
		23:16	CV[23:16]							
		15:8	CV[15:8]							
		7:0	CV[7:0]							
0x94	TC_RA2	31:24	RA[31:24]							
		23:16	RA[23:16]							
		15:8	RA[15:8]							
		7:0	RA[7:0]							
0x98	TC_RB2	31:24	RB[31:24]							
		23:16	RB[23:16]							
		15:8	RB[15:8]							
		7:0	RB[7:0]							
0x9C	TC_RC2	31:24	RC[31:24]							
		23:16	RC[23:16]							
		15:8	RC[15:8]							
		7:0	RC[7:0]							
0xA0	TC_SR2	31:24								
		23:16						MTIOB	MTIOA	CLKSTA
		15:8						SECE	RXBUFF	ENDRX
		7:0	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS

© 2026 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xA4	TC_IER2	31:24								
		23:16								
		15:8						SECE	RXBUFF	ENDRX
		7:0	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS
0xA8	TC_IDR2	31:24								
		23:16								
		15:8						SECE	RXBUFF	ENDRX
		7:0	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS
0xAC	TC_IMR2	31:24								
		23:16								
		15:8						SECE	RXBUFF	ENDRX
		7:0	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS
0xB0	TC_EMR2	31:24								
		23:16								
		15:8								NODIVCLK
		7:0			TRIGSRCB[1:0]				TRIGSRCA[1:0]	
0xB4	TC_CSR2	31:24								
		23:16						MTIOB	MTIOA	CLKSTA
		15:8								
		7:0								
0xB8	TC_SSR2	31:24	ECLASS				SWETYP[3:0]			
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0					SWE	SEQE	CGD	WPVS
0xBC ... 0xBF	Reserved									
0xC0	TC_BCR	31:24								
		23:16								
		15:8								
		7:0								SYNC
0xC4	TC_BMR	31:24			MAXCMP[3:0]				MAXFILT[5:4]	
		23:16	MAXFILT[3:0]					AUTOC	IDXPHB	SWAP
		15:8	INVIDX	INVB	INVA	EDGPHA	QDTRANS	SPEEDEN	POSEN	QDEN
		7:0			TC2XC2S[1:0]		TC1XC1S[1:0]		TC0XC0S[1:0]	
0xC8	TC_QIER	31:24								
		23:16								
		15:8								
		7:0	FMP	FIDX	FPHB	FPHA	MPE	QERR	DIRCHG	IDX
0xCC	TC_QIDR	31:24								
		23:16								
		15:8								
		7:0	FMP	FIDX	FPHB	FPHA	MPE	QERR	DIRCHG	IDX
0xD0	TC_QIMR	31:24								
		23:16								
		15:8								
		7:0	FMP	FIDX	FPHB	FPHA	MPE	QERR	DIRCHG	IDX
0xD4	TC_QISR	31:24								
		23:16								
		15:8								DIR
		7:0	FMP	FIDX	FPHB	FPHA	MPE	QERR	DIRCHG	IDX
0xD8	TC_FMR	31:24								
		23:16								
		15:8								
		7:0							ENCF1	ENCF0
0xDC	TC_QSR	31:24								
		23:16								
		15:8								DIR
		7:0								

© 2026 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xE0 ... 0xE3	Reserved									
0xE4	TC_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0				FIRSTE		WPCREN	WPITEN	WPEN

© 2026 Microchip Technology Inc. and its subsidiaries

36.7.1. TC Channel Control Register

Name: TC_CCRx
Offset: 0x00 + x*0x40 [x=0..2]
Reset: –
Property: Write-only

This register can only be written if the WPCREN bit is cleared in the TC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						SWTRG	CLKDIS	CLKEN
Access						W	W	W
Reset						–	–	–

Bit 2 – SWTRG Software Trigger Command

Value	Description
0	No effect.
1	A software trigger is performed: the counter is reset and the clock is started.

Bit 1 – CLKDIS Counter Clock Disable Command

Value	Description
0	No effect.
1	Disables the clock.

Bit 0 – CLKEN Counter Clock Enable Command

Value	Description
0	No effect.
1	Enables the clock if CLKDIS is not 1.

© 2026 Microchip Technology Inc. and its subsidiaries

36.7.2. TC Channel Mode Register: Capture Mode

Name: TC_CMRx (CAPTURE MODE)
Offset: 0x04 + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read/Write

This register can be written only if the WPEN bit is cleared in the TC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
		SBSMPLR[2:0]			LDRB[1:0]		LDRA[1:0]	
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WAVE	CPCTRG				ABETRG	ETRGEDG[1:0]	
Access	R/W	R/W				R/W	R/W	R/W
Reset	0	0				0	0	0

Bit	7	6	5	4	3	2	1	0
	LDBDIS	LDBSTOP	BURST[1:0]		CLKI	TCCLKS[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 22:20 – SBSMPLR[2:0] Loading Edge Subsampling Ratio

Value	Name	Description
0	ONE	Load a Capture register each selected edge.
1	HALF	Load a Capture register every 2 selected edges.
2	FOURTH	Load a Capture register every 4 selected edges.
3	EIGHTH	Load a Capture register every 8 selected edges.
4	SIXTEENTH	Load a Capture register every 16 selected edges.

Bits 19:18 – LDRB[1:0] RB Loading Edge Selection

Value	Name	Description
0	NONE	None
1	RISING	Rising edge of TIOAx
2	FALLING	Falling edge of TIOAx
3	EDGE	Each edge of TIOAx

Bits 17:16 – LDRA[1:0] RA Loading Edge Selection

Value	Name	Description
0	NONE	None
1	RISING	Rising edge of TIOAx
2	FALLING	Falling edge of TIOAx
3	EDGE	Each edge of TIOAx

Bit 15 – WAVE Waveform Mode

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	Capture mode is enabled.
1	Capture mode is disabled (Waveform mode is enabled).

Bit 14 – CPCTRG RC Compare Trigger Enable

Value	Description
0	RC Compare has no effect on the counter and its clock.
1	RC Compare resets the counter and starts the counter clock.

Bit 10 – ABETRG TIOAx or TIOBx External Trigger Selection

Value	Description
0	TIOBx is used as an external trigger.
1	TIOAx is used as an external trigger.

Bits 9:8 – ETRGEDG[1:0] External Trigger Edge Selection

Value	Name	Description
0	NONE	The clock is not gated by an external signal.
1	RISING	Rising edge
2	FALLING	Falling edge
3	EDGE	Each edge

Bit 7 – LDBDIS Counter Clock Disable with RB Loading

Value	Description
0	Counter clock is not disabled when RB loading occurs.
1	Counter clock is disabled when RB loading occurs.

Bit 6 – LDBSTOP Counter Clock Stopped with RB Loading

Value	Description
0	Counter clock is not stopped when RB loading occurs.
1	Counter clock is stopped when RB loading occurs.

Bits 5:4 – BURST[1:0] Burst Signal Selection

Value	Name	Description
0	NONE	The clock is not gated by an external signal.
1	XC0	XC0 is ANDed with the selected clock.
2	XC1	XC1 is ANDed with the selected clock.
3	XC2	XC2 is ANDed with the selected clock.

Bit 3 – CLKI Clock Invert

Value	Description
0	Counter is incremented on rising edge of the clock.
1	Counter is incremented on falling edge of the clock.

Bits 2:0 – TCCLKS[2:0] Clock Selection
To operate at maximum peripheral clock frequency, see TC_EMRx.

Value	Name	Description
0	TIMER_CLOCK1	Clock selected: internal GCLK clock signal (from PMC)
1	TIMER_CLOCK2	Clock selected: internal MCK/8 clock signal (from PMC)
2	TIMER_CLOCK3	Clock selected: internal MCK/32 clock signal (from PMC)
3	TIMER_CLOCK4	Clock selected: internal MCK/128 clock signal (from PMC)

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Name	Description
4	TIMER_CLOCK5	Clock selected: internal MD_SLCK clock signal (from PMC)
5	XC0	Clock selected: XC0
6	XC1	Clock selected: XC1
7	XC2	Clock selected: XC2

© 2026 Microchip Technology Inc. and its subsidiaries

36.7.3. TC Channel Mode Register: Waveform Mode

Name: TC_CMRx (WAVEFORM MODE)
Offset: 0x04 + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the TC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	BSWTRG[1:0]		BEEVT[1:0]		BCPC[1:0]		BCPB[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ASWTRG[1:0]		AEEVT[1:0]		ACPC[1:0]		ACPA[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WAVE	WAVSEL[1:0]		ENETRG	EEVT[1:0]		EEVTEDG[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CPCDIS	CPCSTOP	BURST[1:0]		CLKI	TCCLKS[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:30 – BSWTRG[1:0] Software Trigger Effect on TIOBx

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

Bits 29:28 – BEEVT[1:0] External Event Effect on TIOBx

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

Bits 27:26 – BCPC[1:0] RC Compare Effect on TIOBx

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

Bits 25:24 – BCPB[1:0] RB Compare Effect on TIOBx

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

Bits 23:22 – ASWTRG[1:0] Software Trigger Effect on TIOAx

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

Bits 21:20 – AEEVT[1:0] External Event Effect on TIOAx

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

Bits 19:18 – ACPC[1:0] RC Compare Effect on TIOAx

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

Bits 17:16 – ACPA[1:0] RA Compare Effect on TIOAx

Value	Name	Description
0	NONE	None
1	SET	Set
2	CLEAR	Clear
3	TOGGLE	Toggle

Bit 15 – WAVE Waveform Mode

Value	Description
0	Waveform mode is disabled (Capture mode is enabled).
1	Waveform mode is enabled.

Bits 14:13 – WAVSEL[1:0] Waveform Selection

Value	Name	Description
0	UP	UP mode without automatic trigger on RC Compare
1	UPDOWN	UPDOWN mode without automatic trigger on RC Compare
2	UP_RC	UP mode with automatic trigger on RC Compare
3	UPDOWN_RC	UPDOWN mode with automatic trigger on RC Compare

Bit 12 – ENETRG External Event Trigger Enable

Whatever the value programmed in ENETRG, the selected external event only controls the TIOAx output and TIOBx if not used as input (trigger event input or other input used).

Value	Description
0	The external event has no effect on the counter and its clock.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
1	The external event resets the counter and starts the counter clock.

Bits 11:10 – EEVT[1:0] External Event Selection
Signal selected as external event.

Value	Name	Description	TIOB Direction
0	TIOB	TIOB	Input
1	XC0	XC0	Output
2	XC1	XC1	Output
3	XC2	XC2	Output

Note: If TIOB is chosen as the external event signal, it is configured as an input and no longer generates waveforms and subsequently no IRQs.

Bits 9:8 – EEVTEDG[1:0] External Event Edge Selection

Value	Name	Description
0	NONE	None
1	RISING	Rising edge
2	FALLING	Falling edge
3	EDGE	Each edge

Bit 7 – CPCDIS Counter Clock Disable with RC Compare

Value	Description
0	Counter clock is not disabled when counter reaches RC.
1	Counter clock is disabled when counter reaches RC.

Bit 6 – CPCSTOP Counter Clock Stopped with RC Compare

Value	Description
0	Counter clock is not stopped when counter reaches RC.
1	Counter clock is stopped when counter reaches RC.

Bits 5:4 – BURST[1:0] Burst Signal Selection

Value	Name	Description
0	NONE	The clock is not gated by an external signal.
1	XC0	XC0 is ANDed with the selected clock.
2	XC1	XC1 is ANDed with the selected clock.
3	XC2	XC2 is ANDed with the selected clock.

Bit 3 – CLKI Clock Invert

Value	Description
0	Counter is incremented on rising edge of the clock.
1	Counter is incremented on falling edge of the clock.

Bits 2:0 – TCCLKS[2:0] Clock Selection
To operate at maximum peripheral clock frequency, see TC_EMRx.

Value	Name	Description
0	TIMER_CLOCK1	Clock selected: internal GCLK clock signal (from PMC)
1	TIMER_CLOCK2	Clock selected: internal MCK/8 clock signal (from PMC)
2	TIMER_CLOCK3	Clock selected: internal MCK/32 clock signal (from PMC)
3	TIMER_CLOCK4	Clock selected: internal MCK/128 clock signal (from PMC)
4	TIMER_CLOCK5	Clock selected: internal MD_SLCK clock signal (from PMC)

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Name	Description
5	XC0	Clock selected: XC0
6	XC1	Clock selected: XC1
7	XC2	Clock selected: XC2

© 2026 Microchip Technology Inc. and its subsidiaries

36.7.4. TC Stepper Motor Mode Register

Name: TC_SMMRx
Offset: 0x08 + x*0x40 [x=0..2]
Reset: 0x00000000
Property: R/W

This register can only be written if the WPEN bit is cleared in the TC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							DOWN	GCEN
Access							R/W	R/W
Reset							0	0

Bit 1 – DOWN Down Count

Value	Description
0	Up counter.
1	Down counter.

Bit 0 – GCEN Gray Count Enable

Value	Description
0	TIOAx [x=0..2] and TIOBx [x=0..2] are driven by internal counter of channel x.
1	TIOAx [x=0..2] and TIOBx [x=0..2] are driven by a 2-bit Gray counter.

© 2026 Microchip Technology Inc. and its subsidiaries

36.7.5. TC Register AB

Name: TC_RABx
Offset: 0x0C + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	RAB[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RAB[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RAB[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RAB[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – RAB[31:0] Register A or Register B
RAB contains the next unread capture Register A or Register B value in real time. It is usually read by the DMA after a request due to a valid load edge on TIOAx.
When DMA is used, the RAB register address must be configured as source address of the transfer.

© 2026 Microchip Technology Inc. and its subsidiaries

36.7.6. TC Counter Value Register

Name: TC_CVx
Offset: 0x10 + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	CV[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CV[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CV[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CV[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CV[31:0] Counter Value
CV contains the counter value in real time.

Important:
For 16-bit channels, the CV field size is limited to register bits 15:0.

© 2026 Microchip Technology Inc. and its subsidiaries

36.7.7. TC Register A

Name: TC_RAx
Offset: 0x14 + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read/Write

This register has access Read-only if TC_CMRx.WAVE = 0, Read/Write if TC_CMRx.WAVE = 1.
This register can only be written if the WPEN bit is cleared in the TC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	RA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – RA[31:0] Register A
RA contains the Register A value in real time.

© 2026 Microchip Technology Inc. and its subsidiaries

36.7.8. TC Register B

Name: TC_RBx
Offset: 0x18 + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read/Write

This register has access Read-only if TC_CMRx.WAVE = 0, Read/Write if TC_CMRx.WAVE = 1.
This register can only be written if the WPEN bit is cleared in the TC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	RB[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RB[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RB[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RB[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – RB[31:0] Register B
RB contains the Register B value in real time.

© 2026 Microchip Technology Inc. and its subsidiaries

36.7.9. TC Register C

Name: TC_RCx
Offset: 0x1C + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the TC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	RC[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RC[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RC[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RC[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – RC[31:0] Register C
RC contains the Register C value in real time.

© 2026 Microchip Technology Inc. and its subsidiaries

36.7.10. TC Interrupt Status Register

Name: TC_SRx
Offset: 0x20 + x*0x40 [x=0..2]
Reset: 0x00000300
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						MTIOB	MTIOA	CLKSTA
Access						R	R	R
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
						SECE	RXBUFF	ENDRX
Access						R	R	R
Reset						0	1	1

Bit	7	6	5	4	3	2	1	0
	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 18 – MTIOB TIOBx Mirror

Value	Description
0	TIOBx is low. If TC_CMRx.WAVE = 0, TIOBx pin is low. If TC_CMRx.WAVE = 1, TIOBx is driven low.
1	TIOBx is high. If TC_CMRx.WAVE = 0, TIOBx pin is high. If TC_CMRx.WAVE = 1, TIOBx is driven high.

Bit 17 – MTIOA TIOAx Mirror

Value	Description
0	TIOAx is low. If TC_CMRx.WAVE = 0, TIOAx pin is low. If TC_CMRx.WAVE = 1, TIOAx is driven low.
1	TIOAx is high. If TC_CMRx.WAVE = 0, TIOAx pin is high. If TC_CMRx.WAVE = 1, TIOAx is driven high.

Bit 16 – CLKSTA Clock Enabling Status

Value	Description
0	The clock is disabled.
1	The clock is enabled.

Bit 10 – SECE Security and/or Safety Event (cleared on read)

Value	Description
0	No security or safety event occurred.
1	One or more safety or security event occurred since the last read of TC_SRx. For details on the event, see TC_SSRx.

Bit 9 – RXBUFF Reception Buffer Full (cleared by writing TC_RCR or TC_RNCR)
Note: TC_RCR and TC_RNCR are PDC registers.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	TC_RCR or TC_RNCR have a value other than 0.
1	Both TC_RCR and TC_RNCR have a value of 0.

Bit 8 – ENDRX End of Receiver Transfer (cleared by writing TC_RCR or TC_RNCR)
Note: TC_RCR and TC_RNCR are PDC registers.

Value	Description
0	The Receive Counter Register has not reached 0 since the last write in TC_RCR or TC_RNCR.
1	The Receive Counter Register has reached 0 since the last write in TC_RCR or TC_RNCR.

Bit 7 – ETRGS External Trigger Status (cleared on read)

Value	Description
0	External trigger has not occurred since the last read of the Status register.
1	External trigger has occurred since the last read of the Status register.

Bit 6 – LDRBS RB Loading Status (cleared on read)

Value	Description
0	RB Load has not occurred since the last read of the Status register or TC_CMRx.WAVE = 1.
1	RB Load has occurred since the last read of the Status register, if TC_CMRx.WAVE = 0.

Bit 5 – LDRAS RA Loading Status (cleared on read)

Value	Description
0	RA Load has not occurred since the last read of the Status register or TC_CMRx.WAVE = 1.
1	RA Load has occurred since the last read of the Status register, if TC_CMRx.WAVE = 0.

Bit 4 – CPCS RC Compare Status (cleared on read)

Value	Description
0	RC Compare has not occurred since the last read of the Status register.
1	RC Compare has occurred since the last read of the Status register.

Bit 3 – CPBS RB Compare Status (cleared on read)

Value	Description
0	RB Compare has not occurred since the last read of the Status register or TC_CMRx.WAVE = 0.
1	RB Compare has occurred since the last read of the Status register, if TC_CMRx.WAVE = 1.

Bit 2 – CPAS RA Compare Status (cleared on read)

Value	Description
0	RA Compare has not occurred since the last read of the Status register or TC_CMRx.WAVE = 0.
1	RA Compare has occurred since the last read of the Status register, if TC_CMRx.WAVE = 1.

Bit 1 – LOVRS Load Overrun Status (cleared on read)

Value	Description
0	Load overrun has not occurred since the last read of the Status register or TC_CMRx.WAVE = 1.
1	RA or RB have been loaded at least twice without any read of the corresponding register since the last read of the Status register, if TC_CMRx.WAVE = 0.

Bit 0 – COVFS Counter Overflow Status (cleared on read)

Value	Description
0	No counter overflow has occurred since the last read of the Status register.
1	A counter overflow has occurred since the last read of the Status register.

© 2026 Microchip Technology Inc. and its subsidiaries

36.7.11. TC Interrupt Enable Register

Name: TC_IERx
Offset: 0x24 + x*0x40 [x=0..2]
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the TC Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						SECE	RXBUFF	ENDRX
Access						W	W	W
Reset						–	–	–

Bit	7	6	5	4	3	2	1	0
	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 10 – SECE Security and/or Safety Event Interrupt Enable

Bit 9 – RXBUFF Reception Buffer Full

Bit 8 – ENDRX End of Receiver Transfer

Bit 7 – ETRGS External Trigger

Bit 6 – LDRBS RB Loading

Bit 5 – LDRAS RA Loading

Bit 4 – CPCS RC Compare

Bit 3 – CPBS RB Compare

Bit 2 – CPAS RA Compare

Bit 1 – LOVRS Load Overrun

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 0 – COVFS Counter Overflow

© 2026 Microchip Technology Inc. and its subsidiaries

36.7.12. TC Interrupt Disable Register

Name: TC_IDRx
Offset: 0x28 + x*0x40 [x=0..2]
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the TC Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						SECE	RXBUFF	ENDRX
Access						W	W	W
Reset						–	–	–

Bit	7	6	5	4	3	2	1	0
	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 10 – SECE Security and/or Safety Event Interrupt Disable

Bit 9 – RXBUFF Reception Buffer Full

Bit 8 – ENDRX End of Receiver Transfer

Bit 7 – ETRGS External Trigger

Bit 6 – LDRBS RB Loading

Bit 5 – LDRAS RA Loading

Bit 4 – CPCS RC Compare

Bit 3 – CPBS RB Compare

Bit 2 – CPAS RA Compare

Bit 1 – LOVRS Load Overrun

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 0 – COVFS Counter Overflow

© 2026 Microchip Technology Inc. and its subsidiaries

36.7.13. TC Interrupt Mask Register

Name: TC_IMRx
Offset: 0x2C + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						SECE	RXBUFF	ENDRX
Access						R	R	R
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 10 – SECE Security and/or Safety Event Interrupt Mask

Bit 9 – RXBUFF Reception Buffer Full

Bit 8 – ENDRX End of Receiver Transfer

Bit 7 – ETRGS External Trigger

Bit 6 – LDRBS RB Loading

Bit 5 – LDRAS RA Loading

Bit 4 – CPCS RC Compare

Bit 3 – CPBS RB Compare

Bit 2 – CPAS RA Compare

Bit 1 – LOVRS Load Overrun

Bit 0 – COVFS Counter Overflow

© 2026 Microchip Technology Inc. and its subsidiaries

36.7.14. TC Extended Mode Register

Name: TC_EMRx
Offset: 0x30 + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the TC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
								NODIVCLK
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
			TRIGSRCB[1:0]				TRIGSRCA[1:0]	
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit 8 – NODIVCLK No Divided Clock

Value	Description
0	The selected clock is defined by field TCCLKS in TC_CMRx.
1	The selected clock is peripheral clock and TCCLKS field (TC_CMRx) has no effect.

Bits 5:4 – TRIGSRCB[1:0] Trigger Source for Input B

Value	Name	Description
0	EXTERNAL_TIOBx	The trigger/capture input B is driven by external pin TIOBx
1	PWMx	For all channels: The trigger/capture input B is driven internally by the comparator output (see Synchronization with PWM) of the PWMx.

Bits 1:0 – TRIGSRCA[1:0] Trigger Source for Input A

Value	Name	Description
0	EXTERNAL_TIOAx	The trigger/capture input A is driven by external pin TIOAx
1	PWMx	The trigger/capture input A is driven internally by PWMx.

© 2026 Microchip Technology Inc. and its subsidiaries

36.7.15. TC Channel Status Register

Name: TC_CSRx
Offset: 0x34 + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read-only

Note: The flags in this register are a copy of the similar flags in the TC_SRx register. Reading the TC_CSRx does not perform a clear-on-read of TC_SRx flags.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						MTIOB	MTIOA	CLKSTA
Access						R	R	R
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 18 – MTIOB TIOBx Mirror

Value	Description
0	TIOBx is low. If TC_CMRx.WAVE = 0, TIOBx is low. If TC_CMRx.WAVE = 1, TIOBx is driven low.
1	TIOBx is high. If TC_CMRx.WAVE = 0, TIOBx is high. If TC_CMRx.WAVE = 1, TIOBx is driven high.

Bit 17 – MTIOA TIOAx Mirror

Value	Description
0	TIOAx is low. If TC_CMRx.WAVE = 0, TIOAx is low. If TC_CMRx.WAVE = 1, TIOAx is driven low.
1	TIOAx is high. If TC_CMRx.WAVE = 0, TIOAx is high. If TC_CMRx.WAVE = 1, TIOAx is driven high.

Bit 16 – CLKSTA Clock Enabling Status

Value	Description
0	Clock is disabled.
1	Clock is enabled.

© 2026 Microchip Technology Inc. and its subsidiaries

36.7.16. TC Safety Status Register

Name: TC_SSRx
Offset: 0x38 + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	ECLASS				SWETYP[3:0]			
Access	R				R	R	R	R
Reset	0				0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPVSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
					SWE	SEQE	CGD	WPVS
Access					R	R	R	R
Reset					0	0	0	0

Bit 31 – ECLASS Software Error Class

Value	Name	Description
0	WARNING	An abnormal access that does not have any impact.
1	ERROR	An abnormal access that may have an impact.

Bits 27:24 – SWETYP[3:0] Software Error Type (cleared on read)

Value	Name	Description
0	READ_WO	TC Channel x is enabled and a write-only register has been read (Warning).
1	WRITE_RO	TC Channel x is enabled and a write access has been performed on a read-only register (Warning).
2	UNDEF_RW	Access to an undefined address of the TC (Warning).
3	W_RARB_CAPT	TC_RAx or TC_RBx are written while channel is enabled and configured in capture mode (Error).

Bits 23:8 – WPVSRC[15:0] Write Protection Violation Source (cleared on read)
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 3 – SWE Software Control Error (cleared on read)

Value	Description
0	No software error has occurred since the last read of TC_SSRx.
1	A software error has occurred since the last read of TC_SSRx. The field SWETYP details the type of software error encountered.

Bit 2 – SEQE Internal Sequencer Error (cleared on read)

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	No internal counter error has occurred since the last read of TC_SSRx. In normal operating conditions, SEQE is cleared.
1	An internal counter error has occurred since the last read of TC_SSRx. This flag can be set only under abnormal operating conditions resulting in clock glitch, etc. The detection is enabled if TC_CSRx.CLKSTA=1, TC_CMRx.WAVE=1, TC_CMRx.CPCTRG=1 and flag is set if TC_CVx > TC_RCx.

Bit 1 – CGD Clock Glitch Detected (cleared on read)

Value	Description
0	The clock monitoring has not been corrupted since the last read of TC_SSRx.
1	The clock monitoring has been corrupted since the last read of TC_SSRx. This flag can be set under abnormal operating conditions.

Bit 0 – WPVS Write Protection Violation Status (cleared on read)

Value	Description
0	No write protection violation has occurred since the last read of TC_SSRx.
1	A write protection violation has occurred since the last read of TC_SSRx. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

© 2026 Microchip Technology Inc. and its subsidiaries

36.7.17. TC Block Control Register

Name: TC_BCR
Offset: 0xC0
Reset: –
Property: Write-only

This register can only be written if the WPCREN bit is cleared in the TC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								SYNC
Access								W
Reset								–

Bit 0 – SYNC Synchro Command

Value	Description
0	No effect.
1	Asserts the SYNC signal which generates a software trigger simultaneously for each of the channels.

© 2026 Microchip Technology Inc. and its subsidiaries

36.7.18. TC Block Mode Register

Name: TC_BMR
Offset: 0xC4
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the TC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
			MAXCMP[3:0]				MAXFILT[5:4]	
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	MAXFILT[3:0]					AUTOC	IDXPHB	SWAP
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bit	15	14	13	12	11	10	9	8
	INVIDX	INVB	INVA	EDGPHA	QDTRANS	SPEEDEN	POSEN	QDEN
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
			TC2XC2S[1:0]		TC1XC1S[1:0]		TC0XC0S[1:0]	
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 29:26 – MAXCMP[3:0] Maximum Consecutive Missing Pulses

Value	Description
0	The flag MPE in TC_QISR never rises.
1–15	Defines the number of consecutive missing pulses before a flag report.

Bits 25:20 – MAXFILT[5:0] Maximum Filter
Pulses with a period shorter than MAXFILT+1 peripheral clock cycles are discarded. For more details on MAXFILT constraints, see Input Preprocessing.

Value	Description
1–63	Defines the filtering capabilities.

Bit 18 – AUTOC AutoCorrection of missing pulses
0 (DISABLED): The detection and autocorrection function is disabled.
1 (ENABLED): The detection and autocorrection function is enabled.

Bit 17 – IDXPHB Index Pin is PHB Pin

Value	Description
0	IDX pin of the rotary sensor must drive TIOA1.
1	IDX pin of the rotary sensor must drive TIOB0.

Bit 16 – SWAP Swap PHA and PHB

Value	Description
0	No swap between PHA and PHB.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
1	Swap PHA and PHB internally, prior to driving the QDEC.

Bit 15 – INVIDX Inverted Index

Value	Description
0	IDX (TIOA1) is directly driving the QDEC.
1	IDX is inverted before driving the QDEC.

Bit 14 – INVB Inverted PHB

Value	Description
0	PHB (TIOB0) is directly driving the QDEC.
1	PHB is inverted before driving the QDEC.

Bit 13 – INVA Inverted PHA

Value	Description
0	PHA (TIOA0) is directly driving the QDEC.
1	PHA is inverted before driving the QDEC.

Bit 12 – EDGPHA Edge on PHA Count Mode

Value	Description
0	Edges are detected on PHA only.
1	Edges are detected on both PHA and PHB.

Bit 11 – QDTRANS Quadrature Decoding Transparent

Value	Description
0	Full quadrature decoding logic is active (direction change detected).
1	Quadrature decoding logic is inactive (direction change inactive) but input filtering and edge detection are performed.

Bit 10 – SPEEDEN Speed Enabled

Value	Description
0	Disabled.
1	Enables the speed measure on channel 0, the time base being provided by channel 2.

Bit 9 – POSEN Position Enabled

Value	Description
0	Disable position.
1	Enables the position measure on channel 0 and 1.

Bit 8 – QDEN Quadrature Decoder Enabled

Quadrature decoding (direction change) can be disabled using QDTRANS bit.
One of the POSEN or SPEEDEN bits must be also enabled.

Value	Description
0	Disabled.
1	Enables the QDEC (filter, edge detection and quadrature decoding).

Bits 5:4 – TC2XC2S[1:0] External Clock Signal 2 Selection

Value	Name	Description
0	TCLK2	Signal connected to XC2: TCLK2
1	–	Reserved

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Name	Description
2	TIOA0	Signal connected to XC2: TIOA0
3	TIOA1	Signal connected to XC2: TIOA1

Bits 3:2 – TC1XC1S[1:0] External Clock Signal 1 Selection

Value	Name	Description
0	TCLK1	Signal connected to XC1: TCLK1
1	–	Reserved
2	TIOA0	Signal connected to XC1: TIOA0
3	TIOA2	Signal connected to XC1: TIOA2

Bits 1:0 – TC0XC0S[1:0] External Clock Signal 0 Selection

Value	Name	Description
0	TCLK0	Signal connected to XC0: TCLK0
1	–	Reserved
2	TIOA1	Signal connected to XC0: TIOA1
3	TIOA2	Signal connected to XC0: TIOA2

© 2026 Microchip Technology Inc. and its subsidiaries

36.7.19. TC QDEC Interrupt Enable Register

Name: TC_QIER
Offset: 0xC8
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the TC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	FMP	FIDX	FPHB	FPHA	MPE	QERR	DIRCHG	IDX
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 7 – FMP Filtered Missing Pulse

Value	Description
0	No effect.
1	Enables the interrupt when phase A or phase B has a corrected missing pulse.

Bit 6 – FIDX Filtered Index Line

Value	Description
0	No effect.
1	Enables the interrupt when index line has a filtered contamination.

Bit 5 – FPHB Filtered Phase B Line

Value	Description
0	No effect.
1	Enables the interrupt when phase B line has a filtered contamination.

Bit 4 – FPHA Filtered Phase A Line

Value	Description
0	No effect.
1	Enables the interrupt when phase A line has a filtered contamination.

Bit 3 – MPE Consecutive Missing Pulse Error

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	No effect.
1	Enables the interrupt when an occurrence of MAXCMP consecutive missing pulses is detected.

Bit 2 – QERR Quadrature Error

Value	Description
0	No effect.
1	Enables the interrupt when a quadrature error occurs on PHA, PHB.

Bit 1 – DIRCHG Direction Change

Value	Description
0	No effect.
1	Enables the interrupt when a change on rotation direction is detected.

Bit 0 – IDX Index

Value	Description
0	No effect.
1	Enables the interrupt when a rising edge occurs on IDX input.

© 2026 Microchip Technology Inc. and its subsidiaries

36.7.20. TC QDEC Interrupt Disable Register

Name: TC_QIDR
Offset: 0xCC
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the TC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	FMP	FIDX	FPHB	FPHA	MPE	QERR	DIRCHG	IDX
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 7 – FMP Filtered Missing Pulse

Value	Description
0	No effect.
1	Disables the interrupt when phase A or phase B has a corrected missing pulse.

Bit 6 – FIDX Filtered Index Line

Value	Description
0	No effect.
1	Disables the interrupt when index line has a filtered contamination.

Bit 5 – FPHB Filtered Phase B Line

Value	Description
0	No effect.
1	Disables the interrupt when phase B line has a filtered contamination.

Bit 4 – FPHA Filtered Phase A Line

Value	Description
0	No effect.
1	Disables the interrupt when phase A line has a filtered contamination.

Bit 3 – MPE Consecutive Missing Pulse Error

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	No effect.
1	Disables the interrupt when an occurrence of MAXCMP consecutive missing pulses has been detected.

Bit 2 – QERR Quadrature Error

Value	Description
0	No effect.
1	Disables the interrupt when a quadrature error occurs on PHA, PHB.

Bit 1 – DIRCHG Direction Change

Value	Description
0	No effect.
1	Disables the interrupt when a change on rotation direction is detected.

Bit 0 – IDX Index

Value	Description
0	No effect.
1	Disables the interrupt when a rising edge occurs on IDX input.

© 2026 Microchip Technology Inc. and its subsidiaries

36.7.21. TC QDEC Interrupt Mask Register

Name: TC_QIMR
Offset: 0xD0
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	FMP	FIDX	FPHB	FPHA	MPE	QERR	DIRCHG	IDX
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 7 – FMP Filtered Missing Pulse

Value	Description
0	The interrupt on auto-corrected missing pulse is disabled.
1	The interrupt on auto-corrected missing pulse is enabled.

Bit 6 – FIDX Filtered Index Line

Value	Description
0	The interrupt on index line filtered contamination is disabled.
1	The interrupt on index line filtered contamination is enabled.

Bit 5 – FPHB Filtered Phase B Line

Value	Description
0	The interrupt on phase B line filtered contamination is disabled.
1	The interrupt on phase B line filtered contamination is enabled.

Bit 4 – FPHA Filtered Phase A Line

Value	Description
0	The interrupt on phase A line filtered contamination is disabled.
1	The interrupt on phase A line filtered contamination is enabled.

Bit 3 – MPE Consecutive Missing Pulse Error

Value	Description
0	The interrupt on the maximum number of consecutive missing pulses specified in MAXCMP is disabled.
1	The interrupt on the maximum number of consecutive missing pulses specified in MAXCMP is enabled.

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 2 – QERR Quadrature Error

Value	Description
0	The interrupt on quadrature error is disabled.
1	The interrupt on quadrature error is enabled.

Bit 1 – DIRCHG Direction Change

Value	Description
0	The interrupt on rotation direction change is disabled.
1	The interrupt on rotation direction change is enabled.

Bit 0 – IDX Index

Value	Description
0	The interrupt on IDX input is disabled.
1	The interrupt on IDX input is enabled.

© 2026 Microchip Technology Inc. and its subsidiaries

36.7.22. TC QDEC Interrupt Status Register

Name: TC_QISR
Offset: 0xD4
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
								DIR
Access								R
Reset								0

Bit	7	6	5	4	3	2	1	0
	FMP	FIDX	FPHB	FPHA	MPE	QERR	DIRCHG	IDX
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 8 – DIR Direction
Returns an image of the current rotation direction.

Bit 7 – FMP Filtered Missing Pulse

Value	Description
0	No correction of missing pulse on phase A or B lines occurred since the last read of TC_QISR.
1	A correction of missing pulse on phase A or B lines occurred since the last read of TC_QISR.

Bit 6 – FIDX Filtered Index Line

Value	Description
0	No filtered contamination on index line since the last read of TC_QISR.
1	A contamination has been successfully on index line since the last read of TC_QISR.

Bit 5 – FPHB Filtered Phase B Line

Value	Description
0	No filtered contamination on phase B line since the last read of TC_QISR.
1	A contamination has been successfully on phase B line since the last read of TC_QISR.

Bit 4 – FPHA Filtered Phase A Line

Value	Description
0	No filtered contamination on phase A line since the last read of TC_QISR.
1	A contamination has been successfully on phase A line since the last read of TC_QISR.

Bit 3 – MPE Consecutive Missing Pulse Error

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	The number of consecutive missing pulses has not reached the maximum value specified in MAXCMP since the last read of TC_QISR.
1	An occurrence of MAXCMP consecutive missing pulses has been detected since the last read of TC_QISR.

Bit 2 – QERR Quadrature Error

Value	Description
0	No quadrature error since the last read of TC_QISR.
1	A quadrature error occurred since the last read of TC_QISR.

Bit 1 – DIRCHG Direction Change

Value	Description
0	No change on rotation direction since the last read of TC_QISR.
1	The rotation direction changed since the last read of TC_QISR.

Bit 0 – IDX Index

Value	Description
0	No Index input change since the last read of TC_QISR.
1	The IDX input has changed since the last read of TC_QISR.

© 2026 Microchip Technology Inc. and its subsidiaries

36.7.23. TC Fault Mode Register

Name: TC_FMR
Offset: 0xD8
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the TC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							ENCF1	ENCF0
Access							R/W	R/W
Reset							0	0

Bit 1 – ENCF1 Enable Compare Fault Channel 1

Value	Description
0	Disables the FAULT output source (CPCS flag) from channel 1.
1	Enables the FAULT output source (CPCS flag) from channel 1.

Bit 0 – ENCF0 Enable Compare Fault Channel 0

Value	Description
0	Disables the FAULT output source (CPCS flag) from channel 0.
1	Enables the FAULT output source (CPCS flag) from channel 0.

© 2026 Microchip Technology Inc. and its subsidiaries

36.7.24. TC QDEC Status Register

Name: TC_QSR
Offset: 0xDC
Reset: 0x00000000
Property: Read-only

Note: The flag in this register is a copy of the similar flag in the TC_QISR register. Reading the TC_QSR does not perform a clear-on-read of TC_QISR flags.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								R
Reset								0

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 8 – DIR Direction
Returns an image of the current rotation direction.

© 2026 Microchip Technology Inc. and its subsidiaries

36.7.25. TC Write Protection Mode Register

Name: TC_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
				FIRSTE		WPCREN	WPITEN	WPEN
Access				R/W		R/W	R/W	R/W
Reset				0		0	0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x54494D	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0.

Bit 4 – FIRSTE First Error Report Enable

Value	Description
0	The last write protection violation source is reported in TC_SSRx.WPVSRC and the last software control error type is reported in TC_SSRx.SWETYP. The TC_SRx.SECE flag is set at the first error occurrence within a series.
1	Only the first write protection violation source is reported in TC_SSRx.WPVSRC and only the first software control error type is reported in TC_SSRx.SWETYP. The TC_SRx.SECE flag is set at the first error occurrence within a series.

Bit 2 – WPCREN Write Protection Control Enable

Value	Description
0	Disables the write protection on control register if WPKEY corresponds to 0x54494D (“TIM” in ASCII).
1	Enables the write protection on control register if WPKEY corresponds to 0x54494D (“TIM” in ASCII).

Bit 1 – WPITEN Write Protection Interrupt Enable

Value	Description
0	Disables the write protection on interrupt registers if WPKEY corresponds to 0x54494D (“TIM” in ASCII).
1	Enables the write protection on interrupt registers if WPKEY corresponds to 0x54494D (“TIM” in ASCII).

Bit 0 – WPEN Write Protection Enable
The Timer Counter clock of the first channel must be enabled to access this register.

© 2026 Microchip Technology Inc. and its subsidiaries

See Register Write Protection for a list of registers that can be write-protected and Timer Counter clock conditions.

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x54494D (“TIM” in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x54494D (“TIM” in ASCII).

© 2026 Microchip Technology Inc. and its subsidiaries

37. Analog-to-Digital Converter (ADC) Controller

37.1. Description

The ADC is based on a 12-bit Analog-to-Digital Converter (ADC) managed by an ADC Controller providing enhanced resolution up to 16 bits. The conversions extend from the voltage carried on pin GND to the voltage carried on pin VREFP.

Conversion results are reported in a common register for all channels, as well as in a channel-dedicated register.

The 13-bit, 14-bit, 15-bit and 16-bit resolution modes are obtained by averaging multiple samples to decrease quantization noise. For the 13-bit mode, 4 samples are used, which gives a real sample rate of 1/4 of the actual sample frequency. For the 14-bit mode, 16 samples are used, giving a real sample rate of 1/16 of the actual sample frequency. For the 15-bit and 16-bit modes, respectively 64 and 256 samples are used, giving a real sample rate of respectively 1/64 and 1/256 of the actual sample frequency. This arrangement allows conversion speed to be traded off against for better accuracy.

The last channel is internally connected to the temperature sensor output voltage, and the last channel but one is connected to the temperature sensor reference voltage. Processing of this channel can be fully configured for efficient downstream processing due to the slow frequency variation of the value carried on such sensor.

The software trigger or internal triggers from Timer Counter output(s) can be selected for triggering a conversion.

The comparison circuitry allows automatic detection of values below a threshold, higher than a threshold, in a given range or outside the range, thresholds and ranges being fully configurable.

The ADC Controller internal fault output is directly connected to the PWM fault input. This input can be asserted by means of a comparison circuitry to immediately put the PWM output in a safe state (pure combinational path).

The ADC also integrates a Sleep mode and a conversion sequencer and connects with a DMA channel. These features reduce both power consumption and processor intervention.

This ADC has a selectable single-ended or fully differential input.

A supply monitor detects abnormal ADC power supply voltage and an interrupt can be triggered. The detection range can be configured.

Note: In this section, REFP = VREFP and REFN = GND.

37.2. Embedded Characteristics

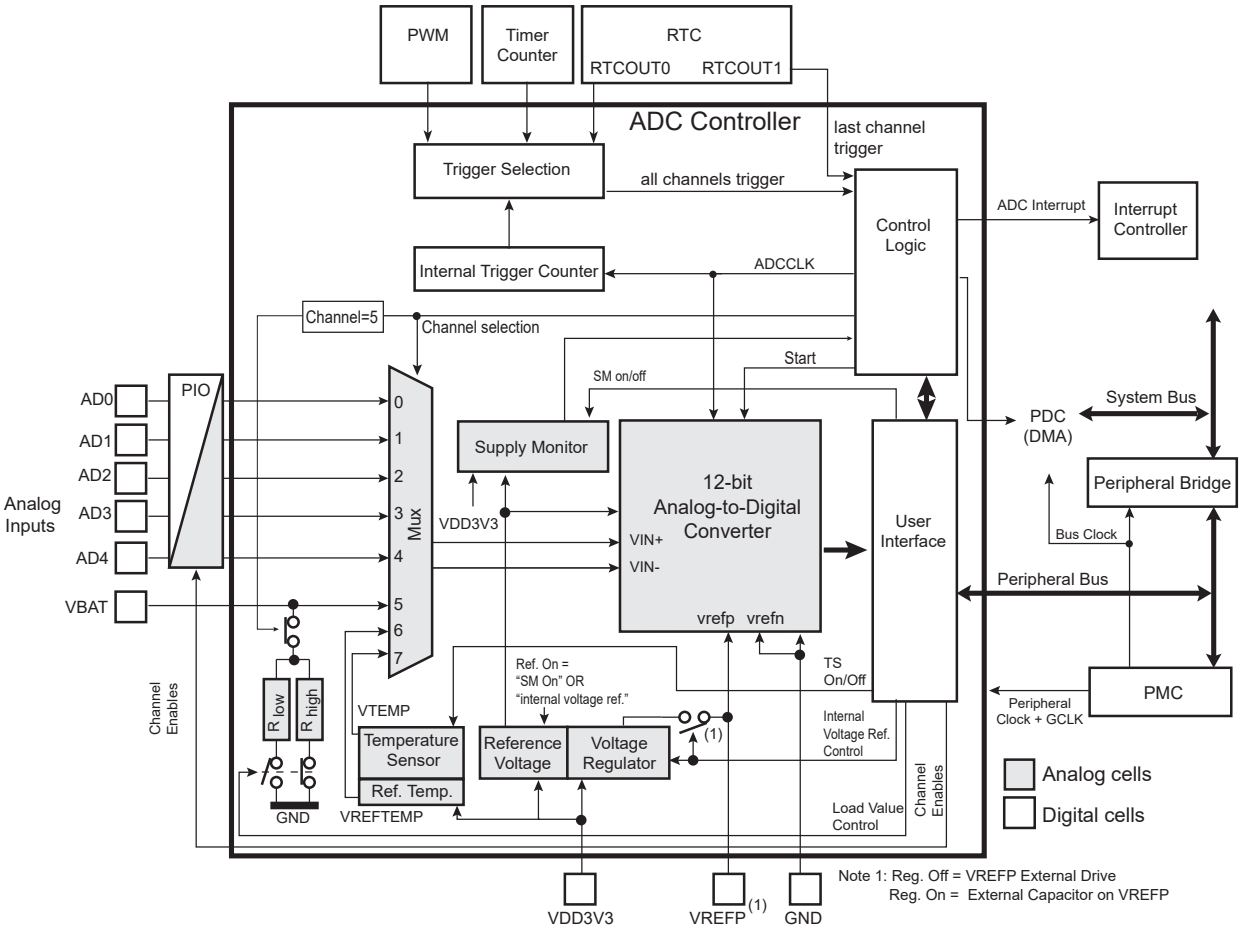
- 12-bit Resolution with Enhanced Mode up to 16 bits
- 1 Msps Conversion Rate
- Digital Averaging Function providing Enhanced Resolution Mode up to 16 bits
- On-chip Temperature Sensor Management
- Selectable Single-Ended or Differential Input Voltage
- Supply Monitor with Configurable Detection Range and Interrupt
- Digital Correction of Offset and Gain Errors
- Individual Enable and Disable of Each Channel
- Hardware or Software Trigger from:
  - ADC internal trigger counter
  - Timer Counter outputs

© 2026 Microchip Technology Inc. and its subsidiaries

  - PWM event line
- Drive of PWM Fault Input
- Peripheral DMA Support
- Two Sleep Modes (Automatic Wake-up on Trigger)
  - Lowest power consumption (voltage reference OFF between conversions)
  - Fast wake-up time response on trigger event (voltage reference ON between conversions)
- Channel Sequence Customizing
- Automatic Window Comparison of Converted Values
- Register Write Protection

37.3. Block Diagram

Figure 37-1. ADC Block Diagram



37.4. Product Dependencies

37.4.1. Power Management

The ADC Controller is not continuously clocked. The programmer must first enable the ADC Controller peripheral clock in the Power Management Controller (PMC) before using the ADC Controller. However, if the application does not require ADC operations, the ADC Controller clock can be stopped when not needed and restarted when necessary. Configuring the ADC Controller does not require the ADC Controller clock to be enabled.

© 2026 Microchip Technology Inc. and its subsidiaries

37.4.2. Interrupt Sources

The ADC interrupt line is connected on one of the internal sources of the Interrupt Controller. Using the ADC interrupt requires the interrupt controller to be programmed first.

37.4.3. Battery Voltage

The battery voltage is internally connected to the channel with the highest index minus two. A fixed division factor is applied on VBAT voltage to fit the CH5 maximum input range (refer to the section “Electrical Characteristics” for the ratio value).

When the channel corresponding to VBAT is selected for conversion, a configurable load is internally connected to VBAT. The load is automatically disconnected from VBAT when the channel is not selecting VBAT or when the ADC is disabled.

37.4.4. Temperature Sensor

The temperature sensor is internally connected to the channel with the highest index of the ADC.

The temperature sensor provides an output voltage VTEMP that is proportional to the absolute temperature (PTAT).

To activate the temperature sensor, the bit TEMPON in the Temperature Sensor Mode register (ADC_TEMPMR) must be set. After setting the bit, the start-up time of the temperature sensor must be achieved prior to initiating any measurements.

37.4.5. Supply Monitor

To activate the supply monitor, the bit SMEN in the Analog Control Register (ADC_ACR) must be written to 1.

The detection voltage can be adjusted by configuring ADC_ACR.SMVT. Refer to the section “Electrical Characteristics” for further details.

When the supply monitor detects an under voltage, an interrupt can be triggered if the bit SMEV in the Interrupt Enable register (ADC_IER) is written to 1.

37.4.6. ADC Voltage Reference Selection

The ADC positive voltage reference can be externally driven on the VREFP pin or internally driven by the embedded regulator. Refer to the section “Electrical Characteristics” for the driving capability of pin VREFP when it is internally driven.

When the voltage is internally driven, the embedded regulator must be enabled and an external decoupling capacitor must be connected to VREFP.

When the voltage is externally driven, the internal regulator must be turned off.

The internal regulator is enabled by writing ADC_ACR.INTVREFEN to 1.

Refer to the section “Electrical Characteristics” for further details.

37.4.7. I/O Lines

The digital inputs ADx are multiplexed with digital functions on the I/O lines. The analog mode for ADx I/O pins is automatically selected when the corresponding conversion channel is enabled.

37.4.8. Hardware Triggers

The ADC can use internal signals to start conversions. See the field TRGSEL for details on the wiring of internal triggers.

37.4.9. Fault Output

The ADC Controller has the FAULT output connected to the FAULT input of PWM. Refer to sections Fault Event and “Pulse Width Modulation Controller (PWM)”.

© 2026 Microchip Technology Inc. and its subsidiaries

37.5. Functional Description

37.5.1. Analog-to-Digital Conversion

Once the programmed start-up time (ADC_MR.STARTUP) has elapsed, ADC conversions are sequenced by three operating times:

- Tracking time—the time for the ADC to charge its input sampling capacitor to the input voltage. When several channels are converted consecutively, the inherent tracking time is 6 ADC clock cycles. However, the tracking time can be increased using ADC_MR.TRACKTIM and the TRACKX field in the Extended Mode register (ADC_EMR).
- ADC inherent conversion time—the time for the ADC to convert the sampled analog voltage. This time is constant and is defined from start of conversion to end of conversion.
- Channel conversion period—the effective time between the end of the current channel conversion and the end of the next channel conversion.

Figure 37-2. Sequence of Consecutive ADC Conversions with TRACKTIM = 0

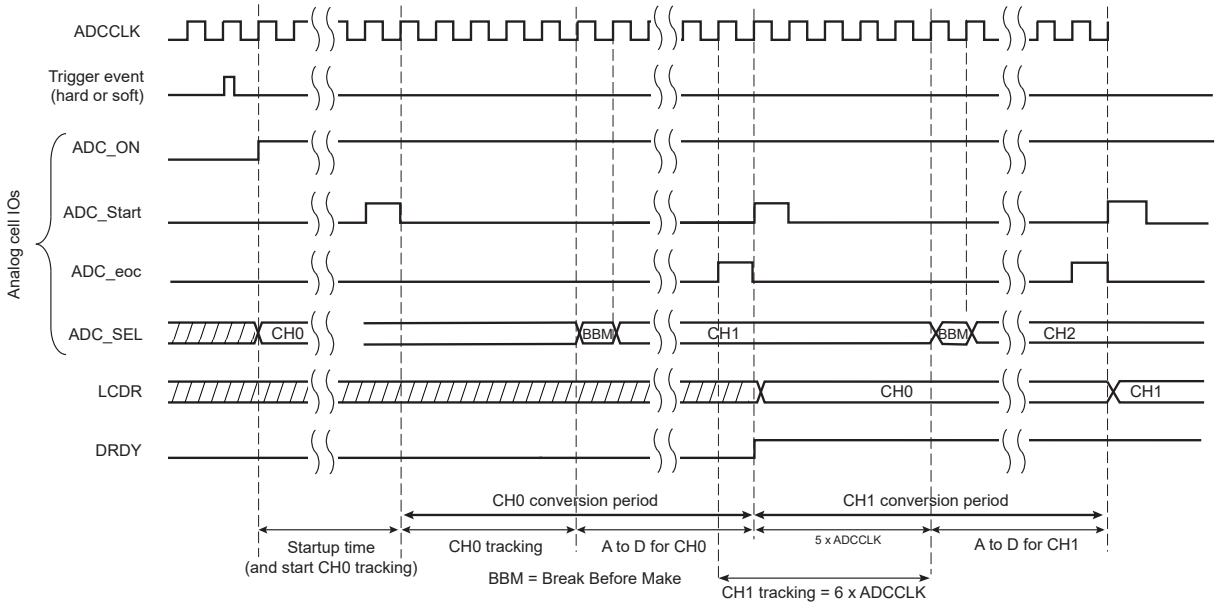
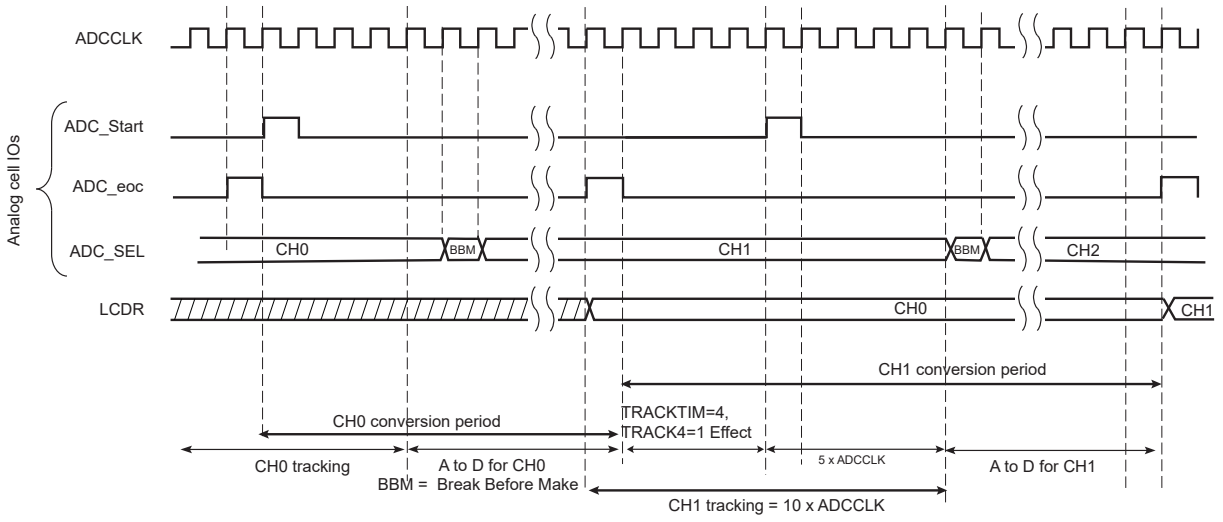


Figure 37-3. Sequence of Consecutive ADC Conversions with TRACKTIM = 4 and TRACKX = 1



© 2026 Microchip Technology Inc. and its subsidiaries

37.5.2. ADC Clock

The ADC uses the ADC clock (ADCCLK) to perform conversions. The ADC clock frequency is selected in ADC_MR.PRESCAL.

To generate the ADC clock, the prescaler has two clock sources: peripheral clock and GCLK. The clock source is selected using ADC_EMR.SRCCLK.

Regardless of the prescaler clock source, peripheral clock must always be running in the PMC.

If GCLK is selected as a clock source, the ADC clock frequency is independent of the processor/bus clock. At reset, peripheral clock is selected.

If ADC_EMR.SRCCLK is cleared, the prescaler clock (presc_clk) is driven by the peripheral clock. If ADC_EMR.SRCCLK is set, the prescaler clock is driven by GCLK. The ADC clock frequency is between fpresc_clk/2, if PRESCAL is 0, and fpresc_clk/512, if PRESCAL is set to 255 (0xFF).

ADC_MR.PRESCAL must be programmed to provide the ADC clock frequency parameter given in the section “Electrical Characteristics”.

37.5.3. ADC Reference Voltage

The voltage reference input of the ADC is the VREFP pin. Refer to the section “Electrical Characteristics” for further details.

37.5.4. Conversion Resolution

The ADC has a native resolution of 12 bits.

The ADC Controller provides enhanced resolution up to 16 bits by means of digital averaging.

37.5.5. Conversion Results

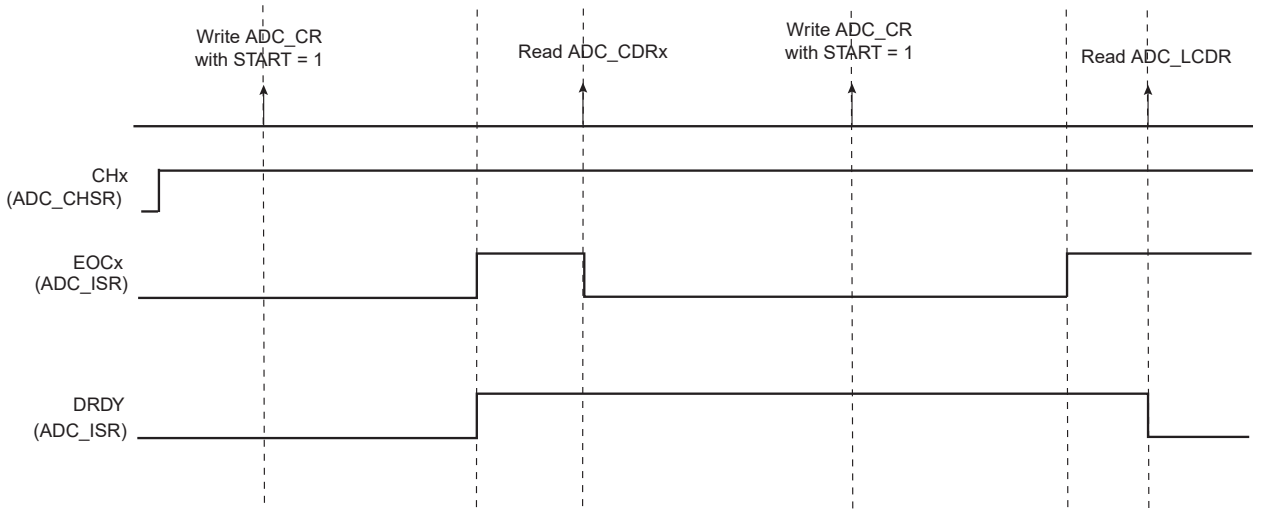
When a conversion is completed, the resulting digital value is stored in the Channel Data register (ADC_CDRx) of the current channel and in the Last Converted Data register (ADC_LCDR). By setting ADC_EMR.TAG, ADC_LCDR presents the channel number associated with the last converted data in NO_OSR_CHNB/CHNBOSR.

When a conversion is completed, EOCx in the End of Conversion Interrupt Status register (ADC_EOC_ISR) and the bit DRDY in the Interrupt Status register (ADC_ISR) are set. In the case of a connected DMA channel, DRDY rising triggers a data request. In any case, either EOC or DRDY can trigger an interrupt.

Reading one of the ADC_CDRx clears the corresponding EOC. Reading ADC_LCDR clears DRDY.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 37-4. EOCx and DRDY Flag Behavior



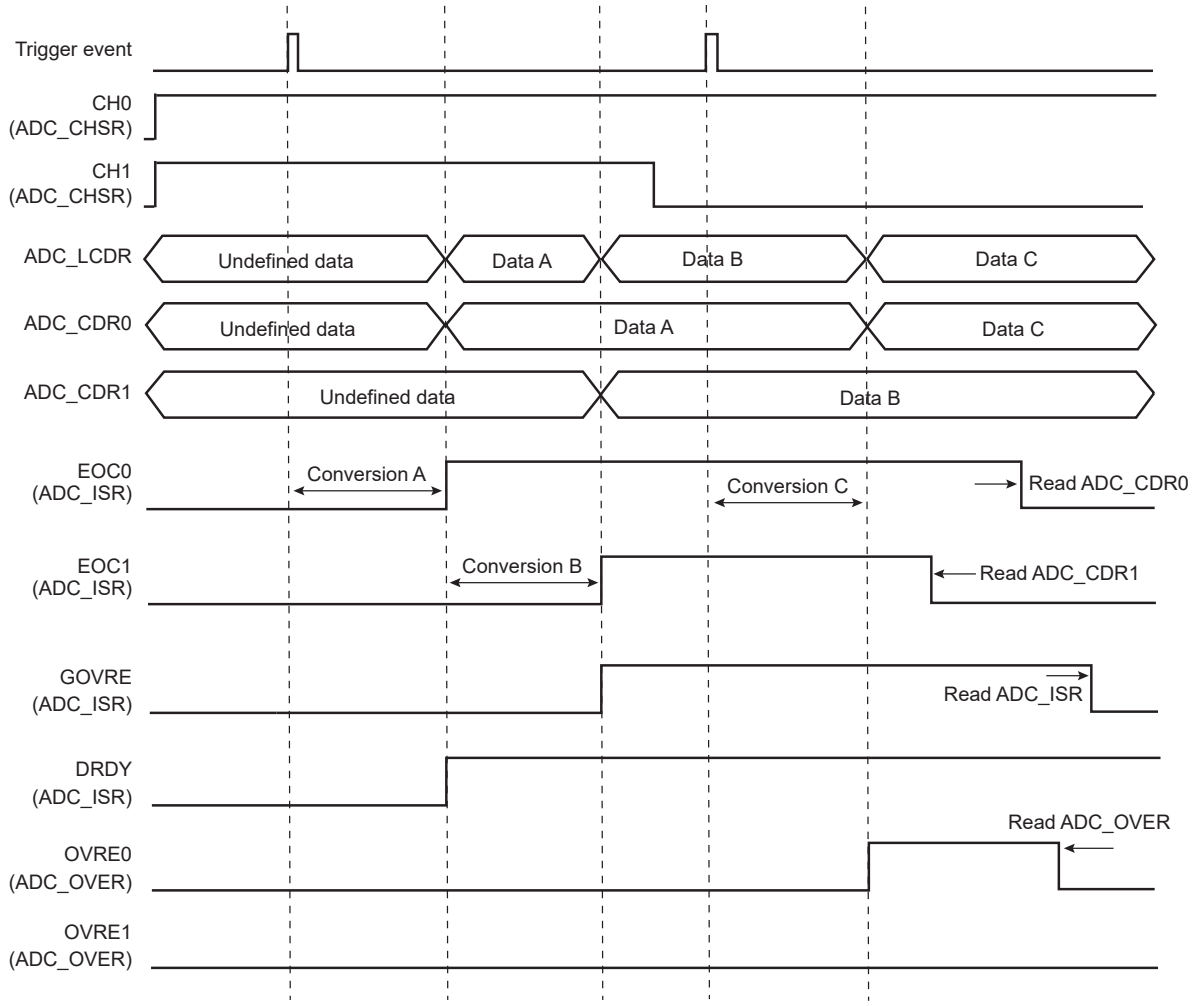
If ADC_CDRx is not read before further incoming data is converted, the corresponding OVREx flag is set in the Overrun Status register (ADC_OVER).

If new data is converted when DRDY is high, ADC_ISR.GOVRE is set.

The OVREx flag is automatically cleared when ADC_OVER is read, and the GOVRE flag is automatically cleared when ADC_ISR is read.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 37-5. EOCx, OVREx and GOVRE Flag Behavior



WARNING If the corresponding channel is disabled during a conversion or if it is disabled and then re-enabled during a conversion, its associated data and corresponding ADC_EOC_ISR.EOCx, ADC_ISR.GOVRE and ADC_OVER.OVREx flags are unpredictable.

37.5.6. Conversion Results Format

The conversion results can be signed (2's complement) or unsigned, depending on the value of ADC_EMR.SIGNMODE.

If conversion results are signed and resolution is less than 16 bits, the sign is extended up to bit 15 (e.g., 0xF43 for 12-bit resolution is read as 0xFF43, and 0x467 is read as 0x0467).

37.5.7. Conversion Triggers

Conversions of the active analog channels are started with a software or hardware trigger. The software trigger is provided by writing the Control register (ADC_CR) with ADC_CR.START at 1.

The list of external/internal events is provided in ADC_MR. The hardware trigger is selected using ADC_MR.TRGSEL. The selected hardware trigger is enabled if TRGMOD = 1, 2 or 3 in the Trigger register (ADC_ TRGR).

© 2026 Microchip Technology Inc. and its subsidiaries

The ADC also provides a dual trigger mode (ADC_TEMPMR.TEMPON=1) in which the highest index channel can be sampled at a rhythm different from the other channels. The trigger of the last channel is generated by the RTC. See Temperature Sensor.

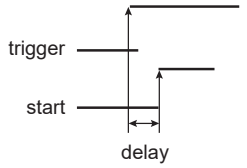
ADC_TRGR.TRGMOD selects the hardware trigger from the following:

- any edge, either rising or falling or both, detected on internal triggers (provided by other peripherals)
- a continuous trigger, meaning the ADC Controller restarts the next sequence as soon as it finishes the current one
- a periodic trigger (generated by the ADC Controller), which is defined by programming ADC_TRGR.TRGPER

The minimum time between two consecutive trigger events must be strictly greater than the duration time of the longest conversion sequence according to configuration of registers ADC_MR, ADC_SEQRx (Channel Sequence register), ADC_CHSR (Channel Status register).

If a hardware trigger is selected, the start of a conversion is triggered after a delay starting at each rising edge of the selected signal. Due to asynchronous handling, the delay may vary in a range of two peripheral clock periods to one ADC clock period. This delay introduces sampling jitter in the A/D conversion process and may therefore degrade the conversion performance (e.g., SNR, THD).

Figure 37-6. Hardware Trigger Delay



Only one start command is necessary to initiate a conversion sequence on all the enabled channels. The ADC hardware logic automatically performs the conversions on the active channels, then waits for a new request. The Channel Enable (ADC_CHER) and Channel Disable (ADC_CHDR) registers enable the analog channels to be enabled or disabled independently.

If the ADC is used with a DMA channel, only the transfers of converted data from enabled channels are performed and the resulting data buffers should be interpreted accordingly.

37.5.8. Sleep Mode and Conversion Sequencer

The ADC Sleep mode maximizes power saving by automatically deactivating the ADC when it is not being used for conversions. Sleep mode is selected by setting ADC_MR.SLEEP.

Sleep mode is managed by a conversion sequencer, which automatically processes the conversions of all channels at the lowest power consumption.

This mode can be used when the minimum period of time between two successive trigger events is greater than the start-up period of the ADC. Refer to the section “Electrical Characteristics”.

When a start conversion request occurs, the ADC is automatically activated. As the analog cell requires a start-up time, the logic waits during this time and starts the conversion on the enabled channels. When all conversions are complete, the ADC is deactivated until the next trigger. Events triggered during the sequence are ignored.

The conversion sequencer allows automatic processing with minimum processor intervention and optimized power consumption. Conversion sequences can be performed periodically using the internal timer (ADC_TRGR) or the PWM event line. The periodic acquisition of several samples can be processed automatically without any intervention of the processor via the DMA channel.

© 2026 Microchip Technology Inc. and its subsidiaries

The sequence can be customized by programming ADC_SEQR1 and setting ADC_MR.USEQ. This sequence is limited to 8 channels, from 0 to 7. The user can choose a specific order of channels and can program up to 8 conversions by sequence. The user is free to create a personal sequence by writing channel numbers in ADC_SEQR1. Not only can channel numbers be written in any sequence, channel numbers can be repeated several times. When ADC_MR.USEQ is set, ADC_SEQR1.USCHx is used to define the sequence. Only enabled USCHx fields will be part of the sequence. Each USCHx field has a corresponding enable, CHx-1, in ADC_CHER.

If 8 channels are used on an application board, there is no restriction of usage of the user sequence. However, if some ADC channels are not enabled for conversion but rather used as pure digital inputs, the respective indexes of these channels cannot be used in the user sequence fields (see ADC_SEQR1). For example, if channel 4 is disabled (ADC_CHSR[4] = 0), ADC_SEQRx fields USCH1 up to USCH8 must not contain the value 4. Thus the length of the user sequence may be limited by this behavior.

As an example, if only four channels (CH0 up to CH3) are selected for ADC conversions, the user sequence length cannot exceed four channels. Each trigger event may launch up to four successive conversions of any combination of channels 0 up to 3, but no more (i.e., in this case the sequence CH0, CH0, CH1, CH1, CH1 is impossible).

A sequence that repeats the same channel several times requires more enabled channels than channels actually used for conversion. For example, the sequence CH0, CH0, CH1, CH1 requires four enabled channels (four free channels on application boards) whereas only CH0, CH1 are really converted.

Note: The reference voltage pins always remain connected in Normal mode as in Sleep mode.

37.5.9. Comparison Window

The ADC Controller features automatic comparison functions. It compares converted values to a low threshold, a high threshold or both, depending on the value of ADC_EMR.CMPMODE. The comparison can be done on all channels or only on the channel specified in ADC_EMR.CMPSEL. To compare all channels, ADC_EMR.CMPALL must be set.

If set, ADC_EMR.CMPTYPE can be used to discard all conversion results that do not match the comparison conditions. Once a conversion result matches the comparison conditions, all the subsequent conversion results are stored in ADC_LCDR (even if these results do not meet the comparison conditions). Setting ADC_CR.CMPRST immediately stops the conversion result storage until the next comparison match.

If ADC_EMR.CMPTYPE is cleared, all conversions are stored in ADC_LCDR. Only the conversions that match the comparison conditions trigger the ADC_ISR.COMPE flag.

Moreover, a filtering option can be set by writing the number of consecutive comparison matches needed to raise the flag. This number can be written and read in ADC_EMR.CMPFILTER. The filtering option is dedicated to reinforcing the detection of an analog signal exceeding a predefined threshold. The filter is cleared as soon as ADC_ISR is read, so this filtering function must be used with the DMA controller and works only when using Interrupt mode (no polling).

The flag can be read on ADC_ISR.COMPE and can trigger an interrupt.

The high threshold and the low threshold can be read/write in the Compare Window register (ADC_CWR).

Depending on the sign of the conversion, chosen with ADC_EMR.SIGNMODE, the high threshold and low threshold values must be signed or unsigned to maintain consistency during the comparison. If the conversion is signed, both thresholds must also be signed; if the conversion is unsigned, both thresholds must be unsigned. If comparison occurs on all channels, SIGNMODE must be set to ALL_UNSIGNED or ALL_SIGNED and the thresholds must be set accordingly.

© 2026 Microchip Technology Inc. and its subsidiaries

37.5.10. Differential and Single-ended Input Modes

37.5.10.1. Input-Output Transfer Functions

The ADC can be configured to operate in the following input voltage modes:

- Single-ended—ADC_CCR.DIFFx = 0. This is the default mode after a reset.
- Differential—ADC_CCR.DIFFx = 1 (see the figure Analog Full Scale Ranges in Single-ended/ Differential Applications). In Differential mode, the ADC requires differential input signals with a VDD/2 common mode voltage (refer to the section “Electrical Characteristics”).

The following equations give the unsigned ADC input-output transfer function in each mode (see Note). With signed conversions (see ADC_EMR.SIGNMODE), subtract 2047 from the ADC_LCDR.LDATA value given below.

In the formulae below, REFP = VREFP, REFN = GND.

Single-ended mode:

$$\text{ADC\_LCDR.LDATA} = \frac{\text{ADx} - \text{REFN}}{\text{REFP} - \text{REFN}} \times 2^{12}$$

Differential mode:

$$\text{ADC\_LCDR.LDATA} = \left(1 + \frac{\text{ADx} - \text{ADx+1}}{\text{REFP} - \text{REFN}}\right) \times 2^{11}$$

Note: Equations assume ADC_EMR.OSR = 1

If ADC_MR.ANACH is set, the ADC can manage both differential channels and single-ended channels. If ADC_MR.ANACH is cleared, the parameters defined in ADC_CCR are applied to all channels.

The following table gives the internal positive and negative ADC inputs assignment with respect to the programmed mode (ADC_CCR.DIFFx).

For example, if Differential mode is required on channel 0, input pins AD0 and AD1 are used. In this case, only channel 0 must be enabled by writing a 1 to ADC_CHER.CH0.

Figure 37-7. Analog Full Scale Ranges in Single-ended/Differential Applications

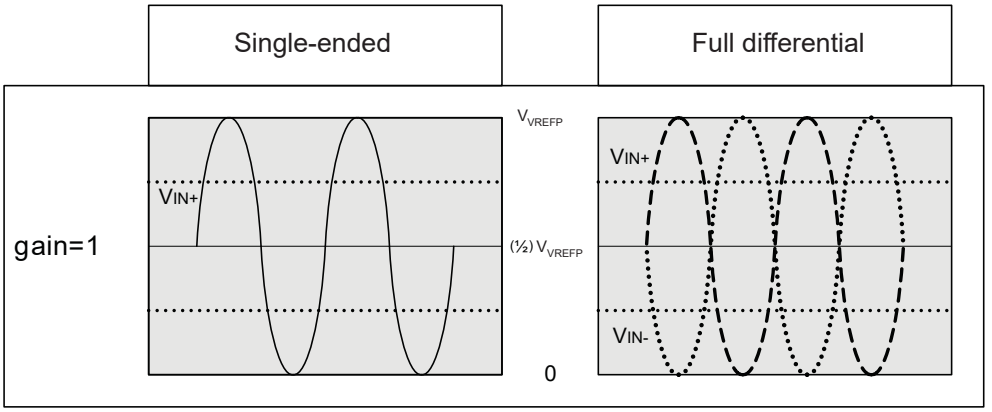


Table 37-1. Input Pins and Channel Numbers in Single-ended and Differential Modes

Internal ADC Inputs (VIN+, VIN-)		Channel Numbers	
Single-ended Mode	Differential Mode	Single-ended Mode	Differential Mode
AD0, GND	AD0, AD1	CH0	CH0
AD1, GND	–	CH1	
AD2, GND	AD2, AD3	CH2	CH2
AD3, GND	–	CH3	

© 2026 Microchip Technology Inc. and its subsidiaries

Table 37-1. Input Pins and Channel Numbers in Single-ended and Differential Modes (continued)

Internal ADC Inputs (VIN+, VIN-)		Channel Numbers	
Single-ended Mode	Differential Mode	Single-ended Mode	Differential Mode
AD4, GND	–	CH4	–
VBAT, GND	–	CH5	
VREFTEMP, GND	–	CH6	–
VTEMP, GND	–	CH7	

37.5.11. ADC Timings

The ADC start-up time is programmed using ADC_MR.STARTUP. Refer to the section “Electrical Characteristics”.

The ADC Controller provides an inherent tracking time of six ADC clock cycles.

A minimal tracking time is necessary for the ADC to ensure the best converted final value between two conversions. The tracking time can be adjusted to accommodate a range of source impedances. If more than six ADC clock cycles are required, the tracking time can be increased by using ADC_MR.TRACKTIM and ADC_EMR.TRACKX.

WARNING No input buffer amplifier to isolate the source is included in the ADC. Refer to the section “Electrical Characteristics”.

37.5.12. Temperature Sensor

The temperature sensor is internally connected to the channel with the highest index. For temperature measurement, ADC_TEMPMR.TEMPON must be set.

Temperature measurement can be performed in different ways through the ADC Controller. The different methods of measurement depend on configuration bits ADC_TRGR.TRGMOD and ADC_CHSR.CHmax (max=highest index).

Temperature measurement can be triggered like the other channels by enabling its associated conversion channel, writing 1 in ADC_CHER.CHmax (max=highest index).

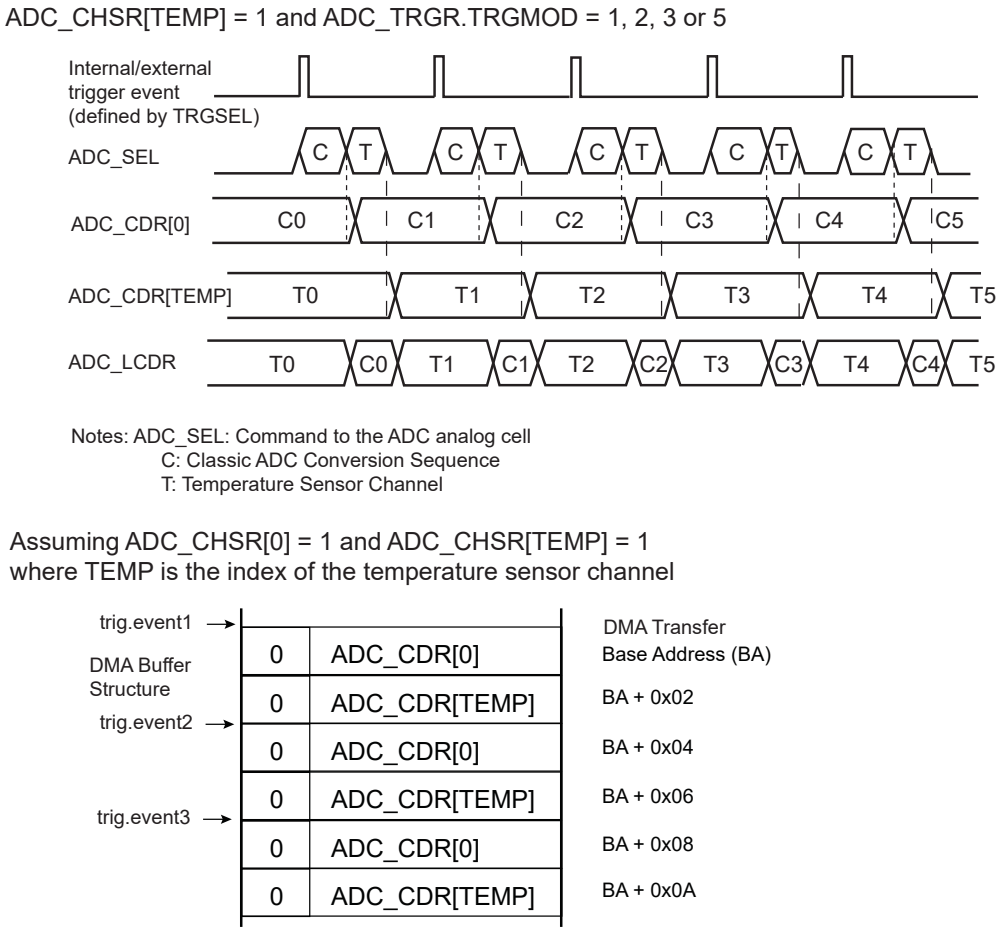
The manual start can only be performed if ADC_TRGR.TRGMOD = 0. When ADC_CR.START is set, the temperature sensor channel conversion is scheduled together with the other enabled channels (if any). The result of the conversion is placed in the ADC_CDRmax (max=highest index) register and the associated ADC_EOC_ISR.EOCmax (max=highest index) flag is set.

If ADC_TRGR.TRGMOD = 1, 2, 3 or 5, the temperature sensor channel is periodically converted together with other enabled channels and the result is placed on the ADC_LCDR and ADC_CDRmax (max=highest index) registers. Thus, the temperature conversion result is part of the DMA Controller buffer. The temperature channel can be enabled/disabled at any time but this may not be optimal for downstream processing.

When the conversion result matches the conditions defined in the ADC_TEMPMR and ADC_TEMPCWR, the ADC_ISR.TEMPCHG flag is set.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 37-8. Non-optimized Temperature Conversion



The temperature factor having a slow variation rate and being potentially totally different from the other conversion channels, the ADC Controller allows a different way of triggering the measure when ADC_TEMPMR.TEMPON is set but ADC_CHSR.CHmax (max=highest index) is not set.

Under these conditions, the measure is triggered every second by means of an internal trigger generated by the RTC, always enabled and totally independent of the internal/external triggers. The RTC event will be processed on the next internal/external trigger event, as described in the following figure. The internal/external trigger is selected through ADC_MR.TRGSEL.

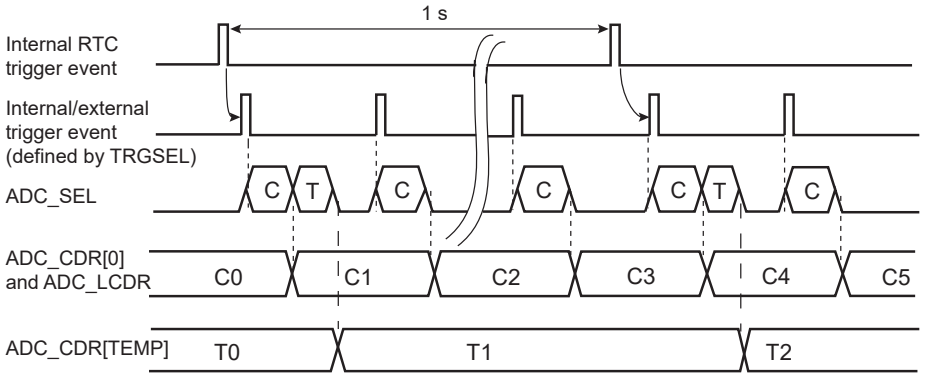
In this mode of operation, the temperature sensor is only powered for a period of time covering the start-up time and conversion time (see the figure Temperature Conversion Only for more details).

Every second, a conversion is scheduled for the temperature channel but the result of the conversion is only uploaded in the ADC_CDRmax (max=highest index) register and not in ADC_LCDR. Therefore, there is no change in the structure of the DMA Controller buffer due to the conversion of the temperature channel, only the enabled channels are kept in the buffer. The end of conversion of the temperature channel is reported by the ADC_EOC_ISR.EOCmax (max=highest index) flag.

© 2026 Microchip Technology Inc. and its subsidiaries

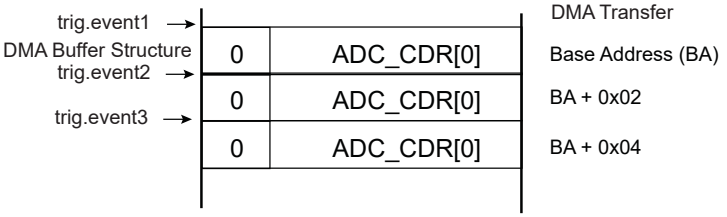
Figure 37-9. Optimized Temperature Conversion Combined With Classical Conversions

ADC_CHSR[TEMP] = 0 and ADC_TRGR.TRGMOD = 1, 2, 3 or 5
TEMPON = 1



Notes: ADC_SEL: Command to ADC analog cell
C: Classic ADC Conversion Sequence
T: Temperature Sensor Channel

Assuming ADC_CHSR[0] = 1

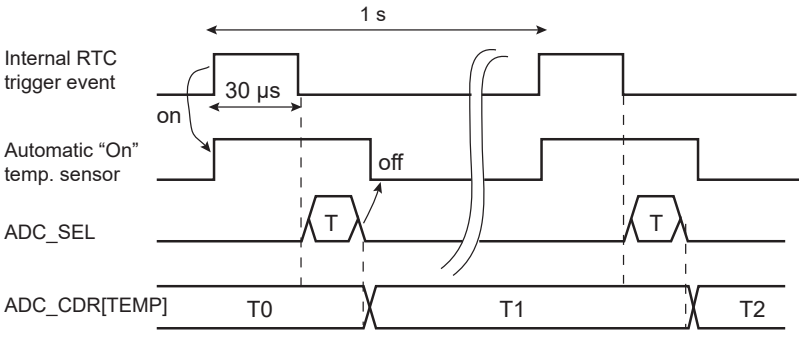


If TEMPON = 1, TRGMOD = 0 and none of the channels are enabled in ADC_CHSR (ADC_CHSR = 0), then only the temperature channel is converted at a rate of 1 conversion per second (see the figure Temperature Conversion Only).

This mode of operation, when combined with the Sleep mode operation of the ADC Controller, provides a low-power mode for temperature measurement (assuming there is no other ADC conversion to schedule at a high sampling rate, or simply no other channel to convert).

Figure 37-10. Temperature Conversion Only

ADC_CHSR = 0 and ADC_TRGR.TRGMOD = 0
TEMPON = 1



Note: ADC_SEL: Command to the ADC analog cell

© 2026 Microchip Technology Inc. and its subsidiaries

Furthermore, it is possible to raise a flag only if there is a predefined change in the temperature measure. The user can define a range of temperatures or a threshold in ADC_TEMPCWR, and the mode of comparison that can be programmed in ADC_TEMPMR.TEMPCMPMOD. These values define how the ADC_ISR.TEMPCHG flag is raised.

In any case, if TEMPON is set and a conversion trigger event occurs, the temperature can be read in ADC_CDRmax (max=highest index).

37.5.12.1. Disabling the Temperature Sensor to Put the System in Low-Power Mode

To put the system in Low-Power mode when the temperature sensor is in use, ADC_TEMPMR.TEMPON must be cleared and a software reset must be performed (ADC_CR.SWRST=1).

37.5.13. Enhanced Resolution Mode and Digital Averaging Function

37.5.13.1. Enhanced Resolution Mode

The Enhanced Resolution mode is enabled if ADC_EMR.OSR is configured to 1, 2, 3 or 4. The enhancement is based on a digital averaging function.

There is no averaging on the last index channel if the measure is triggered by an RTC event.

In this mode, the ADC Controller will trade off conversion speed against accuracy by averaging multiple samples, thus providing a digital low-pass filter function.

The selected oversampling ratio applies to all enabled channels when triggered by an RTC event. The following formula applies:

$$\text{ADC\_LCDR.LDATA} = \frac{1}{M} \times \sum_{k=0}^{k=N-1} ADC(k)$$

where N and M are given in the table below.

Table 37-2. Digital Averaging Function Configuration Versus OSR Values

ADC_EMR.OSR Value	ADC_LCDR.LDATA Length	N Value	M Value	Full Scale Value	Maximum Value
0	12 bits	1	1	4095	4095
1	13 bits	4	2	8191	8190
2	14 bits	16	4	16383	16381
3	15 bits	64	8	32767	32766
4	16 bits	256	16	65535	65533

The average result is valid in ADC_CDRx (x = channel index) only if the ADC_EOC_ISR.EOCn flag is set and if the ADC_OVER.OVREn flag is cleared. The average result for all channels is valid in ADC_LCDR only if ADC_ISR.DRDY is set and ADC_ISR.GOVRE is cleared.

Note that ADC_CDRs are not buffered. Therefore, when an averaging sequence is ongoing, the value in these registers changes after each averaging sample. However, overrun flags in ADC_OVER rise as soon as the first sample of an averaging sequence is received. Thus, the previous averaged value is not read, even if the new averaged value is not ready.

Consequently, when an overrun flag rises in ADC_OVER, it means that the previous unread data is lost but it does not mean that this data has been overwritten by the new averaged value, as the averaging sequence concerning this channel can still be ongoing.

When an oversampling is performed, the maximum value that can be read on ADC_CDRx or ADC_LCDR is not the full-scale value, even if the maximum voltage is supplied on the analog input. See the table Digital Averaging Function Configuration Versus OSR Values.

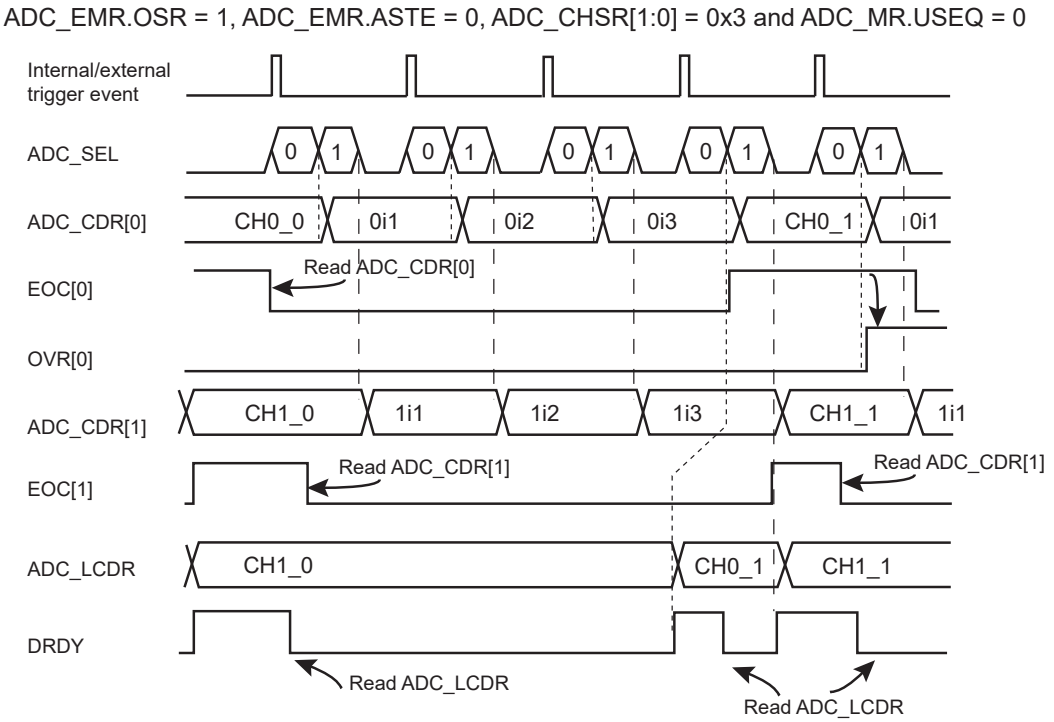
© 2026 Microchip Technology Inc. and its subsidiaries

37.5.13.2. Averaging Function Versus Trigger Events

The samples can be defined in different ways for the averaging function, depending on the configuration of ADC_EMR.ASTE and ADC_MR.USEQ.

When ADC_MR.USEQ = 0, there are two possible ways to generate the averaging through the trigger event. If ADC_EMR.ASTE = 0, every trigger event generates one sample for each enabled channel, as described in the following figure. Therefore, four trigger events are requested to obtain the result of averaging if ADC_EMR.OSR = 1.

Figure 37-11. Digital Averaging Function Waveforms Over Multiple Trigger Events

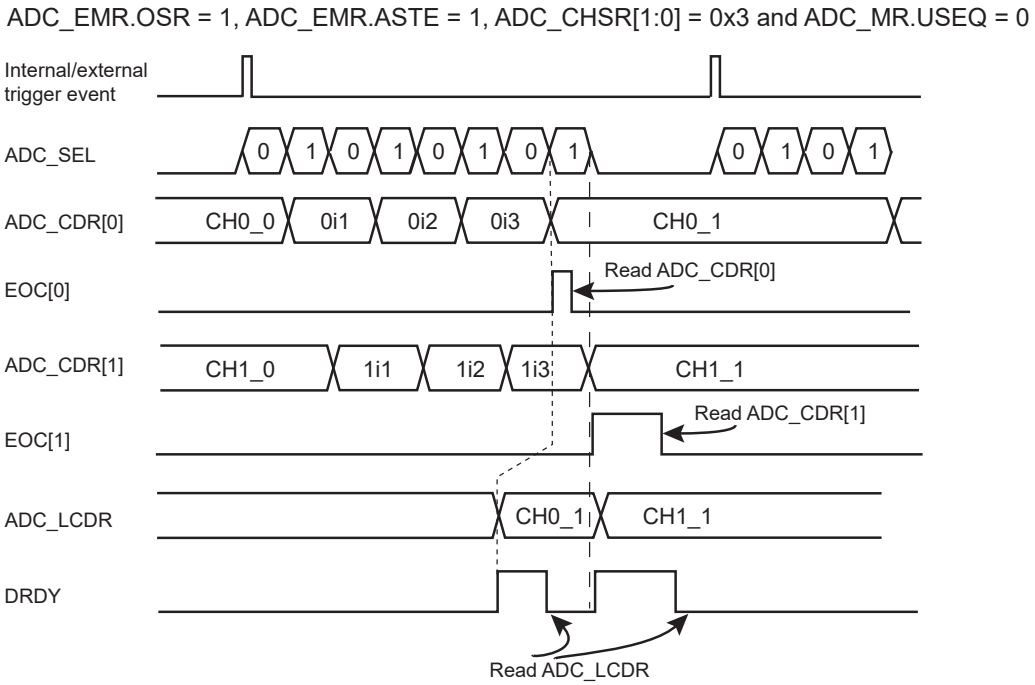


Note: ADC_SEL: Command to the ADC analog cell
0i1, 0i2, 0i3, 1i1, 1i2, 1i3 are intermediate results and CH0_0, CH0_1, CH1_0 and CH1_1 are final results of average function.

If ADC_EMR.ASTE = 1 and ADC_MR.USEQ = 0, the sequence to be converted, defined in ADC_CHSR, is automatically repeated n times (where n corresponds to the oversampling ratio defined in ADC_EMR.OSR). As a result, only one trigger is required to obtain the result of the averaging function, as described in the following figure.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 37-12. Digital Averaging Function Waveforms on a Single Trigger Event



Note: ADC_SEL: Command to the ADC analog cell
0i1, 0i2, 0i3, 1i1, 1i2, 1i3 are intermediate results and CH0_0, CH0_1, CH1_0 and CH1_1 are final results of average function.

When USEQ = 1, the user can define the channel sequence to be converted by configuring ADC_SEQRx and ADC_CHER so that channels are not interleaved during the averaging period. Under these conditions, a sample is defined for each end of conversion, as described in the following figure.

When USEQ = 1 and ASTE = 1, OSR can be only configured to 1. Up to three channels can be converted in this mode. The averaging result will be placed in the corresponding ADC_CDRx and in ADC_LCDR for each trigger event. The ADC real sample rate remains the maximum ADC sample rate divided by 4.

It is important that the user sequence follows a specific pattern. The user sequence must be programmed in such a way that it generates a stream of conversion, where a same channel is successively converted.

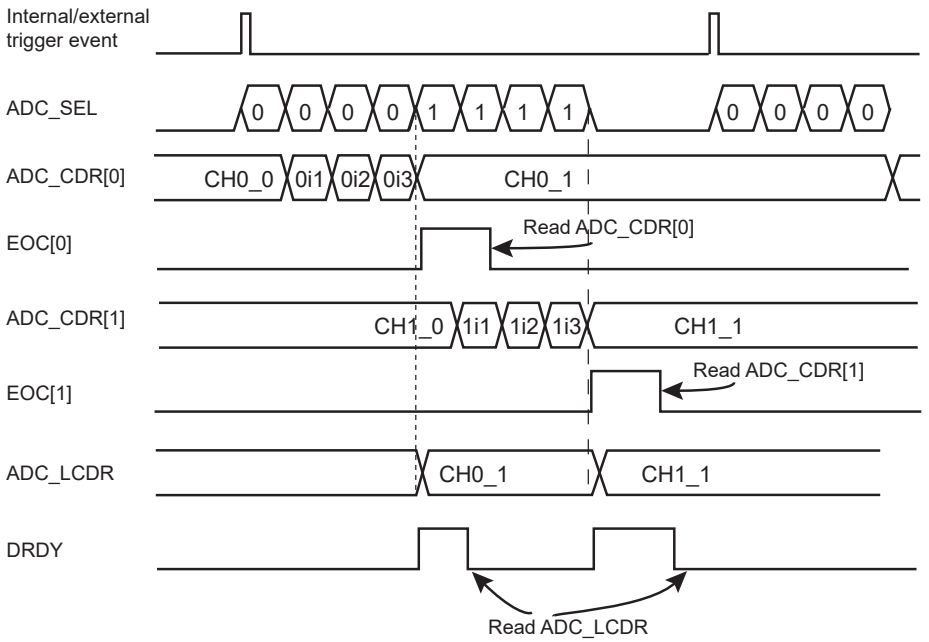
Table 37-3. Example Sequence Configurations (USEQ = 1, ASTE = 1, OSR = 1)

Register	Number of Channels Non-interleaved Averaging - Register Value		
	1 (e.g., CH0)	2 (e.g., CH0, CH1)	3 (e.g., CH0, CH1, CH2)
ADC_CHSR	0x0000000F	0x000000FF	0x00000FFF
ADC_SEQR1	0x00000000	0x11110000	0x11110000

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 37-13. Digital Averaging Function Waveforms on a Single Trigger Event, Non-interleaved

ADC_EMR.OSR = 1, ADC_EMR.ASTE = 1, ADC_CHSR[7:0] = 0xFF and ADC_MR.USEQ = 1
ADC_SEQR1 = 0x1111_0000



Note: ADC_SEL: Command to the ADC analog cell
0i1, 0i2, 0i3, 1i1, 1i2, 1i3 are intermediate results and CH0_0, CH0_1, CH1_0 and CH1_1 are final results of average function.

37.5.14. Automatic Error Correction

The ADC features automatic error correction of conversion results. Offset and gain error corrections are available. The correction can be enabled for each channel and correction values (offset and gain) are the same for all channelsprogrammable per channel.

To enable error correction, the corresponding ECORRx bit must be set in the Channel Error Correction register (ADC_CECR). The offset and gain values used to compensate the results are the same for all correction-enabled channels and programmed in the Correction Values register (ADC_CVR) set on a 'per channel' basis using the Correction Select register (ADC_COSR) and the Correction Values register (ADC_CVR). ADC_COSR is used to select the channel to be displayed in ADC_CVR. This selection applies to both the read and the write operations in ADC_CVR.

ADC_EMR.ADCMODE is used to configure a running mode of the ADC Normal mode, Offset Error mode, or Gain Error mode (see ADC_EMR). ADCMODE uses two internal references (VREFP, GND) to be measured and to extract the offset and gain error from 3 point-measurement codes. If some references already exist on the final application connected to some input channel ADx, they can be used as a replacement of the ADCMODE to generate the 2 or 3 points of calibration and used to extract the GAINCORR and OFFSETCORR.

After a reset, the ADC running mode is Normal mode. Offset Error mode and Gain Error mode are used to determine values of offset compensation and gain compensation, respectively, to apply to conversion results. The following table provides formulas to obtain the compensation values, with:

- OFFSETCORR—the Offset Correction value. OFFSETCORR is a signed value.
- GAINCORR—the Gain Correction value
- GCi—the intermediate Gain Compensation value
- Gs—the value 15

© 2026 Microchip Technology Inc. and its subsidiaries

- ConvValue—the value converted by the ADC (as returned in ADC_LCDR or ADC_CDR)
- Resolution—the resolution used to process the conversion (either RESOLUTION, RESOLUTION+1 or RESOLUTION+2).

Table 37-4. ADC Running Modes

ADC_EMR.ADCMODE	Mode	Description
0	Normal	Normal mode of operation to perform conversions
1	Offset Error	For unsigned conversions: OFFSETCORR = ConvValue – $2^{(\text{Resolution}} - 1)$
		For signed conversions: OFFSETCORR = ConvValue
2	Gain Error	GCi = ConvValue
3		$\text{GAINCORR} = \frac{3584}{\text{GCi} - \text{ConvValue}} \times 2^{(\text{Gs})}$

The final conversion result after error correction is obtained using the following formula:

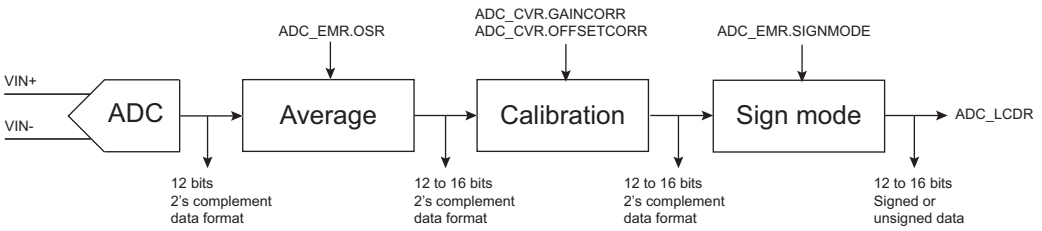
$$\text{Corrected Data} = (\text{Converted Data} + \text{OFFSETCORR}) \times \frac{\text{GAINCORR}}{2^{(\text{Gs})}}$$

37.5.14.1. Calibration Mode of Operation

The ADC is built to convert positive and negative voltages around a common mode voltage (VREFP-GND)/2. This is why the ADC output, averaging and calibration are performed as 2's complements. An ADC positive integer output is generated only in the last stage to generate unsigned data.

The ADC code is first averaged when requested; calibration is applied afterward. See the following figure.

Figure 37-14. ADC Signal Processing



Three internal reference voltages, VM, VL and VH, are generated from VREFP, with:

- VL = (VREFP-GND)x(1/16)
- VH = (VREFP-GND) x(15/16)
- VM = (VREFP-GND)/2

In Single mode non-signed outputs, the references are ideally converted as:

- VL_code = 256
- VH_code = 3840
- VM_code = 2048

In Differential mode signed outputs, the references are ideally converted as:

- (VL-VH)_code = -1792
- (VH-VL)_code = +1792
- VM_code = 0

The difference between the two voltage reference codes is the value 3584.

For calibration, three values are measured using ADCMODE 1,2,3.

© 2026 Microchip Technology Inc. and its subsidiaries

In the table ADC Running Modes, ConvValue is the current converted value.

37.5.14.1.1. First Conversion in ADCMODE=1

The ADC converted value is placed in the OFFSETCORR field (even after averaging) in a Sign mode signed output. If the output is unsigned, subtract half the code dynamic, which depends on the resolution (averaging).

OFFSETCORR is the offset at mid-range in an unsigned configuration.

37.5.14.1.2. Second Conversion in ADCMODE=2

This gives an upper code for gain calculation:

ADC output VH_Code = GCi = ConvValue

37.5.14.1.3. Third Conversion in ADCMODE=3

This gives the lowest code for the gain calculation:

ADC output VL_code = new ConvValue

Finally, VH_Code-VL_Code = GCI-ConvValue is the spread between the two codes, ideally at 3584 when there is no gain error.

37.5.14.1.4. Final GAINCORR Computation

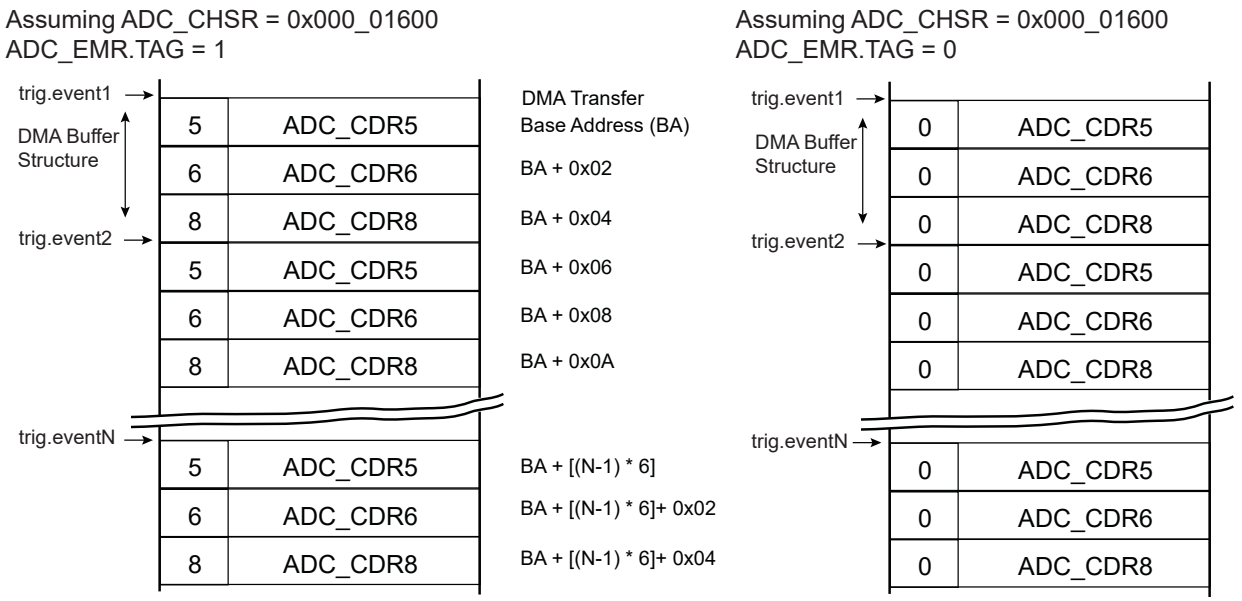
The final formula to compute GAINCORR can be applied. This formula is given without averaging (ADC resolution at 12 bits).

If an averaging is applied to get the ADC code in ADCMODE 2 and 3, then it is needed to scale down the output code to fit a 12-bit resolution.

37.5.15. Buffer Structure without FIFO

The DMA read channel is triggered when ADC_FMR.ENFIFO is set to 0 (see ENFIFO: Enable FIFO) and each time a new data is stored in ADC_LCDR. The same structure of data is repeatedly stored in ADC_LCDR each time a trigger event occurs. Depending on the user mode of operation (ADC_MR, ADC_CHSR, ADC_SEQR1, ADC_TSMR), the structure differs. Each data read to DMA buffer, carried on a half-word (16 bits), consists of last converted data right-aligned and when TAG is set in ADC_EMR, the four most significant bits carry the channel number, thus allowing an easier post-processing in the DMA buffer or a better checking of the DMA buffer integrity.

Figure 37-15. Buffer Structure



© 2026 Microchip Technology Inc. and its subsidiaries

.

37.5.15.1. Classic ADC Channels Only

The structure of data within the buffer is defined by ADC_MR, ADC_CHSR, ADC_SEQRx. See the figure Buffer Structure.

If the user sequence is not used (i.e., ADC_MR.USEQ is cleared), then only the value of ADC_CHSR defines the data structure. For each trigger event, enabled channels are consecutively stored in ADC_LCDR and automatically read to the buffer.

When the user sequence is configured (i.e., ADC_MR.USEQ is set), not only does ADC_CHSR modify the data structure of the buffer, but ADC_SEQRx registers may modify the data structure of the buffer as well.

37.5.16. Buffer Structure with FIFO

The DMA read channel is triggered when the fields ENFIFO and ENLEVEL are used (see ADC_FMR).

If the configuration ENFIFO is high and ENLEVEL is low, the DMA read channel is triggered as soon as one data is written.

When at least one data is ready in the FIFO, the RXRDY flag rises and remains high as long as there is at least one data to be read in the FIFO. When the entire FIFO has been filled with data, the RXFULL flag rises. If a new data is written into the RX FIFO while RXFULL is high, then the RXOVR flag rises. This flag remains high until ADC_ISR is read.

Once all data contained in the RX FIFO have been read, the RXEMPTY flag rises. If a data is read while the RX FIFO is empty, an underrun occurs and the RXUDR flag rises. This flag remains high until ADC_ISR is read.

Once a data is written in the RX FIFO, the RXRDY flag rises. If the corresponding interrupt has been enabled, an interrupt is generated and remains high as long as a data is available in the RX FIFO.

The DMA read channel is triggered when a new data is stored in FIFO. The same structure of data is repeatedly stored in FIFO each time a trigger event occurs. Depending on the user mode of operation (ADC_MR, ADC_CHSR, ADC_SEQR1, ADC_TSMR), the structure differs. Each data read to DMA buffer, carried on a half-word (16 bits), consists of last converted data right-aligned, and when ADC_EMR.TAG is set, the four most significant bits carry the channel number, thus allowing an easier post-processing in the DMA buffer or a better checking of the DMA buffer integrity.

ADC_FMR.FIFOCNT gives the number of conversions available in the FIFO.

37.5.17. Fault Event

The ADC Controller internal fault output is directly connected to the PWM fault input. The fault event may be asserted depending on the configuration of comparison registers and converted values.

The comparison based on ADC_CWR settings, i.e., on all converted channels except the last one, triggers a fault event sent to the PWM.

As an example, overcurrent or temperature values exceeding user-defined limits trigger a fault to the PWM.

When the comparison event occurs, the ADC fault output generates a pulse of one peripheral clock cycle to the PWM fault input. This fault line can be enabled or disabled within PWM. Should it be activated and asserted by the ADC Controller, the PWM outputs are immediately placed in a safe state (pure combinational path). Note that the ADC fault output connected to the PWM is not the COMPE bit. Thus, the Fault mode (FMOD) within the PWM configuration must be FMOD = 1.

37.5.18. Battery Voltage Measurement

The battery voltage is connected to one input of the ADC to perform voltage measurement. To obtain an accurate measurement under conditions that are close to real-life, an internal resistive

© 2026 Microchip Technology Inc. and its subsidiaries

load is automatically attached to the battery voltage output when the channel is selected for conversion.

When VBAT is selected for measurement, the conversion result is 0.6 x VBAT.

Two resistive load values can be selected by configuring ADC_ACR.ZBAT.

For information on the ADC channel to which VBAT is connected, see the figure Block Diagram.

Refer to the section “Electrical Characteristics” for the resistive load value.

37.5.19. Register Write Protection

To prevent any single software error from corrupting ADC behavior, certain registers in the address space can be write-protected by setting the bits WPEN and WPITEN in the ADC Write Protection Mode Register (ADC_WPMR).

If a write access to the protected registers is detected, the WPVS flag in the ADC Write Protection Status Register (ADC_WPSR) is set and the field WPVSRC indicates the register in which the write access has been attempted.

The WPVS flag is automatically reset by reading ADC_WPSR.

The following registers are write-protected when ADC_WPMR.WPEN is set:

- ADC Mode Register
- ADC Channel Sequence Register 1
- ADC Channel Enable Register
- ADC Channel Disable Register
- ADC Temperature Sensor Mode Register
- ADC Temperature Compare Window Register
- ADC Extended Mode Register
- ADC FIFO Mode Register
- ADC Compare Window Register
- ADC Channel Configuration Register
- ADC Analog Control Register
- ADC Trigger Register
- ADC Correction Select Register
- ADC Correction Values Register
- ADC Channel Error Correction Register

The following registers are write-protected when ADC_WPMR.WPITEN is set:

- ADC Interrupt Enable Register
- ADC Interrupt Disable Register

The following register is write-protected when ADC_WPMR.WPCREN is set:

- ADC Control Register

© 2026 Microchip Technology Inc. and its subsidiaries

37.6. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	ADC_CR	31:24								
		23:16								
		15:8								
		7:0				CMPRST	SWFIFO		START	SWRST
0x04	ADC_MR	31:24	USEQ	ALWAYS1	TRANSFER[1:0]		TRACKTIM[3:0]			
		23:16	ANACH				STARTUP[3:0]			
		15:8	PRESCAL[7:0]							
		7:0		FWUP	SLEEP		TRGSEL[2:0]			
0x08	ADC_SEQR1	31:24	USCH8[3:0]				USCH7[3:0]			
		23:16	USCH6[3:0]				USCH5[3:0]			
		15:8	USCH4[3:0]				USCH3[3:0]			
		7:0	USCH2[3:0]				USCH1[3:0]			
0x0C ... 0x0F	Reserved									
0x10	ADC_CHER	31:24								
		23:16								
		15:8								
		7:0	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
0x14	ADC_CHDR	31:24								
		23:16								
		15:8								
		7:0	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
0x18	ADC_CHSR	31:24								
		23:16								
		15:8								
		7:0	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
0x1C ... 0x1F	Reserved									
0x20	ADC_LCDR	31:24				CHNBOSR[4:0]				
		23:16								
		15:8	LDATA[15:8]							
		7:0	LDATA[7:0]							
0x20	ADC_LCDR (NO_OSR)	31:24								
		23:16								
		15:8	NO_OSR_CHNBOSR[3:0]				NO_OSR_LDATA[11:8]			
		7:0	NO_OSR_LDATA[7:0]							
0x24	ADC_IER	31:24				RXBUFF	ENDRX	COMPE	GOVRE	DRDY
		23:16					TEMPCHG	EOS		
		15:8								
		7:0		SMEV	RXOVR	RXUDR		RXFULL	RXEMPTY	RXRDY
0x28	ADC_IDR	31:24				RXBUFF	ENDRX	COMPE	GOVRE	DRDY
		23:16					TEMPCHG	EOS		
		15:8								
		7:0		SMEV	RXOVR	RXUDR		RXFULL	RXEMPTY	RXRDY
0x2C	ADC_IMR	31:24				ENDRX	COMPE	GOVRE	DRDY	
		23:16					EOS	SMEV		
		15:8								
		7:0			RXOVR	RXUDR		RXFULL	RXEMPTY	RXRDY
0x30	ADC_ISR	31:24				RXBUFF	ENDRX	COMPE	GOVRE	DRDY
		23:16					TEMPCHG	EOS		
		15:8								
		7:0		SMEV	RXOVR	RXUDR		RXFULL	RXEMPTY	RXRDY
0x34	ADC_EOC_IER	31:24								
		23:16								
		15:8								
		7:0	EOC7	EOC6	EOC5	EOC4	EOC3	EOC2	EOC1	EOC0

© 2026 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x38	ADC_EOC_IDR	31:24								
		23:16								
		15:8								
		7:0	EOC7	EOC6	EOC5	EOC4	EOC3	EOC2	EOC1	EOC0
0x3C	ADC_EOC_IMR	31:24								
		23:16								
		15:8								
		7:0	EOC7	EOC6	EOC5	EOC4	EOC3	EOC2	EOC1	EOC0
0x40	ADC_EOC_ISR	31:24								
		23:16								
		15:8								
		7:0	EOC7	EOC6	EOC5	EOC4	EOC3	EOC2	EOC1	EOC0
0x44	ADC_TEMPMR	31:24								
		23:16								
		15:8								
		7:0			TEMPCMPMOD[1:0]					TEMPON
0x48	ADC_TEMPCWR	31:24					THIGHTHRES[11:8]			
		23:16	THIGHTHRES[7:0]							
		15:8					TLOWTHRES[11:8]			
		7:0	TLOWTHRES[7:0]							
0x4C	ADC_OVER	31:24								
		23:16								
		15:8								
		7:0	OVRE7	OVRE6	OVRE5	OVRE4	OVRE3	OVRE2	OVRE1	OVRE0
0x50	ADC_EMR	31:24			ADCMODE[1:0]			SIGNMODE[1:0]		TAG
		23:16	TRACKX[1:0]		SRCCLK	ASTE		OSR[2:0]		
		15:8			CMPFILTER[1:0]				CMPALL	CMPSEL[4]
		7:0	CMPSEL[3:0]					CMPTYPE	CMPMODE[1:0]	
0x54	ADC_CWR	31:24	HIGHTHRES[15:8]							
		23:16	HIGHTHRES[7:0]							
		15:8	LOWTHRES[15:8]							
		7:0	LOWTHRES[7:0]							
0x58 ... 0x5B	Reserved									
0x5C	ADC_CCR	31:24								
		23:16								
		15:8								
		7:0	DIFF7	DIFF6	DIFF5	DIFF4	DIFF3	DIFF2	DIFF1	DIFF0
0x60	ADC_CDR0	31:24								
		23:16								
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x64	ADC_CDR1	31:24								
		23:16								
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x68	ADC_CDR2	31:24								
		23:16								
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x6C	ADC_CDR3	31:24								
		23:16								
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x70	ADC_CDR4	31:24								
		23:16								
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							

© 2026 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x74	ADC_CDR5	31:24								
		23:16								
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x78	ADC_CDR6	31:24								
		23:16								
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x7C	ADC_CDR7	31:24								
		23:16								
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x80 ... 0xDF	Reserved									
0xE0	ADC_ACR	31:24								
		23:16		SMVT	SMEN	INTVREFEN				
		15:8								
		7:0						ZBAT		
0xE4	ADC_FMR	31:24								
		23:16	FIFOCNT[7:0]							
		15:8								
		7:0							ENLEVEL	ENFIFO
0xE8 ... 0x012F	Reserved									
0x0130	ADC_TRGR	31:24	TRGPER[23:16]							
		23:16	TRGPER[15:8]							
		15:8	TRGPER[7:0]							
		7:0						TRGMOD[2:0]		
0x0134	ADC_COSR	31:24								
		23:16								
		15:8								
		7:0				CSEL[4:0]				
0x0138	ADC_CVR	31:24	GAINCORR[15:8]							
		23:16	GAINCORR[7:0]							
		15:8	OFFSETCORR[15:8]							
		7:0	OFFSETCORR[7:0]							
0x013C	ADC_CECR	31:24								
		23:16								
		15:8								
		7:0	ECORR7	ECORR6	ECORR5	ECORR4	ECORR3	ECORR2	ECORR1	ECORR0
0x0140 ... 0x0143	Reserved									
0x0144	ADC_SR	31:24								
		23:16								
		15:8								
		7:0								VADCSM
0x0148 ... 0x014B	Reserved									
0x014C	ADC_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0						WPCREN	WPITEN	WPEN
0x0150	ADC_WPSR	31:24								
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0								WPVS

© 2026 Microchip Technology Inc. and its subsidiaries

37.6.1. ADC Control Register

Name: ADC_CR
Offset: 0x00
Reset: –
Property: Write-only

This register can only be written if the WPCREN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				CMPRST	SWFIFO		START	SWRST
Access				W	W		W	W
Reset				–	–		–	–

Bit 4 – CMPRST Comparison Restart

Value	Description
0	No effect.
1	Stops the conversion result storage until the next comparison match.

Bit 3 – SWFIFO Software FIFO Reset

Value	Description
0	No effect.
1	Resets the internal FIFO, simulating a hardware reset.

Bit 1 – START Start Conversion

Value	Description
0	No effect.
1	Begins analog-to-digital conversion.

Bit 0 – SWRST Software Reset

Value	Description
0	No effect.
1	Resets the ADC, simulating a hardware reset.

© 2026 Microchip Technology Inc. and its subsidiaries

37.6.2. ADC Mode Register

Name: ADC_MR
Offset: 0x04
Reset: 0x20000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	USEQ	ALWAYS1	TRANSFER[1:0]		TRACKTIM[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	ANACH				STARTUP[3:0]			
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0
Bit	15	14	13	12	11	10	9	8
	PRESCAL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
		FWUP	SLEEP		TRGSEL[2:0]			
Access		R/W	R/W		R/W	R/W	R/W	
Reset		0	0		0	0	0	

Bit 31 – USEQ User Sequence Enable

Value	Name	Description
0	NUM_ORDER	Normal mode: The controller converts channels in a simple numeric order depending only on the channel index.
1	REG_ORDER	User Sequence mode: The sequence respects what is defined in ADC_SEQR1 and can be used to convert the same channel several times.

Bit 30 – ALWAYS1 Must always be written to 1

Bits 29:28 – TRANSFER[1:0] Transfer Time
Must be set to 2 to ensure the optimal transfer time.

Bits 27:24 – TRACKTIM[3:0] Tracking Time

ADC_EMR.TRACKX	TRACKTIM		
	0 to 3	4 to 14	15
0	$6 \times t_{ADCCLK}$		$7 \times t_{ADCCLK}$
1	$6 \times t_{ADCCLK}$	$([4 \times (TRACKTIM + 1)] - 10) \times t_{ADCCLK}$	
2	Not Applicable	$([8 \times (TRACKTIM + 1)] - 10) \times t_{ADCCLK}$	
3	Not Applicable	$([16\times (TRACKTIM + 1)] - 10) \times t_{ADCCLK}$	

Bit 23 – ANACH Analog Change

Value	Name	Description
0	NONE	No analog change on channel switching: DIFF0 is used for all channels.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Name	Description
1	ALLOWED	Allows different analog settings for each channel. See ADC_CCR.

Bits 19:16 – STARTUP[3:0] Start-Up Time

Value	Name	Description
0	SUT0	0 period of ADCCLK
1	SUT8	8 periods of ADCCLK
2	SUT16	16 periods of ADCCLK
3	SUT24	24 periods of ADCCLK
4	SUT64	64 periods of ADCCLK
5	SUT80	80 periods of ADCCLK
6	SUT96	96 periods of ADCCLK
7	SUT112	112 periods of ADCCLK
8	SUT512	512 periods of ADCCLK
9	SUT576	576 periods of ADCCLK
10	SUT640	640 periods of ADCCLK
11	SUT704	704 periods of ADCCLK
12	SUT768	768 periods of ADCCLK
13	SUT832	832 periods of ADCCLK
14	SUT896	896 periods of ADCCLK
15	SUT960	960 periods of ADCCLK

Bits 15:8 – PRESCAL[7:0] Prescaler Rate Selection

PRESCAL = ($f_{peripheral\ clock}$ / (2 × f_{ADCCLK})) – 1.

Bit 6 – FWUP Fast Wakeup

Value	Name	Description
0	OFF	If SLEEP is 1, then both ADC core and reference voltage circuitry are OFF between conversions.
1	ON	If SLEEP is 1, then Fast Wake-up Sleep mode: The voltage reference is ON between conversions and ADC core is OFF.

Bit 5 – SLEEP Sleep Mode

Value	Name	Description
0	NORMAL	Normal mode: The ADC core and reference voltage circuitry are kept ON between conversions.
1	SLEEP	Sleep mode: The wake-up time can be modified by programming the FWUP bit.

Bits 3:1 – TRGSEL[2:0] Trigger Selection

The trigger selection can be performed only if ADC_TRGR.TRGMOD = 1, 2 or 3.

Value	Name	Description
0	ADC_TRIG0	PWM event line 0
1	ADC_TRIG1	TIOA0 TC0
2	ADC_TRIG2	TIOA1 TC0
3	ADC_TRIG3	TIOA2 TC0
4	ADC_TRIG4	TIOA0 TC1
5	ADC_TRIG5	TIOA1 TC1
6	ADC_TRIG6	RTCOUT0

© 2026 Microchip Technology Inc. and its subsidiaries

37.6.3. ADC Channel Sequence Register 1

Name: ADC_SEQR1
Offset: 0x08
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	USCH8[3:0]				USCH7[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	USCH6[3:0]				USCH5[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	USCH4[3:0]				USCH3[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	USCH2[3:0]				USCH1[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0:3, 4:7, 8:11, 12:15, 16:19, 20:23, 24:27, 28:31, 32:35 – USCHx User Sequence Number x
This register can be used only if ADC_MR.USEQ is set to '1'.
Any USCHx field is processed only if the bit CHx-1 in ADC_CHSR reads logical '1', else any value written in USCHx does not add the corresponding channel in the conversion sequence.
Configuring the same value in different fields leads to multiple samples of the same channel during the conversion sequence. This can be done consecutively, or not, according to user needs.
Example: For each trigger event, to obtain the "CH3 CH1 CH0 CH4 CH4" conversion sequence, use the following settings:
ADC_SEQR1.USCH1=3, ADC_CHSR.CH0=1
ADC_SEQR1.USCH2=1, ADC_CHSR.CH1=1
ADC_SEQR1.USCH3=0, ADC_CHSR.CH2=1
ADC_SEQR1.USCH4=4, ADC_CHSR.CH3=1
ADC_SEQR1.USCH5=4, ADC_CHSR.CH4=1

© 2026 Microchip Technology Inc. and its subsidiaries

37.6.4. ADC Channel Enable Register

Name: ADC_CHER
Offset: 0x10
Reset: –
Property: Write-only

If ADC_MR.USEQ = 1, CHx corresponds to the enable of sequence number x+1 described in ADC_SEQR1 (e.g. CH0 enables sequence number USCH1). For example, if Differential mode is required on channel 0, input pins AD0 and AD1 are used. In this case, only channel 0 must be enabled by writing a 1 to ADC_CHER.CH0.

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7 – CHx Channel x Enable

Value	Description
0	No effect.
1	Enables the corresponding channel.

© 2026 Microchip Technology Inc. and its subsidiaries

37.6.5. ADC Channel Disable Register

Name: ADC_CHDR
Offset: 0x14
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

WARNING If the corresponding channel is disabled during a conversion, or if it is disabled and then reenabled during a conversion, its associated data and corresponding ADC_EOC_ISR.EOCx, ADC_ISR.GOVRE and ADC_OVER.OVREx flags are unpredictable.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7 – CHx Channel x Disable

Value	Description
0	No effect.
1	Disables the corresponding channel.

© 2026 Microchip Technology Inc. and its subsidiaries

37.6.6. ADC Channel Status Register

Name: ADC_CHSR
Offset: 0x18
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7 – CHx Channel x Status
As an example, when ADC_MR.USEQ=1 and ADC_CHSR.CH2=1, the channel configured in ADC_SEQ1R.USCH3 is part of the sequence of conversions.

Value	Description
0	The corresponding channel (or part of sequence, see ADC_SEQyR.USCHx) is disabled..
1	The corresponding channel (or part of sequence, see ADC_SEQyR.USCHx) is enabled.

© 2026 Microchip Technology Inc. and its subsidiaries

37.6.7. ADC Last Converted Data Register

Name: ADC_LCDR
Offset: 0x20
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
				CHNBOSR[4:0]				
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	LDATA[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	LDATA[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 28:24 – CHNBOSR[4:0] Channel Number in Oversampling Mode
Indicates the last converted channel when ADC_EMR.TAG is set and ADC_EMR.OSR is not equal to 0. If ADC_EMR.TAG is not set, CHNBOSR = 0.

Bits 15:0 – LDATA[15:0] Last Data Converted
The analog-to-digital conversion data is placed into this register at the end of a conversion and remains until a new conversion is completed.
If OSR = 0 and TAG = 1 in ADC_EMR, the 4 MSBs of LDATA carry the channel number to obtain a packed system memory buffer made of 1 converted data stored in a half-word (16 bits) instead of 1 converted data in a 32-bit word, thus dividing by 2 the size of the memory buffer. See ADC_LCDR (NO_OSR).

© 2026 Microchip Technology Inc. and its subsidiaries

37.6.8. ADC Last Converted Data Register (NO_OSR)

Name: ADC_LCDR (NO_OSR)
Offset: 0x20
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	NO_OSR_CHNBOSR[3:0]				NO_OSR_LDATA[11:8]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	NO_OSR_LDATA[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:12 – NO_OSR_CHNBOSR[3:0] Channel Number when No Oversampling
Indicates the last converted channel when ADC_EMR.TAG is set and the ADC_EMR.OSR = 0. If ADC_EMR.TAG is not set, NO_OSR_CHNB = 0.

Bits 11:0 – NO_OSR_LDATA[11:0] Last Data Converted when No Oversampling
The analog-to-digital conversion data is placed into this register at the end of a conversion and remains until a new conversion is completed.

© 2026 Microchip Technology Inc. and its subsidiaries

37.6.9. ADC Interrupt Enable Register

Name: ADC_IER
Offset: 0x24
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the ADC Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
				RXBUFF	ENDRX	COMPE	GOVRE	DRDY
Access				W	W	W	W	W
Reset				–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
					TEMPCHG	EOS		
Access					W	W		
Reset					–	–		

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		SMEV	RXOVR	RXUDR		RXFULL	RXEMPTY	RXRDY
Access		W	W	W		W	W	W
Reset		–	–	–		–	–	–

Bit 28 – RXBUFF Receive Buffer Full Interrupt Enable

Bit 27 – ENDRX End of Receive Transfer Interrupt Enable

Bit 26 – COMPE Comparison Event Interrupt Enable

Bit 25 – GOVRE General Overrun Error Interrupt Enable

Bit 24 – DRDY Data Ready Interrupt Enable

Bit 19 – TEMPCHG Temperature Change Interrupt Enable

Bit 18 – EOS End Of Sequence Interrupt Enable

Bit 6 – SMEV Supply Monitor Event Interrupt Enable

Bit 5 – RXOVR Receive Over Flow Interrupt Enable

Bit 4 – RXUDR Receive Under Flow Interrupt Enable

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 2 – RXFULL Receive FIFO Full Interrupt Enable

Bit 1 – RXEMPTY Receive FIFO Empty Interrupt Enable

Bit 0 – RXRDY Receive Ready Interrupt Enable

© 2026 Microchip Technology Inc. and its subsidiaries

37.6.10. ADC Interrupt Disable Register

Name: ADC_IDR
Offset: 0x28
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the ADC Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
				RXBUFF	ENDRX	COMPE	GOVRE	DRDY
Access				W	W	W	W	W
Reset				–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
					TEMPCHG	EOS		
Access					W	W		
Reset					–	–		

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		SMEV	RXOVR	RXUDR		RXFULL	RXEMPTY	RXRDY
Access		W	W	W		W	W	W
Reset		–	–	–		–	–	–

Bit 28 – RXBUFF Receive Buffer Full Interrupt Disable

Bit 27 – ENDRX End of Receive Transfer Interrupt Disable

Bit 26 – COMPE Comparison Event Interrupt Disable

Bit 25 – GOVRE General Overrun Error Interrupt Disable

Bit 24 – DRDY Data Ready Interrupt Disable

Bit 19 – TEMPCHG Temperature Change Interrupt Disable

Bit 18 – EOS End Of Sequence Interrupt Disable

Bit 6 – SMEV Supply Monitor Event Interrupt Disable

Bit 5 – RXOVR Receive Over Flow Interrupt Disable

Bit 4 – RXUDR Receive Under Flow Interrupt Disable

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 2 – RXFULL Receive FIFO Full Interrupt Disable

Bit 1 – RXEMPTY Receive FIFO Empty Interrupt Disable

Bit 0 – RXRDY Receive Ready Interrupt Disable

© 2026 Microchip Technology Inc. and its subsidiaries

37.6.11. ADC Interrupt Mask Register

Name: ADC_IMR
Offset: 0x2C
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is disabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
				ENDRX	COMPE	GOVRE	DRDY	
Access				R	R	R	R	
Reset				0	0	0	0	

Bit	23	22	21	20	19	18	17	16
					EOS	SMEV		
Access					R	W		
Reset					0	0		

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			RXOVR	RXUDR		RXFULL	RXEMPTY	RXRDY
Access			R	R		R	R	R
Reset			0	0		0	0	0

Bit 28 – ENDRX End of Receive Transfer Interrupt Mask

Bit 27 – COMPE Comparison Event Interrupt Mask

Bit 26 – GOVRE General Overrun Error Interrupt Mask

Bit 25 – DRDY Data Ready Interrupt Mask

Bit 19 – EOS End Of Sequence Interrupt Mask

Bit 18 – SMEV Supply Monitor Event Interrupt Mask

Bit 5 – RXOVR Receive Over Flow Interrupt Mask

Bit 4 – RXUDR Receive Under Flow Interrupt Mask

Bit 2 – RXFULL Receive FIFO Full Interrupt Mask

Bit 1 – RXEMPTY Receive FIFO Empty Interrupt Mask

Bit 0 – RXRDY Receive Ready Interrupt Mask

© 2026 Microchip Technology Inc. and its subsidiaries

37.6.12. ADC Interrupt Status Register

Name: ADC_ISR
Offset: 0x30
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
				RXBUFF	ENDRX	COMPE	GOVRE	DRDY
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
					TEMPCHG	EOS		
Access					R	R		
Reset					0	0		

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		SMEV	RXOVR	RXUDR		RXFULL	RXEMPTY	RXRDY
Access		W	R	R		R	R	R
Reset		0	0	0		0	0	0

Bit 28 – RXBUFF Receive Buffer Full (cleared by writing ADC_RCR or ADC_RNCR)
ADC_RCR and ADC_RNCR are PDC registers

Value	Description
0	ADC_RCR or ADC_RNCR has a value other than 0.
1	Both ADC_RCR and ADC_RNCR have a value of 0.

Bit 27 – ENDRX End of Receive Transfer (cleared by writing ADC_RCR or ADC_RNCR)
ADC_RCR and ADC_RNCR are PDC registers

Value	Description
0	The Receive Counter register has not reached 0 since the last write in ADC_RCR or ADC_RNCR.
1	The Receive Counter register has reached 0 since the last write in ADC_RCR or ADC_RNCR.

Bit 26 – COMPE Comparison Event (cleared on read)

Value	Description
0	No comparison event occurred since the last read of ADC_ISR.
1	At least one comparison event (defined in ADC_EMR and ADC_CWR) has occurred since the last read of ADC_ISR.

Bit 25 – GOVRE General Overrun Error (cleared on read)

Value	Description
0	No general overrun error occurred since the last read of ADC_ISR.
1	At least one general overrun error has occurred since the last read of ADC_ISR.

Bit 24 – DRDY Data Ready (automatically set / cleared)

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	No data has been converted since the last read of ADC_LCDR.
1	At least one data has been converted and is available in ADC_LCDR.

Bit 19 – TEMPCHG Temperature Change (cleared on read)

Value	Description
0	There is no comparison match (defined in the Temperature Compare Window register (ADC_TEMPCWR) since the last read of ADC_ISR.
1	The temperature value reported on ADC_CDRmax (max=highest index) has changed since the last read of ADC_ISR, according to what is defined in ADC_TEMPTMR and ADC_TEMPCWR.

Bit 18 – EOS End Of Sequence (cleared on read)

Value	Description
0	No sequence is in progress or the sequence is not finished. This flag is cleared when reading ADC_ISR.
1	The sequence is complete.

Bit 6 – SMEV Supply Monitor Event (cleared on read)

Value	Description
0	No supply monitor event occurred since the last read of ADC_ISR.
1	At least one supply monitor event has occurred since the last read of ADC_ISR.

Bit 5 – RXOVR Receive Over Flow (cleared on read)

Value	Description
0	No general overrun error occurred since the last read of ADC_ISR.
1	At least one general overrun error has occurred since the last read of ADC_ISR.

Bit 4 – RXUDR Receive Under Flow (cleared on read)

Value	Description
0	No general underrun error occurred since the last read of ADC_ISR.
1	At least one general underrun error has occurred since the last read of ADC_ISR.

Bit 2 – RXFULL Receive FIFO Full (cleared on read)

Value	Description
0	FIFO has not been full since the last read of ADC_ISR.
1	FIFO has been full since the last read of ADC_ISR.

Bit 1 – RXEMPTY Receive FIFO Empty (cleared on read)

Value	Description
0	FIFO has not been empty since the last read of ADC_ISR.
1	FIFO has been empty since the last read of ADC_ISR.

Bit 0 – RXRDY Receive Ready (cleared on read)

Value	Description
0	FIFO has been empty since the last read of ADC_ISR.
1	One element has been written since the last read of ADC_ISR.

© 2026 Microchip Technology Inc. and its subsidiaries

37.6.13. ADC End Of Conversion Interrupt Enable Register

Name: ADC_EOC_IER
Offset: 0x34
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	EOC7	EOC6	EOC5	EOC4	EOC3	EOC2	EOC1	EOC0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7 – EOCx End of Conversion Interrupt Enable x

Value	Description
0	No effect.
1	Enables the corresponding interrupt.

© 2026 Microchip Technology Inc. and its subsidiaries

37.6.14. ADC End Of Conversion Interrupt Disable Register

Name: ADC_EOC_IDR
Offset: 0x38
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	EOC7	EOC6	EOC5	EOC4	EOC3	EOC2	EOC1	EOC0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7 – EOCx End of Conversion Interrupt Disable x

Value	Description
0	No effect.
1	Disables the corresponding interrupt.

© 2026 Microchip Technology Inc. and its subsidiaries

37.6.15. ADC End Of Conversion Interrupt Mask Register

Name: ADC_EOC_IMR
Offset: 0x3C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	EOC7	EOC6	EOC5	EOC4	EOC3	EOC2	EOC1	EOC0
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7 – EOCx End of Conversion Interrupt Mask x

Value	Description
0	The corresponding interrupt is disabled.
1	The corresponding interrupt is enabled.

© 2026 Microchip Technology Inc. and its subsidiaries

37.6.16. ADC End Of Conversion Interrupt Status Register

Name: ADC_EOC_ISR
Offset: 0x40
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	EOC7	EOC6	EOC5	EOC4	EOC3	EOC2	EOC1	EOC0
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7 – EOCx End of Conversion x (automatically set / cleared)

Value	Description
0	The corresponding analog channel is disabled, or the conversion is not finished. This flag is cleared when reading the corresponding ADC_CDRx registers.
1	The corresponding analog channel is enabled and conversion is complete.

© 2026 Microchip Technology Inc. and its subsidiaries

37.6.17. ADC Temperature Sensor Mode Register

Name: ADC_TEMPMR
Offset: 0x44
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			TEMPCMPMOD[1:0]					TEMPON
Access			R/W	R/W				R/W
Reset			0	0				0

Bits 5:4 – TEMPCMPMOD[1:0] Temperature Comparison Mode

Value	Name	Description
0	LOW	Generates the TEMPCHG flag in ADC_ISR when the converted data is lower than the low threshold of the window.
1	HIGH	Generates the TEMPCHG flag in ADC_ISR when the converted data is higher than the high threshold of the window.
2	IN	Generates the TEMPCHG flag in ADC_ISR when the converted data is in the comparison window.
3	OUT	Generates the TEMPCHG flag in ADC_ISR when the converted data is out of the comparison window.

Bit 0 – TEMPON Temperature Sensor On

WARNING To put the system in Low-Power mode when the temperature sensor is in use, TEMPON must be cleared and a software reset must be performed (ADC_CR.SWRST=1).

Value	Description
0	Disables the temperature sensor.
1	Enables the temperature sensor and the measurements are performed as soon as a trigger event occurs.

© 2026 Microchip Technology Inc. and its subsidiaries

37.6.18. ADC Temperature Compare Window Register

Name: ADC_TEMPCWR
Offset: 0x48
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
					THIGHTHRES[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
	THIGHTHRES[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
					TLOWTHRES[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TLOWTHRES[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 27:16 – THIGHTHRES[11:0] Temperature High Threshold
High threshold associated to ADC_TEMPMR compare settings.

Bits 11:0 – TLOWTHRES[11:0] Temperature Low Threshold
Low threshold associated to ADC_TEMPMR compare settings.

© 2026 Microchip Technology Inc. and its subsidiaries

37.6.19. ADC Overrun Status Register

Name: ADC_OVER
Offset: 0x4C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	OVRE7	OVRE6	OVRE5	OVRE4	OVRE3	OVRE2	OVRE1	OVRE0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7 – OVREx Overrun Error x
An overrun error does not always mean that the unread data has been replaced by new valid data. See Enhanced Resolution Mode and Digital Averaging Function for details.

Value	Description
0	No overrun error has occurred on the corresponding channel since the last read of ADC_OVER.
1	An overrun error has occurred on the corresponding channel since the last read of ADC_OVER.

© 2026 Microchip Technology Inc. and its subsidiaries

37.6.20. ADC Extended Mode Register

Name: ADC_EMR
Offset: 0x50
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
			ADCMODE[1:0]			SIGNMODE[1:0]		TAG
Access			R/W	R/W		R/W	R/W	R/W
Reset			0	0		0	0	0

Bit	23	22	21	20	19	18	17	16
	TRACKX[1:0]		SRCCLK	ASTE		OSR[2:0]		
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bit	15	14	13	12	11	10	9	8
			CMPFILTER[1:0]				CMPALL	CMPSEL[4]
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit	7	6	5	4	3	2	1	0
	CMPSEL[3:0]					CMPTYPE	CMPMODE[1:0]	
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bits 29:28 – ADCMODE[1:0] ADC Running Mode
See Automatic Error Correction for details on ADC Running mode.

Value	Name	Description
0	NORMAL	Normal mode of operation.
1	OFFSET_ERROR	Offset Error mode to measure the offset error. See the table ADC Running Modes.
2	GAIN_ERROR_HIGH	Gain Error mode to measure the gain error. See the table ADC Running Modes.
3	GAIN_ERROR_LOW	Gain Error mode to measure the gain error. See the table ADC Running Modes.

Bits 26:25 – SIGNMODE[1:0] Sign Mode
If conversion results are signed and resolution is below 16 bits, the sign is extended up to bit 15 (for example, 0xF43 for 12-bit resolution is read as 0xFF43 and 0x467 is read as 0x0467). See Conversion Results Format.

Value	Name	Description
0	SE_UNSG_DF_SIGN	Single-ended channels: unsigned conversions Differential channels: signed conversions
1	SE_SIGN_DF_UNSG	Single-ended channels: signed conversions Differential channels: unsigned conversions
2	ALL_UNSIGNED	All channels: unsigned conversions
3	ALL_SIGNED	All channels: signed conversions

Bit 24 – TAG ADC_LCDR Tag

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	Sets ADC_LCDR.NO_OSR_CHNB/CHNBOSR to zero.
1	Appends the channel number to the conversion result in ADC_LCDR.

Bits 23:22 – TRACKX[1:0] Tracking Time x4, x8 or x16

Value	Name	Description
0	TRACKTIMx1	ADC_MR.TRACKTIM effect is multiplied by 1.
1	TRACKTIMx4	ADC_MR.TRACKTIM effect is multiplied by 4.
2	TRACKTIMx8	ADC_MR.TRACKTIM effect is multiplied by 8.
3	TRACKTIMx16	ADC_MR.TRACKTIM effect is multiplied by 16.

Bit 21 – SRCCLK External Clock Selection

Value	Name	Description
0	PERIPH_CLK	The peripheral clock is the source for the ADC prescaler.
1	GCLK	GCLK is the source clock for the ADC prescaler, thus the ADC clock can be independent of the core/peripheral clock.

Bit 20 – ASTE Averaging on Single Trigger Event

Value	Name	Description
0	MULTI_TRIG_AVERAGE	The average requests several trigger events.
1	SINGLE_TRIG_AVERAGE	The average requests only one trigger event.

Bits 18:16 – OSR[2:0] Over Sampling Rate

Value	Name	Description
0	NO_AVERAGE	No averaging. ADC sample rate is maximum.
1	OSR4	1-bit enhanced resolution by averaging. ADC sample rate divided by 4.
2	OSR16	2-bit enhanced resolution by averaging. ADC sample rate divided by 16.
3	OSR64	1-bit enhanced resolution by averaging. ADC sample rate divided by 64.
4	OSR256	2-bit enhanced resolution by averaging. ADC sample rate divided by 256.

Bits 13:12 – CMPFILTER[1:0] Compare Event Filtering

Number of consecutive compare events necessary to raise the flag = CMPFILTER+1
When programmed to 0, the flag rises as soon as an event occurs.
See Comparison Window when using the filtering option (CMPFILTER > 0).

Bit 9 – CMPALL Compare All Channels

Value	Description
0	Only the channel indicated in CMPSEL is compared.
1	All channels are compared.

Bits 8:4 – CMPSEL[4:0] Comparison Selected Channel

If CMPALL = 0: CMPSEL indicates which channel has to be compared.
If CMPALL = 1: No effect.

Bit 2 – CMPTYPE Comparison Type

Value	Name	Description
0	FLAG_ONLY	Any conversion is performed and comparison function drives the COMPE flag.
1	START_CONDITION	Comparison conditions must be met to start the storage of all conversions until ADC_CR.CMPRST is set.

Bits 1:0 – CMPMODE[1:0] Comparison Mode

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Name	Description
0	LOW	When the converted data is lower than the low threshold of the window, generates the COMPE flag in ADC_ISR or, in Partial Wake-up mode, defines the conditions to exit the system from Wait mode.
1	HIGH	When the converted data is higher than the high threshold of the window, generates the COMPE flag in ADC_ISR or, in Partial Wake-up mode, defines the conditions to exit the system from Wait mode.
2	IN	When the converted data is in the comparison window, generates the COMPE flag in ADC_ISR or, in Partial Wake-up mode, defines the conditions to exit the system from Wait mode.
3	OUT	When the converted data is out of the comparison window, generates the COMPE flag in ADC_ISR or, in Partial Wake-up mode, defines the conditions to exit the system from Wait mode.

© 2026 Microchip Technology Inc. and its subsidiaries

37.6.21. ADC Compare Window Register

Name: ADC_CWR
Offset: 0x54
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	HIGHTHRES[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	HIGHTHRES[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	LOWTHRES[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	LOWTHRES[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – HIGHTHRES[15:0] High Threshold
High threshold associated to ADC_EMR compare settings.

Bits 15:0 – LOWTHRES[15:0] Low Threshold
Low threshold associated to ADC_EMR compare settings.

© 2026 Microchip Technology Inc. and its subsidiaries

37.6.22. ADC Channel Configuration Register

Name: ADC_CCR
Offset: 0x5C
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	DIFF7	DIFF6	DIFF5	DIFF4	DIFF3	DIFF2	DIFF1	DIFF0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7 – DIFFx Differential Inputs for Channel x

Value	Description
0	Corresponding channel is set in Single-ended mode.
1	Corresponding channel is set in Differential mode.

© 2026 Microchip Technology Inc. and its subsidiaries

37.6.23. ADC Channel Data Register

Name: ADC_CDRx
Offset: 0x60 + x*0x04 [x=0..7]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – DATA[15:0] Converted Data
The analog-to-digital conversion data is placed into this register at the end of a conversion and remains until a new conversion is completed. ADC_CDRx is only loaded if the corresponding analog channel is enabled.

© 2026 Microchip Technology Inc. and its subsidiaries

37.6.24. ADC Analog Control Register

Name: ADC_ACR
Offset: 0xE0
Reset: 0x00001100
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

By default, bits 12 and 13 are set to 1 and 0, respectively, and must not be modified.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
		SMVT	SMEN	INTVREFEN				
Access		R/W	R/W	R/W				
Reset		0	0	0				

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						ZBAT		
Access						R/W		
Reset						0		

Bit 22 – SMVT Supply Monitor Voltage Threshold

Value	Description
0	Voltage threshold is in low range. Refer to the section “Electrical Characteristics”.
1	Voltage threshold is in high range. Refer to the section “Electrical Characteristics”.

Bit 21 – SMEN Supply Monitor Enable

Value	Description
0	Disables the supply monitor.
1	Enables both the voltage reference and the supply monitor.

Bit 20 – INTVREFEN ADC Internal Positive Voltage Reference Enable

Value	Description
0	Disables the internal positive voltage reference for the ADC conversions (the VREFP pin can be externally driven).
1	Enables the internal positive reference voltage for the ADC conversions (a capacitor can be connected to the VREFP pin).

Bit 2 – ZBAT VBAT Resistive Load Selection
There is no internal load on VBAT when the VBAT channel is not selected for conversion (only leakage current).

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	Selects the high value of the resistive load only attached on VBAT when the channel is selected for conversion. Refer to the section “Electrical Characteristics” for further details.
1	Selects the low value of the resistive load only attached on VBAT when the channel is selected for conversion.

© 2026 Microchip Technology Inc. and its subsidiaries

37.6.25. ADC FIFO Mode Register

Name: ADC_FMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	FIFOCNT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							ENLEVEL	ENFIFO
Access							R/W	R/W
Reset							0	0

Bits 23:16 – FIFOCNT[7:0] FIFO Count
Number of conversions available in the FIFO (not a source of interrupt).

Bit 1 – ENLEVEL Enable Level
This bit must always be written to ‘0’.

Bit 0 – ENFIFO Enable FIFO

Value	Description
0	FIFO is disabled.
1	FIFO is enabled.

© 2026 Microchip Technology Inc. and its subsidiaries

37.6.26. ADC Trigger Register

Name: ADC_TRGR
Offset: 0x130
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	TRGPER[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	TRGPER[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	TRGPER[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
						TRGMOD[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 31:8 – TRGPER[23:0] Trigger Period
Effective only if TRGMOD defines a periodic trigger.
Defines the periodic trigger period, with the following equation:
Trigger Period = (TRGPER + 1) / ADCCLK
The minimum time between two consecutive trigger events must be strictly greater than the duration time of the longest conversion sequence depending on the configuration of registers ADC_MR, ADC_CHSR, ADC_SEQRx and ADC_TSMR.

Bits 2:0 – TRGMOD[2:0] Trigger Mode

Value	Name	Description
0	NO_TRIGGER	No trigger, only software trigger can start conversions
1	EXT_TRIG_RISE	Rising edge of the selected trigger event, defined in ADC_MR.TRGSEL
2	EXT_TRIG_FALL	Falling edge of the selected trigger event
3	EXT_TRIG_ANY	Any edge of the selected trigger event
4		–
5	PERIOD_TRIG	ADC internal periodic trigger (see TRGPER)
6	CONTINUOUS	Continuous mode, Free Run mode

© 2026 Microchip Technology Inc. and its subsidiaries

37.6.27. ADC Correction Select Register

Name: ADC_COSR
Offset: 0x134
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				CSEL[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bits 4:0 – CSEL[4:0] Channel Correction Select
Selects the channel to be displayed in ADC_CVR.

© 2026 Microchip Technology Inc. and its subsidiaries

37.6.28. ADC Correction Values Register

Name: ADC_CVR
Offset: 0x138
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	GAINCORR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	GAINCORR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	OFFSETCORR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	OFFSETCORR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – GAINCORR[15:0] Gain Correction
Gain correction to apply on converted data. Only bits 0 to 15 are relevant (other bits are ignored and read as 0).

Bits 15:0 – OFFSETCORR[15:0] Offset Correction
Offset correction to apply on converted data. The offset is signed (2's complement), only bits 0 to 11 are relevant (other bits are ignored and read as 0).

© 2026 Microchip Technology Inc. and its subsidiaries

37.6.29. ADC Channel Error Correction Register

Name: ADC_CECR
Offset: 0x13C
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	ECORR7	ECORR6	ECORR5	ECORR4	ECORR3	ECORR2	ECORR1	ECORR0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7 – ECORRx Error Correction Enable for Channel x

Value	Description
0	Automatic error correction is disabled for channel x.
1	Automatic error correction is enabled for channel x.

© 2026 Microchip Technology Inc. and its subsidiaries

37.6.30. ADC Status Register

Name: ADC_SR
Offset: 0x144
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								VADCSM
Access								R
Reset								0

Bit 0 – VADCSM VDD ADC Supply Monitor Output

Value	Description
0	The VDD ADC voltage is not valid.
1	The VDD ADC voltage is valid.

© 2026 Microchip Technology Inc. and its subsidiaries

37.6.31. ADC Write Protection Mode Register

Name: ADC_WPMR
Offset: 0x14C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
						WPCREN	WPITEN	WPEN
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x414443	PASSWD	Writing any other value in this field aborts the write operation of the WPEN and WPITEN bits. Always reads as 0.

Bit 2 – WPCREN Write Protection Control Register Enable
See Register Write Protection for the list of write-protected registers.

Value	Description
0	Disables the write protection on control registers if WPKEY corresponds to 0x414443 (“ADC” in ASCII).
1	Enables the write protection on control registers if WPKEY corresponds to 0x414443 (“ADC” in ASCII).

Bit 1 – WPITEN Write Protection Interrupt Enable
See Register Write Protection for the list of write-protected registers.

Value	Description
0	Disables the write protection on interrupt registers if WPKEY corresponds to 0x414443 (“ADC” in ASCII).
1	Enables the write protection on interrupt registers if WPKEY corresponds to 0x414443 (“ADC” in ASCII).

Bit 0 – WPEN Write Protection Enable
See Register Write Protection for the list of write-protected registers.

Value	Description
0	Disables write protection if WPKEY corresponds to 0x414443 (“ADC” in ASCII).
1	Enables write protection if WPKEY corresponds to 0x414443 (“ADC” in ASCII).

© 2026 Microchip Technology Inc. and its subsidiaries

37.6.32. ADC Write Protection Status Register

Name: ADC_WPSR
Offset: 0x150
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	WPVSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPVS
Access								R
Reset								0

Bits 23:8 – WPVSRC[15:0] Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protection Violation Status

Value	Description
0	No write protection violation has occurred since the last read of ADC_WPSR.
1	A write protection violation has occurred since the last read of ADC_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

© 2026 Microchip Technology Inc. and its subsidiaries

38. Analog Comparator Controller (ACC)

38.1. Description

The Analog Comparator Controller (ACC) configures the analog comparator and generates an interrupt depending on user settings. The analog comparator embeds two 8-to-1 multiplexers that generate two internal inputs. These inputs are compared, resulting in a compare output. The hysteresis level, edge detection and polarity are configurable.

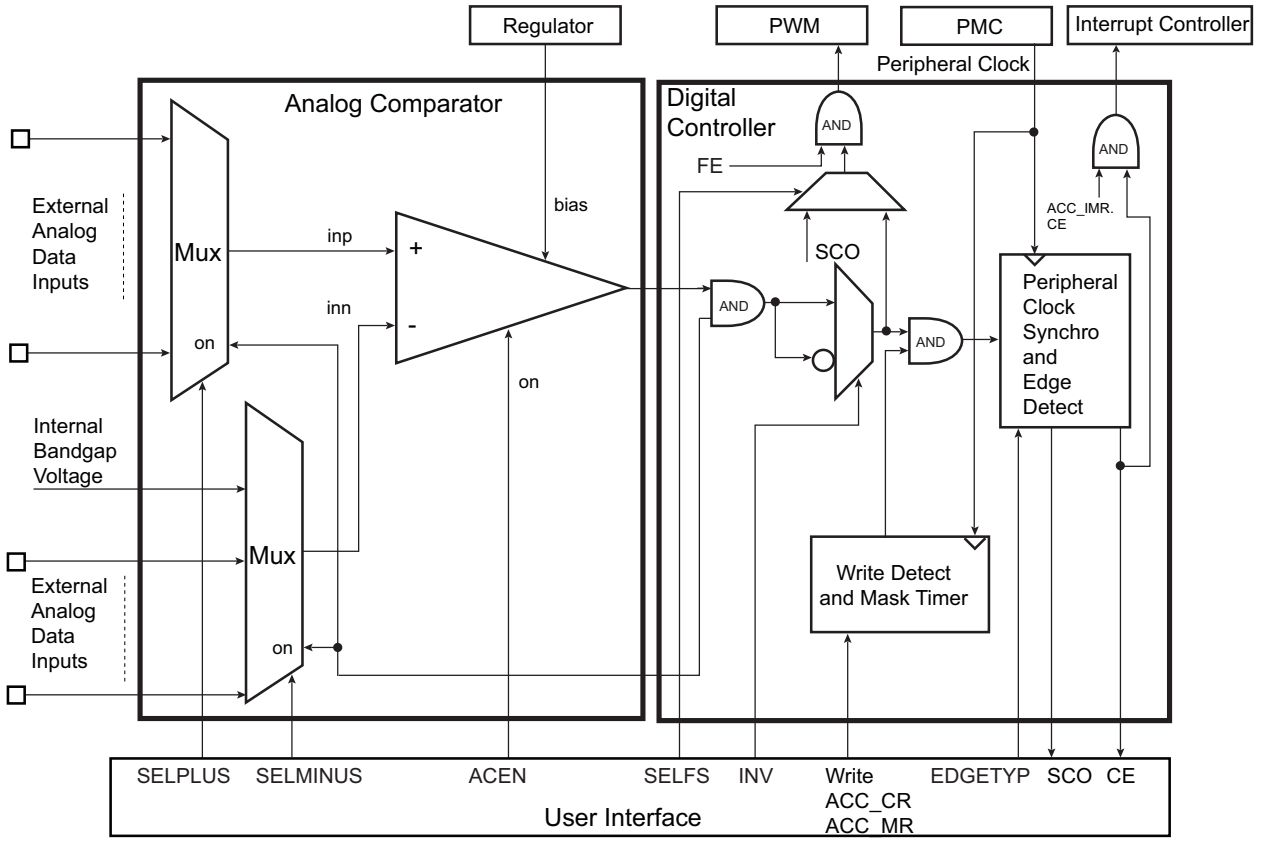
The ACC also generates a compare event which can be used by the Pulse Width Modulator (PWM).

38.2. Embedded Characteristics

- 5 User Analog Inputs Selectable for Comparison
- Bandgap Voltage Reference Selectable for Comparison
- Interrupt Generation
- Compare Event Fault Generation for PWM

38.3. Block Diagram

Figure 38-1. ACC Block Diagram



38.4. Signal Description

Table 38-1. ACC Signal Description

Pin Name	Description	Type
AD3..4, AD0..2	External analog data inputs	Input

© 2026 Microchip Technology Inc. and its subsidiaries

Table 38-1. ACC Signal Description (continued)

Pin Name	Description	Type
VREFP, VREFTEMP[1]	Internal bandgap voltage	Input

Note:

1. To enable the VBG band gap voltage (refer to the section Analog-to-Digital Converter (ADC) Controller) and enable the temperature sensor VTEMP. This also enables the VBG.

38.5. Product Dependencies

38.5.1. I/O Lines

The analog input pins (AD3..4 and AD0..2) are multiplexed with digital functions (PIO) on the IO line. By writing the SELMINUS and SELPLUS fields in the ACC Mode Register (ACC_MR), the associated IO lines are set to Analog mode.

38.5.2. Power Management

The ACC is clocked through the Power Management Controller (PMC), thus the programmer must first configure the PMC to enable the ACC clock.

Note that the voltage regulator must be activated to use the analog comparator.

38.5.3. Interrupt Sources

The ACC has an interrupt line connected to the Interrupt Controller (IC). In order to handle interrupts, the Interrupt Controller must be programmed before configuring the ACC.

38.5.4. Fault Output

The ACC has the FAULT output connected to the FAULT input of PWM. See Fault Mode and the implementation of the PWM in the product.

38.6. Functional Description

38.6.1. Description

The Analog Comparator Controller (ACC) controls the analog comparator settings and performs postprocessing of the analog comparator output.

When the analog comparator settings are modified, the output of the analog cell may be invalid. The ACC masks the output for the invalid period.

A comparison flag is triggered by an event on the output of the analog comparator and an interrupt is generated. The event on the analog comparator output can be selected among falling edge, rising edge or any edge.

The ACC registers are listed in the Register Summary.

38.6.2. Fault Mode

In Fault mode, a comparison match event is communicated by the ACC fault output which is directly and internally connected to a PWM fault input.

The source of the fault output can be configured as either a combinational value derived from the analog comparator output or as the peripheral clock resynchronized value. Refer to Block Diagram.

38.6.3. Output Masking Period

As soon as the analog comparator settings change, the output is invalid for a duration. A masking period is automatically triggered as soon as a write access is performed on the ACC_MR and ACC_CR regardless of the register data content.

The mask period is $8 \times t_{\text{peripheral clock}}$ if ACC_ACR.MSEL = 0.

© 2026 Microchip Technology Inc. and its subsidiaries

The mask period is 128 × $t_{peripheral\ clock}$ if ACC_ACR.MSEL = 1.

The masking period is reported by reading a negative value (MASK='1') on the ACC Interrupt Status register (ACC_ISR).

38.6.4. Register Write Protection

To prevent any single software error from corrupting ACC behavior, certain registers in the address space can be write-protected by setting the WPEN bit in the ACC Write Protection Mode Register (ACC_WPMR).

If a write access to a write-protected register is detected, the WPVS flag in the ACC Write Protection Status Register is set and the field WPVSRC indicates the register in which the write access has been attempted.

The WPVS bit is automatically cleared after reading the ACC_WPSR register.

The following registers can be write-protected:

- ACC Mode Register
- ACC Analog Control Register

© 2026 Microchip Technology Inc. and its subsidiaries

38.7. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	ACC_CR	31:24								
		23:16								
		15:8								
		7:0								SWRST
0x04	ACC_MR	31:24								
		23:16								
		15:8		FE	SELFS	INV		EDGETYP[1:0]		ACEN
		7:0		SELPLUS[2:0]				SELMINUS[2:0]		
0x08 ... 0x23	Reserved									
0x24	ACC_IER	31:24								
		23:16								
		15:8								
		7:0								CE
0x28	ACC_IDR	31:24								
		23:16								
		15:8								
		7:0								CE
0x2C	ACC_IMR	31:24								
		23:16								
		15:8								
		7:0								CE
0x30	ACC_ISR	31:24	MASK							
		23:16								
		15:8								
		7:0							SCO	CE
0x34 ... 0x93	Reserved									
0x94	ACC_ACR	31:24								
		23:16								
		15:8								
		7:0								MSEL
0x98 ... 0xE3	Reserved									
0xE4	ACC_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0								WPEN
0xE8	ACC_WPSR	31:24								
		23:16								
		15:8								
		7:0								WPVS

© 2026 Microchip Technology Inc. and its subsidiaries

38.7.1. ACC Control Register

Name: ACC_CR
Offset: 0x00
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								SWRST
Access								W
Reset								–

Bit 0 – SWRST Software Reset

Value	Description
0	No effect.
1	Resets the module.

© 2026 Microchip Technology Inc. and its subsidiaries

38.7.2. ACC Mode Register

Name: ACC_MR
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the ACC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		FE	SELFS	INV		EDGETYP[1:0]		ACEN
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bit	7	6	5	4	3	2	1	0
		SELPLUS[2:0]				SELMINUS[2:0]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bit 14 – FE Fault Enable

Value	Name	Description
0	DIS	The FAULT output is tied to 0.
1	EN	The FAULT output is driven by the signal defined by SELFS.

Bit 13 – SELFS Selection Of Fault Source

Value	Name	Description
0	CE	The CE flag is used to drive the FAULT output.
1	OUTPUT	The output of the analog comparator flag is used to drive the FAULT output.

Bit 12 – INV Invert Comparator Output

Value	Name	Description
0	DIS	Analog comparator output is directly processed.
1	EN	Analog comparator output is inverted prior to being processed.

Bits 10:9 – EDGETYP[1:0] Edge Type

Value	Name	Description
0	RISING	Only rising edge of comparator output
1	FALLING	Falling edge of comparator output
2	ANY	Any edge of comparator output

Bit 8 – ACEN Analog Comparator Enable

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Name	Description
0	DIS	Analog comparator disabled.
1	EN	Analog comparator enabled.

Bits 6:4 – SELPLUS[2:0] Selection For Plus Comparator Input
0..3: Selects the input to apply on analog comparator SELPLUS comparison input.

Value	Name	Description
0	AD0	Selects AD0
1	AD1	Selects AD1
2	AD2	Selects AD2
3	VREFP	Selects VREFP

Bits 2:0 – SELMINUS[2:0] Selection for Minus Comparator Input
0..3: Selects the input to apply on analog comparator SELMINUS comparison input.

Value	Name	Description
0	AD3	Selects AD3
1	AD4	Selects AD4
2	VTEMP	Selects VTEMP
3	VREFTEMP	Selects VREFTEMP

© 2026 Microchip Technology Inc. and its subsidiaries

38.7.3. ACC Interrupt Enable Register

Name: ACC_IER
Offset: 0x24
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								CE
Access								W
Reset								–

Bit 0 – CE Comparison Edge

Value	Description
0	No effect.
1	Enables the interrupt when the selected edge (defined by EDGETYP) occurs.

© 2026 Microchip Technology Inc. and its subsidiaries

38.7.4. ACC Interrupt Disable Register

Name: ACC_IDR
Offset: 0x28
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								CE
Access								W
Reset								–

Bit 0 – CE Comparison Edge

Value	Description
0	No effect.
1	Disables the interrupt when the selected edge (defined by EDGETYP) occurs.

© 2026 Microchip Technology Inc. and its subsidiaries

38.7.5. ACC Interrupt Mask Register

Name: ACC_IMR
Offset: 0x2C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								CE
Access								R
Reset								0

Bit 0 – CE Comparison Edge

Value	Description
0	The interrupt is disabled.
1	The interrupt is enabled.

© 2026 Microchip Technology Inc. and its subsidiaries

38.7.6. ACC Interrupt Status Register

Name: ACC_ISR
Offset: 0x30
Reset: 0x80000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	MASK							
Access	R							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							SCO	CE
Access							R	R
Reset							0	0

Bit 31 – MASK Flag Mask

Value	Description
0	The CE flag and SCO value are valid.
1	The CE flag and SCO value are invalid.

Bit 1 – SCO Synchronized Comparator Output

Returns an image of the analog comparator output after being preprocessed (see Block Diagram).
If INV = 0

- SCO = 0 if inn > inp
- SCO = 1 if inp > inn

If INV = 1

- SCO = 1 if inn > inp
- SCO = 0 if inp > inn

Bit 0 – CE Comparison Edge (cleared on read)

Value	Description
0	No edge occurred (defined by EDGETYP) on analog comparator output since the last read of ACC_ISR.
1	A selected edge (defined by EDGETYP) on analog comparator output occurred since the last read of ACC_ISR.

© 2026 Microchip Technology Inc. and its subsidiaries

38.7.7. ACC Analog Control Register

Name: ACC_ACR
Offset: 0x94
Reset: 0
Property: Read/Write

This register can only be written if the WPEN bit is cleared in ACC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								MSEL
Access								R/W
Reset								0

Bit 0 – MSEL Masking Period Selection

Value	Description
0	Masks AC output for 16 peripheral clock periods after any write access in ACC_MR or ACC_CR.
1	Masks AC output for 128 peripheral clock periods after any write access in ACC_MR or ACC_CR.

© 2026 Microchip Technology Inc. and its subsidiaries

38.7.8. ACC Write Protection Mode Register

Name: ACC_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

Refer to Register Write Protection for the list of registers that can be write-protected.

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPEN
Access								R/W
Reset								0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x414343	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0.

Bit 0 – WPEN Write Protection Enable

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x414343 (“ACC” in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x414343 (“ACC” in ASCII).

© 2026 Microchip Technology Inc. and its subsidiaries

38.7.9. ACC Write Protection Status Register

Name: ACC_WPSR
Offset: 0xE8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								WPVS
Access								R
Reset								0

Bit 0 – WPVS Write Protection Violation Status

Value	Description
0	No write protection violation has occurred since the last command ACC_CR.SWRST=1.
1	A write protection violation (WPEN = 1) has occurred since the last command ACC_CR.SWRST=1.

© 2026 Microchip Technology Inc. and its subsidiaries

39. Advanced Encryption Standard (AES)

39.1. Description

The Advanced Encryption Standard (AES) is compliant with the American FIPS (Federal Information Processing Standard) Publication 197 specification.

The AES supports the following confidentiality modes of operation for symmetrical key block cipher algorithms: ECB, CBC, OFB, CFB, CTR, as specified in the NIST Special Publication 800-38A Recommendation, as well as Galois/Counter Mode (GCM) as specified in the NIST Special Publication 800-38D Recommendation. It is compatible with all these modes via Peripheral DMA Controller channels, minimizing processor intervention for large buffer transfers.

The AES key can be either loaded by the software or loaded in an invisible manner from the software.

The 128-bit/192-bit/256-bit AES key is stored in the AES Key register made of four/six/eight 32-bit write-only AES Key Word registers (AES_KEYWR0–7). For a software-invisible key transfer, the Private Key Bus accesses the Private Key Internal Register from the TRNG or Flash Controller. The bit PKRS in the Extended Mode register (AES_EMR) selects either AES_KEYWRx or the Private Key Internal Register.

The 128-bit input data and initialization vector (for some modes) are each stored in four 32-bit write-only AES Input Data registers (AES_IDATAR0–3) and AES Initialization Vector registers (AES_IVR0–3).

As soon as the initialization vector, the input data and the key are configured, the encryption/decryption process may be started. Then the encrypted/decrypted data are ready to be read out on the four 32-bit AES Output Data registers (AES_ODATAR0–3) or through the PDC channels.

39.2. Embedded Characteristics

- Compliant with FIPS Publication 197, Advanced Encryption Standard (AES)
- 128-bit/192-bit/256-bit Cryptographic Key
- 10/12/14 Clock Cycles Encryption/Decryption Inherent Processing Time with a 128-bit/192-bit/256-bit Cryptographic Key
- Double Input Buffer Optimizes Runtime
- Automatic Padding supported for IPSec and SSL standards
- IPSec and SSL Protocol Layers Improved Performances (Tightly coupled with SHA)
- Support of the Modes of Operation Specified in the NIST Special Publication 800-38A and NIST Special Publication 800-38D :
  - Electronic Codebook (ECB)
  - Cipher Block Chaining (CBC) including CBC-MAC
  - Cipher Feedback (CFB)
  - Output Feedback (OFB)
  - Counter (CTR)
  - Galois/Counter Mode (GCM)
- 8, 16, 32, 64 and 128-bit Data Sizes Possible in CFB Mode
- Last Output Data Mode Allows Optimized Message Authentication Code (MAC) Generation
- Abnormal Software Access and Internal Sequencer Integrity Check Reports
- Register Write Protection
- Temporary Secure Storage for Keys

© 2026 Microchip Technology Inc. and its subsidiaries

- Private Key Bus Access to the Private Key Internal Register Not Readable from any Peripheral or Software
- Connection to PDC Channel Capabilities Optimizes Data Transfers for all Operating Modes
  - One channel for the receiver, one channel for the transmitter
  - Next buffer support

39.3. Product Dependencies

39.3.1. Power Management

The AES is clocked through the Power Management Controller (PMC), so the programmer must first configure the PMC to enable the AES clock.

39.3.2. Interrupt Sources

The AES interface has one interrupt line used for standard functions and one interrupt line used to trigger any safety or security event that may occur. Both lines are connected to the Interrupt Controller. The interrupt line for standard functions is driven by the Interrupt Mask register (AES_IMR) and the Interrupt Status register (AES_ISR), whereas the interrupt line for safety/security functions is driven by Write Protection Status register (AES_WPSR) flags. If one of the flags AES_WPSR.WPVS, AES_WPSR.CGD, AES_WPSR.SEQE or AES_WPSR.SWE is set, the interrupt line is asserted. In order to handle interrupts, the Interrupt Controller must be programmed before configuring the AES.

39.4. Functional Description

The Advanced Encryption Standard (AES) specifies a FIPS-approved cryptographic algorithm that can be used to protect electronic data. The AES algorithm is a symmetric block cipher that can encrypt (encipher) and decrypt (decipher) information.

Encryption converts data to an unintelligible form called ciphertext. Decrypting the ciphertext converts the data back into its original form, called plaintext. The CIPHER bit in the AES Mode register (AES_MR) allows selection between the encryption and the decryption processes.

The AES is capable of using cryptographic keys of 128/192/256 bits to encrypt and decrypt data in blocks of 128 bits. This 128-bit/192-bit/256-bit key is defined in the user interface AES_KEYWRx register or in the Private Key Internal Register that is only writable from the Private Key Bus.

The input to the encryption processes of the CBC, CFB, and OFB modes includes, in addition to the plaintext, a 128-bit data block called the initialization vector (IV), which must be set in AES_IVRx. The initialization vector is used in an initial step in the encryption of a message and in the corresponding decryption of the message. AES_IVRx are also used by the CTR mode to set the counter value.

39.4.1. AES Register Endianness

In Arm processor-based products, the system bus and processors manipulate data in little-endian form. The AES interface requires little-endian format words. However, in accordance with the protocol of the FIPS 197 specification, data is collected, processed and stored by the AES algorithm in big-endian form.

The following example illustrates how to configure the AES:

If the first 64 bits of a message (according to FIPS 197, i.e., big-endian format) to be processed is 0xcafedeca_01234567, then AES_IDATAR0 and AES_IDATAR1 registers must be written with the following pattern:

- AES_IDATAR0 = 0xcadefeca
- AES_IDATAR1 = 0x67452301

39.4.2. Operating Modes

The AES supports the following modes of operation:

© 2026 Microchip Technology Inc. and its subsidiaries

- ECB: Electronic Codebook
- CBC: Cipher Block Chaining
  - CBC-MAC: Useful for CMAC hardware acceleration
- OFB: Output Feedback
- CFB: Cipher Feedback
  - CFB8 (CFB where the length of the data segment is 8 bits)
  - CFB16 (CFB where the length of the data segment is 16 bits)
  - CFB32 (CFB where the length of the data segment is 32 bits)
  - CFB64 (CFB where the length of the data segment is 64 bits)
  - CFB128 (CFB where the length of the data segment is 128 bits)
- CTR: Counter
- GCM: Galois/Counter Mode

Data pre-processing, data post-processing and data chaining for the concerned modes are performed automatically. Refer to the *NIST Special Publication 800-38A* and *NIST Special Publication 800-38D* for more complete information.

Mode selection is done by configuring AES_MR.OPMOD.

When switching from an operating mode requiring the initialization vectors (e.g. CBC, GCM) to another operating mode that does not require initialization vectors (e.g. ECB) and a message of one block has been processed, initialization vector registers (AES_IVRx) must be cleared before switching to the new mode.

In CFB mode, five data sizes are possible (8, 16, 32, 64 or 128 bits), configurable by means of AES_MR.CFBS.

In CTR mode, the size of the block counter embedded in the module is 16 bits. Therefore, there is a rollover after processing 1 Mbyte of data. If the file to be processed is greater than 1 Mbyte, this file must be split into fragments of 1 Mbyte or less for the first fragment if the initial value of the counter is greater than 0. Prior to loading the first fragment into AES_IDATARx, AES_IVRx must be fully programmed with the initial counter value. For any fragment, after the transfer is completed and prior to transferring the next fragment, AES_IVRx must be programmed with the appropriate counter value.

If the initial value of the counter is greater than 0 and the data buffer size to be processed is greater than 1 Mbyte, the size of the first fragment to be processed must be 1 Mbyte minus 16 × (initial value) to prevent a rollover of the internal 16-bit counter.

To have a sequential increment, the counter value must be programmed with the value programmed for the previous fragment + 2^{16} (or less for the first fragment).

All AES_IVRx fields must be programmed to take into account the possible carry propagation.

39.4.3. Last Output Data Mode (CBC-MAC)

This mode is used to generate cryptographic checksums on data (MAC) by means of cipher block chaining encryption algorithm (CBC-MAC algorithm for example).

The CMAC algorithm is a variant of CBC-MAC with post-processing requiring one-block encryption in ECB mode. Thus CBC-MAC is useful to accelerate CMAC.

After each end of encryption/decryption, the output data are available either on AES_ODATARx for Manual and Auto mode, or at the address specified in the receive buffer pointer for PDC mode (see the table Last Output Data Mode Behavior versus Start Modes).

AES_MR.LOD allows retrieval of only the last data of several encryption/decryption processes.

© 2026 Microchip Technology Inc. and its subsidiaries

Therefore, there is no need to define a read buffer in PDC mode.

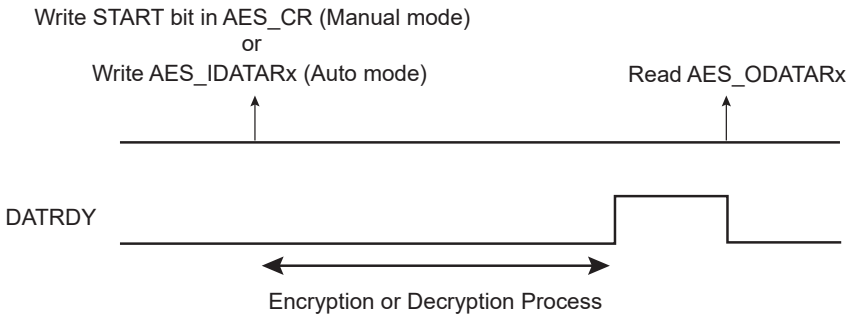
This data are only available in AES_ODATARx.

39.4.3.1. Manual and Auto Modes

39.4.3.1.1. If AES_MR.LOD = 0

The DATRDY flag is cleared when at least one AES_ODATARx is read (see the following figure).

Figure 39-1. Manual and Auto Modes with AES_MR.LOD = 0



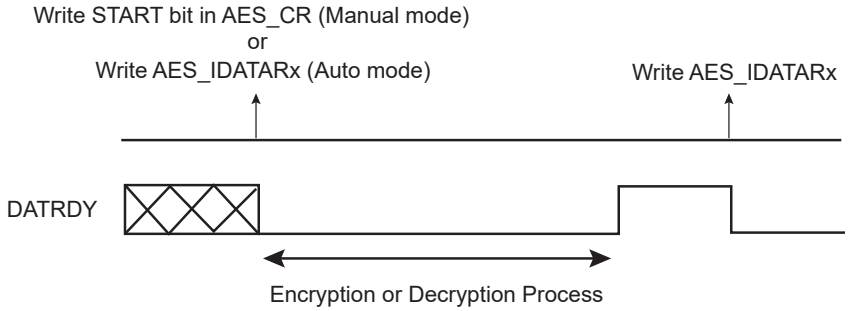
If the user does not want to read AES_ODATARx between each encryption/decryption, the DATRDY flag will not be cleared. If the DATRDY flag is not cleared, the user cannot know the end of the following encryptions/decryptions.

39.4.3.1.2. If AES_MR.LOD = 1

This mode is optimized to process AES CBC-MAC operating mode.

The DATRDY flag is cleared when at least one AES_IDATAR is written (see the following figure). No additional AES_ODATAR reads are necessary between consecutive encryptions/decryptions.

Figure 39-2. Manual and Auto Modes with AES_MR.LOD = 1



39.4.3.2. PDC Mode

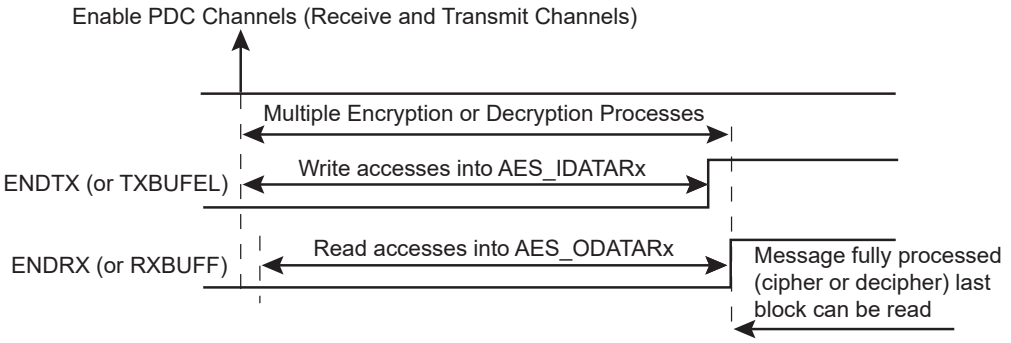
39.4.3.2.1. If AES_MR.LOD = 0

This mode may be used for all AES operating modes except CBC-MAC where AES_MR.LOD = 1 mode is recommended.

The end of the encryption/decryption is indicated when AES_ISR.ENDRX (or AES_ISR.RXBUFF) flag is raised (see the following figure).

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 39-3. PDC Transfer with AES_MR.LOD = 0



39.4.3.2.2. If AES_MR.LOD = 1

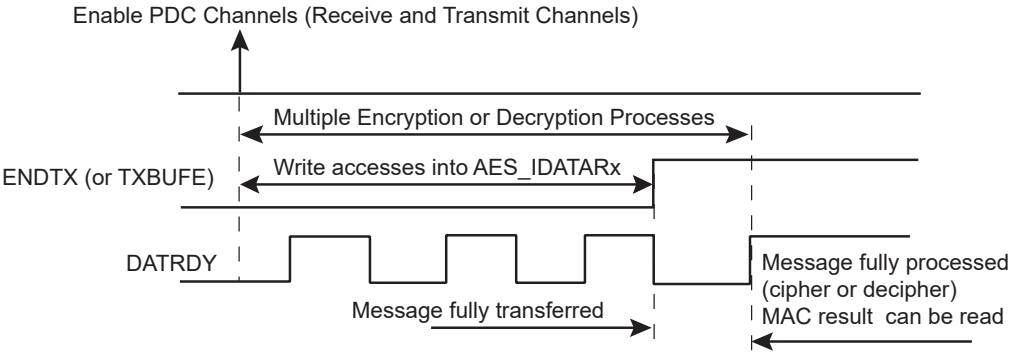
This mode is optimized to process AES CBC-MAC operating mode.

The user must first wait for the AES_ISR.ENDTX (or AES_ISR.TXBUFE) flag to be raised, then for DATRDY to ensure that the encryption/decryption is completed (see the following figure).

The PDC receive channel must not be used. Prior to reading the CBC-MAC result, AES_MR.SMOD must be written to 0. To restart a CBC-MAC on a new buffer, AES_MR.SMOD must be written to 2.

The output data are only available on AES_ODATARx.

Figure 39-4. PDC Transfer with AES_MR.LOD = 1



The following table summarizes the different cases.

Table 39-1. Last Output Data Mode Behavior versus Start Modes

Sequence	Manual and Auto Modes		PDC Mode	
	AES_MR.LOD = 0	AES_MR.LOD = 1	AES_MR.LOD = 0	AES_MR.LOD = 1
DATRDY Flag Clearing Condition[(1)]	At least one AES_ODATAR must be read	At least one AES_IDATAR must be written	Not used	Managed by the PDC
End of Encryption/ Decryption Notification	DATRDY	DATRDY	ENDRX (or RXBUFF)	ENDTX (or TXBUFE) then DATRDY
Encrypted/Decrypted Data Result Location	In AES_ODATARx	In AES_ODATARx	At the address specified in AES_RPR	In AES_ODATARx

Note: Depending on the mode, there are other ways of clearing the AES_ISR.DATRDY flag. See AES_ISR.

© 2026 Microchip Technology Inc. and its subsidiaries

WARNING In PDC mode, reading AES_ODATARx before the last data transfer may lead to unpredictable results.

39.4.4. Galois/Counter Mode (GCM)

39.4.4.1. Description

GCM comprises the AES engine in CTR mode along with a universal hash function (GHASH engine) that is defined over a binary Galois field to produce a message authentication tag (the AES CTR engine and the GHASH engine are depicted in the following figure.

The GHASH engine processes data packets after the AES operation. GCM assures the confidentiality of data through the AES Counter mode of operation for encryption. Authenticity of the confidential data is assured through the GHASH engine. GCM can also provide assurance of data that is not encrypted. Refer to *NIST Special Publication 800-38D* for more complete information.

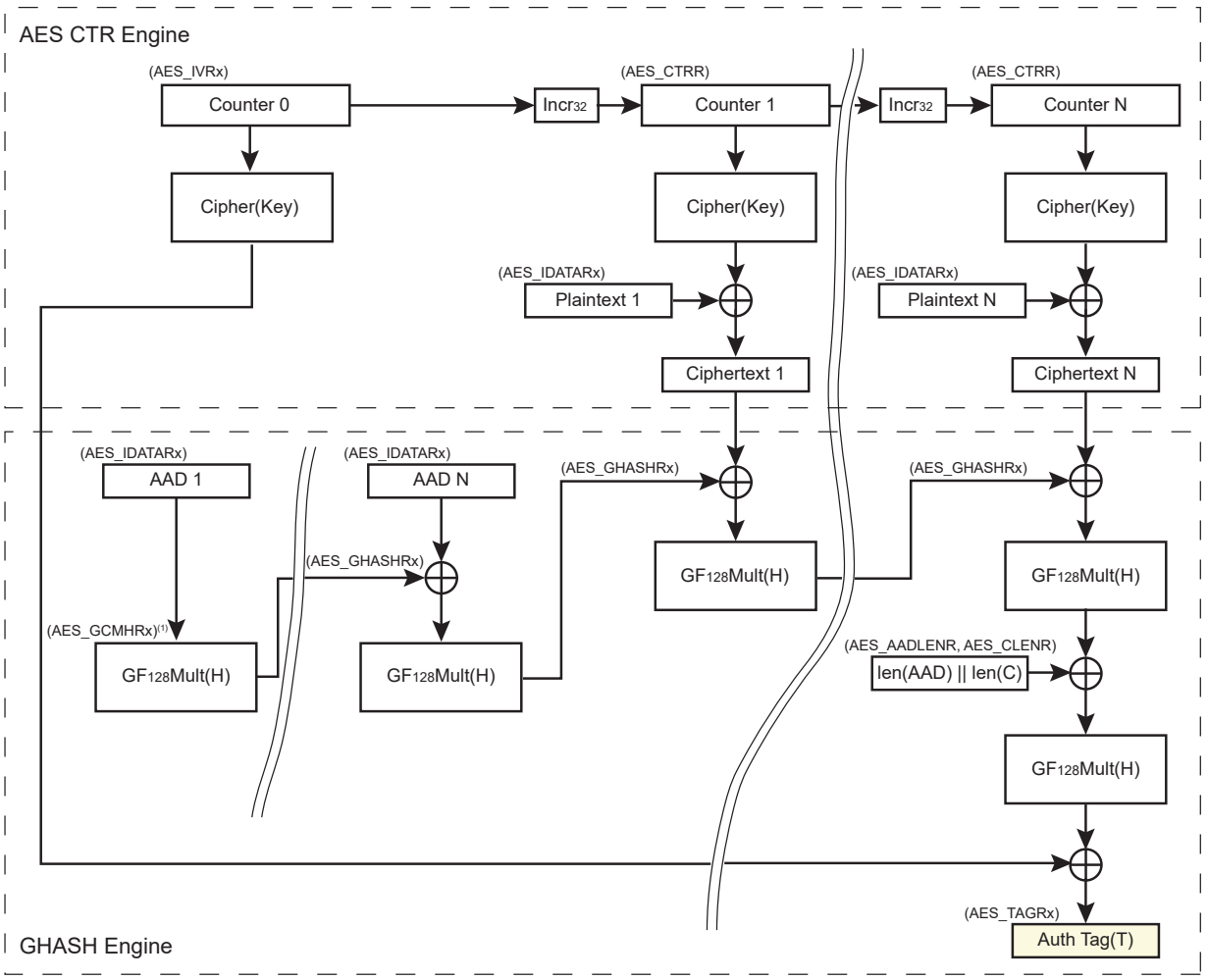
GCM can be used with or without the PDC host. Messages may be processed as a single complete packet of data or they may be broken into multiple packets of data over time.

GCM processing is computed on 128-bit input data fields. There is no support for unaligned data. The AES key length can be whatever length is supported by the AES module.

The recommended programming procedure when using PDC is described in the section GCM Processing.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 39-5. GCM Block Diagram



Note: 1. Optional

39.4.4.2. Key Writing and Automatic Hash Subkey Calculation

Whenever a new key is written to the hardware, two automatic actions are processed:

- GCM Hash Subkey *H* generation—The GCM hash subkey (*H*) is automatically generated. The GCM hash subkey generation must be complete before doing any other action. AES_ISR.DATRDY indicates when the subkey generation is complete (with interrupt if configured). The GCM hash subkey calculation is processed with the formula *H* = CIPHER(Key, <128 bits to zero>). The generated GCM *H* value is then available in AES_GCMHRx. If the application software requires a specific hash subkey, the automatically generated *H* value can be overwritten in AES_GCMHRx. AES_GCMHRx can be written after the end of the hash subkey generation (see AES_ISR.DATRDY) and prior to starting the input data feed.
- AES_GHASHRx Clear—AES_GHASHRx are automatically cleared. If a hash initial value is needed for the GHASH, it must be written to AES_GHASHRx
  - after writing AES_KEYWRx, if any
  - before starting the input data feed

39.4.4.3. GCM Processing

GCM processing is made up of three phases:

© 2026 Microchip Technology Inc. and its subsidiaries

1. Processing the Additional Authenticated Data (*AAD*), hash computation only.
2. Processing the Ciphertext (C), hash computation + ciphering/deciphering.
3. Generating the Tag using length of *AAD*, length of C and J_0 (refer to NIST documentation for details).

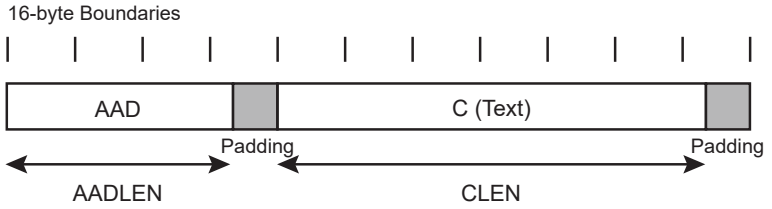
The Tag generation can be done either automatically, after the end of *AAD*/C processing if AES_MR.GTAGEN is set, or manually using AES_GHASHRx.GHASH (see subsections Processing a Complete Message with Tag Generation and Manual GCM Tag Generation for details).

39.4.4.3.1. Processing a Complete Message with Tag Generation

Use this procedure only if J0 four LSB bytes ≠ 0xFFFFFFFF.

Note: If J_0 four LSB bytes = 0xFFFFFFFF or if the value is unknown, use the procedure described in Processing a Complete Message without Tag Generation followed by the procedure in Manual GCM Tag Generation.

Figure 39-6. Full Message Alignment



To process a complete message with Tag generation, the sequence is as follows:

1. Set AES_MR.OPMOD to GCM and AES_MR.GTAGEN to '1'.
2. Write the key and wait until AES_ISR.DATRDY is set (GCM hash subkey generation complete); use interrupt if needed. See Key Writing and Automatic Hash Subkey Calculation.
3. Calculate the J_0 value as described in NIST documentation $J_0 = IV \,||\, 0^{31} \,||\, 1$ when len(IV) = 96 and $J_0 = GHASH_H(IV \,||\, 0^{s+64} \,||\, [len(IV)]64)$ if len(IV) ≠ 96. See Processing a Message with only AAD (GHASHH) for J_0 generation.
4. Set AES_IVRx.IV with inc32(J_0) (J_0 + 1 on 32 bits).
5. Configure AES_AADLENR.AADLEN and AES_CLENR.CLEN.
6. Fill AES_IDATARx.IDATA with the message to process according to the SMOD configuration used. If Manual Mode or Auto Mode is used, the DATRDY bit indicates when the data have been processed (however, no output data are generated when processing *AAD*).
7. Wait for TAGRDY to be set (use interrupt if needed), then read AES_TAGRx.TAG to obtain the authentication tag of the message.

39.4.4.3.2. Processing a Complete Message without Tag Generation

Processing a message without generating the Tag can be used to customize the Tag generation, or to process a fragmented message. To manually generate the GCM Tag, see Manual GCM Tag Generation.

To process a complete message without Tag generation, the sequence is as follows:

1. Set AES_MR.OPMOD to GCM and AES_MR.GTAGEN to '0'.
2. Write the key and wait until AES_ISR.DATRDY is set (GCM hash subkey generation complete); use interrupt if needed. After the GCM hash subkey generation is complete the GCM hash subkey can be read or overwritten with specific value in AES_GCMHRx. See Key Writing and Automatic Hash Subkey Calculation.

© 2026 Microchip Technology Inc. and its subsidiaries

3. Calculate the J0 value as described in NIST documentation J_0 = IV || 0^{31} || 1 when len(IV) = 96 and J_0 = $GHASH_H$(IV || 0^{s+64} || [len(IV)]64) if len(IV) ≠ 96. See Processing a Message with only AAD (GHASHH) for J_0 generation example when len(IV) ≠ 96.
4. Set AES_IVRx.IV with inc32(J_0) (J_0 + 1 on 32 bits).
5. Configure AES_AADLENR.AADLEN and AES_CLENR.CLEN.
6. Fill AES_IDATARx.IDATA with the message to process according to the SMOD configuration used. If Manual Mode or Auto Mode is used, the DATRDY bit indicates when the data have been processed (however, no output data are generated when processing *AAD*).
7. Make sure the last output data have been read if AES_CLENR.CLEN ≠ 0 (or wait for DATRDY), then read AES_GHASHRx.GHASH to obtain the hash value after the last processed data.

39.4.4.3.3. Processing a Fragmented Message without Tag Generation

If needed, a message can be processed by fragments, in such case automatic GCM Tag generation is not supported.

To process a message by fragments, the sequence is as follows:

- First fragment:

1. Set AES_MR.OPMOD to GCM and AES_MR.GTAGEN to '0'.
2. Write the key and wait for AES_ISR.DATRDY to be set (GCM hash subkey generation complete); use interrupt if needed. After the GCM hash subkey generation is complete the GCM hash subkey can be read or overwritten with specific value in AES_GCMHRx. See Key Writing and Automatic Hash Subkey Calculation.
3. Calculate the J_0 value as described in NIST documentation J_0 = IV || 0^{31} || 1 when len(IV) = 96 and J_0 = $GHASH_H$(IV || 0^{s+64} || [len(IV)]64) if len(IV) ≠ 96. See Processing a Message with only AAD (GHASHH) for J_0 generation example when len(IV) ≠ 96.
4. Set AES_IVRx.IV with inc32(J_0) (J_0 + 1 on 32 bits).
5. Configure AES_AADLENR.AADLEN and AES_CLENR.CLEN according to the length of the first fragment, or set the fields with the full message length (both configurations work).
6. Fill AES_IDATARx.IDATA with the first fragment of the message to process (aligned on 16-byte boundary) according to the SMOD configuration used. If Manual Mode or Auto Mode is used the DATRDY bit indicates when the data have been processed (however, no output data are generated when processing *AAD*).
7. Make sure the last output data have been read if the fragment ends in C phase (or wait for DATRDY if the fragment ends in *AAD* phase), then read AES_GHASHRx.GHASH to obtain the value of the hash after the last processed data and finally read AES_CTR.CTR to obtain the value of the CTR encryption counter (not needed when the fragment ends in AAD phase).

- Next fragment (or last fragment):

1. Set AES_MR.OPMOD to GCM and AES_MR.GTAGEN to '0'.
2. Write the key and wait until AES_ISR.DATRDY is set (GCM hash subkey generation complete); use interrupt if needed. After the GCM hash subkey generation is complete the GCM hash subkey can be read or overwritten with specific value in AES_GCMHRx. See Key Writing and Automatic Hash Subkey Calculation.
3. Set AES_IVRx.IV as follows:
   - If the first block of the fragment is a block of Additional Authenticated data, set AES_IVRx.IV with the J_0 initial value
   - If the first block of the fragment is a block of Plaintext data, set AES_IVRx.IV with a value constructed as follows: 'LSB96(J_0) || CTR' value, (96 bit LSB of J_0 concatenated with saved CTR value from previous fragment).

© 2026 Microchip Technology Inc. and its subsidiaries

4. Configure AES_AADLENR.AADLEN and AES_CLENR.CLEN according to the length of the current fragment, or set the fields with the remaining message length, both configurations work.
5. Fill AES_GHASHRx.GHASH with the value stored after the previous fragment.
6. Fill AES_IDATARx.IDATA with the current fragment of the message to process (aligned on 16 byte boundary) according to the SMOD configuration used. If Manual Mode or Auto Mode is used, the DATRDY bit indicates when the data have been processed (however, no output data are generated when processing *AAD*).
7. Make sure the last output data have been read if the fragment ends in C phase (or wait for DATRDY if the fragment ends in AAD phase), then read AES_GHASHRx.GHASH to obtain the value of the hash after the last processed data and finally read AES_CTR.CTR to obtain the value of the CTR encryption counter (not needed when the fragment ends in *AAD* phase).

Note: Step 1 and 2 are required only if the value of the concerned registers has been modified.

Once the last fragment has been processed, the GHASH value will allow manual generation of the GCM tag. See Manual GCM Tag Generation.

39.4.4.3.4. Manual GCM Tag Generation

This section describes the last steps of the GCM Tag generation.

The Manual GCM Tag Generation is used to complete the GCM Tag Generation when the message has been processed without Tag Generation.

Note: The Message Processing without Tag Generation must be finished before processing the Manual GCM Tag Generation.

To generate a GCM Tag manually, the sequence is as follows:

Processing S = $GHASH_H$ (*AAD* || 0v || C || 0u || [len(*AAD*)]64 || [len(C)]64):

1. Set AES_MR.OPMOD to GCM and AES_MR.GTAGEN to '0'.
2. Write the key and wait for AES_ISR.DATRDY to be set (GCM hash subkey generation complete); use interrupt if needed. After the GCM hash subkey generation is complete the GCM hash subkey can be read or overwritten with specific value in AES_GCMHRx. See Key Writing and Automatic Hash Subkey Calculation.
3. Configure AES_AADLENR.AADLEN to 0x10 (16 bytes) and AES_CLENR.CLEN to '0'. This will allow running a single $GHASH_H$ on a 16-byte input data (see the following figure).
4. Fill AES_GHASHRx.GHASH with the state of the GHASH field stored at the end of the message processing.
5. Fill AES_IDATARx.IDATA according to the SMOD configuration used with 'len(*AAD*)64 || len(*C*)64' value as described in the NIST documentation and wait for DATRDY to be set; use interrupt if needed.
6. Read AES_GHASHRx.GHASH to obtain the current value of the hash.

Processing *T* = GCTRK(J_0, S):

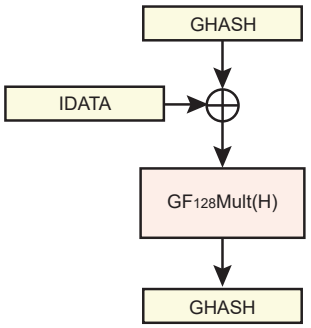
1. Set AES_MR.OPMOD to CTR.
2. Set AES_IVRx.IV with 'J_0' value.
3. Fill AES_IDATARx.IDATA with the GHASH value read at step 6 and wait for DATRDY to be set (use interrupt if needed).
4. Read AES_ODATARx.ODATA to obtain the GCM Tag value.

Note: Step 4 is optional if the GHASH field is to be filled with value '0' (0 length packet for instance).

© 2026 Microchip Technology Inc. and its subsidiaries

39.4.4.3.5. Processing a Message with only AAD (GHASHH)

Figure 39-7. Single $GHASH_H$ Block Diagram (AADLEN ≤ 0x10 and CLEN = 0)



It is possible to process a message with only *AAD* setting the CLEN field to '0' in AES_CLENR, this can be used for J0 generation when len(*IV*) ≠ 96 for instance.

Example: Processing J_0 when len(*IV*) ≠ 96

To process $J_0 = GHASH_H(IV \,||\, 0^{s+64} \,||\, [len(IV)]64)$, the sequence is as follows:

1. Set AES_MR.OPMOD to GCM and AES_MR.GTAGEN to '0'.
2. Write AES_KEYWRx and wait until AES_ISR.DATRDY is set (GCM hash subkey generation complete); use interrupt if needed. After the GCM hash subkey generation is complete the GCM hash subkey can be read or overwritten with specific value in AES_GCMHRx. See Key Writing and Automatic Hash Subkey Calculation.
3. Configure AES_AADLENR.AADLEN with '$len(IV \,||\, 0^{s+64} \,||\, [len(IV)]64)$' in and AES_CLENR.CLEN to '0'. This will allow running a $GHASH_H$ only.
4. Fill AES_IDATARx.IDATA with the message to process ($IV \,||\, 0^{s+64} \,||\, [len(IV)]64$) according to the SMOD configuration used. If Manual Mode or Auto Mode is used, the DATRDY bit indicates when a $GHASH_H$ step is over (use interrupt if needed).
5. Read AES_GHASHRx.GHASH to obtain the J_0 value.
   Note: The GHASH value can be overwritten at any time by writing the value of AES_GHASHRx.GHASH, used to perform a $GHASH_H$ with an initial value for GHASH (write GHASH field between step 3 and step 4 in this case).

39.4.4.3.6. Processing a Single GF128 Multiplication

The AES can also be used to process a single multiplication in the Galois field on 128 bits (GF_{128}) using a single $GHASH_H$ with custom *H* value (see the figure above).

To run a GF_{128} multiplication (A x B), the sequence is as follows:

1. Set AES_MR.OPMOD to GCM and AES_MR.GTAGEN to '0'.

1. Configure AES_AADLENR.AADLEN with 0x10 (16 bytes) and AES_CLENR.CLEN to '0'. This will allow running a single $GHASH_H$.
2. Fill AES_GCMHRx.H with B value.
3. Fill AES_IDATARx.IDATA with the A value according to the SMOD configuration used. If Manual Mode or Auto Mode is used, the DATRDY bit indicates when a GHASHH computation is over (use interrupt if needed).
4. Read AES_GHASHRx.GHASH to obtain the result.

Note: AES_GHASHRx.GHASH can be initialized with a value C between step 3 and step 4 to run a ((A XOR C) x B) GF_{128} multiplication.

© 2026 Microchip Technology Inc. and its subsidiaries

39.4.5. Double Input Buffer

AES_IDATARx can be double-buffered to reduce the runtime of large files.

This mode allows a new message block to be written when the previous message block is being processed. This is only possible when DMA accesses are performed (AES_MR.SMOD = 2).

AES_MR.DUALBUFF must be set to '1' to access the double buffer.

39.4.6. Temporary Secured Storage for Keys

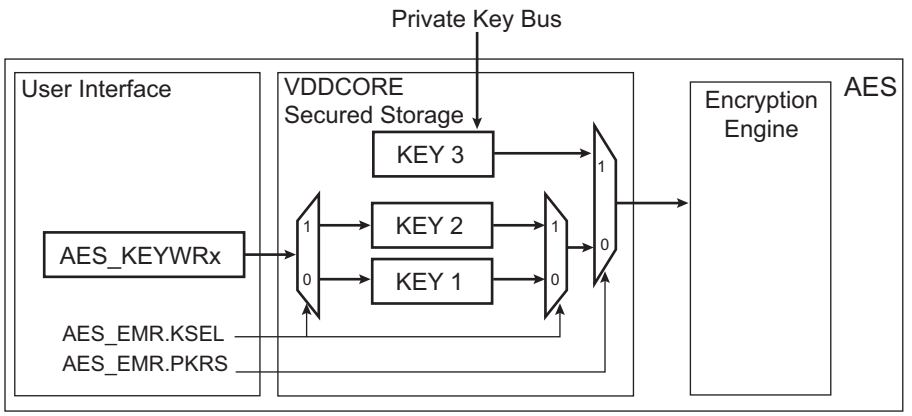
The AES provides secure storage for up to 256-bit keys. The storage is available while VDDCORE voltage is supplied.

The keys can be only written in AES internal registers and are not readable. Moreover, the internal registers holding the keys are buried in the overall product logic area during the physical implementation.

Two keys can be loaded by software by writing the Key Word registers (AES_KEYWRx).

One key can be loaded by Private Key bus only.

Figure 39-8. Temporary Secured Storage for Keys



39.4.7. Key Selection

AES_EMR.KSEL selects one of the two keys written by software via AES_KEYWRx.

AES_EMR.PKRS selects the key written by Private Key bus or the key resulting from the selection of the keys written by software.

When AES_EMR.PKRS is modified, it is mandatory to perform either a key write or a write in AES_CR.KSWP. The key write is mandatory when a new key value must be used. Writing AES_CR.KSWP to '1' is mandatory if the key has been previously written and selected again after using another key.

If Private Key internal registers and software-loaded keys are already written, selecting one or the other requires that AES_EMR.PKRS/KSEL are configured prior to writing AES_CR.KSWP=1.

When AES_EMR.KSEL is modified, it is mandatory to perform either a key write (write KEY1 or KEY2) or a write in AES_CR.KSWP. The key write is mandatory when a new key value must be used. Writing AES_CR.KSWP to '1' is mandatory if the key has been previously written and selected again after using another key.

When KEY1 and KEY2 are already loaded, selecting one or the other requires that AES_EMR.KSEL is configured prior to writing AES_CR.KSWP=1.

If two keys must be consecutively loaded, the sequence of actions is as follows:

1. Verify no processing is in progress (AES_ISR.DATRDY=0).

© 2026 Microchip Technology Inc. and its subsidiaries

2. Configure AES_EMR.KSEL and/or AES_EMR.PKRS according to the first key to load.
3. Write AES_KEYWRx or initiate a transfer on the Private Key bus for the first key.
4. Verify no processing is in progress (AES_ISR.DATRDY=0).
5. Modify AES_EMR.KSEL and/or AES_EMR.PKRS according to the second key to load.
6. Verify the new configuration AES_EMR.KSEL and/or AES_EMR.PKRS.
7. Write AES_KEYWRx or initiate a transfer on the Private Key bus for the second key.

Note: AES_CR.KSWP must not be written to 1 after writing AES_KEYWRx or Private Key bus registers.

39.4.8. Start Modes

AES_MR.SMOD allows selection of the encryption (or decryption) Start mode.

39.4.8.1. Manual Mode

The sequence of actions is as follows:

1. Write AES_MR with all required fields, including but not limited to SMOD and OPMOD. (Write AES_EMR.PKRS according to the type of key to be loaded).
2. Write the 128-bit/192-bit/256-bit AES key in AES_KEYWRx or in the Private Key internal registers.
3. Write the initialization vector (or counter) in AES_IVRx.
   Note: AES_IVRx concerns all modes except ECB.
4. Set the bit DATRDY (Data Ready) in the AES Interrupt Enable register (AES_IER), depending on whether an interrupt is required or not at the end of processing.
5. Write the data to be encrypted/decrypted in the authorized AES_IDATARx (see the following table).
6. Set the START bit in the AES Control register (AES_CR) to begin the encryption or the decryption process.
7. When processing completes, the DATRDY flag in the AES Interrupt Status register (AES_ISR) is raised. If an interrupt has been enabled by setting AES_IER.DATRDY, the interrupt line of the AES is activated.
8. When software reads one of AES_ODATARx, AES_IER.DATRDY is automatically cleared.

Table 39-2. Authorized Input Data Registers

Operating Mode	Input Data Registers to Write
ECB	All
CBC	All
OFB	All
128-bit CFB	All
64-bit CFB	AES_IDATAR0 and AES_IDATAR1
32-bit CFB	AES_IDATAR0
16-bit CFB	AES_IDATAR0
8-bit CFB	AES_IDATAR0
CTR	All
GCM	All

Notes:

1. In 64-bit CFB mode, writing to AES_IDATAR2 and AES_IDATAR3 is not allowed and may lead to errors in processing.
2. In 32, 16, and 8-bit CFB modes, writing to AES_IDATAR1, AES_IDATAR2 and AES_IDATAR3 is not allowed and may lead to errors in processing.

© 2026 Microchip Technology Inc. and its subsidiaries

39.4.8.2. Auto Mode

The Auto Mode is similar to the manual one, except that in this mode, as soon as the correct number of AES_IDATARx is written, processing is automatically started without any action in AES_CR.

39.4.8.3. PDC Mode

The Peripheral DMA Controller (PDC) can be used in association with the AES to perform an encryption/decryption of a buffer without any action by software during processing.

AES_MR.SMOD must be configured to 2.

For all operating modes except CBC-MAC (AES_MR.LOD=1), 2 DMA channels must be programmed (transmit and receive). In CBC-MAC, only 1 transmit channel must be programmed.

The sequence order is as follows:

1. Write AES_MR with all required fields, including but not limited to SMOD and OPMOD.
2. Write the key.
3. Write the initialization vector or counter in AES_IVRx.
   Note: IVRx concerns all modes except ECB.
4. Set the Transmit Pointer register (AES_TPR) to the address where the data buffer to encrypt/decrypt is stored and the Receive Pointer register (AES_RPR) where it must be encrypted/decrypted.
   Note: Transmit and receive buffers can be identical.
5. Set the Transmit and the Receive Counter registers (AES_TCR and AES_RCR) to the same value. This value must be a multiple of the data transfer type size (see the following table).
   Note: The same requirements are necessary for the Next Pointer(s) and Counter(s) of the PDC (AES_TNPR, AES_RNPR, AES_TNCR, AES_RNCR).
6. If not already done, set AES_IER.ENDRX (or AES_IER.RXBUFF if the next pointers and counters are used), depending on whether an interrupt is required or not at the end of processing.
7. Enable the PDC in transmission and reception to start the processing (AES_PTCR).

When the processing completes, AES_ISR.ENDRX (or AES_ISR.RXBUFF) is raised. If an interrupt has been enabled by setting the corresponding bit in AES_IER, the interrupt line of the AES is activated.

When PDC is used, the data size to transfer (byte, half-word or word) depends on the AES mode of operations. This size is automatically configured by the AES.

Table 39-3. Data Transfer Type for the Different Operating Modes

Operating Mode	Data Transfer Type
ECB	Word
CBC	Word
OFB	Word
CFB 128-bit	Word
CFB 64-bit	Word
CFB 32-bit	Word
CFB 16-bit	Half-word
CFB 8-bit	Byte
CTR	Word
GCM	Word

39.4.9. Automatic Padding Mode

When Automatic Padding mode is configured, the message is automatically padded after the last block is written. Depending on the size of the message, either a padding is performed after the last part of the message and padding blocks are added, or only padding blocks are added.

© 2026 Microchip Technology Inc. and its subsidiaries

IPSec and SSL padding standards are both supported.

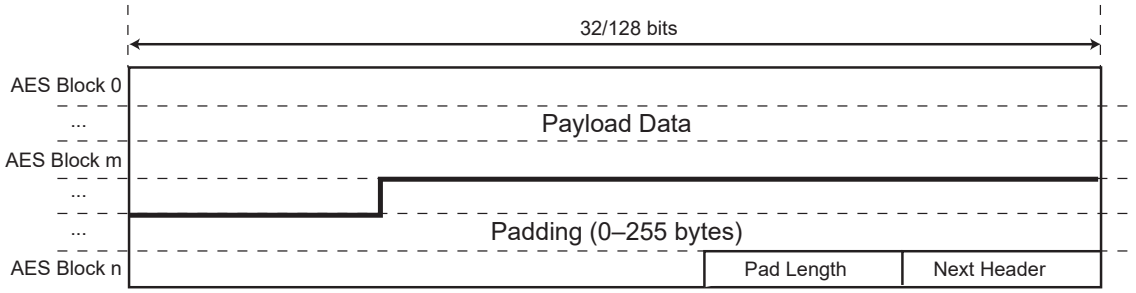
The auto padding feature only supports CBC and CTR modes.

Note: When automatic padding is enabled and AES_MR.SMOD=2, AES_MR.DUALBUFF must be cleared.

39.4.9.1. IPSec Padding

Automatic Padding is enabled by writing a '1' to AES_EMR.APEN. IPSEC padding mode is selected by writing a '0' to AES_EMR.APM.

Figure 39-9. IPSec Padding



Each byte of the padding area contains incremental integer values.

The "Pad Length" in bytes is configured in AES_EMR.PADLEN and the "Next Header" value is configured in AES_EMR.NHEAD. AES_EMR.PADLEN must be configured with the length of the padding section, not including the length of the "Pad Length" and "Next Header" sections.

The BCNT field in the AES Byte Counter register (AES_BCNT) defines the length, in bytes, of the message to process. It must be configured before writing the first data in AES_IDATARx and the remaining bytes to process can be read at anytime (BCNT value is decremented after each AES_IDATARx access).

AES_BCNT.BCNT and AES_EMR.PADLEN must be configured so that the sum of the length of the message (Payload Data) and of the length of the Padding, Pad Length (1 byte) and Next Header (1 byte) sections is a multiple of the AES block size (128 bits).

To process an IPSec message using auto-padding, the sequence is as follows:

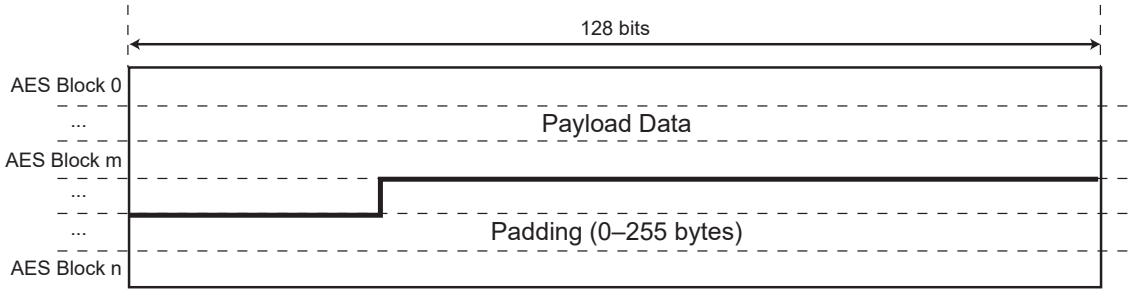
1. Set AES_MR.OPMOD to either CBC or CTR mode.
2. Set AES_EMR.APEN to '1', AES_EMR.APM to '0', AES_EMR.PADLEN to the desired padding length in byte and AES_EMR.NHEAD to the desired Next Header field value.
3. Configure AES_BCNT.BCNT with the whole message length, without padding, in byte.
4. Write the key.
5. Set AES_IVRx.IV if needed.
6. Fill AES_IDATARx.IDATA with the message to process according to the SMOD configuration used. On the last data block, write only what is necessary (e.g., write only AES_IDATAR0 if last block size is ≤ 32 bits).
7. Wait for the DATRDY flag to be raised, meaning auto-padding completion and last block processing.

39.4.9.2. SSL Padding

Auto Padding is enabled by writing a '1' to AES_EMR.APEN and SSL padding mode is selected by writing a '1' to AES_EMR.APM.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 39-10. SSL Padding



Each byte of the padding area contains the padding length.

The padding length is configured in AES_EMR.PADLEN.

AES_BCNT.BCNT defines the length, in bytes, of the message to process. It must be configured before writing the first data in AES_IDATARx and the remaining bytes to process can be read at anytime (BCNT value is decremented after each AES_IDATARx access).

AES_BCNT.BCNT and AES_EMR.PADLEN must be configured so that the length of the message plus the length of the padding section is a multiple of the AES block size (128 bits).

To process a complete SSL message, the sequence is as follows:

1. Set AES_MR.OPMOD to either CBC or CTR mode.
2. Set AES_EMR.APEN to '1', AES_EMR.APM to '1', AES_EMR.PADLEN to the desired padding length in bytes.
3. Set AES_BCNT.BCNT with the whole message length, without padding, in bytes.
4. Write the key.
5. Set AES_IVRx.IV if needed.
6. Fill AES_IDATARx.IDATA with the message to process according to the SMOD configuration used. On the last data block write only what is necessary (e.g., write only AES_IDATAR0 if last block size is ≤ 32 bits).
7. Wait for the DATRDY flag to be raised, meaning auto-padding completion and last block processing.

39.4.9.3. Flags

AES_ISR.EOPAD rises as soon as the automatic padding phase is over, meaning that all the extra padding blocks have been processed. Reading AES_ISR clears this flag.

AES_ISR.PLENERR indicates an error in the frame configuration, meaning that the whole message length including padding does not respect the standard selected. AES_ISR.PLENERR rises at the end of the frame in case of wrong message length and is cleared reading AES_ ISR.

In IPSec/SSL standard message length including padding must be a multiple of the AES block size when CBC mode is used and multiple of 32-bit if CTR mode is used.

39.4.10. ARIA Encryption Algorithm

The AES is able to encrypt/decrypt data using either the AES or the ARIA algorithm. AES_EMR.ALGO selects the algorithm.

39.4.11. Secure Protocol Layers Improved Performances

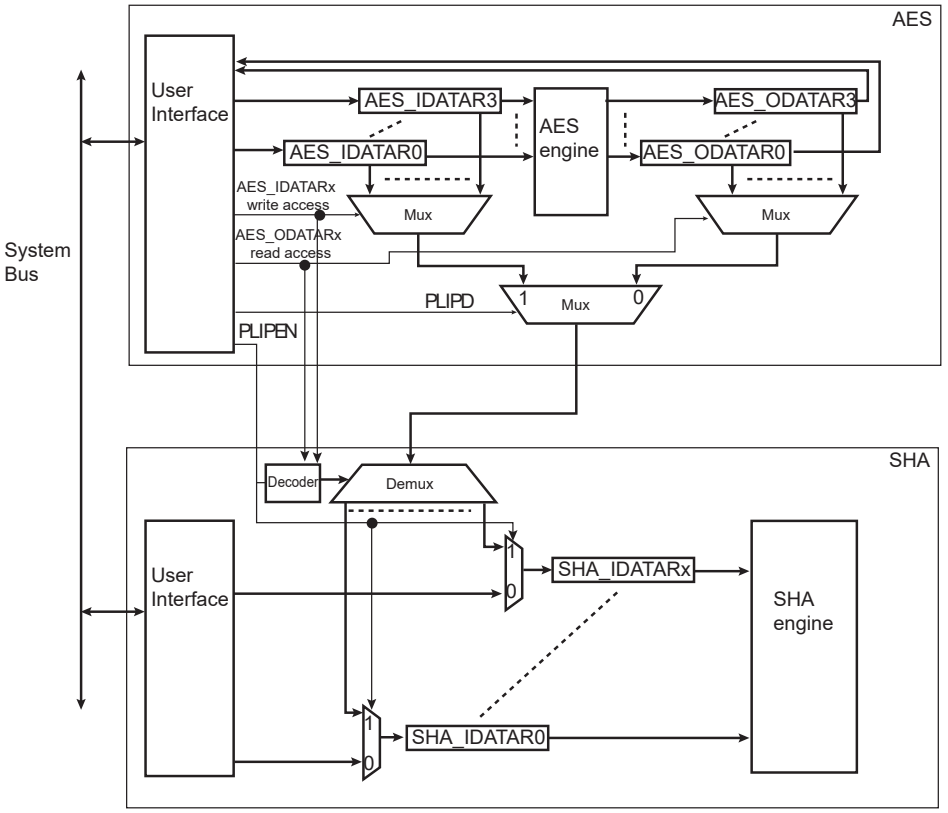
Secure protocol layers such as IPSec require encryption and authentication. For IPSec, the authentication is based on HMAC, thus SHA is required. To optimize performance, the AES embeds a mode of operation, used with DMA only, that enables the SHA module to process the input or output data of the AES module. If this mode is enabled, write access is required only into

© 2026 Microchip Technology Inc. and its subsidiaries

AES_IDATARx registers, since SHA input data registers are automatically written by AES without software intervention. When the DMA is configured to transfer a buffer of data (input frame), only one transfer descriptor is required for both authentication and encryption/decryption processes and only one buffer is transferred through the system bus (reducing the load of the system bus).

Improved performance for secure protocol layers requires AES_EMR.PLIPEN to be set.

Figure 39-11. Secure Protocol Layers Improved Performances Block Diagram



39.4.11.1. Cipher Mode

When AES_EMR.PLIPD is cleared and AES_EMR.PLIPEN=1, the message written into AES_IDATARx is first encrypted with the AES module and the encrypted message is authenticated with the SHA module. Therefore, when AES_EMR.PLIPD is cleared, AES_ODATARx are selected and sent to SHA_IDATARx as soon as AES_ODATARx are read. A read access in AES corresponds to a write access to the corresponding SHA_IDATARx. The number of SHA_IDATARx is greater than the number of AES_ODATARx, but the SHA module embeds the decoding logic to automatically dispatch AES_ODATARx values into the corresponding SHA_IDATARx without software intervention.

39.4.11.2. Decipher Mode

When AES_EMR.PLIPD is written to '1' and AES_EMR.PLIPEN=1, the message written into AES_IDATARx is decrypted with the AES module and also sent to SHA for authentication. Therefore, when AES_EMR.PLIPD=1, AES_IDATARx are selected and sent to SHA_IDATARx as soon as AES_IDATARx are written. A write access in AES corresponds to a write access to the corresponding SHA_IDATARx. The number of SHA_IDATARx is greater than the number of AES_ODATARx, but the SHA module embeds the decoding logic to automatically dispatch AES_IDATARx values into the corresponding SHA_IDATARx without software intervention.

39.4.11.3. Encapsulating Security Payload (ESP) IPSec Examples

The following examples describe how to configure AES and SHA to optimize processing an ESP IPSec frame for maximum performance.

© 2026 Microchip Technology Inc. and its subsidiaries

The cipher (or decipher) of an ESP IPSec frame requires both encryption (or decryption) and authentication.

For cipher, the input frame located in the system memory must first be padded and the resulting buffer encrypted. The encrypted frame must be written back to the system memory and sent to the authentication module.

When the AES module is configured to improve the performance of the secure protocol layers (AES_EMR.PLIPEN = 1), the data transfers are simplified, limiting the bandwidth requirements on the system bus.

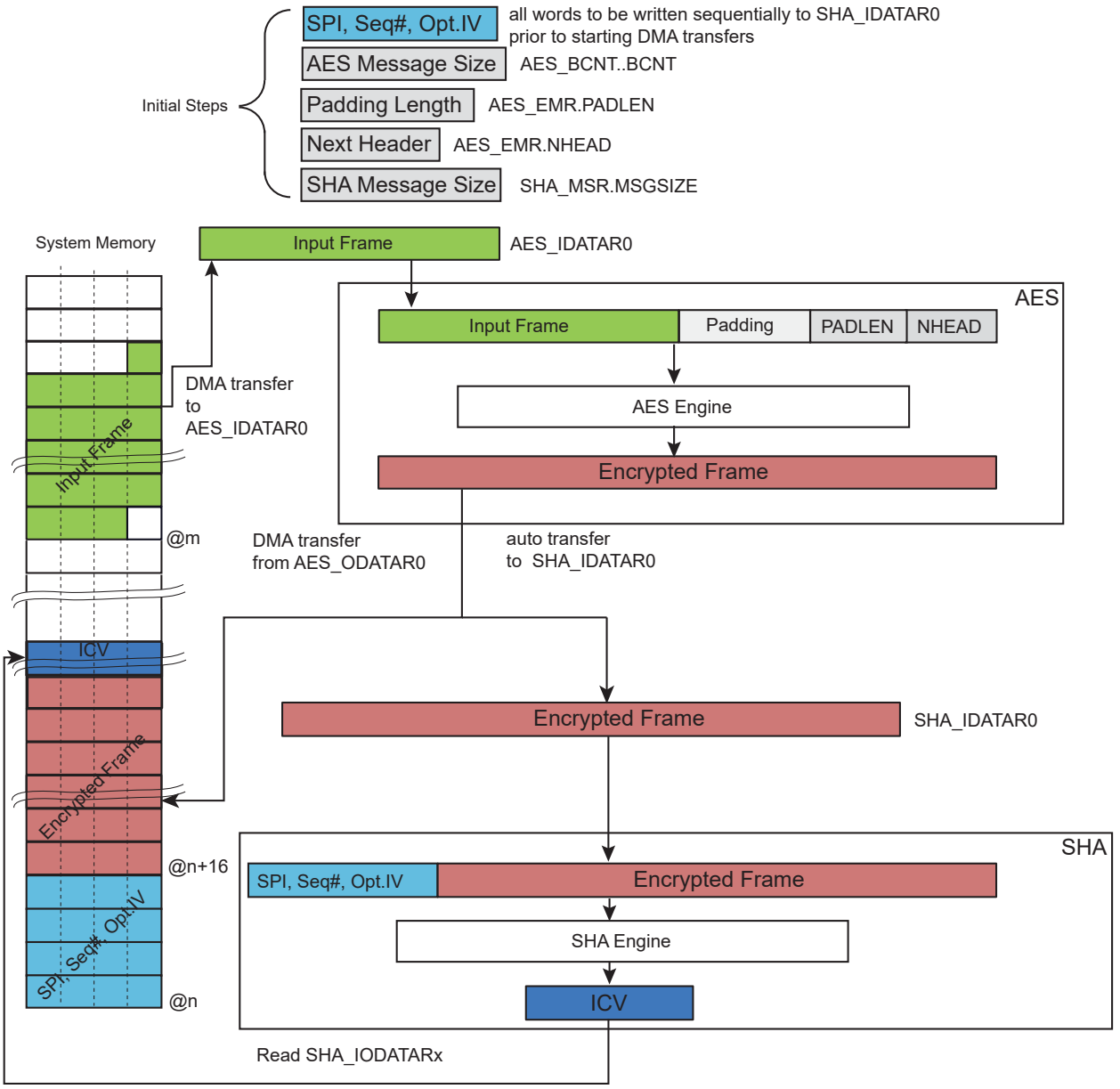
Before configuring the DMA to start the transfer of the data buffer (input frame) to the AES, the following actions must be taken in registers:

- AES_BCNT.BCNT must be configured with the length of the message (Input Frame).
- The padding length of the AES must be configured in AES_EMR.PADLEN. See Automatic Padding Mode to configure Automatic Padding mode.
- The next header value must be configured in AES_EMR.NHEAD.
- AES_MR.SMOD and SHA_MR.SMOD must be configured to 2.
  Note: When automatic padding is enabled and AES_MR.SMOD = 2 , AES_MR.DUALBUFF must be cleared.
- The SHA_MSR.MSGSIZE must be configured with the length of the authentication message including the optional extended sequence number (ESN) and header and trailer information required by the authentication algorithm used (HMAC, etc.). Refer to the section “Secure Hash Algorithm (SHA)” for more details on configuration for optimized processing of header information.
- The Security Parameter Index (SPI, sequence number (SEQ#)) and the optional Initialization Vector (IV) must be configured sequentially in SHA_IDATAR0.
- A first DMA transfer descriptor must be configured to transfer the input frame from the system memory to the AES input data registers (AES_IDATARx), and a second DMA descriptor must be configured to transfer the encrypted frame from AES to the system memory.
  Note: If AES_EMR.PLIPEN = 1 , there is no need to define a transfer descriptor to load the encrypted frame into the SHA input data registers because the transfer is automatically performed while the second descriptor transfer is in progress.

See the following figures.

© 2026 Microchip Technology Inc. and its subsidiaries

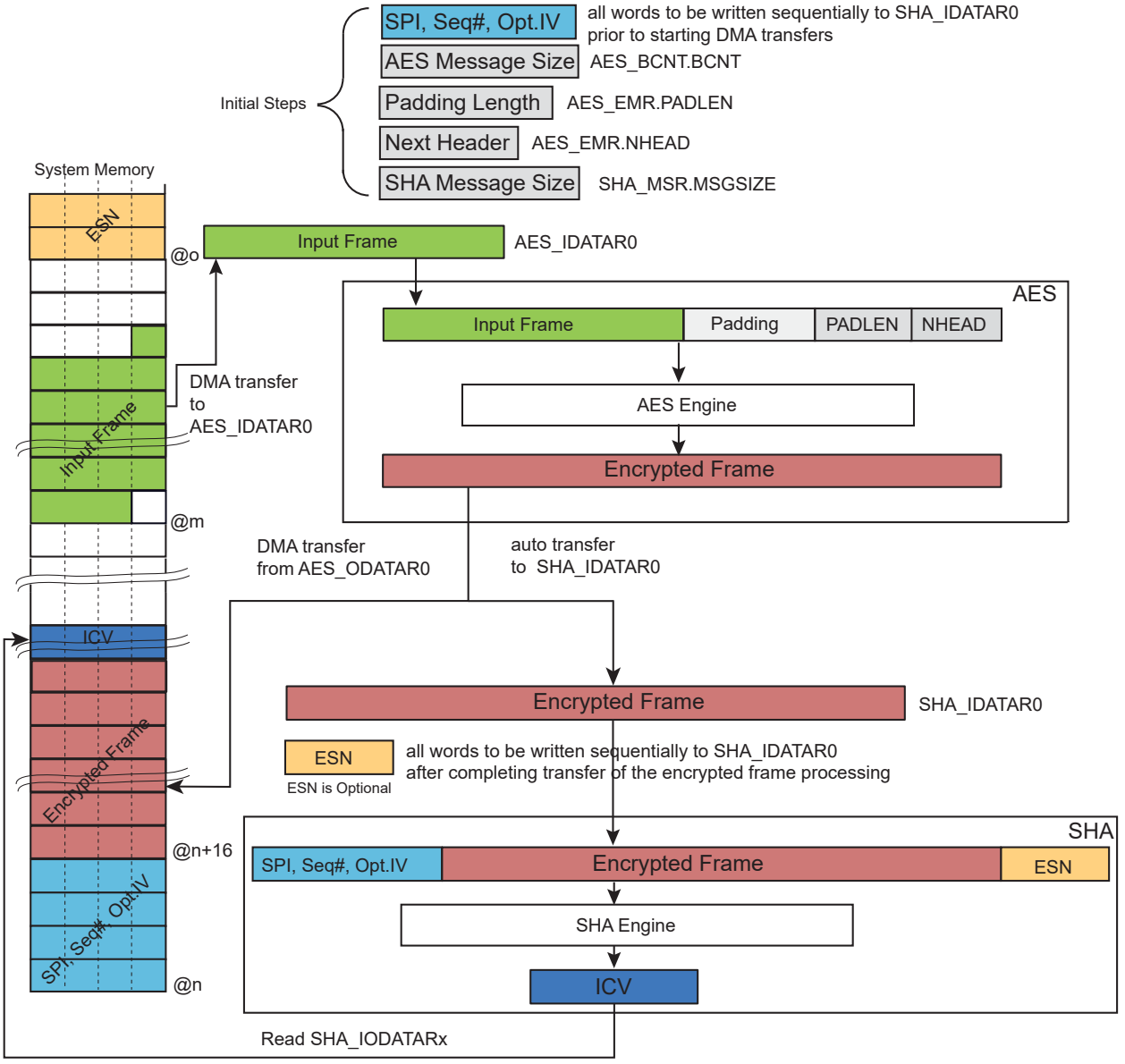
Figure 39-12. Generation of an ESP IPSec Frame without ESN



If the optional extended sequence number is required for authentication, wait for the AES-to-system memory DMA buffer transfer to complete before configuring the ESN value. The ESN value must be configured in the SHA by writing sequentially each 32-bit word of the ESN into the SHA_IDATAR0 register. Wait for SHA_ISR.WRDY=1 before each write in the SHA_IDATAR0 register. See the following figure.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 39-13. Generation of an ESP IPSec Frame with ESN



To decipher an ESP IPSec frame without the optional ESN trailer information, two DMA channels are required and the SHA must be configured in Automatic padding mode.

Note: AES automatic padding must be disabled when deciphering a frame.

- A first DMA transfer descriptor must be configured to load the received encrypted frame from the system memory to AES_IDATARx for decryption. The start address of the first transfer descriptor must be defined after the SPI, SEQ#, and optional IV (see the following figure).
- A second DMA descriptor must be configured to transfer the decrypted frame from AES_ODATARx to the system memory.
- AES_EMR.PLIPEN and AES_EMR.PLIPD must be written to '1' so that the data buffer is written in AES_IDATARx and in SHA_IDATARx.

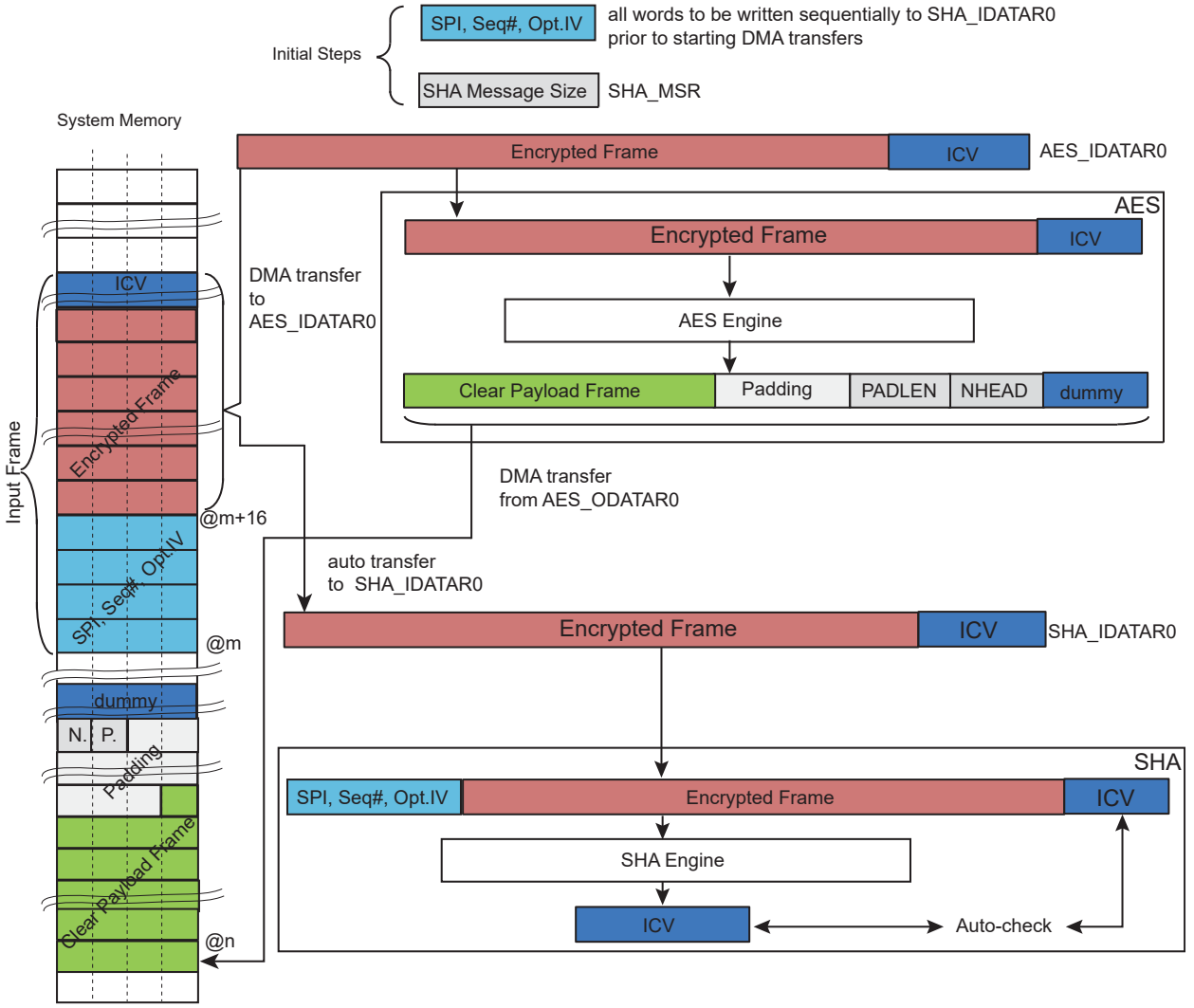
The SHA has the capability to perform an automatic check with an expected integrity check value if this value is appended at the end of the frame buffer (SHA_MR.CHECK=2). Thus, if the first transfer descriptor includes the ICV for SHA, the first DMA transfer allows the decryption and authentication

© 2026 Microchip Technology Inc. and its subsidiaries

processes including the automatic check. The decrypted part resulting from ICV is not required for downstream processing and must be considered as dummy data.

The end of the decryption and authentication processes occur when flag SHA_ISR.CHECKF=1. The authentication status is provided by SHA_ISR.CHKST.

Figure 39-14. Decryption of an ESP IP Sec Frame without ESN

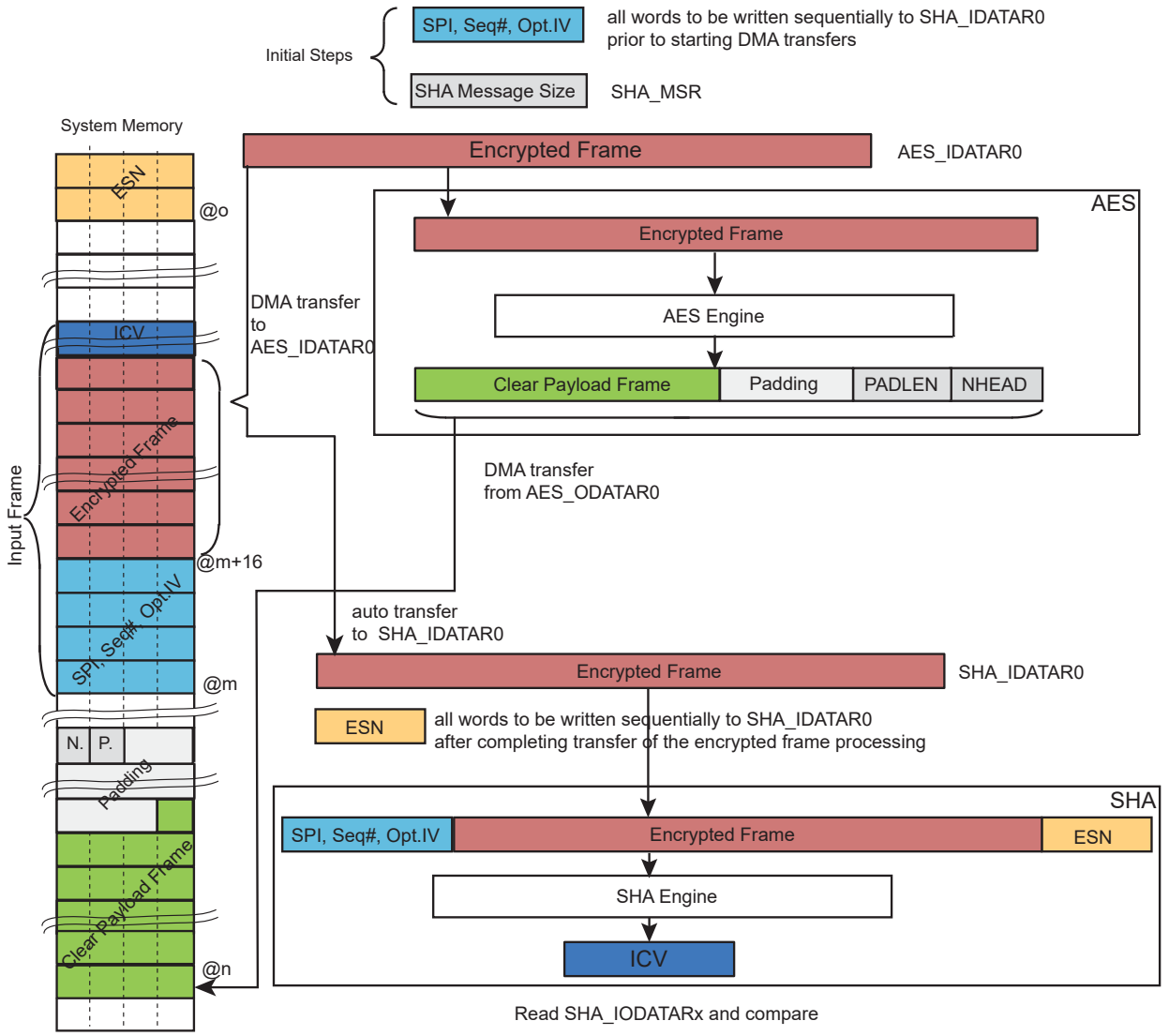


If the optional ESN trailer information is part of the ICV (see the following figure), the ESN must be manually written into SHA_IDATAR0. The ESN value must be written after completion of the system memory-to-AES DMA buffer transfer. The ESN value must be configured in the SHA by writing sequentially each 32-bit word of the ESN into the SHA_IDATAR0 register. Wait for SHA_ISR.WRDY=1 before each write in the SHA_IDATAR0 register.

When the optional ESN trailer information is part of the ICV, it is not possible to include the ICV received in the input frame to the first transfer descriptor. Moreover, if the HMAC algorithm is used for authentication, no automatic check can be performed when optimizing the processing performances of the SHA module. For more details, refer to the section “Secure Hash Algorithm (SHA)”. The result of the HMAC read in the SHA_IODATARx must be manually compared with the ICV value of the input frame. The comparison must be performed after the end of the authentication process. The authentication process is completed when the SHA_ISR.DATRDY flag is set.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 39-15. Decryption of an ESP IPSec Frame with ESN



39.4.12. Security Features

39.4.12.1. Private Key Bus

The AES provides secure key transfer that requires a transfer command only, thus avoiding any manipulation of the key by software.

The AES features a set of Private Key internal registers that can be accessed only through the dedicated Private Key bus from the TRNG or Flash Controller.

The Private Key internal registers cannot be read from any peripheral or from software.

The AES key used by the encryption/decryption engine is either the Private Key internal registers content or the AES_KEYWRx registers loaded via the AES_KEYWRx.

To select the Private Key internal registers as the source of the AES key, AES_EMR.PKRS must be written to '1'.

When AES_EMR.PKRS is modified, it is mandatory to perform either a key write or a write in AES_CR.KSWP. The key write is mandatory when a new key value must be used. Writing

© 2026 Microchip Technology Inc. and its subsidiaries

AES_CR.KSWP to '1' is mandatory if the key has been previously written and selected again after using another key.

If Private Key internal registers and software-loaded keys are already written, selecting one or the other requires only to configure AES_EMR.PKRS prior to writing AES_CR.KSWP=1.

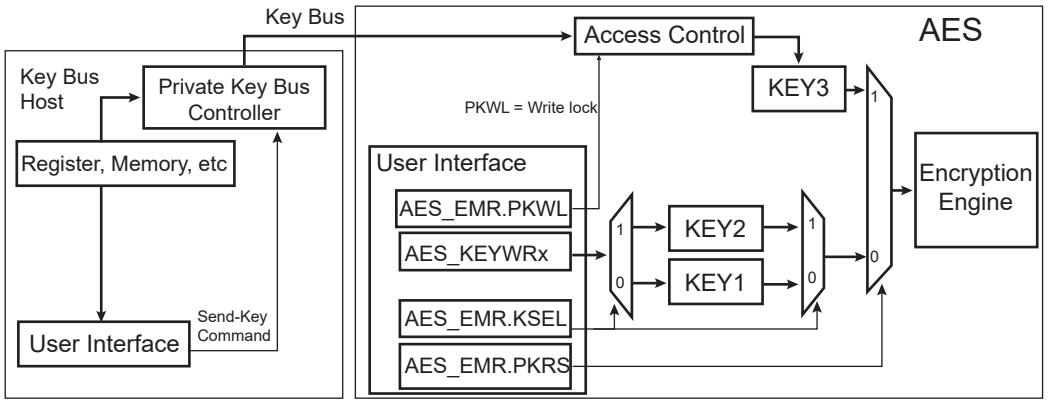
To write the Private Key internal registers, the software must:

1. Write a '1' in AES_EMR.PKRS.
2. Trigger the key transfer over the Private Key bus from the TRNG or Flash Controller key bus host.
3. Wait for completion of the transfer signaled in the TRNG or Flash Controller Status register.
4. Check for any access violation in AES_WPSR.PKRPVS.

While AES_EMR.PKWL=0, it is possible to write the Private Key internal registers as many times as required.

As soon as AES_EMR.PKWL=1, the next write sequence on Private Key internal registers is the last one. Any additional write sequence in the Private Key internal registers has no effect, thus providing write-protection of these registers. A hardware reset is the only way to exit from the write-protected state.

Figure 39-16. Key Selection



39.4.12.2. Unspecified Register Access Detection

When an unspecified register access occurs, AES_ISR.URAD is raised. Its source is then reported in AES_ISR.URAT. Only the last unspecified register access is available through the AES_ISR.URAT.

Several kinds of unspecified register accesses can occur:

- Input Data register written during the data processing when SMOD = IDATAR0_START
- Output Data register read during data processing
- Mode register written during data processing
- Output Data register read during sub-keys generation
- Mode register written during sub-keys generation
- Write-only register read access

AES_ISR.URAD and AES_ISR.URAT can only be reset by AES_CR.SWRST.

39.4.12.3. Clearing Key on Tamper Event

On a tamper detection event on WKUP0..4 pins, an immediate clear of the key (internal registers) can be performed if AES_MR.TAMPCLR=1. For configuration details, refer to the section "Supply Controller (SUPC)".

© 2026 Microchip Technology Inc. and its subsidiaries

39.4.12.4. Register Write Protection

To prevent any single software error from corrupting AES behavior, certain registers in the address space can be write-protected by setting the WPEN (Write Protection Enable), WPITEN (Write Protection Interrupt Enable), and/or WPCREN (Write Protection Control Enable) bits in the AES Write Protection Mode Register (AES_WPMR).

If a write access to a write-protected register is detected, the Write Protection Violation Status (WPVS) flag in the AES Write Protection Status Register (AES_WPSR) is set and the Write Protection Violation Source (WPVSRC) field indicates the register in which the write access has been attempted.

The WPVS bit is automatically cleared after reading AES_WPSR.

The following register(s) can be write-protected when WPEN is set in AES_WPMR:

- AES Mode Register
- AES Key Word Register x
- AES Initialization Vector Register x
- AES Additional Authenticated Data Length Register
- AES Plaintext/Ciphertext Length Register
- AES GCM Intermediate Hash Word Register x
- AES GCM H Word Register x
- AES Extended Mode Register
- AES Byte Counter Register

The following register(s) can be write-protected when WPITEN is set:

- AES Interrupt Enable Register
- AES Interrupt Disable Register

The following register(s) can be write-protected when WPCREN is set:

- AES Control Register

39.4.12.5. Security and Safety Analysis and Reports

Several types of checks are performed when the AES is enabled.

The peripheral clock of the AES is monitored by specific circuitry to detect abnormal waveforms on the internal clock net that may affect the behavior of the AES. Corruption on the triggering edge of the clock or a pulse with a minimum duration may be identified. If the flag AES_WPSR.CGD is set, an abnormal condition occurred on the peripheral clock. This flag is not set under normal operating conditions.

The internal sequencer of the AES is also monitored and if an abnormal state is detected, the flag AES_WPSR.SEQE is set. This flag is not set under normal operating conditions.

The software accesses to the AES are monitored and if an incorrect access is performed, the flag AES_WPSR.SWE is set. The type of incorrect/abnormal software access is reported in AES_WPSR.SWETYP (see AES_WPSR for details). For example, writing the AES_ODATARx is an error, as well as reading the AES_IDATARx, when the AES_ISR.DATRDY flag is cleared. AES_WPSR.ECLASS is an indicator reporting the criticality of the SWETYP report.

The flags CGD, SEQE, SWE and WPVS are automatically cleared when AES_WPSR is read.

If one of these flags is set, the flag AES_ISR.SECE is set and can trigger an interrupt if the AES_IMR.SECE bit is '1'. SECE is cleared by reading AES_ISR.

It is possible to configure an action to be performed by AES as soon as an abnormal event detection occurs. If AES_WPMR.ACTION > 0, either a lock is performed or a lock and immediate clear of the

© 2026 Microchip Technology Inc. and its subsidiaries

AES_KEYWRx key. If a lock is performed, the current processing is ended normally but any new processing is not performed whatever the start mode of operation (see AES_MR.SMOD).

A locked state of the AES is unlocked as follows:

1. Read AES_WPSR.
2. Disable the source of tamper if the tamper is enabled to perform a clear of the key.
3. Write a '1' to AES_CR.UNLOCK.

It is possible to select the type of event that will lock the AES in case of abnormal event detection. See AES_WPMR.ACTION for details.

If the AES_MR.TMPCLR=1 and the tamper pin is active, the AES is locked.

© 2026 Microchip Technology Inc. and its subsidiaries

39.5. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	AES_CR	31:24								UNLOCK
		23:16								
		15:8								SWRST
		7:0							KSWP	START
0x04	AES_MR	31:24	TAMPCLR							
		23:16	CKEY[3:0]					CFBS[2:0]		
		15:8	LOD	OPMOD[2:0]			KEYSIZE[1:0]		SMOD[1:0]	
		7:0	PROCDLY[3:0]				DUALBUFF		GTAGEN	CIPHER
0x08 ... 0x0F	Reserved									
0x10	AES_IER	31:24								
		23:16					SECE	PLENERR	EOPAD	TAGRDY
		15:8								URAD
		7:0				TXBUFE	RXBUFF	ENDTX	ENDRX	DATRDY
0x14	AES_IDR	31:24								
		23:16					SECE	PLENERR	EOPAD	TAGRDY
		15:8								URAD
		7:0				TXBUFE	RXBUFF	ENDTX	ENDRX	DATRDY
0x18	AES_IMR	31:24								
		23:16					SECE	PLENERR	EOPAD	TAGRDY
		15:8								URAD
		7:0				TXBUFE	RXBUFF	ENDTX	ENDRX	DATRDY
0x1C	AES_ISR	31:24								
		23:16					SECE	PLENERR	EOPAD	TAGRDY
		15:8	URAT[3:0]							URAD
		7:0				TXBUFE	RXBUFF	ENDTX	ENDRX	DATRDY
0x20	AES_KEYWR0	31:24	KEYW[31:24]							
		23:16	KEYW[23:16]							
		15:8	KEYW[15:8]							
		7:0	KEYW[7:0]							
0x24	AES_KEYWR1	31:24	KEYW[31:24]							
		23:16	KEYW[23:16]							
		15:8	KEYW[15:8]							
		7:0	KEYW[7:0]							
0x28	AES_KEYWR2	31:24	KEYW[31:24]							
		23:16	KEYW[23:16]							
		15:8	KEYW[15:8]							
		7:0	KEYW[7:0]							
0x2C	AES_KEYWR3	31:24	KEYW[31:24]							
		23:16	KEYW[23:16]							
		15:8	KEYW[15:8]							
		7:0	KEYW[7:0]							
0x30	AES_KEYWR4	31:24	KEYW[31:24]							
		23:16	KEYW[23:16]							
		15:8	KEYW[15:8]							
		7:0	KEYW[7:0]							
0x34	AES_KEYWR5	31:24	KEYW[31:24]							
		23:16	KEYW[23:16]							
		15:8	KEYW[15:8]							
		7:0	KEYW[7:0]							
0x38	AES_KEYWR6	31:24	KEYW[31:24]							
		23:16	KEYW[23:16]							
		15:8	KEYW[15:8]							
		7:0	KEYW[7:0]							
0x3C	AES_KEYWR7	31:24	KEYW[31:24]							
		23:16	KEYW[23:16]							
		15:8	KEYW[15:8]							
		7:0	KEYW[7:0]							

© 2026 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x40	AES_IDATAR0	31:24	IDATA[31:24]							
		23:16	IDATA[23:16]							
		15:8	IDATA[15:8]							
		7:0	IDATA[7:0]							
0x44	AES_IDATAR1	31:24	IDATA[31:24]							
		23:16	IDATA[23:16]							
		15:8	IDATA[15:8]							
		7:0	IDATA[7:0]							
0x48	AES_IDATAR2	31:24	IDATA[31:24]							
		23:16	IDATA[23:16]							
		15:8	IDATA[15:8]							
		7:0	IDATA[7:0]							
0x4C	AES_IDATAR3	31:24	IDATA[31:24]							
		23:16	IDATA[23:16]							
		15:8	IDATA[15:8]							
		7:0	IDATA[7:0]							
0x50	AES_ODATAR0	31:24	ODATA[31:24]							
		23:16	ODATA[23:16]							
		15:8	ODATA[15:8]							
		7:0	ODATA[7:0]							
0x54	AES_ODATAR1	31:24	ODATA[31:24]							
		23:16	ODATA[23:16]							
		15:8	ODATA[15:8]							
		7:0	ODATA[7:0]							
0x58	AES_ODATAR2	31:24	ODATA[31:24]							
		23:16	ODATA[23:16]							
		15:8	ODATA[15:8]							
		7:0	ODATA[7:0]							
0x5C	AES_ODATAR3	31:24	ODATA[31:24]							
		23:16	ODATA[23:16]							
		15:8	ODATA[15:8]							
		7:0	ODATA[7:0]							
0x60	AES_IVR0	31:24	IV[31:24]							
		23:16	IV[23:16]							
		15:8	IV[15:8]							
		7:0	IV[7:0]							
0x64	AES_IVR1	31:24	IV[31:24]							
		23:16	IV[23:16]							
		15:8	IV[15:8]							
		7:0	IV[7:0]							
0x68	AES_IVR2	31:24	IV[31:24]							
		23:16	IV[23:16]							
		15:8	IV[15:8]							
		7:0	IV[7:0]							
0x6C	AES_IVR3	31:24	IV[31:24]							
		23:16	IV[23:16]							
		15:8	IV[15:8]							
		7:0	IV[7:0]							
0x70	AES_AADLENR	31:24	AADLEN[31:24]							
		23:16	AADLEN[23:16]							
		15:8	AADLEN[15:8]							
		7:0	AADLEN[7:0]							
0x74	AES_CLENR	31:24	CLEN[31:24]							
		23:16	CLEN[23:16]							
		15:8	CLEN[15:8]							
		7:0	CLEN[7:0]							
0x78	AES_GHASHR0	31:24	GHASH[31:24]							
		23:16	GHASH[23:16]							
		15:8	GHASH[15:8]							
		7:0	GHASH[7:0]							

© 2026 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x7C	AES_GHASHR1	31:24	GHASH[31:24]							
		23:16	GHASH[23:16]							
		15:8	GHASH[15:8]							
		7:0	GHASH[7:0]							
0x80	AES_GHASHR2	31:24	GHASH[31:24]							
		23:16	GHASH[23:16]							
		15:8	GHASH[15:8]							
		7:0	GHASH[7:0]							
0x84	AES_GHASHR3	31:24	GHASH[31:24]							
		23:16	GHASH[23:16]							
		15:8	GHASH[15:8]							
		7:0	GHASH[7:0]							
0x88	AES_TAGR0	31:24	TAG[31:24]							
		23:16	TAG[23:16]							
		15:8	TAG[15:8]							
		7:0	TAG[7:0]							
0x8C	AES_TAGR1	31:24	TAG[31:24]							
		23:16	TAG[23:16]							
		15:8	TAG[15:8]							
		7:0	TAG[7:0]							
0x90	AES_TAGR2	31:24	TAG[31:24]							
		23:16	TAG[23:16]							
		15:8	TAG[15:8]							
		7:0	TAG[7:0]							
0x94	AES_TAGR3	31:24	TAG[31:24]							
		23:16	TAG[23:16]							
		15:8	TAG[15:8]							
		7:0	TAG[7:0]							
0x98	AES_CTRR	31:24	CTR[31:24]							
		23:16	CTR[23:16]							
		15:8	CTR[15:8]							
		7:0	CTR[7:0]							
0x9C	AES_GCMHR0	31:24	H[31:24]							
		23:16	H[23:16]							
		15:8	H[15:8]							
		7:0	H[7:0]							
0xA0	AES_GCMHR1	31:24	H[31:24]							
		23:16	H[23:16]							
		15:8	H[15:8]							
		7:0	H[7:0]							
0xA4	AES_GCMHR2	31:24	H[31:24]							
		23:16	H[23:16]							
		15:8	H[15:8]							
		7:0	H[7:0]							
0xA8	AES_GCMHR3	31:24	H[31:24]							
		23:16	H[23:16]							
		15:8	H[15:8]							
		7:0	H[7:0]							
0xAC ... 0xAF	Reserved									
0xB0	AES_EMR	31:24	BPE							ALGO
		23:16	NHEAD[7:0]							
		15:8	PADLEN[7:0]							
		7:0	PKRS	PKWL	PLIPD	PLIPEN		KSEL	APM	APEN
0xB4	AES_BCNT	31:24	BCNT[31:24]							
		23:16	BCNT[23:16]							
		15:8	BCNT[15:8]							
		7:0	BCNT[7:0]							

© 2026 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xB8 ... 0xE3	Reserved									
0xE4	AES_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0	ACTION[2:0]			FIRSTE		WPCREN	WPITEN	WPEN
0xE8	AES_WPSR	31:24	ECLASS				SWETYP[3:0]			
		23:16								
		15:8	WPVSRC[7:0]							
		7:0				PKRPVS	SWE	SEQE	CGD	WPVS

© 2026 Microchip Technology Inc. and its subsidiaries

39.5.1. AES Control Register

Name: AES_CR
Offset: 0x00
Reset: –
Property: Write-only

This register can only be written if the WPCREN bit is cleared in the AES Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
								UNLOCK
Access								W
Reset								–

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
								SWRST
Access								W
Reset								–

Bit	7	6	5	4	3	2	1	0
							KSWP	START
Access							W	W
Reset							–	–

Bit 24 – UNLOCK Unlock Processing
AES_WPSR must be cleared before performing the unlock command.

Value	Description
0	No effect.
1	Unlocks the processing in case of abnormal event detection if AES_WPMR.ACTION > 0.

Bit 8 – SWRST Software Reset

Value	Description
0	No effect.
1	Resets the AES. A software-triggered reset of the AES interface is performed.

Bit 1 – KSWP Key Swap

Value	Description
0	No effect.
1	Activates the set of key registers defined by AES_EMR if AES_EMR.PKRS has been changed.

Bit 0 – START Start Processing

Value	Description
0	No effect.
1	Starts manual encryption/decryption process.

© 2026 Microchip Technology Inc. and its subsidiaries

39.5.2. AES Mode Register

Name: AES_MR
Offset: 0x04
Reset: 0x00080000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the AES Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	TAMPCLR							
Access	R/W							
Reset	–							

Bit	23	22	21	20	19	18	17	16
	CKEY[3:0]					CFBS[2:0]		
Access	W	W	W	W		R/W	R/W	R/W
Reset	–	–	–	–		0	0	0

Bit	15	14	13	12	11	10	9	8
	LOD	OPMOD[2:0]			KEYSIZE[1:0]		SMOD[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PROCDLY[3:0]				DUALBUFF		GTAGEN	CIPHER
Access	R/W	R/W	R/W	R/W	R/W		R/W	R/W
Reset	0	0	0	0	0		0	0

Bit 31 – TAMPCLR Tamper Clear Enable

Value	Description
0	A tamper detection event has no effect on the AES_KEYWRx key.
1	A tamper detection event immediately clears the AES_KEYWRx key.

Bits 23:20 – CKEY[3:0] Key

Value	Name	Description
0xE	PASSWD	This field must be written with 0xE the first time AES_MR is programmed. For subsequent programming of AES_MR, any value can be written, including that of 0xE. Always reads as 0.

Bits 18:16 – CFBS[2:0] Cipher Feedback Data Size

Value	Name	Description
0	SIZE_128BIT	128-bit
1	SIZE_64BIT	64-bit
2	SIZE_32BIT	32-bit
3	SIZE_16BIT	16-bit
4	SIZE_8BIT	8-bit

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 15 – LOD Last Output Data Mode

> **WARNING** In PDC mode, reading to the Output Data registers before the last data encryption/decryption process may lead to unpredictable results.

Value	Description
0	No effect. After each end of encryption/decryption, the output data are available either on the output data registers (Manual and Auto modes) or at the address specified in the Receive Pointer Register (AES_RPR) for PDC mode. In Manual and Auto modes, the DATRDY flag is cleared when at least one of the Output Data registers is read.
1	The DATRDY flag is cleared when at least one of the Input Data Registers is written. No more Output Data Register reads are necessary between consecutive encryptions/decryptions (see Last Output Data Mode).

Bits 14:12 – OPMOD[2:0] Operating Mode

For CBC-MAC operating mode, set OPMOD to CBC and LOD to 1.
When switching from an operating mode requiring the initialization vectors (e.g. CBC, GCM) to another operating mode that does not require initialization vectors (e.g. ECB) and a message of one block has been processed, initialization vector registers (AES_IVRx) must be cleared before switching to the new mode.

Value	Name	Description
0	ECB	ECB: Electronic Codebook mode
1	CBC	CBC: Cipher Block Chaining mode
2	OFB	OFB: Output Feedback mode
3	CFB	CFB: Cipher Feedback mode
4	CTR	CTR: Counter mode (16-bit internal counter)
5	GCM	GCM: Galois/Counter mode

Bits 11:10 – KEYSIZE[1:0] Key Size

Value	Name	Description
0	AES128	AES Key Size is 128 bits
1	AES192	AES Key Size is 192 bits
2	AES256	AES Key Size is 256 bits

Bits 9:8 – SMOD[1:0] Start Mode

If a PDC transfer is used, configure SMOD to 2. See PDC Mode for more details.

Value	Name	Description
0	MANUAL_START	Manual Mode
1	AUTO_START	Auto Mode
2	IDATAR0_START	AES_IDATAR0 access only Auto Mode (PDC)

Bits 7:4 – PROCDLY[3:0] Processing Delay

Processing Time = N × (PROCDLY + 1)
where

- N = 10 when KEYSIZE = 0
- N = 12 when KEYSIZE = 1
- N = 14 when KEYSIZE = 2

The processing time represents the number of clock cycles that the AES needs in order to perform one encryption/decryption.
Note: The best performance is achieved with PROCDLY equal to 0.

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 3 – DUALBUFF Dual Input Buffer

Value	Name	Description
0	INACTIVE	AES_IDATARx cannot be written during processing of previous block.
1	ACTIVE	AES_IDATARx can be written during processing of previous block when SMOD = 2. It speeds up the overall runtime of large files.

Bit 1 – GTAGEN GCM Automatic Tag Generation Enable

Value	Description
0	Automatic GCM Tag generation disabled.
1	Automatic GCM Tag generation enabled.

Bit 0 – CIPHER Processing Mode

Value	Description
0	Decrypts data.
1	Encrypts data.

© 2026 Microchip Technology Inc. and its subsidiaries

39.5.3. AES Interrupt Enable Register

Name: AES_IER
Offset: 0x10
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the AES Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					SECE	PLENERR	EOPAD	TAGRDY
Access					W	W	W	W
Reset					–	–	–	–

Bit	15	14	13	12	11	10	9	8
								URAD
Access								W
Reset								–

Bit	7	6	5	4	3	2	1	0
				TXBUFE	RXBUFF	ENDTX	ENDRX	DATRDY
Access				W	W	W	W	W
Reset				–	–	–	–	–

Bit 19 – SECE Security and/or Safety Event Interrupt Enable

Bit 18 – PLENERR Padding Length Error Interrupt Enable

Bit 17 – EOPAD End of Padding Interrupt Enable

Bit 16 – TAGRDY GCM Tag Ready Interrupt Enable

Bit 8 – URAD Unspecified Register Access Detection Interrupt Enable

Bit 4 – TXBUFE Transmit Buffer Empty Interrupt Enable

Bit 3 – RXBUFF Receive Buffer Full Interrupt Enable

Bit 2 – ENDTX End of Transmit Buffer Interrupt Enable

Bit 1 – ENDRX End of Receive Buffer Interrupt Enable

Bit 0 – DATRDY Data Ready Interrupt Enable

© 2026 Microchip Technology Inc. and its subsidiaries

39.5.4. AES Interrupt Disable Register

Name: AES_IDR
Offset: 0x14
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the AES Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					SECE	PLENERR	EOPAD	TAGRDY
Access					W	W	W	W
Reset					–	–	–	–

Bit	15	14	13	12	11	10	9	8
								URAD
Access								W
Reset								–

Bit	7	6	5	4	3	2	1	0
				TXBUFE	RXBUFF	ENDTX	ENDRX	DATRDY
Access				W	W	W	W	W
Reset				–	–	–	–	–

Bit 19 – SECE Security and/or Safety Event Interrupt Disable

Bit 18 – PLENERR Padding Length Error Interrupt Disable

Bit 17 – EOPAD End of Padding Interrupt Disable

Bit 16 – TAGRDY GCM Tag Ready Interrupt Disable

Bit 8 – URAD Unspecified Register Access Detection Interrupt Disable

Bit 4 – TXBUFE Transmit Buffer Empty Interrupt Disable

Bit 3 – RXBUFF Receive Buffer Full Interrupt Disable

Bit 2 – ENDTX End of Transmit Buffer Interrupt Disable

Bit 1 – ENDRX End of Receive Buffer Interrupt Disable

Bit 0 – DATRDY Data Ready Interrupt Disable

© 2026 Microchip Technology Inc. and its subsidiaries

39.5.5. AES Interrupt Mask Register

Name: AES_IMR
Offset: 0x18
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					SECE	PLENERR	EOPAD	TAGRDY
Access					R	R	R	R
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
								URAD
Access								R
Reset								0

Bit	7	6	5	4	3	2	1	0
				TXBUFE	RXBUFF	ENDTX	ENDRX	DATRDY
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit 19 – SECE Security and/or Safety Event Interrupt Mask

Bit 18 – PLENERR Padding Length Error Interrupt Mask

Bit 17 – EOPAD End of Padding Interrupt Mask

Bit 16 – TAGRDY GCM Tag Ready Interrupt Mask

Bit 8 – URAD Unspecified Register Access Detection Interrupt Mask

Bit 4 – TXBUFE Transmit Buffer Empty Interrupt Mask

Bit 3 – RXBUFF Receive Buffer Full Interrupt Mask

Bit 2 – ENDTX End of Transmit Buffer Interrupt Mask

Bit 1 – ENDRX End of Receive Buffer Interrupt Mask

Bit 0 – DATRDY Data Ready Interrupt Mask

© 2026 Microchip Technology Inc. and its subsidiaries

39.5.6. AES Interrupt Status Register

Name: AES_ISR
Offset: 0x1C
Reset: 0x0000001E
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					SECE	PLENERR	EOPAD	TAGRDY
Access					R	R	R	R
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	URAT[3:0]							URAD
Access	R	R	R	R				R
Reset	0	0	0	0				0

Bit	7	6	5	4	3	2	1	0
				TXBUFE	RXBUFF	ENDTX	ENDRX	DATRDY
Access				R	R	R	R	R
Reset				1	1	1	1	0

Bit 19 – SECE Security and/or Safety Event (cleared on read)

Value	Description
0	There is no security report in AES_WPSR.
1	One security flag is set in AES_WPSR.

Bit 18 – PLENERR Padding Length Error

Value	Description
0	No Padding Length Error occurred.
1	Padding Length Error detected.

Bit 17 – EOPAD End of Padding

Value	Description
0	Padding is not over.
1	Padding phase is over.

Bit 16 – TAGRDY GCM Tag Ready

Value	Description
0	GCM Tag is not valid.
1	GCM Tag generation is complete (cleared by reading GCM Tag, starting another processing or when writing a new key).

Bits 15:12 – URAT[3:0] Unspecified Register Access (cleared by writing SWRST in AES_CR)
Only the last Unspecified Register Access Type is available through the URAT field.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Name	Description
0	IDR_WR_PROCESSING	Input Data register written during the data processing when SMOD = 2 mode.
1	ODR_RD_PROCESSING	Output Data register read during the data processing.
2	MR_WR_PROCESSING	Mode register written during the data processing.
3	ODR_RD_SUBKGEN	Output Data register read during the sub-keys generation.
4	MR_WR_SUBKGEN	Mode register written during the sub-keys generation.
5	WOR_RD_ACCESS	Write-only register read access.

Bit 8 – URAD Unspecified Register Access Detection Status (cleared by writing SWRST in AES_CR)

Value	Description
0	No unspecified register access has been detected since the last SWRST.
1	At least one unspecified register access has been detected since the last SWRST.

Bit 4 – TXBUFE TX Buffer Empty (cleared by writing AES_TCR or AES_TNCR)

Value	Description
0	AES_TCR or AES_TNCR has a value other than 0.
1	AES_TCR and AES_TNCR both have a value of 0.

Note: This flag must be used only in PDC mode with AES_MR.LOD bit set.

Bit 3 – RXBUFF RX Buffer Full (cleared by writing AES_RCR or AES_RNCR)

Value	Description
0	AES_RCR or AES_RNCR has a value other than 0.
1	AES_RCR and AES_RNCR both have a value of 0.

Note: This flag must be used only in PDC mode with AES_MR.LOD bit cleared.

Bit 2 – ENDTX End of TX Buffer (cleared by writing AES_TCR or AES_TNCR)

Value	Description
0	The Transmit Counter register has not reached 0 since the last write in AES_TCR or AES_TNCR.
1	The Transmit Counter register has reached 0 since the last write in AES_TCR or AES_TNCR.

Note: This flag must be used only in PDC mode with AES_MR.LOD bit set.

Bit 1 – ENDRX End of RX Buffer (cleared by writing AES_RCR or AES_RNCR)

Value	Description
0	The Receive Counter register has not reached 0 since the last write in AES_RCR or AES_RNCR.
1	The Receive Counter register has reached 0 since the last write in AES_RCR or AES_RNCR.

Note: This flag must be used only in PDC mode with AES_MR.LOD bit cleared.

Bit 0 – DATRDY Data Ready (cleared by setting bit START or bit SWRST in AES_CR or by reading AES_ODATARx)

Value	Description
0	Output data not valid.
1	Encryption or decryption process is completed.

Note: If AES_MR.LOD = 1: In Manual and Auto mode, the DATRDY flag can also be cleared by writing at least one AES_IDATARx.

© 2026 Microchip Technology Inc. and its subsidiaries

39.5.7. AES Key Word Register x

Name: AES_KEYWRx
Offset: 0x20 + x*0x04 [x=0..7]
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the AES Write Protection Mode Register.

These registers are write-only to prevent the key from being read by another application.

Note: AES_KEYWRx registers are not used if the Private Key internal registers are selected (AES_EMR.PKRS=1).

Bit	31	30	29	28	27	26	25	24
	KEYW[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	KEYW[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	KEYW[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	KEYW[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – KEYW[31:0] Key Word

The four/six/eight 32-bit Key Word registers set the 128-bit/192-bit/256-bit cryptographic key used for AES encryption/decryption. Depending on AES_EMR.KSEL, the first key or the second key is written. See Temporary Secured Storage for Keys.
AES_KEYWR0 corresponds to the first word of the key and respectively AES_KEYWR3/AES_KEYWR5/AES_KEYWR7 to the last one.
Whenever a new key (AES_KEYWRx) is written to the hardware, two automatic actions are processed:

- GCM hash subkey generation
- AES_GHASHRx Clear

See Key Writing and Automatic Hash Subkey Calculation for details.
These registers are write-only to prevent the key from being read by another application.
Note: To write AES_KEYWRx and start using the key immediately, AES_EMR.PKRS must be written to 0 prior to writing AES_KEYWRx.

© 2026 Microchip Technology Inc. and its subsidiaries

39.5.8. AES Input Data Register x

Name: AES_IDATARx
Offset: 0x40 + x*0x04 [x=0..3]
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	IDATA[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	IDATA[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	IDATA[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	IDATA[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – IDATA[31:0] Input Data Word
The four 32-bit Input Data registers set the 128-bit data block used for encryption/decryption. AES_IDATAR0 corresponds to the first word of the data to be encrypted/decrypted, and AES_IDATAR3 to the last one.
These registers are write-only to prevent the input data from being read by another application.

© 2026 Microchip Technology Inc. and its subsidiaries

39.5.9. AES Output Data Register x

Name: AES_ODATARx
Offset: 0x50 + x*0x04 [x=0..3]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	ODATA[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ODATA[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ODATA[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ODATA[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ODATA[31:0] Output Data
The four 32-bit Output Data registers contain the 128-bit data block that has been encrypted/decrypted.
AES_ODATAR0 corresponds to the first word, AES_ODATAR3 to the last one.

© 2026 Microchip Technology Inc. and its subsidiaries

39.5.10. AES Initialization Vector Register x

Name: AES_IVRx
Offset: 0x60 + x*0x04 [x=0..3]
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the AES Write Protection Mode Register.

The four 32-bit Initialization Vector registers set the 128-bit Initialization Vector data block that is used by some modes of operation as an additional initial input.

AES_IVR0 corresponds to the first word of the Initialization Vector, AES_IVR3 to the last one.

These registers are write-only to prevent the Initialization Vector from being read by another application.

For CBC, OFB and CFB modes, the IV input value corresponds to the initialization vector.

For CTR mode, the IV input value corresponds to the initial counter value.

These registers are not used in ECB mode and must not be written.

When switching from an operating mode requiring the initialization vectors (e.g. CBC, GCM) to another operating mode that does not require initialization vectors (e.g. ECB) and a message of one block has been processed, AES_IVRx must be cleared before switching to the new mode

Bit	31	30	29	28	27	26	25	24
	IV[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	IV[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	IV[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	IV[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – IV[31:0] Initialization Vector

© 2026 Microchip Technology Inc. and its subsidiaries

39.5.11. AES Additional Authenticated Data Length Register

Name: AES_AADLENR
Offset: 0x70
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the AES Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	AADLEN[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	AADLEN[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	AADLEN[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	AADLEN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – AADLEN[31:0] Additional Authenticated Data Length
Length in bytes of the Additional Authenticated Data (AAD) that is to be processed.
Note: The maximum byte length of the AAD portion of a message is limited to the 32-bit counter length.

© 2026 Microchip Technology Inc. and its subsidiaries

39.5.12. AES Plaintext/Ciphertext Length Register

Name: AES_CLENR
Offset: 0x74
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the AES Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	CLEN[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CLEN[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CLEN[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CLEN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CLEN[31:0] Plaintext/Ciphertext Length
Length in bytes of the plaintext/ciphertext (C) data that is to be processed.
Note: The maximum byte length of the C portion of a message is limited to the 32-bit counter length.

© 2026 Microchip Technology Inc. and its subsidiaries

39.5.13. AES GCM Intermediate Hash Word Register x

Name: AES_GHASHRx
Offset: 0x78 + x*0x04 [x=0..3]
Reset: 0x00000000
Property: R/W

This register can only be written if the WPEN bit is cleared in the AES Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	GHASH[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	GHASH[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	GHASH[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	GHASH[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – GHASH[31:0] Intermediate GCM Hash Word x

The four 32-bit Intermediate Hash Word registers expose the intermediate GHASH value. May be read to save the current GHASH value so processing can later be resumed, presumably on a later message fragment. Whenever a new key is written in AES_KEYWRx, two automatic actions are processed:

- GCM hash subkey generation
- AES_GHASHRx Clear

See Key Writing and Automatic Hash Subkey Calculation for details.
If an application software-specific hash initial value is needed for the GHASH, it must be written to AES_GHASHRx:

- after writing AES_KEYWRx, if any
- before starting the input data feed.

© 2026 Microchip Technology Inc. and its subsidiaries

39.5.14. AES GCM Authentication Tag Word Register x

Name: AES_TAGRx
Offset: 0x88 + x*0x04 [x=0..3]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	TAG[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	TAG[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TAG[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TAG[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – TAG[31:0] GCM Authentication Tag x
The four 32-bit Tag registers contain the final 128-bit GCM Authentication tag (*T*) when GCM processing is complete. TAG0 corresponds to the first word, TAG3 to the last word.

© 2026 Microchip Technology Inc. and its subsidiaries

39.5.15. AES GCM Encryption Counter Value Register

Name: AES_CTRR
Offset: 0x98
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	CTR[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CTR[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CTR[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CTR[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CTR[31:0] GCM Encryption Counter
Reports the current value of the 32-bit GCM counter.

© 2026 Microchip Technology Inc. and its subsidiaries

39.5.16. AES GCM H Word Register x

Name: AES_GCMHRx
Offset: 0x9C + x*0x04 [x=0..3]
Reset: 0x00000000
Property: R/W

This register can only be written if the WPEN bit is cleared in the AES Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	H[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	H[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	H[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	H[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – H[31:0] GCM H Word x

The four 32-bit H Word registers contain the 128-bit GCM hash subkey *H*value.
Whenever a new key is written in AES_KEYWRx, two automatic actions are processed:

- GCM hash subkey *H* generation
- AES_GHASHRx Clear

If the application software requires a specific hash subkey, the automatically-generated H value can be overwritten in AES_GCMHRx. See Key Writing and Automatic Hash Subkey Calculation for details.
Generating a GCM hash subkey *H* by a write in AES_GCMHRx enables to:

- select the GCM hash subkey *H* for GHASH operations,
- select one operand to process a single GF128 multiply.

© 2026 Microchip Technology Inc. and its subsidiaries

39.5.17. AES Extended Mode Register

Name: AES_EMR
Offset: 0xB0
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the AES Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	BPE							ALGO
Access	R/W							R/W
Reset	0							0

Bit	23	22	21	20	19	18	17	16
	NHEAD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	PADLEN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PKRS	PKWL	PLIPD	PLIPEN		KSEL	APM	APEN
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bit 31 – BPE Block Processing End

Value	Description
0	AES_ISR.DATRDY flag reports only the end message encryption processing. No intermediate block processing is reported when SMOD=2. When a DMA is used to transfer data, BPE must be cleared.
1	AES_ISR.DATRDY flag reports each end of block processing when SMOD=2. When AES_IDATARx are not loaded by a DMA and SMOD=2, this bit can be written to 1 to rise the AES_ISR.DATRDY flag when a new data block can be written.

Bit 24 – ALGO Encryption Algorithm

Value	Name	Description
0	AES	The AES algorithm is used for encryption.
1	ARIA	The ARIA algorithm is used for encryption.

Bits 23:16 – NHEAD[7:0] IPSec Next Header

Value	Description
0–255	IPSec Next Header field

Bits 15:8 – PADLEN[7:0] Auto Padding Length

Value	Description
0–255	Padding length in bytes

Bit 7 – PKRS Private Key Internal Register Select

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	The key used by the AES is in the AES_KEYWRx registers.
1	The key used by the AES is in the Private Key internal registers written through the Private Key bus.

Bit 6 – PKWL Private Key Write Lock

Once PKWL is set to '1', only a hardware reset sets this bit to '0' internally. Writing it to '0' with a register access has no impact (although the field will be read to value '0').

Value	Description
0	The Private Key internal registers can be written multiple times via the Private Key bus.
1	The Private Key internal registers can be written only once via the Private Key bus until hardware reset.

Bit 5 – PLIPD Protocol Layer Improved Performance Decipher

Value	Description
0	Protocol layer improved performance is in ciphering mode.
1	Protocol layer improved performance is in deciphering mode.

Bit 4 – PLIPEN Protocol Layer Improved Performance Enable

Value	Description
0	Protocol layer improved performance is disabled.
1	Protocol layer improved performance is enabled.

Bit 2 – KSEL Key Selection

Value	Description
0	Selects the first key loaded by software.
1	Selects the second key loaded by software.

Bit 1 – APM Auto Padding Mode

Value	Description
0	Auto Padding performed according to IPSec standard.
1	Auto Padding performed according to SSL standard.

Bit 0 – APEN Auto Padding Enable

Value	Description
0	Auto Padding feature is disabled.
1	Auto Padding feature is enabled.

© 2026 Microchip Technology Inc. and its subsidiaries

39.5.18. AES Byte Counter Register

Name: AES_BCNT
Offset: 0xB4
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the AES Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	BCNT[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	BCNT[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	BCNT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BCNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – BCNT[31:0] Auto Padding Byte Counter
Auto padding byte counter value. BCNT must be greater than 0.

© 2026 Microchip Technology Inc. and its subsidiaries

39.5.19. AES Write Protection Mode Register

Name: AES_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	ACTION[2:0]			FIRSTE		WPCREN	WPITEN	WPEN
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x414553	PASSWD	Writing any other value in this field aborts the write operation of the WPEN,WPITEN,WPCREN bits. Always reads as 0.

Bits 7:5 – ACTION[2:0] Action on Abnormal Event Detection

When the field AES_WPMR.ACTION differs from 0 and an abnormal event or internal state is detected, the AES is locked until the unlock command is issued (AES_CR.UNLOCK=1). The lock source must be cleared before performing the unlock command. If AES_WPSR.SEQE=1, the following two actions must be performed:
1/ Read AES_WPSR.
2/ Issue software reset by writing a 1 in AES_CR.SWRST.
A specific configuration applies where the sequence does not clear the lock source (AES_WPSR=0). If AES_WPSR.SEQE remains high after the clearing sequence, then only a hardware reset will unlock the AES. A hardware reset can be performed by issuing a reset controller software reset (refer to the section “Reset Controller (RSTC)”). This condition can be met when AES_EMR.PKWL=1 and a key has been loaded through the Private Key bus. The key loaded through the key bus is corrupted, but it is impossible to reload a new key unless a hardware reset is issued.

Value	Name	Description
0	REPORT_ONLY	No action (stop or clear key) is performed when one of PKRPVS, WPVS, CGD, SEQE, or SWE flags is set.
1	LOCK_PKRPVS_WPVS_SWE	If a processing is in progress when the AES_WPSR.PKRPVS/WPVS/SWE event detection occurs, the current processing is ended normally but no other processing is started while a AES_CR.UNLOCK command is issued.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Name	Description
2	LOCK_CGD_SEQE	If a processing is in progress when the AES_WPSR.CGD/SEQE event detection occurs, the current processing is ended normally but no other processing is started while a AES_CR.UNLOCK command is issued.
3	LOCK_ANY_EV	If a processing is in progress when the AES_WPSR.PKRPVS/WPVS/CGD/SEQE/SWE events detection occurs, the current processing is ended normally but no other processing is started while a AES_CR.UNLOCK command is issued.
4	CLEAR_PKRPVS_WPVS_SWE	If a processing is in progress when the AES_WPSR.PKRPVS/WPVS/SWE events detection occurs, the current processing is ended normally but no other processing is started while a AES_CR.UNLOCK command is issued. Moreover, the AES_KEYWRx key is immediately cleared.
5	CLEAR_CGD_SEQE	If a processing is in progress when the AES_WPSR.CGD/SEQE events detection occurs, the current processing is ended normally but no other processing is started while a AES_CR.UNLOCK command is issued. Moreover, the AES_KEYWRx key is immediately cleared.
6	CLEAR_ANY_EV	If a processing is in progress when the AES_WPSR.PKRPVS/WPVS/CGD/SEQE/SWE events detection occurs, the current processing is ended normally but no other processing is started while a AES_CR.UNLOCK command is issued. Moreover, the AES_KEYWRx key is immediately cleared.

Bit 4 – FIRSTE First Error Report Enable

Value	Description
0	The last write protection violation source is reported in AES_WPSR.WPVSRC and the last software control error type is reported in AES_WPSR.SWETYP. The AES_ISR.SECE flag is set at the first error occurrence within a series.
1	Only the first write protection violation source is reported in AES_WPSR.WPVSRC and only the first software control error type is reported in AES_WPSR.SWETYP. The AES_ISR.SECE flag is set at the first error occurrence within a series.

Bit 2 – WPCREN Write Protection Control Enable

Value	Description
0	Disables the write protection on control register if WPKEY corresponds to 0x414553 (“AES” in ASCII).
1	Enables the write protection on control register if WPKEY corresponds to 0x414553 (“AES” in ASCII).

Bit 1 – WPITEN Write Protection Interruption Enable

Value	Description
0	Disables the write protection on interrupt registers if WPKEY corresponds to 0x414553 (“AES” in ASCII).
1	Enables the write protection on interrupt registers if WPKEY corresponds to 0x414553 (“AES” in ASCII).

Bit 0 – WPEN Write Protection Configuration Enable
See Register Write Protection for the list of registers that can be write-protected.

Value	Description
0	Disables the write protection on configuration registers if WPKEY corresponds to 0x414553 (“AES” in ASCII).
1	Enables the write protection on configuration registers if WPKEY corresponds to 0x414553 (“AES” in ASCII).

© 2026 Microchip Technology Inc. and its subsidiaries

39.5.20. AES Write Protection Status Register

Name: AES_WPSR
Offset: 0xE8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	ECLASS				SWETYP[3:0]			
Access	R				R	R	R	R
Reset	0				0	0	0	0

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
				PKRPVS	SWE	SEQE	CGD	WPVS
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit 31 – ECLASS Software Error Class (cleared on read)
0 (WARNING): An abnormal access that does not affect system functionality
1 (ERROR): An access is performed into key, input data, control registers while the AES is performing an encryption/decryption or a start is request by software or DMA while the key is not fully configured.

Bits 27:24 – SWETYP[3:0] Software Error Type (cleared on read)

Value	Name	Description
0	READ_WO	A write-only register has been read (Warning).
1	WRITE_RO	AES is enabled and a write access has been performed on a read-only register (Warning).
2	UNDEF_RW	Access to an undefined address (Warning).
3	CTRL_START	Abnormal use of AES_CR.START command when DMA access is configured.
4	WEIRD_ACTION	A key write, init value write, output data read, AES_MR and AES_EMR write, GCM configuration registers write, AES_BCNT write, Private Key Bus access has been performed while a current processing is in progress (abnormal).
5	INCOMPLETE_KEY	A tentative of start is required while the key is not fully loaded into the AES_KEYWRx registers.

Bits 15:8 – WPVSRC[7:0] Write Protection Violation Source
When WPVS=1, WPVSRC indicates the register address offset at which a write access has been attempted.
When WPVS=0 and SWE=1, WPVSRC reports the address of the incorrect software access. As soon as WPVS=1, WPVSRC returns the address of the write-protected violation.

Bit 4 – PKRPVS Private Key Internal Register Protection Violation Status (cleared on read)

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	No Private Key Internal Register access violation has occurred since the last read of AES_WPSR.
1	A Private Key Internal Register access violation has occurred since the last read of AES_WPSR.

Bit 3 – SWE Software Control Error (cleared on read)

Value	Description
0	No software error has occurred since the last read of AES_WPSR.
1	A software error has occurred since the last read of AES_WPSR. The field SWETYP details the type of software error; the associated incorrect software access is reported in the field WPVSRC (if WPVS=0).

Bit 2 – SEQE Internal Sequencer Error (cleared on read)

Value	Description
0	No peripheral internal sequencer error has occurred since the last read of AES_WPSR.
1	A peripheral internal sequencer error has occurred since the last read of AES_WPSR. This flag can only be set under abnormal operating conditions.

Bit 1 – CGD Clock Glitch Detected (cleared on read)

Value	Description
0	The clock monitoring circuitry has not been corrupted since the last read of AES_WPSR. Under normal operating conditions, this bit is always cleared.
1	The clock monitoring circuitry has been corrupted since the last read of AES_WPSR. This flag can only be set in case of abnormal clock signal waveform (glitch).

Bit 0 – WPVS Write Protection Violation Status (cleared on read)

Value	Description
0	No write protect violation has occurred since the last read of AES_WPSR.
1	A write protect violation has occurred since the last read of AES_WPSR. The address offset of the violated register is reported into field WPVSRC.

© 2026 Microchip Technology Inc. and its subsidiaries

40. Advanced Encryption Standard Bridge (AESB)

40.1. Description

The Advanced Encryption Standard Bridge (AESB) provides on-the-fly off-chip memory encryption/decryption compliant with the American *FIPS (Federal Information Processing Standard) Publication 197* specification.

The AESB supports three confidentiality modes of operation for symmetrical key block cipher algorithms (ECB, CBC and CTR), as specified in the *NIST Special Publication 800-38A Recommendation*.

The AESB key can be either loaded by the software or loaded in an invisible manner from the software.

The 128-bit AESB key is stored in the AESB Key register made of four 32-bit write-only AESB Key Word registers (AESB_KEYWR0–3). For a software-invisible key transfer, the private key bus accesses the private key internal register from the TRNG. PKRS in the Extended Mode register (AESB_EMR) selects either AESB_KEYWRx or the private key internal register.

The 128-bit input data and initialization vector (for some modes) are each stored in four 32-bit registers (AESB_IDATARx and AESB_IVRx) which are all write-only.

As soon as the initialization vector, the input data and the key are configured, the encryption/decryption process may be started. Then the encrypted/decrypted data will be ready to be read out on the four 32-bit output data registers (AESB_ODATARx).

40.2. Embedded Characteristics

- On-The-Fly Off-Chip Encryption/Decryption
- Compliant with FIPS Publication 197, Advanced Encryption Standard (AES)
- 128-bit Cryptographic Key
- 10 Clock Cycles Encryption/Decryption Inherent Processing Time
- Double Input Buffer Optimizes Runtime
- Support of the Three Standard Modes of Operation Specified in *NIST Special Publication 800-38A, Recommendation for Block Cipher Modes of Operation - Methods and Techniques*:
  - Electronic Code Book (ECB)
  - Cipher Block Chaining (CBC) including CBC-MAC
  - Counter (CTR)
- Last Output Data Mode Allows Optimized Message Authentication Code (MAC) Generation
- Abnormal Software Access Reports
- Two Separate Interrupt Lines: One Line Driven by Standard Functions and the Second Line Driven by Safety Checks
- Register Write Protection
- Private Key Bus Access to the Private Key Internal Register Not Readable from any Peripheral or Software

40.3. Product Dependencies

40.3.1. Power Management

The AESB may be clocked through the Power Management Controller (PMC), so the programmer must first configure the PMC to enable the AESB clock.

© 2026 Microchip Technology Inc. and its subsidiaries

40.3.2. Interrupt Sources

The AESB has one interrupt line used for standard functions and one interrupt line used to trigger any safety or security events that may occur. Both lines are connected to the Interrupt Controller.

Separate lines ease management of interrupt priorities related to safety/security checks.

The interrupt line for standard functions is driven by AEB_IMR and AESB_ISR, whereas the interrupt line for safety/security functions is driven by AESB_WPSR flags. If one of the flags in AESB_WPSR is set, the interrupt line is asserted. Handling the AESB interrupt requires programming the Interrupt Controller before configuring the AESB.

40.4. Functional Description

The Advanced Encryption Standard Bridge (AESB) specifies a FIPS-approved cryptographic algorithm that can be used to protect electronic data. The AESB algorithm is a symmetric block cipher that can encrypt (encipher) and decrypt (decipher) information.

Encryption converts data to an unintelligible form called ciphertext. Decrypting the ciphertext converts the data back into its original form, called plaintext. CIPHER in the AESB Mode register (AESB_MR) allows selection between the encryption and the decryption processes.

The AESB is capable of using cryptographic keys of 128 bits to encrypt and decrypt data in blocks of 128 bits. This 128-bit key is defined in the Key registers (AESB_KEYWRx) or in the private key internal register that is only writable from the private key bus.

The input to the encryption processes of the CBC mode includes, in addition to the plaintext, a 128-bit data block called the initialization vector (IV), which must be set in the Initialization Vector Registers (AESB_IVRx). The initialization vector is used in an initial step in the encryption of a message and in the corresponding decryption of the message. The Initialization Vector registers are also used by the CTR mode to set the counter value.

40.4.1. Operating Modes

The AESB supports the following modes of operation:

- ECB—Electronic Code Book
- CBC—Cipher Block Chaining
- CTR—Counter

The data pre-processing, post-processing and data chaining for the operating modes are performed automatically. Refer to *NIST Special Publication 800-38A Recommendation* for more complete information.

The modes are selected in AESB_MR.OPMOD.

In CTR mode, the size of the block counter embedded in the module is 16 bits. Therefore, there is a rollover after processing 1 megabyte of data. If the file to be processed is greater than 1 megabyte, this file must be split into fragments of 1 megabyte or less for the first fragment if the initial value of the counter is greater than 0. Prior to loading the first fragment into AESB_IDATARx registers, the AESB_IVRx registers must be cleared. For any fragment, after the transfer is completed and prior to transferring the next fragment, AESB_IVR0 must be programmed so that the fragment number (0 for the first fragment, 1 for the second one, and so on) is written in the 16 MSB of AESB_IVR0.

If the initial value of the counter is greater than 0 and the data buffer size to be processed is greater than 1 megabyte, the size of the first fragment to be processed must be 1 megabyte minus 16x(initial value) to prevent a rollover of the internal 1-bit counter.

40.4.2. Double Input Buffer

The input data register can be double-buffered to reduce the runtime of large files.

© 2026 Microchip Technology Inc. and its subsidiaries

This mode allows writing a new message block when the previous message block is being processed.

AESB_MR.DUALBUFF must be set to '1' to access the double buffer.

40.4.3. Start Modes

AESB_MR.SMOD allows selection of the Encryption or Decryption Start mode.

40.4.3.1. Manual Mode

The sequence is as follows:

1. Write AESB_MR with all required fields, including but not limited to SMOD and OPMOD.
2. Write the 128-bit key in the Key registers (AESB_KEYWRx) or private key internal register.
3. Write the initialization vector (or counter) in the Initialization Vector registers (AESB_IVRx).
   Note: The Initialization Vector registers concern all modes except ECB.
4. Set DATRDY (Data Ready) in the AESB Interrupt Enable register (AESB_IER) depending on whether an interrupt is required, or not, at the end of processing.
5. Write the data to be encrypted/decrypted in the authorized Input Data registers (see the table below).

Table 40-1. Authorized Input Data Registers

Operating Mode	Input Data Registers to Write
ECB	All
CBC	All
CTR	All

6. Set the START bit in the AESB Control register (AESB_CR) to begin the encryption or decryption process.
7. When processing is complete, DATRDY in the AESB Interrupt Status register (AESB_ISR) rises. If an interrupt has been enabled by setting AESB_IER.DATRDY, the interrupt line of the AESB is activated.
8. When the software reads one of the Output Data registers (AESB_ODATARx), AESB_ISR.DATRDY is automatically cleared.

40.4.3.2. Auto Mode

In Auto mode, as soon as the correct number of Input Data registers is written, processing starts automatically without any action in AESB_CR.

40.4.4. Last Output Data Mode

Last Output Data mode is used to generate cryptographic checksums on data (MAC) by means of a cipher block chaining encryption algorithm (e.g., the CBC-MAC algorithm).

After each end of encryption/decryption, the output data are available on the output data registers for Manual and Auto modes.

AESB_MR.LOD allows retrieval of only the last data of several encryption/decryption processes.

Those data are only available on the Output Data registers (AESB_ODATARx).

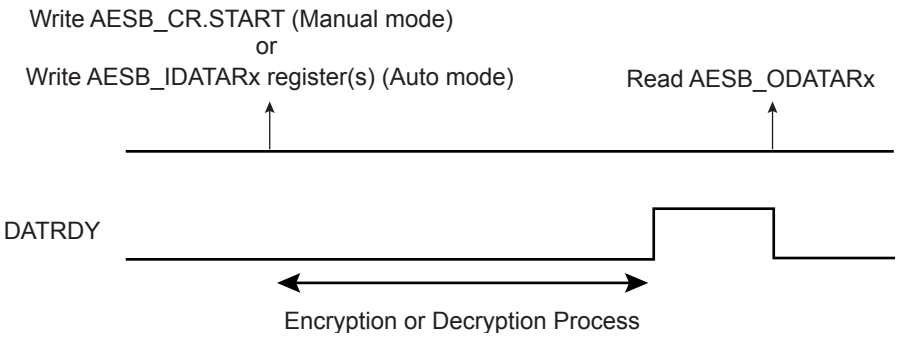
40.4.5. Manual and Auto Modes

40.4.5.1. If AESB_MR.LOD = 0

AESB_ISR.DATRDY is cleared when at least one of the Output Data registers is read (see the figure below).

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 40-1. Manual and Auto Modes with AESB_MR.LOD = 0

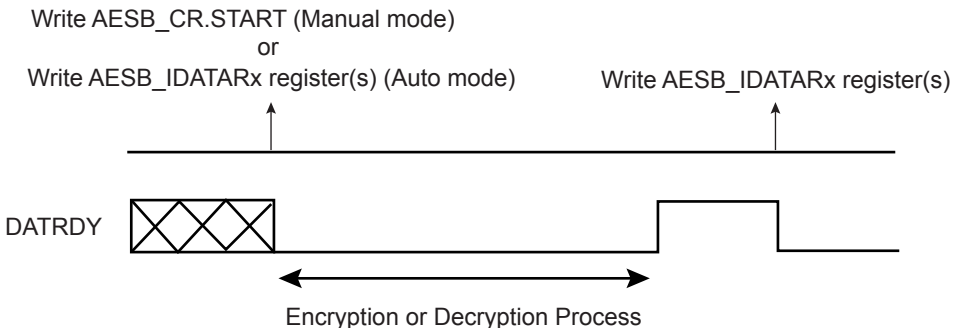


If the user does not want to read the output data registers between each encryption/decryption, AESB_ISR.DATRDY will not be cleared. If AESB_ISR.DATRDY is not cleared, the user cannot know the end of the following encryptions/decryptions.

40.4.5.2. If AESB_MR.LOD = 1

AESB_ISR.DATRDY is cleared when at least one Input Data register is written, so before the start of a new transfer (see the figure below). No more Output Data register reads are necessary between consecutive encryptions/decryptions.

Figure 40-2. Manual and Auto Modes with AESB_MR.LOD = 1



40.4.6. Automatic Bridge Mode

40.4.6.1. Description

The Automatic Bridge mode, when the AESB block is connected between the system bus and a DDR port and the QSPI, provides automatic encryption/decryption without any action on the part of the user. For Automatic Bridge mode, AESB_MR.OPMOD must be configured to 0x4 (see AESB Mode Register). If AESB_MR.AAHB is set and AESB_MR.OPMOD = 0x4, there is no compliance with the standard CTR mode of operation.

In case of write transfer, this mode automatically encrypts the data before writing it to the final client destination. In case of read transfer, this mode automatically decrypts the data read from the target client before putting it on the system bus.

Therefore, this mode does not work if the automatically encrypted data is moved at another address outside of the AESB. This means that for a given data, the encrypted value is not the same if written at different addresses.

40.4.6.2. Configuration

The Automatic Bridge mode can be enabled by setting AESB_MR.AAHB.

The IV (Initialization Vector) field of the AESB Initialization Vector register x (AESB_IVRx) can be used to add a nonce in the encryption process in order to bring even more security (ignored if not filled). In this case, any value encrypted with a given nonce can only be decrypted with this nonce.

© 2026 Microchip Technology Inc. and its subsidiaries

If another nonce is set for the AESB_IVRx.IV, any value encrypted with the previous nonce can no longer be decrypted (see AESB Initialization Vector Register x).

Dual buffer usage (AESB_MR.DUALBUFF='1') is recommended for improved performance.

40.4.7. Security Features

40.4.7.1. Private Key Bus

The AESB features a private key internal register that can be accessed only through the dedicated private key bus from the TRNG.

The private key internal register cannot be read from any peripheral or from software.

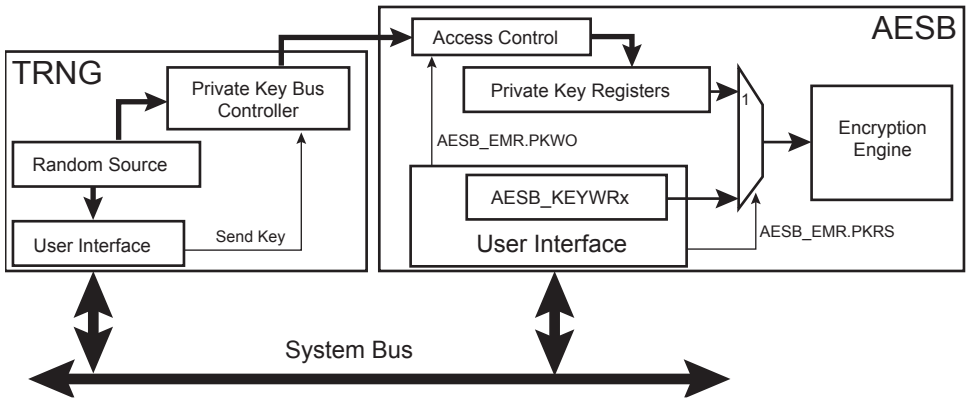
The AESB key used by the encryption/decryption engine is either the private key internal register content or the AESB_KEYWRx registers content.

By default, after hardware reset, the AESB key is provided by the AESB_KEYWRx registers. The software can select the private key internal register by setting AESB_EMR.PKRS. The key stored in the AESB_KEYWRx registers remain available for later use by clearing AESB_EMR.PKRS.

Before selecting the private key internal register, the software must:

1. Trigger the key transfer over the private key bus from the TRNG Key Bus host.
2. Wait for completion of the transfer signaled in the TRNG Status register.
3. Check for any access violation in AESB_WPSR.PKRPVS.

Figure 40-3. Key Selection



40.4.7.2. Unspecified Register Access Detection

When an unspecified register access occurs, AESB_ISR.URAD rises. Its source is then reported in AESB_ISR.URAT. Only the last unspecified register access is available through the URAT field.

Several kinds of unspecified register accesses can occur:

- Input Data register written during the data processing when SMOD = IDATAR0_START
- Output Data register read during data processing
- Mode register written during data processing
- Output Data register read during sub-keys generation
- Mode register written during sub-keys generation
- Write-only register read access

URAD and URAT can only be reset by AESB_CR.SWRST.

© 2026 Microchip Technology Inc. and its subsidiaries

40.4.7.3. Clearing Key on Tamper Event

On a tamper detection event, an immediate clear of the scrambling key (AESB_KEYWRx) can be performed if the bit AESB_MR.TAMPCLR=1.

40.4.7.4. Register Write Protection

To prevent any single software error from corrupting AESB behavior, certain registers in the address space can be write-protected by setting WPEN (Write Protection Enable), WPITEN (Write Protection Interrupt Enable), and/or WPCREN (Write Protection Control Enable) in the AESB Write Protection Mode register (AESB_WPMR).

If a write access to a write-protected register is detected, the Write Protection Violation Status (WPVS) flag in the AESB Write Protection Status register (AESB_WPSR) is set and the Write Protection Violation Source (WPVSRC) field indicates the register in which the write access has been attempted.

WPVS is automatically cleared after reading AESB_WPSR.

The following registers can be write-protected when WPEN is set in AESB_WPMR:

- AESB Mode Register
- AESB Key Word Register x
- AESB Initialization Vector Register x
- AESB Extended Mode Register

The following registers can be write-protected when WPITEN is set:

- AESB Interrupt Enable Register
- AESB Interrupt Disable Register

The following register can be write-protected when WPCREN is set:

- AESB Control Register

40.4.7.5. Security and Safety Analysis and Reports

Several types of checks are performed when the AESB is enabled.

The peripheral clock of the AESB is monitored by specific circuitry to detect abnormal waveforms on the internal clock net that may affect the behavior of the AESB. Corruption on the triggering edge of the clock or a pulse with a minimum duration may be identified. If the flag AESB_WPSR.CGD is set, an abnormal condition occurred on the peripheral clock. This flag is not set under normal operating conditions.

The internal sequencer of the AESB is also monitored and if an abnormal state is detected, the flag AESB_WPSR.SEQE is set. This flag is not set under normal operating conditions.

The software accesses to the AESB are monitored and if an incorrect access is performed, the flag AESB_WPSR.SWE is set. The type of incorrect/abnormal software access is reported in AESB_WPSR.SWETYP (see AESB Write Protection Status Register for details). e.g., writing AESB_ODATARx is an error, as well as reading AESB_IDATARx, when the AESB_ISR.DATRDY flag is cleared. AESB_WPSR.ECLASS is an indicator reporting the criticality of the SWETYP report.

The flags CGD, SEQE, SWE and WPVS are automatically cleared when AESB_WPSR is read.

If one of these flags is set, the flag AESB_ISR.SECE is set and can trigger an interrupt if AESB_IMR.SECE is '1'. SECE is cleared by reading AESB_ISR.

It is possible to configure an action to be performed by AESB as soon as an abnormal event detection occurs. If AESB_WPMR.ACTION > 0, either a lock is performed or a lock and immediate clear of the AESB_KEYWRx key. If a lock is performed, the current processing is ended normally but any new processing is not performed whatever the start mode of operation (see AESB_MR.SMOD).

A locked state of the AESB is unlocked as follows:

© 2026 Microchip Technology Inc. and its subsidiaries

1. Read AESB_WPSR.
2. Disable the source of tamper if the tamper is enabled to perform a clear of the key.
3. Write a ‘1’ to AESB_CR.UNLOCK.

It is possible to select the type of event that will lock the AESB in case of abnormal event detection. See AESB_WPMR.ACTION for details.

If AESB_MR.TMPCLR=1 and the tamper pin is active, the AESB is locked.

© 2026 Microchip Technology Inc. and its subsidiaries

40.5. Register Summary

This is the start of your topic.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	AESB_CR	31:24								UNLOCK
		23:16								
		15:8								SWRST
		7:0								START
0x00	AESB_IDATAR0	31:24	IDATA[31:24]							
		23:16	IDATA[23:16]							
		15:8	IDATA[15:8]							
		7:0	IDATA[7:0]							
0x00	AESB_ODATAR0	31:24	ODATA[31:24]							
		23:16	ODATA[23:16]							
		15:8	ODATA[15:8]							
		7:0	ODATA[7:0]							
0x00	AESB_IVR0	31:24	IV[31:24]							
		23:16	IV[23:16]							
		15:8	IV[15:8]							
		7:0	IV[7:0]							
0x04	AESB_MR	31:24	TAMPCLR							
		23:16	CKEY[3:0]							
		15:8	LOD	OPMOD[2:0]					SMOD[1:0]	
		7:0	PROCDLY[3:0]				DUALBUFF	AAHB		CIPHER
0x04	AESB_IDATAR1	31:24	IDATA[31:24]							
		23:16	IDATA[23:16]							
		15:8	IDATA[15:8]							
		7:0	IDATA[7:0]							
0x04	AESB_ODATAR1	31:24	ODATA[31:24]							
		23:16	ODATA[23:16]							
		15:8	ODATA[15:8]							
		7:0	ODATA[7:0]							
0x04	AESB_IVR1	31:24	IV[31:24]							
		23:16	IV[23:16]							
		15:8	IV[15:8]							
		7:0	IV[7:0]							
0x08	AESB_IDATAR2	31:24	IDATA[31:24]							
		23:16	IDATA[23:16]							
		15:8	IDATA[15:8]							
		7:0	IDATA[7:0]							
0x08	AESB_ODATAR2	31:24	ODATA[31:24]							
		23:16	ODATA[23:16]							
		15:8	ODATA[15:8]							
		7:0	ODATA[7:0]							
0x08	AESB_IVR2	31:24	IV[31:24]							
		23:16	IV[23:16]							
		15:8	IV[15:8]							
		7:0	IV[7:0]							
0x0C	AESB_IDATAR3	31:24	IDATA[31:24]							
		23:16	IDATA[23:16]							
		15:8	IDATA[15:8]							
		7:0	IDATA[7:0]							
0x0C	AESB_ODATAR3	31:24	ODATA[31:24]							
		23:16	ODATA[23:16]							
		15:8	ODATA[15:8]							
		7:0	ODATA[7:0]							
0x0C	AESB_IVR3	31:24	IV[31:24]							
		23:16	IV[23:16]							
		15:8	IV[15:8]							
		7:0	IV[7:0]							

© 2026 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x10	AESB_IER	31:24								
		23:16					SECE			
		15:8								URAD
		7:0								DATRDY
0x14	AESB_IDR	31:24								
		23:16					SECE			
		15:8								URAD
		7:0								DATRDY
0x18	AESB_IMR	31:24								
		23:16					SECE			
		15:8								URAD
		7:0								DATRDY
0x1C	AESB_ISR	31:24								
		23:16					SECE			
		15:8	URAT[3:0]							URAD
		7:0								DATRDY
0x20	AESB_KEYWR0	31:24	KEYW[31:24]							
		23:16	KEYW[23:16]							
		15:8	KEYW[15:8]							
		7:0	KEYW[7:0]							
0x24	AESB_KEYWR1	31:24	KEYW[31:24]							
		23:16	KEYW[23:16]							
		15:8	KEYW[15:8]							
		7:0	KEYW[7:0]							
0x28	AESB_KEYWR2	31:24	KEYW[31:24]							
		23:16	KEYW[23:16]							
		15:8	KEYW[15:8]							
		7:0	KEYW[7:0]							
0x2C	AESB_KEYWR3	31:24	KEYW[31:24]							
		23:16	KEYW[23:16]							
		15:8	KEYW[15:8]							
		7:0	KEYW[7:0]							
0x30 ... 0xAF	Reserved									
0xB0	AESB_EMR	31:24								
		23:16								
		15:8								
		7:0	PKRS	PKWO						
0xB4 ... 0xE3	Reserved									
0xE4	AESB_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0	ACTION[2:0]			FIRSTE		WPCREN	WPITEN	WPEN
0xE8	AESB_WPSR	31:24	ECLASS				SWETYP[3:0]			
		23:16								
		15:8	WPVSRC[7:0]							
		7:0				PKRPVS	SWE	SEQE	CGD	WPVS

© 2026 Microchip Technology Inc. and its subsidiaries

40.5.1. AESB Control Register

Name: AESB_CR
Offset: 0x00
Reset: –
Property: Write-only

This register can only be written if the WPCREN bit is cleared in the AESB Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
								UNLOCK
Access								W
Reset								–

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
								SWRST
Access								W
Reset								–

Bit	7	6	5	4	3	2	1	0
								START
Access								W
Reset								–

Bit 24 – UNLOCK Unlock Processing
AESB_WPSR must be cleared before performing the unlock command.

Value	Description
0	No effect.
1	Unlocks the processing in case of abnormal event detection if AESB_WPMR.ACTION > 0.

Bit 8 – SWRST Software Reset

Value	Description
0	No effect
1	Resets the AESB. A software triggered hardware reset of the AESB interface is performed.

Bit 0 – START Start Processing

Value	Description
0	No effect
1	Starts manual encryption/decryption process

© 2026 Microchip Technology Inc. and its subsidiaries

40.5.2. AESB Mode Register

Name: AESB_MR
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the AESB Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	TAMPCLR							
Access	R/W							
Reset	0							

Bit	23	22	21	20	19	18	17	16
	CKEY[3:0]							
Access	W	W	W	W				
Reset	–	–	–	–				

Bit	15	14	13	12	11	10	9	8
	LOD	OPMOD[2:0]					SMOD[1:0]	
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	0	0	0			0	0

Bit	7	6	5	4	3	2	1	0
	PROCDLY[3:0]				DUALBUFF	AAHB		CIPHER
Access	R/W	R/W	R/W	R/W	R/W	R/W		R/W
Reset	0	0	0	0	0	0		0

Bit 31 – TAMPCLR Tamper Clear Enable

Value	Description
0	A tamper detection event has no effect on the AESB_KEYWRx key.
1	A tamper detection event immediately clears the AESB_KEYWRx key.

Bits 23:20 – CKEY[3:0] Key

Value	Name	Description
0xE	PASSWD	Must be written with 0xE the first time that AESB_MR is programmed. For subsequent programming of the AESB_MR, any value can be written, including that of 0xE. Always reads as 0.

Bit 15 – LOD Last Output Data Mode

Value	Description
0	No effect. After each end of encryption/decryption, the output data will be available either on the output data registers (Manual and Auto modes). In Manual and Auto modes, AESB_ISR.DATRDY is cleared when at least one of the Output Data registers is read.
1	AESB_ISR.DATRDY is cleared when at least one of the Input Data registers is written. No additional Output Data register reads are necessary between consecutive encryptions/decryptions (see Last Output Data Mode).

© 2026 Microchip Technology Inc. and its subsidiaries

Bits 14:12 – OPMOD[2:0] Operating Mode

Values which are not listed in the table must be considered as “reserved”.
For CBC-MAC operating mode, configure OPMOD to 0x1 (CBC) and set LOD to 1.
If OPMOD is set to 4 and AAHB = 1, there is no compliance with the standard CTR mode of operation.

Value	Name	Description
0	ECB	Electronic Code Book mode
1	CBC	Cipher Block Chaining mode
2	–	Reserved
3	–	Reserved
4	CTR	Counter mode (16-bit internal counter)

Bits 9:8 – SMOD[1:0] Start Mode

Value	Name	Description
0	MANUAL_START	Manual mode
1	AUTO_START	Auto mode
2	IDATAR0_START	AESB_IDATAR0 access only Auto mode

Bits 7:4 – PROCDLY[3:0] Processing Delay

Processing Time = 12 × (PROCDLY + 1)
The Processing Time represents the number of clock cycles that the AESB needs in order to perform one encryption/decryption .
Note: The best performance is achieved with PROCDLY = 0.

Bit 3 – DUALBUFF Dual Input Buffer

Value	Name	Description
0	INACTIVE	AESB_IDATARx cannot be written during processing of previous block.
1	ACTIVE	AESB_IDATARx can be written during processing of previous block when SMOD = 2. It speeds up the overall runtime of large files.

Bit 2 – AAHB Automatic Bridge Mode

Value	Description
0	Automatic Bridge mode disabled.
1	Automatic Bridge mode enabled.

Bit 0 – CIPHER Processing Mode

Value	Description
0	Decrypts data.
1	Encrypts data.

© 2026 Microchip Technology Inc. and its subsidiaries

40.5.3. AESB Interrupt Enable Register

Name: AESB_IER
Offset: 0x10
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the AESB Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					SECE			
Access					W			
Reset					–			

Bit	15	14	13	12	11	10	9	8
								URAD
Access								W
Reset								–

Bit	7	6	5	4	3	2	1	0
								DATRDY
Access								W
Reset								–

Bit 19 – SECE Security and/or Safety Event

Bit 8 – URAD Unspecified Register Access Detection Interrupt Enable

Bit 0 – DATRDY Data Ready Interrupt Enable

© 2026 Microchip Technology Inc. and its subsidiaries

40.5.4. AESB Interrupt Disable Register

Name: AESB_IDR
Offset: 0x14
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the AESB Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					SECE			
Access					W			
Reset					–			

Bit	15	14	13	12	11	10	9	8
								URAD
Access								W
Reset								–

Bit	7	6	5	4	3	2	1	0
								DATRDY
Access								W
Reset								–

Bit 19 – SECE Security and/or Safety Event

Bit 8 – URAD Unspecified Register Access Detection Interrupt Disable

Bit 0 – DATRDY Data Ready Interrupt Disable

© 2026 Microchip Technology Inc. and its subsidiaries

40.5.5. AESB Interrupt Mask Register

Name: AESB_IMR
Offset: 0x18
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					SECE			
Access					R			
Reset					0			

Bit	15	14	13	12	11	10	9	8
								URAD
Access								R
Reset								0

Bit	7	6	5	4	3	2	1	0
								DATRDY
Access								R
Reset								0

Bit 19 – SECE Security and/or Safety Event

Bit 8 – URAD Unspecified Register Access Detection Interrupt Mask

Bit 0 – DATRDY Data Ready Interrupt Mask

© 2026 Microchip Technology Inc. and its subsidiaries

40.5.6. AESB Interrupt Status Register

Name: AESB_ISR
Offset: 0x1C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					SECE			
Access					R			
Reset					0			

Bit	15	14	13	12	11	10	9	8
	URAT[3:0]							URAD
Access	R	R	R	R				R
Reset	0	0	0	0				0

Bit	7	6	5	4	3	2	1	0
								DATRDY
Access								R
Reset								0

Bit 19 – SECE Security and/or Safety Event

Value	Description
0	There is no security report in AESB_WPSR.
1	One security flag is set in AESB_WPSR.

Bits 15:12 – URAT[3:0] Unspecified Register Access
Only the last Unspecified Register Access Type is available through URAT.
URAT field is reset only by AESB_CR.SWRST.

Value	Name	Description
0x0	IDR_WR_PROCESSING	Input Data register written during the data processing when SMOD = 0x2 mode
0x1	ODR_RD_PROCESSING	Output Data register read during the data processing
0x2	MR_WR_PROCESSING	Mode register written during the data processing
0x3	ODR_RD_SUBKGEN	Output Data register read during the sub-keys generation
0x4	MR_WR_SUBKGEN	Mode register written during the sub-keys generation
0x5	WOR_RD_ACCESS	Write-only register read access

Bit 8 – URAD Unspecified Register Access Detection Status
URAD is reset only by AESB_CR.SWRST.

Value	Description
0	No unspecified register access has been detected since the last SWRST.
1	At least one unspecified register access has been detected since the last SWRST.

Bit 0 – DATRDY Data Ready
DATRDY is cleared when a Manual encryption/decryption occurs (AESB_CR.START) or when a software triggered hardware reset of the AESB interface is performed (AESB_CR.SWRST).

© 2026 Microchip Technology Inc. and its subsidiaries

AESB_MR.LOD = 0: In Manual and Auto modes, DATRDY can also be cleared when at least one of the Output Data registers is read.
AESB_MR.LOD = 1: In Manual and Auto modes, DATRDY can also be cleared when at least one of the Input Data registers is written.

Value	Description
0	Output data not valid.
1	Encryption or decryption process is completed.

© 2026 Microchip Technology Inc. and its subsidiaries

40.5.7. AESB Key Word Register x

Name: AESB_KEYWRx
Offset: 0x20 + x*0x04 [x=0..3]
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the AESB Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	KEYW[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	KEYW[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	KEYW[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	KEYW[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – KEYW[31:0] Key Word
The four 32-bit Key Word registers set the 128-bit cryptographic key used for encryption/decryption. AESB_KEYWR0 corresponds to the first word of the key, AESB_KEYWR3 to the last one.
These registers are write-only to prevent the key from being read by another application.
Note: AESB_KEYWRx registers are not used if the private key internal register is selected instead by writing a 1 to AESB_EMR.PKRS.

© 2026 Microchip Technology Inc. and its subsidiaries

40.5.8. AESB Input Data Register x

Name: AESB_IDATARx
Offset: 0x00 + x*0x04 [x=0..3]
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	IDATA[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	IDATA[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	IDATA[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	IDATA[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – IDATA[31:0] Input Data Word
The four 32-bit Input Data registers set the 128-bit data block used for encryption/decryption. AESB_IDATAR0 corresponds to the first word of the data to be encrypted/decrypted, AESB_IDATAR3 to the last one.
These registers are write-only to prevent the input data from being read by another application.

© 2026 Microchip Technology Inc. and its subsidiaries

40.5.9. AESB Output Data Register x

Name: AESB_ODATARx
Offset: 0x00 + x*0x04 [x=0..3]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	ODATA[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	–	–	–	–	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ODATA[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ODATA[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ODATA[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ODATA[31:0] Output Data
The four 32-bit Output Data registers contain the 128-bit data block that has been encrypted/decrypted.
AESB_ODATAR0 corresponds to the first word, AESB_ODATAR3 to the last one.

© 2026 Microchip Technology Inc. and its subsidiaries

40.5.10. AESB Initialization Vector Register x

Name: AESB_IVRx
Offset: 0x00 + x*0x04 [x=0..3]
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the AESB Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	IV[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	IV[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	IV[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	IV[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – IV[31:0] Initialization Vector
The four 32-bit Initialization Vector registers set the 128-bit Initialization Vector data block that is used by some modes of operation as an additional initial input.
AESB_IVR0 corresponds to the first word of the Initialization Vector, AESB_IVR3 to the last one.
These registers are write-only to prevent the Initialization Vector from being read by another application.
For CBC mode, the IV input value corresponds to the initialization vector.
For CTR mode, the IV input value corresponds to the initial counter value.
Note: These registers are not used in ECB mode and must not be written. For Automatic Bridge dedicated mode, the IV input value corresponds to the initial nonce.

© 2026 Microchip Technology Inc. and its subsidiaries

40.5.11. AESB Extended Mode Register

Name: AESB_EMR
Offset: 0xB0
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPEN is cleared in the AESB Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	PKRS	PKWO						
Access	R/W	R/W						
Reset	–	–						

Bit 7 – PKRS Private Key Internal Register Select

Value	Description
0	The key used by the AESB is in the AESB_KEYWRx registers.
1	The key used by the AESB is in the private key internal register written through the private key bus.

Bit 6 – PKWO Private Key Write Once

Value	Description
0	The private key internal register can be written multiple times through the private key bus.
1	The private key internal register can be written only once through the private key bus until hardware reset.

© 2026 Microchip Technology Inc. and its subsidiaries

40.5.12. AESB Write Protection Mode Register

Name: AESB_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ACTION[2:0]			FIRSTE		WPCREN	WPITEN	WPEN
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x414553	PASSWD	Writing any other value in this field aborts the write operation of the WPEN,WPITEN,WPCREN bits. Always reads as 0.

Bits 7:5 – ACTION[2:0] Action on Abnormal Event Detection

Value	Name	Description
0	REPORT_ONLY	No action (stop or clear key) is performed when one of PKRPVS, WPVS, CGD, SEQE, or SWE flags is set.
1	LOCK_PKRPVS_WPVS_SWE	If a processing is in progress when the AESB_WPSR.PKRPVS/WPVS/SWE event detection occurs, the current processing is ended normally but no other processing is started while a AESB_CR.UNLOCK command is issued.
2	LOCK_CGD_SEQE	If a processing is in progress when the AESB_WPSR.CGD/SEQE event detection occurs, the current processing is ended normally but no other processing is started while a AESB_CR.UNLOCK command is issued.
3	LOCK_ANY_EV	If a processing is in progress when the AESB_WPSR.PKRPVS/WPVS/CGD/SEQE/SWE events detection occurs, the current processing is ended normally but no other processing is started while a AESB_CR.UNLOCK command is issued.
4	CLEAR_PKRPVS_WPVS_SWE	If a processing is in progress when the AESB_WPSR.PKRPVS/WPVS/SWE events detection occurs, the current processing is ended normally but no other processing is started while a AESB_CR.UNLOCK command is issued. Moreover, the AESB_KEYWRx key is immediately cleared.
5	CLEAR_CGD_SEQE	If a processing is in progress when the AESB_WPSR.CGD/SEQE events detection occurs, the current processing is ended normally but no other processing is started while a AESB_CR.UNLOCK command is issued. Moreover, the AESB_KEYWRx key is immediately cleared.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Name	Description
6	CLEAR_ANY_EV	If a processing is in progress when the AESB_WPSR.PKRPVS/WPVS/CGD/SEQE/SWE events detection occurs, the current processing is ended normally but no other processing is started while a AESB_CR.UNLOCK command is issued. Moreover, the AESB_KEYWRx key is immediately cleared.

Bit 4 – FIRSTE First Error Report Enable

Value	Description
0	The last write protection violation source is reported in AESB_WPSR.WPVSRC and the last software control error type is reported in AESB_WPSR.SWETYP. The AESB_ISR.SECE flag is set at the first error occurrence within a series.
1	Only the first write protection violation source is reported in AESB_WPSR.WPVSRC and only the first software control error type is reported in AESB_WPSR.SWETYP. The AESB_ISR.SECE flag is set at the first error occurrence within a series.

Bit 2 – WPCREN Write Protection Control Enable

Value	Description
0	Disables the write protection on control register if WPKEY corresponds to 0x414553 (“AES” in ASCII).
1	Enables the write protection on control register if WPKEY corresponds to 0x414553 (“AES” in ASCII).

Bit 1 – WPITEN Write Protection Interruption Enable

Value	Description
0	Disables the write protection on interrupt registers if WPKEY corresponds to 0x414553 (“AES” in ASCII).
1	Enables the write protection on interrupt registers if WPKEY corresponds to 0x414553 (“AES” in ASCII).

Bit 0 – WPEN Write Protection Configuration Enable
See Register Write Protection for the list of registers that can be write-protected.

Value	Description
0	Disables the write protection on configuration registers if WPKEY corresponds to 0x414553 (“AES” in ASCII).
1	Enables the write protection on configuration registers if WPKEY corresponds to 0x414553 (“AES” in ASCII).

© 2026 Microchip Technology Inc. and its subsidiaries

40.5.13. AESB Write Protection Status Register

Name: AESB_WPSR
Offset: 0xE8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	ECLASS				SWETYP[3:0]			
Access	R				R	R	R	R
Reset	0				0	0	0	0

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
				PKRPVS	SWE	SEQE	CGD	WPVS
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit 31 – ECLASS Software Error Class (cleared on read)
0 (WARNING): An abnormal access that does not affect system functionality
1 (ERROR): An access is performed into key, input data, control registers while the AESB is performing an encryption/decryption or a start is request by software or DMA while the key is not fully configured.

Bits 27:24 – SWETYP[3:0] Software Error Type (cleared on read)

Value	Name	Description
0	READ_WO	A write-only register has been read (Warning).
1	WRITE_RO	AESB is enabled and a write access has been performed on a read-only register (Warning).
2	UNDEF_RW	Access to an undefined address (Warning).
3	CTRL_START	Abnormal use of AESB_CR.START command when DMA access is configured.
4	WEIRD_ACTION	A private key bus access, key write, init value write, output data read, AESB_MR and AESB_EMR write has been performed while a current processing is in progress (abnormal).
5	INCOMPLETE_KEY	A tentative of start is required while the key is not fully loaded into the AESB_KEYWRx registers.

Bits 15:8 – WPVSRC[7:0] Write Protection Violation Source
When WPVS=1, WPVSRC indicates the register address offset at which a write access has been attempted.
When WPVS=0 and SWE=1, WPVSRC reports the address of the incorrect software access. As soon as WPVS=1, WPVSRC returns the address of the write-protected violation.

Bit 4 – PKRPVS Private Key Internal Register Protection Violation Status (cleared on read)

Value	Description
0	No private key internal register access violation has occurred since the last read of AESB_WPSR.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
1	A private key internal register access violation has occurred since the last read of AESB_WPSR.

Bit 3 – SWE Software Control Error (cleared on read)

Value	Description
0	No software error has occurred since the last read of AESB_WPSR.
1	A software error has occurred since the last read of AESB_WPSR. The field SWETYP details the type of software error; the associated incorrect software access is reported in the field WPVSRC (if WPVS=0).

Bit 2 – SEQE Internal Sequencer Error (cleared on read)

Value	Description
0	No peripheral internal sequencer error has occurred since the last read of AESB_WPSR.
1	A peripheral internal sequencer error has occurred since the last read of AESB_WPSR. This flag can only be set under abnormal operating conditions.

Bit 1 – CGD Clock Glitch Detected (cleared on read)

Value	Description
0	No clock glitch has occurred since the last read of AESB_WPSR. Under normal operating conditions, this bit is always cleared.
1	A clock glitch has occurred since the last read of AESB_WPSR. This flag can only be set in case of an abnormal clock signal waveform (glitch).

Bit 0 – WPVS Write Protection Violation Status (cleared on read)

Value	Description
0	No write protect violation has occurred since the last read of AESB_WPSR.
1	A write protect violation has occurred since the last read of AESB_WPSR. The address offset of the violated register is reported into field WPVSRC.

© 2026 Microchip Technology Inc. and its subsidiaries

41. Secure Hash Algorithm (SHA)

41.1. Description

The Secure Hash Algorithm (SHA) is compliant with the American *FIPS (Federal Information Processing Standard) Publication 180-4* specification.

The 512/1024-bit block of message is respectively stored in 16/32 x 32-bit registers, (SHA_IDATARx/SHA_IODATARx) which are write-only.

As soon as the input data is written, hash processing can be started. The registers comprising the block of a message must be entered consecutively. Then, after the processing period, the message digest is ready to be read out on the 5 up to 8/16 x 32-bit output data registers (SHA_IODATARx) or through the PDC channels.

The SHA supports the SHA512 derivative algorithms SHA512/224 and SHA512/256 which are based on the SHA512 algorithm with specific initial vectors and a truncated digest. Unless specified otherwise, features and functionality of the SHA512 algorithm also apply to the SHA512/224 and SHA512/256 algorithms.

The SHA supports the HMAC-SHA512 derivative algorithms HMAC-SHA512/224 and HMAC-SHA512/256 which are based on the HMAC-SHA512 algorithm with specific initial vectors and a truncated digest. Unless specified otherwise, features and functionality of the HMAC-SHA512 algorithm also apply to the HMAC-SHA512/224 and HMAC-SHA512/256 algorithms.

41.2. Embedded Characteristics

- Supports Secure Hash Algorithm (SHA1, SHA224, SHA256, SHA384, SHA512, SHA512/224, SHA512/256)
- Supports Hash-based Message Authentication Code (HMAC) Algorithm (HMAC-SHA1, HMAC-SHA224, HMAC-SHA256, HMAC-SHA384, HMAC-SHA512, HMAC-SHA512/224, HMAC-SHA512/256)
- Compliant with FIPS Publication 180-4
- Supports Automatic Padding of Messages
- Supports Up to 2 Sets of Initial Hash Values Registers (HMAC Acceleration or other)
- Supports Automatic Check of the Hash (HMAC Acceleration or other)
- Tightly Coupled to AES for Protocol Layers Improved Performances
- Configurable Processing Period:
  - 85 clock cycles to obtain a fast SHA1 runtime, 88 clock cycles for SHA384, SHA512 or 209 Clock Cycles for Maximizing Bandwidth of Other Applications
  - 72 clock cycles to obtain a fast SHA224, SHA256 runtime or 194 clock cycles for maximizing bandwidth of other applications
- Connection to PDC Channel Capabilities Optimizes Data Transfers
- Double Input Buffer Optimizes Runtime
- Register Write Protection

41.3. Product Dependencies

41.3.1. Power Management

The SHA may be clocked through the Power Management Controller (PMC), so the programmer must first configure the PMC to enable the SHA clock.

© 2026 Microchip Technology Inc. and its subsidiaries

41.3.2. Interrupt Sources

The SHA interface has one interrupt line used for standard functions and one interrupt line used to trigger any safety or security event that may occur. Both lines are connected to the Interrupt Controller. The interrupt line for standard functions is driven by the Interrupt Mask register (SHA_IMR) and the Interrupt Status register (SHA_ISR), whereas the interrupt line for safety/security functions is driven by Write Protection Status register (SHA_WPSR) flags. If one of the flags SHA_WPSR.WPVS, SHA_WPSR.CGD, SHA_WPSR.SEQE or SHA_WPSR.SWE is set, the interrupt line is asserted. In order to handle interrupts, the Interrupt Controller must be programmed before configuring the SHA.

41.4. Functional Description

The Secure Hash Algorithm (SHA) module requires a padded message according to the FIPS 180 specification. This message can be provided with the padding to the SHA module, or the padding can be automatically computed by the SHA module if the size of the message is provided. The first block of the message must be indicated to the module by a specific command. The SHA module produces an N-bit message digest each time a block is written and processing period ends, where N is 160 for SHA1, 224 for SHA224, 256 for SHA256, 384 for SHA384, 512 for SHA512. The SHA module is also capable of computing a Hash-based Message Authentication Code (HMAC) algorithm.

41.4.1. SHA Algorithm

The SHA can process SHA1, SHA224, SHA256, SHA384, SHA512 by configuring the ALGO field in the Mode register (SHA_MR).

41.4.2. HMAC Algorithm

The HMAC algorithm is as follows:

$HMAC_K(m) = h((K_0 \oplus opad) || h((K_0 \oplus ipad) || m))$

where:

- h = SHA function
- K_0 = the key K after any necessary pre-processing to form a block size key
- m = message to authenticate
- || = concatenation operator
- ⊕ = XOR operator
- ipad = predefined constant (0x3636...3636)
- opad = predefined constant (0x5C5C...5C5C)

The SHA provides a fully optimized processing of the HMAC algorithm by executing the following operations:

- starting the SHA algorithm from any user predefined hash value, thus 'h(K_0 ⊕ ipad)' for first HMAC hash and 'h(K_0 ⊕ opad)' for second HMAC hash
- performing automatic padding
- routing automatically the first hash result 'h((K_0 ⊕ ipad) || m)' to the source of the second hash processing 'h((K_0 ⊕ opad) || (first hash result))' including the concatenation of the first hash result to 'K_0 ⊕ opad'.

To perform the HMAC operation, the ALGO field value must be greater than 7, the automatic padding feature must be enabled (MSGSIZE and BYTCNT fields differ from 0) and the SHA internal initial hash value registers 0 and 1 must be configured, respectively, with the hash results of input blocks "K_0 ⊕ ipad" and "K_0 ⊕ opad" (see Internal Registers for Initial Hash Value or Expected Hash Result).

The size of the message ('m') must be written in the MSGSIZE and BYTCNT fields.

© 2026 Microchip Technology Inc. and its subsidiaries

The FIRST bit in the SHA Control register (SHA_CR) should be set before writing the first block of the message.

The SHA can process HMAC-SHA1, HMAC-SHA224, HMAC-SHA256, HMAC-SHA384, HMAC-SHA512 by configuring the SHA_MR.ALGO field.

41.4.3. Processing Period

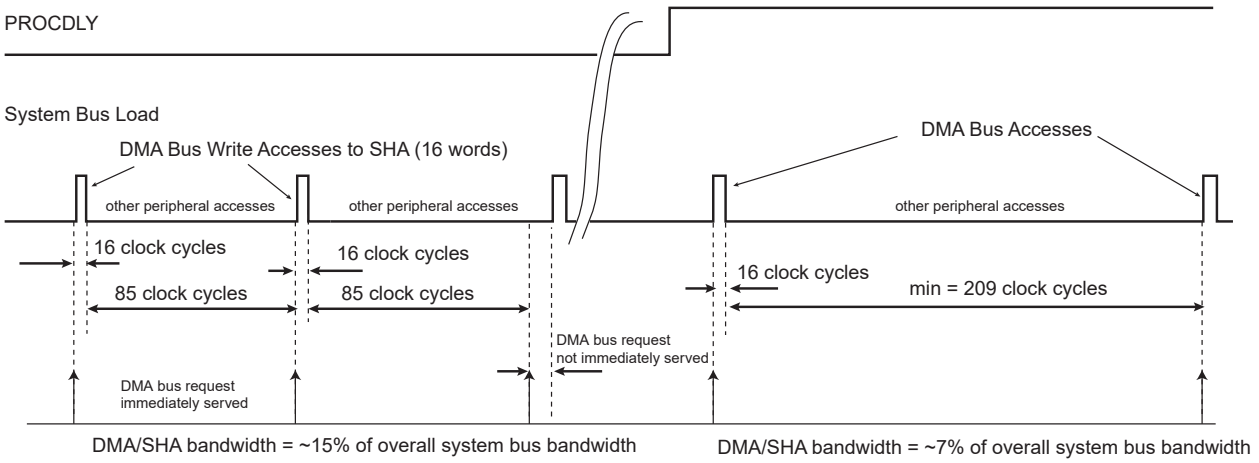
When SHA is enabled and PDC is used to write the messages, the inherent processing period may result, depending on the application, in a significant bandwidth usage at system bus level. In some applications, it may be important to keep as much bandwidth as possible for the other peripherals (e.g. CPU, other PDC channels). The SHA engine inherent processing period can be configured to reduce the bandwidth required by writing SHA_MR.PROCDLY=1.

In SHA1 mode, the shortest processing period is 85 clock cycles + 2 clock cycles for start command synchronization (SHA_MR.PROCDLY=0). The longest period is 209 clock cycles + 2 clock cycles when SHA_MR.PROCDLY=1 (see the figure below).

In SHA256 mode, the shortest processing period is 72 clock cycles + 2 clock cycles for start command synchronization (SHA_MR.PROCDLY=0). The longest period is 194 clock cycles + 2 clock cycles when SHA_MR.PROCDLY=1.

In SHA384 or SHA512 mode, the shortest processing period is 88 clock cycles + 2 clock cycles for start command synchronization. The longest period is 209 clock cycles + 2 clock cycles.

Figure 41-1. Bandwidth Usage in SHA-1 Mode



41.4.4. Double Input Buffer

The SHA Input Data registers (SHA_IDATARx) can be double-buffered to reduce the runtime of large messages.

Double-buffering allows a new message block to be written while the previous message block is being processed. This is only possible when DMA accesses are performed (SMOD = 2).

The DUALBUFF bit in the SHA_MR must be set to have double input buffer access.

41.4.5. Internal Registers for Initial Hash Value or Expected Hash Result

The SHA module embeds two sets of internal registers (IR0, IR1) to store different data used by the SHA or HMAC algorithms (see the figure User Initial Hash Value and Expected Hash Internal Register Access). These internal registers are accessed through SHA Input Data registers (SHA_IDATARx).

When the ALGO field selects SHA algorithms, IR0 can be configured with a user initial hash value. This initial hash value can be used to compute a custom hash algorithm with two sets of different

© 2026 Microchip Technology Inc. and its subsidiaries

initial constants, or to continue a hash computation by providing the intermediate hash value previously returned by the SHA module.

When the ALGO field selects SHA algorithms, IR1 can be configured with either a user initial hash value or an expected hash result. The expected hash result must be configured in the IR1 if the field CHECK = 1 (see Automatic Check). If the field CHECK = 0 or 2, IR1 can be configured with a user initial hash value that differs from IR0 value.

When the ALGO field selects HMAC algorithms, IR0 must be configured with the hash result of $K_0 \oplus$ ipad and IR1 must be configured with the hash result of $K_0 \oplus$ opad. These pre-computed first blocks speed up the HMAC computation by saving the time to compute the intermediate hash values of the first block which is constant while the secret key is constant (see HMAC Algorithm).

Table 41-1. Configuration Values of Internal Registers

Register	SHA Modes (ALGO < 8)			HMAC Modes (ALGO > 7)
	CHECK = 0	CHECK = 1	CHECK = 2	
IR0	User Initial Hash	User Initial Hash	User Initial Hash	hash($K_0 \oplus$ ipad)
IR1	User Initial Hash	Expected Hash Result	User Initial Hash	hash($K_0 \oplus$ opad)

To calculate the initial HMAC values, follow this sequence:

1. Calculate K_0.
2. Calculate $K_0 \oplus$ ipad and $K_0 \oplus$ opad.
3. Perform a hash of the result of $K_0 \oplus$ ipad and $K_0 \oplus$ opad (auto-padding must be disabled for that type of hash).
4. Write h($K_0 \oplus$ ipad) and h($K_0 \oplus$ opad) in IR0 and IR1 respectively.

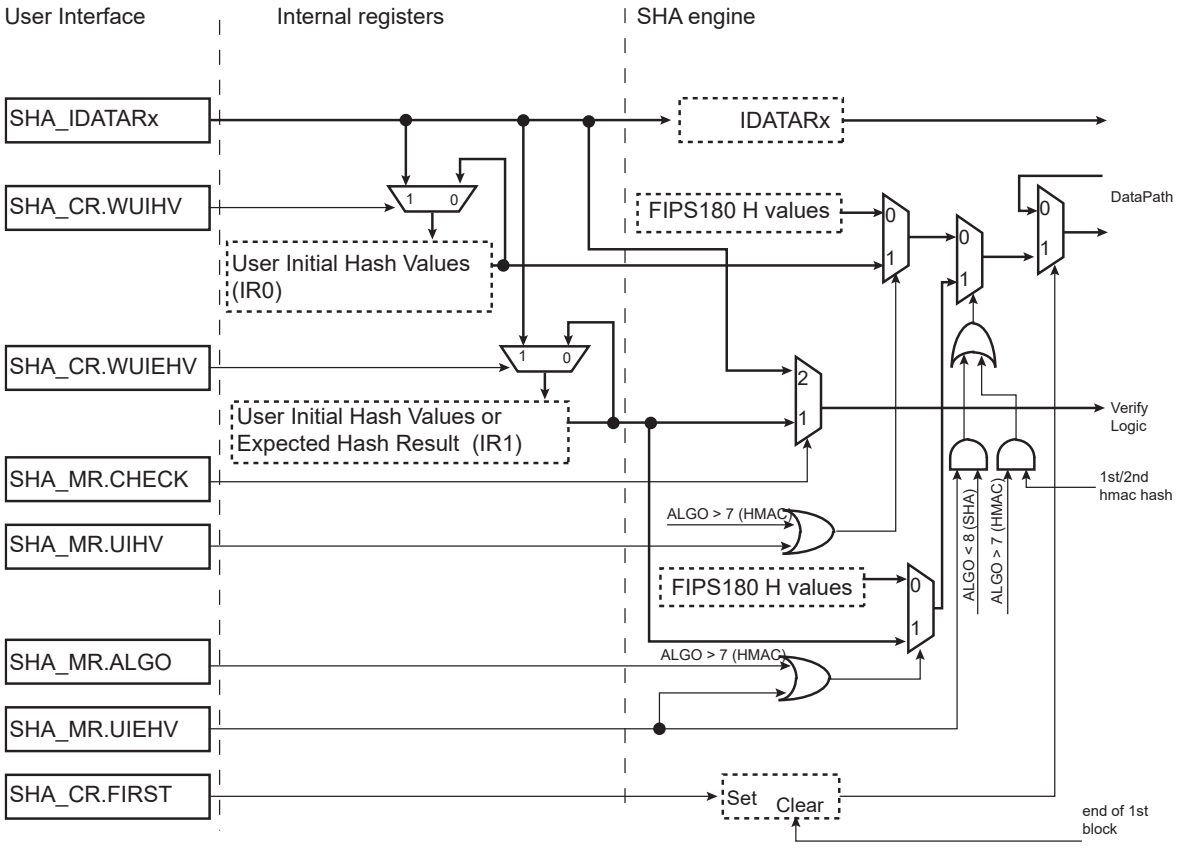
To write IR0 or IR1, follow this sequence:

1. Set SHA_CR. WUIHV (IR0) or SHA_CR.WUIEHV (IR1).
2. Write the data in SHA_IDATARx. The number of registers to write depends on the type of data (user initial hash values or expected hash result) and on the type of algorithm selected:
   - For user initial hash values:
     - SHA_IDATAR0 to SHA_IDATAR4 for SHA1
     - SHA_IDATAR0 to SHA_IDATAR7 for SHA224 or SHA256
     - SHA_IDATAR0 to SHA_IDATAR15 for SHA384, SHA512, SHA512/224, SHA512/256
   - For expected hash result:
     - SHA_IDATAR0 to SHA_IDATAR4 for SHA1
     - SHA_IDATAR0 to SHA_IDATAR6 for SHA224 or SHA512/224
     - SHA_IDATAR0 to SHA_IDATAR7 for SHA256 or SHA512/256
     - SHA_IDATAR0 to SHA_IDATAR11 for SHA384
     - SHA_IDATAR0 to SHA_IDATAR16 for SHA512
3. Clear SHA_CR.WUIHV or SHA_CR.WUIEHV.

IR0 and IR1 are automatically selected for HMAC processing if the field ALGO selects HMAC algorithms. If SHA algorithms are selected, the internal registers are selected if the corresponding UIHV or UIEHV bits are set.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 41-2. User Initial Hash Value and Expected Hash Internal Register Access



41.4.6. Automatic Padding

The SHA module features an automatic padding computation to speed up the execution of the algorithm.

The automatic padding function requires the following information:

- Complete message size in bytes to be written in the MSGSIZE field of the SHA Message Size register (SHA_MSR).
  The size of the message is written at the end of the last block, as required by the FIPS 180 specification (the size is automatically converted into a bit-size).
  Note: SHA_MSR is a 32-bit register, thus the automatic padding capability is limited to messages of less than 4 gigabytes. For messages greater than 4 gigabytes, padding must be performed by the software.
- Number of remaining bytes (to write in the SHA_IDATARx) to be written in the BYTCNT field of the SHA Bytes Count register (SHA_BCR).
  Automatic padding occurs when the BYTCNT field reaches 0. At each write in the SHA Input registers, the BYTCNT field value is decreased by the number of bytes written.

The BYTCNT field value must be written with the same value as the MSGSIZE field value if the full message is processed. If the message is partially preprocessed and an initial hash value is used, BYTCNT must be written with the remaining bytes to hash while MSGSIZE holds the message size.

To disable the automatic padding feature, the MSGSIZE and BYTCNT fields must be configured with 0.

© 2026 Microchip Technology Inc. and its subsidiaries

41.4.7. Automatic Check

The SHA module features an automatic check of the hash result with the expected hash. A check failure can generate an interrupt if configured in the SHA Interrupt Enable register (SHA_IER).

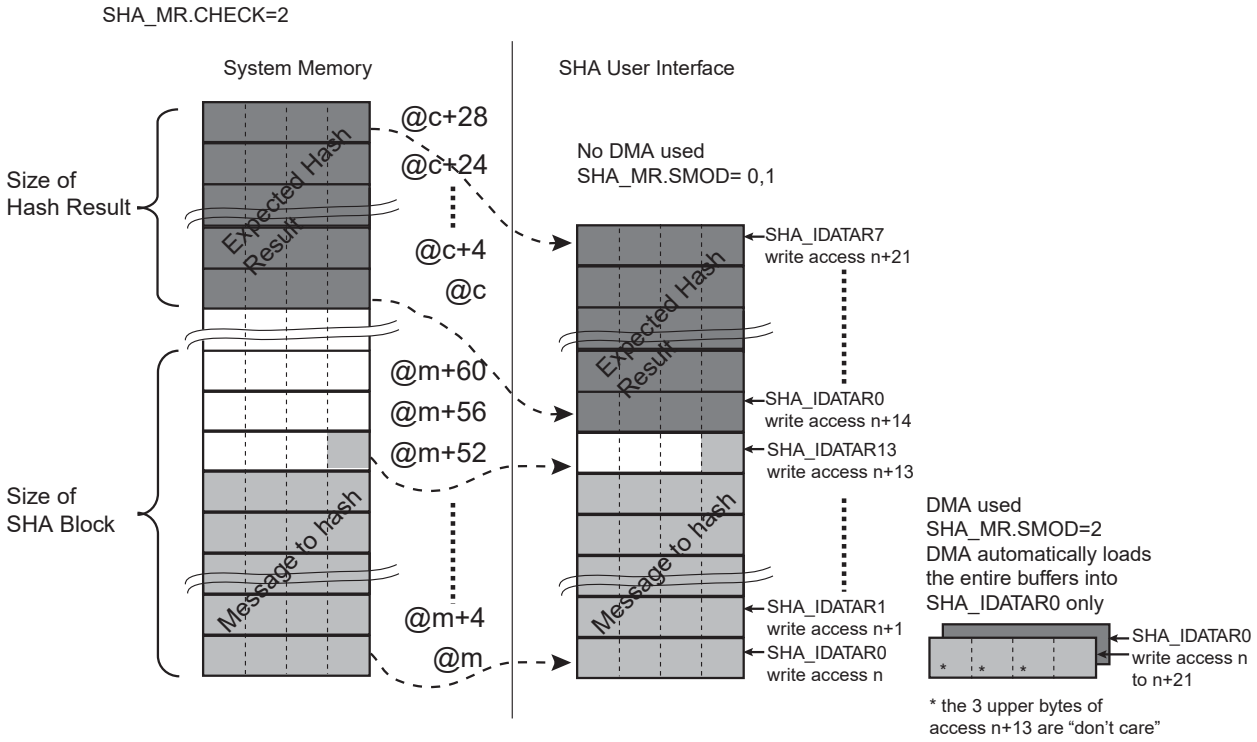
Automatic check requires the automatic padding feature to be enabled (MSGSIZE and BYTCNT fields must be greater than 0).

There are two methods to configure the expected hash result:

- if SHA_MR.CHECK = 1, the expected hash result is read from the internal register (IR1). This method cannot be used when HMAC algorithms is selected because this register is already used to store user initial hash values for the second hash processing. IR1 cannot be read by software.
- If SHA_MR.CHECK = 2, the expected hash result is written in the SHA_IDATARx after the message.

When SHA_MR.CHECK = 2, the method can provide more flexibility of use if a message is stored in system memory together with its expected hash result. A PDC can be used to ease the transfer of the message and its expected hash result.

Figure 41-3. Message and Expected Hash Result Memory Mapping



The number of 32-bit words of the hash result to check with the expected hash can be selected with SHA_MR.CHKCNT. The status of the check is available in the CHKST field in the SHA Interrupt Status register (SHA_ISR).

An interrupt can be generated (if enabled) when the check is completed. The check occurs several clock cycles after the computation of the requested hash, so the interrupt and the CHECKF bit are set several clock cycles after the DATRDY flag of the SHA_ISR.

41.4.8. Protocol Layers Improved Performances

The SHA can be tightly coupled to the AES module to improve performances when processing protocol layers such as IPsec or OpenSSL.

When the AES is configured to be tightly coupled to SHA (AES_MR), SHA must be always configured in Double Buffer mode (SHA_MR.DUALBUFF = 1).

© 2026 Microchip Technology Inc. and its subsidiaries

Refer to the section “Advanced Encryption Standard (AES)” for details.

41.4.9. Start Modes

SHA_MR.SMOD is used to select the Hash Processing Start mode.

41.4.9.1. Manual Mode

In Manual mode, the sequence is as follows:

1. Set SHA_IER.DATRDY (Data Ready) , depending on whether an interrupt is required at the end of processing.
2. If the initial hash values differ from the FIPS standard, set SHA_MR.UIHV and/or SHA_MR.UIEHV. If the initial hash values comply with the FIPS180 specification, clear SHA_MR.UIHV and/or SHA_MR.UIEHV.
3. If automatic padding is required, configure SHA_MSR.MSGSIZE with the number of bytes of the message, and configure SHA_BCR.BYTCNT with the remaining number of bytes to write. The BYTCNT field must be written with a value different from MSGSIZE field value if the message is preprocessed and completed by using user initial hash values.
   If automatic padding is not required, configure SHA_MSR.MSGSIZE and SHA_BCR.BYTCNT to 0.
4. The FIRST command must be set by writing a 1 into the corresponding bit of the Control register (SHA_CR) to start a hash computation with initial constants (first block of a message) or to resume after message processing was interrupted. When a first message processing is interrupted to process another message, the intermediate hash results must be stored in the system memory and they must be reloaded in user initial values registers (IR0 accessed via SHA_IDATAR when SHA_CR.WUIHV=1) prior to resume and continue the processing of the first message. For the other blocks, there is nothing to write.
5. Write the block to process in SHA_IDATARx.
6. To begin processing, set SHA_CR.START.
7. When processing is completed, the bit DATRDY in the Interrupt Status register (SHA_ISR) rises. If an interrupt has been enabled by setting SHA_IER.DATRDY, the interrupt line of the SHA is activated.
8. Repeat the write procedure for each block (step 5), start procedure (step 6) and wait for the interrupt procedure (step 7) up to the last block of the entire message. Each time the start procedure is complete, the DATRDY flag is cleared.
9. After the last block is processed (the DATRDY flag is set, if an interrupt was enabled by setting SHA_IER.DATRDY, the interrupt line of the SHA is activated), read the message digest in the Output Data registers. The DATRDY flag is automatically cleared when reading the SHA_IODATARx registers.

41.4.9.2. Auto Mode

In Auto mode, processing starts as soon as the correct number of SHA_IDATARx is written. No action is required in SHA_CR.

41.4.9.3. PDC Mode

The Peripheral Data Controller (PDC or Peripheral DMA Controller) can be used in association with the SHA to perform the algorithm on a complete message without any action by software during processing.

In this mode, the type of data transfer is set in words.

The sequence is as follows:

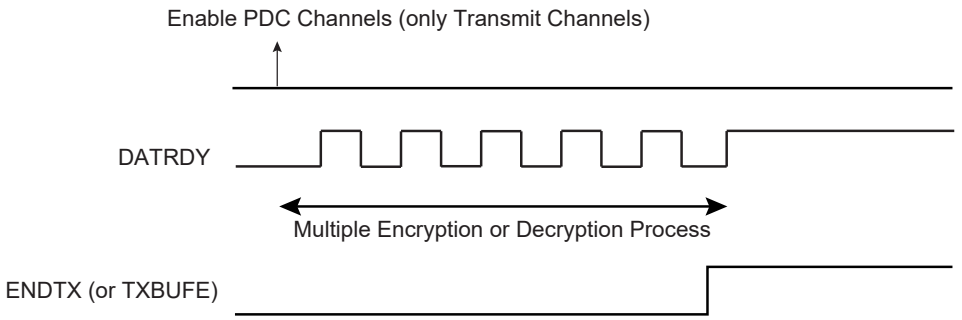
1. Set the Transmit Pointer register (SHA_TPR) to the address where the data buffer to process is stored.
2. Set the Transmit Counter registers (SHA_TCR) to the same value. This value must be a multiple of words.

© 2026 Microchip Technology Inc. and its subsidiaries

Note: The same requirements are necessary for the Next Pointer(s) and Counter(s) of the PDC (SHA_TNPR, SHA_TNCR).

3. If not already done, set SHA_IER.ENDTX (or TXBUFF if the next pointers and counters are used), depending on whether an interrupt is required at the end of processing.
4. If not already done, set SHA_IER.DATRDY.
5. Set the FIRST command by writing a '1' into the corresponding bit of the SHA_CR.
   Note: The FIRST bit command is also used to resume after message processing was interrupted. When a first message processing is interrupted to process another message, the intermediate hash results must be stored in the system memory and they must be reloaded in user initial values registers (IR0 accessed via SHA_IDATAR when SHA_CR.WUIHV=1) prior to resume and continue the processing of the first message. Thus, the PDC data buffers and SHA_CR.FIRST command must be managed accordingly.
6. Enable the PDC in transmission to start the processing (SHA_PTCR).
7. When the processing completes, ENDTX (or TXBUFF) in the SHA_ISR rises. If an interrupt has been enabled by setting the corresponding bit in SHA_IER, the interrupt line of the SHA is activated.
8. As soon as ENDTX (or TXBUFF) or interrupt is triggered, the DATRDY bit or interrupt line must be triggered.
9. The message digest can be read from the Output Data registers (SHA_IODATARx read-only registers).

Figure 41-4. Enable PDC Channels



41.4.9.4. SHA Register Endianness

In Arm processor-based products, the system bus and processors manipulate data in little-endian form. The SHA interface requires little-endian format words. However, in accordance with the protocol of the FIPS 180 specification, data is collected, processed and stored by the SHA algorithm in big-endian form.

The following example illustrates how to configure the SHA:

If the first 64 bits of a message (according to FIPS 180, i.e., big-endian format) to be processed is 0xcafedeca_01234567, then the SHA_IDATAR0 and SHA_IDATAR1 registers must be written with the following pattern:

- SHA_IDATAR0 = 0xcadefeca
- SHA_IDATAR1 = 0x67452301

In a little-endian system, the message (according to FIPS 180) starting with pattern 0xcafedeca_01234567 is stored into memory as follows:

- 0xca stored at initial offset (for example 0x00),
- then 0xfe stored at initial offset + 1 (i.e., 0x01),

© 2026 Microchip Technology Inc. and its subsidiaries

- 0xde stored at initial offset + 2 (i.e., 0x02),
- 0xca stored at initial offset + 3 (i.e., 0x03).

If the message is received through a serial-to-parallel communication channel, the first received character is 0xca and it is stored at the first memory location (initial offset). The second byte, 0xfe, is stored at initial offset + 1.

When reading on a 32-bit little-endian system bus, the first word read back from system memory is 0xcadefeca.

When the SHA_IODATARx registers are read, the hash result is organized in little-endian format, allowing system memory storage in the same format as the message.

Taking an example from the FIPS 180 specification Appendix B.1, the endian conversion can be observed.

For this example, the 512-bit message is:

0x61626380018

and the expected SHA-256 result is:

0xba7816bf_8f01cfea_414140de_5dae2223_b00361a3_96177a9c_b410ff61_f20015ad

If the message has not already been stored in the system memory, the first step is to convert the input message to little-endian before writing to the SHA_IDATARx registers. This would result in a write of:

SHA_IDATAR0 = 0x80636261..... SHA_IDATAR15 = 0x18000000

The data in the output message digest registers, SHA_IODATARx, contain SHA_IODATAR0 = 0xbf1678ba... SHA_IODATAR7 = 0xad1500f2 which is the little-endian format of 0xba7816bf,..., 0xf20015ad.

Reading SHA_IODATAR0 to SHA_IODATAR1 and storing into a little-endian memory system forces hash results to be stored in the same format as the message.

When the output message is read, the user can convert back to big-endian for a resulting message value of:

0xba7816bf_8f01cfea_414140de_5dae2223_b00361a3_96177a9c_b410ff61_f20015ad

41.4.10. Security Features

41.4.10.1. Unspecified Register Access Detection

When an unspecified register access occurs, the URAD bit in the SHA_ISR is set. Its source is then reported in the Unspecified Register Access Type field (URAT). Only the last unspecified register access is available through the URAT field.

Several kinds of unspecified register accesses can occur:

- SHA_IDATARx written during data processing in PDC mode
- SHA_IODATARx read during data processing
- SHA_MR written during data processing
- Write-only register read access

The URAD bit and the URAT field can only be reset by the SWRST bit in the SHA_CR.

41.4.10.2. Register Write Protection

To prevent any single software error from corrupting SHA behavior, certain registers in the address space can be write-protected by setting the WPEN (Write Protection Enable), WPITEN (Write

© 2026 Microchip Technology Inc. and its subsidiaries

Protection Interrupt Enable), and/or WPCREN (Write Protection Control Enable) bits in the SHA Write Protection Mode register (SHA_WPMR).

If a write access to a write-protected register is detected, the Write Protection Violation Status (WPVS) flag in the SHA Write Protection Status register (SHA_WPSR) is set and the Write Protection Violation Source (WPVSRC) field indicates the register in which the write access has been attempted.

The WPVS bit is automatically cleared after reading SHA_WPSR.

The following register(s) can be write-protected when SHA_WPMR.WPEN is set:

- SHA Mode Register
- SHA Message Size Register
- SHA Bytes Count Register

The following register(s) can be write-protected when WPITEN is set:

- SHA Interrupt Enable Register
- SHA Interrupt Disable Register

The following register(s) can be write-protected when WPCREN is set:

- SHA Control Register

41.4.10.3. Security and Safety Analysis and Reports

Several types of checks are performed when the SHA is enabled.

The peripheral clock of the SHA is monitored by a specific circuitry to detect abnormal waveforms on the internal clock net that may affect the behavior of the SHA. Corruption on the triggering edge of the clock or a pulse with a minimum duration may be identified. If the SHA_WPSR.CGD flag is set, an abnormal condition occurred on the peripheral clock. This flag is not set under normal operating conditions.

The internal sequencer of the SHA is also monitored, and if an abnormal state is detected, the SHA_WPSR.SEQE flag is set. This flag is not set under normal operating conditions.

Software accesses to the SHA are monitored and if an incorrect access is performed, the SHA_WPSR.SWE flag is set. The type of incorrect/abnormal software access is reported in the SHA_WPSR.SWETYP field (see SHA Write Protection Status Register for details), e.g., reading the SHA_ODATARx when the SHA_ISR.DATRDY flag is cleared is an error. SHA_WPSR.ECLASS is an indicator reporting the criticality of the SWETYP report.

The CGD, SEQE, SWE and WPVS flags are automatically cleared when SHA_WPSR is read.

If one of these flags is set, the SHA_ISR.SECE flag is set and can trigger an interrupt if SHA_IMR.SECE is ‘1’. SECE is cleared by reading SHA_ISR.

It is possible to configure an action to be performed by SHA as soon as an abnormal event detection occurs. If SHA_WPMR.ACTION > 0, a lock is performed. When a lock occurs, the current processing is ended normally but any new processing is not performed whatever the start mode of operation (see SHA_MR.SMOD).

A locked state of the SHA is unlocked as follows:

1. Read SHA_WPSR.
2. Disable the source of tamper if the tamper is enabled.
3. Write a ‘1’ to SHA_CR.UNLOCK.

It is possible to select the type of event that will lock the SHA in case of abnormal event detection. See SHA_WPMR.ACTION for details.

© 2026 Microchip Technology Inc. and its subsidiaries

If SHA_MR.TMPLCK=1 and the tamper pin is active, the SHA is locked whatever the value of the field SHA_WPMR.ACTION.

© 2026 Microchip Technology Inc. and its subsidiaries

41.5. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	SHA_CR	31:24								UNLOCK
		23:16								
		15:8			WUIEHV	WUIHV				SWRST
		7:0				FIRST				START
0x04	SHA_MR	31:24	CHKCNT[3:0]						CHECK[1:0]	
		23:16								DUALBUFF
		15:8	TMPLCK				ALGO[3:0]			
		7:0	BPE	UIEHV	UIHV	PROCDLY	AOE		SMOD[1:0]	
0x08 ... 0x0F	Reserved									
0x10	SHA_IER	31:24								SECE
		23:16								CHECKF
		15:8								URAD
		7:0						TXBUFE	ENDTX	DATRDY
0x14	SHA_IDR	31:24								SECE
		23:16								CHECKF
		15:8								URAD
		7:0						TXBUFE	ENDTX	DATRDY
0x18	SHA_IMR	31:24								SECE
		23:16								CHECKF
		15:8								URAD
		7:0						TXBUFE	ENDTX	DATRDY
0x1C	SHA_ISR	31:24								SECE
		23:16	CHKST[3:0]							CHECKF
		15:8		URAT[2:0]						URAD
		7:0				WRDY		TXBUFE	ENDTX	DATRDY
0x20	SHA_MSR	31:24	MSGSIZE[31:24]							
		23:16	MSGSIZE[23:16]							
		15:8	MSGSIZE[15:8]							
		7:0	MSGSIZE[7:0]							
0x24 ... 0x2F	Reserved									
0x30	SHA_BCR	31:24	BYTCNT[31:24]							
		23:16	BYTCNT[23:16]							
		15:8	BYTCNT[15:8]							
		7:0	BYTCNT[7:0]							
0x34 ... 0x3F	Reserved									
0x40	SHA_IDATAR0	31:24	IDATA[31:24]							
		23:16	IDATA[23:16]							
		15:8	IDATA[15:8]							
		7:0	IDATA[7:0]							
...										
0x7C	SHA_IDATAR15	31:24	IDATA[31:24]							
		23:16	IDATA[23:16]							
		15:8	IDATA[15:8]							
		7:0	IDATA[7:0]							
0x80	SHA_IODATAR0	31:24	IODATA[31:24]							
		23:16	IODATA[23:16]							
		15:8	IODATA[15:8]							
		7:0	IODATA[7:0]							
...										
0xBC	SHA_IODATAR15	31:24	IODATA[31:24]							
		23:16	IODATA[23:16]							
		15:8	IODATA[15:8]							
		7:0	IODATA[7:0]							

© 2026 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xC0 ... 0xE3	Reserved									
0xE4	SHA_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0		ACTION[1:0]		FIRSTE		WPCREN	WPITEN	WPEN
0xE8	SHA_WPSR	31:24	ECLASS				SWETYP[3:0]			
		23:16								
		15:8	WPVSRC[7:0]							
		7:0					SWE	SEQE	CGD	WPVS

© 2026 Microchip Technology Inc. and its subsidiaries

41.5.1. SHA Control Register

Name: SHA_CR
Offset: 0x00
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
								UNLOCK
Access								W
Reset								–

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			WUIEHV	WUIHV				SWRST
Access			W	W				W
Reset			–	–				–

Bit	7	6	5	4	3	2	1	0
				FIRST				START
Access				W				W
Reset				–				–

Bit 24 – UNLOCK Unlock Processing
SHA_WPSR must be cleared before performing the unlock command.

Value	Description
0	No effect.
1	Unlocks the processing in case of abnormal event detection if SHA_WPMR.ACTION > 0.

Bit 13 – WUIEHV Write User Initial or Expected Hash Values

Value	Description
0	SHA_IDATARx accesses are routed to the data registers.
1	SHA_IDATARx accesses are routed to the internal registers (IR1).

Bit 12 – WUIHV Write User Initial Hash Values

Value	Description
0	SHA_IDATARx accesses are routed to the data registers.
1	SHA_IDATARx accesses are routed to the internal registers (IR0).

Bit 8 – SWRST Software Reset

Value	Description
0	No effect.
1	Resets the SHA. A software-triggered hardware reset of the SHA interface is performed.

Bit 4 – FIRST First Block of a Message

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	No effect.
1	Indicates that the next block to process is the first one of a message or the first block of a fragment of a message.

Bit 0 – START Start Processing

Value	Description
0	No effect.
1	Starts manual hash algorithm process.

© 2026 Microchip Technology Inc. and its subsidiaries

41.5.2. SHA Mode Register

Name: SHA_MR
Offset: 0x04
Reset: 0x0000100
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	CHKCNT[3:0]						CHECK[1:0]	
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	0	0	0			0	0

Bit	23	22	21	20	19	18	17	16
								DUALBUFF
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
	TMPLCK				ALGO[3:0]			
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	1

Bit	7	6	5	4	3	2	1	0
	BPE	UIEHV	UIHV	PROCDLY	AOE		SMOD[1:0]	
Access	R/W	R/W	R/W	R/W	R/W		R/W	R/W
Reset	0	0	0	0	0		0	0

Bits 31:28 – CHKCNT[3:0] Check Counter
Number of 32-bit words to check. The value 0 indicates that the number of words to compare will be based on the algorithm selected (5 words for SHA1, 7 words for SHA224, 8 words for SHA256, 12 words for SHA384, 16 words for SHA512).

Bits 25:24 – CHECK[1:0] Hash Check
Values not listed in table must be considered as “reserved”.

Value	Name	Description
0	NO_CHECK	No check is performed.
1	CHECK_EHV	Check is performed with expected hash stored in internal expected hash value registers.
2	CHECK_MESSAGE	Check is performed with expected hash provided after the message.

Bit 16 – DUALBUFF Dual Input Buffer

Value	Name	Description
0	INACTIVE	SHA_IDATARx and SHA_IODATARx cannot be written during processing of previous block.
1	ACTIVE	SHA_IDATARx and SHA_IODATARx can be written during processing of previous block when SMOD value = 2. It speeds up the overall runtime of large files.

Bit 15 – TMPLCK Tamper Lock Enable

Value	Description
0	A tamper event has no effect.
1	A tamper event locks the SHA until the tamper root cause is cleared and SHA_CR.UNLOCK is written to 1.

Bits 11:8 – ALGO[3:0] SHA Algorithm
Values not listed in the table must be considered as “reserved”.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Name	Description
0	SHA1	SHA1 algorithm processed
1	SHA256	SHA256 algorithm processed
2	SHA384	SHA384 algorithm processed
3	SHA512	SHA512 algorithm processed
4	SHA224	SHA224 algorithm processed
5	SHA512_224	SHA512/224 algorithm processed
6	SHA512_256	SHA512/256 algorithm processed
8	HMAC_SHA1	HMAC algorithm with SHA1 Hash processed
9	HMAC_SHA256	HMAC algorithm with SHA256 Hash processed
10	HMAC_SHA384	HMAC algorithm with SHA384 Hash processed
11	HMAC_SHA512	HMAC algorithm with SHA512 Hash processed
12	HMAC_SHA224	HMAC algorithm with SHA224 Hash processed
13	HMAC_SHA512_224	HMAC algorithm with SHA512/224 Hash processed
14	HMAC_SHA512_256	HMAC algorithm with SHA512/256 Hash processed

Bit 7 – BPE Block Processing End

When SMOD=2 and ALGO<5, the SHA_ISR.DATRDY flag rises when each block has been processed. When SMOD=2 and ALGO>7, the SHA_ISR.DATRDY rises when all blocks except the last one have been processed.

Value	Description
0	BPE must be cleared when a DMA transfers data. When SMOD=2, SHA_ISR.DATRDY flag rises only when the SHA or HMAC processing cycle has completed. No intermediate block processing is reported.
1	When processing small messages, data transfer by software can improve performance compared to DMA. In this case, BPE can be written to 1, forcing the SHA_ISR.DATRDY to rise when a data must be loaded into SHA_IDATARx.

Bit 6 – UIEHV User Initial or Expected Hash Value Registers

Value	Description
0	The SHA algorithm is started with the standard initial values as defined in the FIPS 180 specification.
1	The SHA algorithm is started with the user initial hash values stored in the internal register 1 (IR1). If HMAC is configured, UIEHV has no effect (i.e. IR1 is always selected).

Bit 5 – UIHV User Initial Hash Values

Value	Description
0	The SHA algorithm is started with the standard initial values as defined in the FIPS 180 specification.
1	The SHA algorithm is started with the user initial hash values stored in the internal register 0 (IR0). If HMAC is configured, UIHV has no effect (i.e. IR0 is selected).

Bit 4 – PROCDLY Processing Delay

When SHA1 algorithm is processed, runtime period is either 85 or 209 clock cycles.
When SHA256 or SHA224 algorithm is processed, runtime period is either 72 or 194 clock cycles.
When SHA384 or SHA512 algorithm is processed, runtime period is either 88 or 209 clock cycles.

Value	Name	Description
0	SHORTEST	SHA processing runtime is the shortest one
1	LONGEST	SHA processing runtime is the longest one (reduces the SHA bandwidth requirement, reduces the system bus overload)

Bit 3 – AOE Always On Enable

Value	Description
0	The SHA operates in functional operating modes.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
1	As soon as a START command is written, the SHA processes dummy calculations until AOE=0, without software intervention. This can be used to create an additional current consumption when AES is used to encrypt/ decrypt.

Bits 1:0 – SMOD[1:0] Start Mode

Values not listed in the table must be considered as “reserved”.
If a PDC transfer is used, configure the SMOD value to 2.

Value	Name	Description
0	MANUAL_START	Manual mode
1	AUTO_START	Auto mode
2	IDATAR0_START	SHA_IDATAR0 access only mode (mandatory when DMA is used)

© 2026 Microchip Technology Inc. and its subsidiaries

41.5.3. SHA Interrupt Enable Register

Name: SHA_IER
Offset: 0x10
Reset: –
Property: Write-only

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
								SECE
Access								W
Reset								–

Bit	23	22	21	20	19	18	17	16
								CHECKF
Access								W
Reset								–

Bit	15	14	13	12	11	10	9	8
								URAD
Access								W
Reset								–

Bit	7	6	5	4	3	2	1	0
						TXBUFE	ENDTX	DATRDY
Access						W	W	W
Reset						–	–	–

Bit 24 – SECE Security and/or Safety Event Interrupt Enable

Bit 16 – CHECKF Check Done Interrupt Enable

Bit 8 – URAD Unspecified Register Access Detection Interrupt Enable

Bit 2 – TXBUFE Transmit Buffer Empty Interrupt Enable

Bit 1 – ENDTX End of Transmit Buffer Interrupt Enable

Bit 0 – DATRDY Data Ready Interrupt Enable

© 2026 Microchip Technology Inc. and its subsidiaries

41.5.4. SHA Interrupt Disable Register

Name: SHA_IDR
Offset: 0x14
Reset: –
Property: Write-only

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
								SECE
Access								W
Reset								–

Bit	23	22	21	20	19	18	17	16
								CHECKF
Access								W
Reset								–

Bit	15	14	13	12	11	10	9	8
								URAD
Access								W
Reset								–

Bit	7	6	5	4	3	2	1	0
						TXBUFE	ENDTX	DATRDY
Access						W	W	W
Reset						–	–	–

Bit 24 – SECE Security and/or Safety Event Interrupt Disable

Bit 16 – CHECKF Check Done Interrupt Disable

Bit 8 – URAD Unspecified Register Access Detection Interrupt Disable

Bit 2 – TXBUFE Transmit Buffer Empty Interrupt Disable

Bit 1 – ENDTX End of Transmit Buffer Interrupt Disable

Bit 0 – DATRDY Data Ready Interrupt Disable

© 2026 Microchip Technology Inc. and its subsidiaries

41.5.5. SHA Interrupt Mask Register

Name: SHA_IMR
Offset: 0x18
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
								SECE
Access								R
Reset								0

Bit	23	22	21	20	19	18	17	16
								CHECKF
Access								R
Reset								0

Bit	15	14	13	12	11	10	9	8
								URAD
Access								R
Reset								0

Bit	7	6	5	4	3	2	1	0
						TXBUFE	ENDTX	DATRDY
Access						R	R	R
Reset						0	0	0

Bit 24 – SECE Security and/or Safety Event Interrupt Mask

Bit 16 – CHECKF Check Done Interrupt Mask

Bit 8 – URAD Unspecified Register Access Detection Interrupt Mask

Bit 2 – TXBUFE Transmit Buffer Empty Interrupt Mask

Bit 1 – ENDTX End of Transmit Buffer Interrupt Mask

Bit 0 – DATRDY Data Ready Interrupt Mask

© 2026 Microchip Technology Inc. and its subsidiaries

41.5.6. SHA Interrupt Status Register

Name: SHA_ISR
Offset: 0x1C
Reset: 0x0000000A
Property: Read-only

Bit	31	30	29	28	27	26	25	24
								SECE
Access								R
Reset								0

Bit	23	22	21	20	19	18	17	16
	CHKST[3:0]							CHECKF
Access	R	R	R	R				R
Reset	0	0	0	0				0

Bit	15	14	13	12	11	10	9	8
		URAT[2:0]						URAD
Access		R	R	R				R
Reset		0	0	0				0

Bit	7	6	5	4	3	2	1	0
				WRDY		TXBUFE	ENDTX	DATRDY
Access				R		R	R	R
Reset				0		0	1	0

Bit 24 – SECE Security and/or Safety Event

Value	Description
0	There is no report in SHA_WPSR.
1	There is a Security and/or Safety Event reported in SHA_WPSR.

Bits 23:20 – CHKST[3:0] Check Status (cleared by writing SHA_CR.START or SHA_CR.SWRST or by reading SHA_IODATARx)
Value 5 indicates identical hash values (expected hash = hash result). Any other value indicates different hash values.

Bit 16 – CHECKF Check Done Status (cleared by writing SHA_CR.START or SHA_CR.SWRST or by reading SHA_IODATARx)

Value	Description
0	Hash check has not been computed.
1	Hash check has been computed, status is available in the CHKST bits.

Bits 14:12 – URAT[2:0] Unspecified Register Access Type (cleared by writing a 1 to SWRST bit in SHA_CR)
Only the last Unspecified Register Access Type is available through the URAT field.

Value	Name
0	SHA_IDATAR0 to SHA_IDATAR15 written during data processing in PDC mode (URAD = 1 and URAT = 0 can occur only if DUALBUFF is cleared in SHA_MR)
1	Output Data Register read during data processing
2	SHA_MR written during data processing
3	Write-only register read access

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 8 – URAD Unspecified Register Access Detection Status (cleared by writing a 1 to SHA_CR.SWRST)

Value	Description
0	No unspecified register access has been detected since the last SWRST.
1	At least one unspecified register access has been detected since the last SWRST.

Bit 4 – WRDY Input Data Register Write Ready

Value	Description
0	SHA_IDATAR0 cannot be written
1	SHA_IDATAR0 can be written

Bit 2 – TXBUFE TX Buffer Empty (cleared by writing SHA_TCR or SHA_TNCR)
SHA_TCR and SHA_TNCR are PDC registers.

Value	Description
0	SHA_TCR or SHA_TNCR has a value other than 0.
1	Both SHA_TCR and SHA_TNCR have a value of 0.

Bit 1 – ENDTX End of TX Buffer (cleared by writing SHA_TCR or SHA_TNCR)
SHA_TCR and SHA_TNCR are PDC registers.

Value	Description
0	The Transmit Counter Register has not reached 0 since the last write in SHA_TCR or SHA_TNCR.
1	The Transmit Counter Register has reached 0 since the last write in SHA_TCR or SHA_TNCR.

Bit 0 – DATRDY Data Ready (cleared by writing a 1 to bit SWRST or START in SHA_CR, or by reading SHA_IODATARx)

Value	Description
0	Output data is not valid.
1	512/1024-bit block process is completed. DATRDY is cleared when one of the following conditions is met: • Bit START in SHA_CR is set. • Bit SWRST in SHA_CR is set. • The hash result is read.

© 2026 Microchip Technology Inc. and its subsidiaries

41.5.7. SHA Message Size Register

Name: SHA_MSR
Offset: 0x20
Reset: 0x0
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	MSGSIZE[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	MSGSIZE[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	MSGSIZE[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MSGSIZE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – MSGSIZE[31:0] Message Size

The size in bytes of the message. When MSGSIZE differs from 0, the SHA appends the corresponding value converted in bits after the padding section, as described in the FIPS180 specification.
To disable automatic padding, MSGSIZE field must be written to 0.
Note: SHA_MSR is a 32-bit register, thus the automatic padding capability is limited to messages of less than 4 gigabytes. For messages greater than 4 gigabytes, padding must be performed by the software.

© 2026 Microchip Technology Inc. and its subsidiaries

41.5.8. SHA Bytes Count Register

Name: SHA_BCR
Offset: 0x30
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	BYTCNT[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	BYTCNT[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	BYTCNT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BYTCNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – BYTCNT[31:0] Remaining Byte Count Before Auto Padding

When the hash processing starts from the beginning of a message (without preprocessed hash part), BYTCNT must be written with the same value as MSGSIZE. If a part of the message has been already hashed and the hash does not start from the beginning, BYTCNT must be configured with the number of bytes remaining to process before the padding section.

When read, provides the size in bytes of the message remaining to be written before the automatic padding starts.

BYTCNT is automatically updated each time a write occurs in SHA_IDATARx and SHA_IODATARx.

When BYTCNT reaches 0, the MSGSIZE is converted into a bit count and appended at the end of the message after the padding, as described in the FIPS 180 specification.

To disable automatic padding, the MSGSIZE and BYTCNT fields must be written to 0.

© 2026 Microchip Technology Inc. and its subsidiaries

41.5.9. SHA Input Data Register x

Name: SHA_IDATARx
Offset: 0x40 + x*0x04 [x=0..15]
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	IDATA[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	IDATA[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	IDATA[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	IDATA[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – IDATA[31:0] Input Data
32-bit Input Data registers load the data block used for hash processing.
These registers are write-only to prevent reading of input data by another application.
SHA_IDATAR0 corresponds to the first word of the block, SHA_IDATAR15 to the last word of the last block in case SHA algorithm is set to SHA1, SHA224, SHA256, or SHA_IODATAR15 to the last word of the block if SHA algorithm is SHA384 or SHA512 (see SHA Input/Output Data Register x).
SHA_IDATARx can be also written to configure the hash result of the previous fragment of a message when starting the processing of the next fragment when the SHA has processed another message in between fragments.

© 2026 Microchip Technology Inc. and its subsidiaries

41.5.10. SHA Input/Output Data Register x

Name: SHA_IODATARx
Offset: 0x80 + x*0x04 [x=0..15]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	IODATA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	IODATA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	IODATA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	IODATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – IODATA[31:0] Input/Output Data

These registers can be used to read the resulting message digest and to write the second part of the message block when the SHA algorithm is SHA-384 or SHA-512.
SHA_IODATAR0 to SHA_IODATAR15 can be written or read but reading these offsets does not return the content of corresponding parts (words) of the message block. Only results from SHA calculation can be read through these registers.
When SHA processing is in progress, these registers return 0x0000.
SHA_IODATAR0 corresponds to the first word of the message digest; SHA_IODATAR4 to the last one in SHA1 mode, SHA_IODATAR6 in SHA224, SHA_IODATAR7 in SHA256, SHA_IODATAR11 in SHA384 or SHA_IODATAR15 in SHA512.
When SHA224 is selected, the content of SHA_IODATAR7 must be ignored.
When SHA384 is selected, the content of SHA_IODATAR12 to SHA_IODATAR15 must be ignored.

© 2026 Microchip Technology Inc. and its subsidiaries

41.5.11. SHA Write Protection Mode Register

Name: SHA_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		ACTION[1:0]		FIRSTE		WPCREN	WPITEN	WPEN
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x534841	PASSWD	Writing any other value in this field aborts the write operation of the WPEN,WPITEN,WPCREN bits. Always reads as 0.

Bits 6:5 – ACTION[1:0] Action on Abnormal Event Detection

Value	Name	Description
0	REPORT_ONLY	No action (stop or clear key) is performed when one of WPVS,CGD,SEQE, or SWE flag is set.
1	LOCK_WPVS_SWE	If a processing is in progress when the SHA_WPSR.WPVS/SWE event detection occurs, the current processing is ended normally but no other processing is started while a SHA_CR.UNLOCK command is issued.
2	LOCK_CGD_SEQE	If a processing is in progress when the SHA_WPSR.CGD/SEQE event detection occurs, the current processing is ended normally but no other processing is started while a SHA_CR.UNLOCK command is issued.
3	LOCK_ANY_EV	If a processing is in progress when the SHA_WPSR.WPVS/CGD/SEQE/SWE events detection occurs, the current processing is ended normally but no other processing is started while a SHA_CR.UNLOCK command is issued.

Bit 4 – FIRSTE First Error Report Enable

Value	Description
0	The last write protection violation source is reported in SHA_WPSR.WPVSRC and the last software control error type is reported in SHA_WPSR.SWETYP. The SHA_ISR.SECE flag is set at the first error occurrence within a series.
1	Only the first write protection violation source is reported in SHA_WPSR.WPVSRC and only the first software control error type is reported in SHA_WPSR.SWETYP. The SHA_ISR.SECE flag is set at the first error occurrence within a series.

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 2 – WPCREN Write Protection Control Enable

Value	Description
0	Disables the write protection on control register if WPKEY corresponds to 0x534841 (“SHA” in ASCII).
1	Enables the write protection on control register if WPKEY corresponds to 0x534841 (“SHA” in ASCII).

Bit 1 – WPITEN Write Protection Interruption Enable

Value	Description
0	Disables the write protection on interrupt registers if WPKEY corresponds to 0x534841 (“SHA” in ASCII).
1	Enables the write protection on interrupt registers if WPKEY corresponds to 0x534841 (“SHA” in ASCII).

Bit 0 – WPEN Write Protection Configuration Enable
See Register Write Protection for the list of registers that can be write-protected.

Value	Description
0	Disables the write protection on configuration registers if WPKEY corresponds to 0x534841 (“SHA” in ASCII).
1	Enables the write protection on configuration registers if WPKEY corresponds to 0x534841 (“SHA” in ASCII).

© 2026 Microchip Technology Inc. and its subsidiaries

41.5.12. SHA Write Protection Status Register

Name: SHA_WPSR
Offset: 0xE8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	ECLASS				SWETYP[3:0]			
Access	R				R	R	R	R
Reset	0				0	0	0	0

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
					SWE	SEQE	CGD	WPVS
Access					R	R	R	R
Reset					0	0	0	0

Bit 31 – ECLASS Software Error Class (cleared on read)
0 (WARNING): An abnormal access that does not affect system functionality
1 (ERROR): An access is performed into key, input data, control registers while the SHA is performing an encryption/decryption or a start is request by software or DMA while the key is not fully configured.

Bits 27:24 – SWETYP[3:0] Software Error Type (cleared on read)

Value	Name	Description
0	READ_WO	A write-only register has been read (Warning).
1	WRITE_RO	SHA is enabled and a write access has been performed on a read-only register (Warning).
2	UNDEF_RW	Access to an undefined address (Warning).
3	CTRL_START	SHA is locked and a start command with SHA_CR.START has been performed.
4	AUTO_START	SHA is locked and a tentative automatic start has been performed by writing input data registers (SHA_MR.SMOD>0).
5	BAD_START	SHA is not locked and a start command with SHA_CR.START has been performed whereas Start mode is automatic (SHA_MR.SMOD>0)

Bits 15:8 – WPVSRC[7:0] Write Protection Violation Source
When WPVS=1, WPVSRC indicates the register address offset at which a write access has been attempted.
When WPVS=0 and SWE=1, WPVSRC reports the address of the incorrect software access. As soon as WPVS=1, WPVSRC returns the address of the write-protected violation.

Bit 3 – SWE Software Control Error (cleared on read)

Value	Description
0	No software error has occurred since the last read of SHA_WPSR.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
1	A software error has occurred since the last read of SHA_WPSR. The field SWETYP details the type of software error; the associated incorrect software access is reported in the field WPVSRC (if WPVS=0).

Bit 2 – SEQE Internal Sequencer Error (cleared on read)

Value	Description
0	No peripheral internal sequencer error has occurred since the last read of SHA_WPSR.
1	A peripheral internal sequencer error has occurred since the last read of SHA_WPSR. This flag can only be set under abnormal operating conditions.

Bit 1 – CGD Clock Glitch Detected (cleared on read)

Value	Description
0	The clock monitoring circuitry has not been corrupted since the last read of SHA_WPSR. Under normal operating conditions, this bit is always cleared.
1	The clock monitoring circuitry has been corrupted since the last read of SHA_WPSR. This flag can only be set in case of an abnormal clock signal waveform (glitch).

Bit 0 – WPVS Write Protection Violation Status (cleared on read)

Value	Description
0	No write protect violation has occurred since the last read of SHA_WPSR.
1	A write protect violation has occurred since the last read of SHA_WPSR. The address offset of the violated register is reported into field WPVSRC.

© 2026 Microchip Technology Inc. and its subsidiaries

42. True Random Number Generator (TRNG)

42.1. Description

The True Random Number Generator (TRNG) passes the American *NIST Special Publication 800-22 (A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications)* and the *Diehard Suite of Tests*.

The TRNG may be used as an entropy source for seeding an NIST approved DRNG (Deterministic RNG) as required by FIPS PUB 140-2 and 140-3.

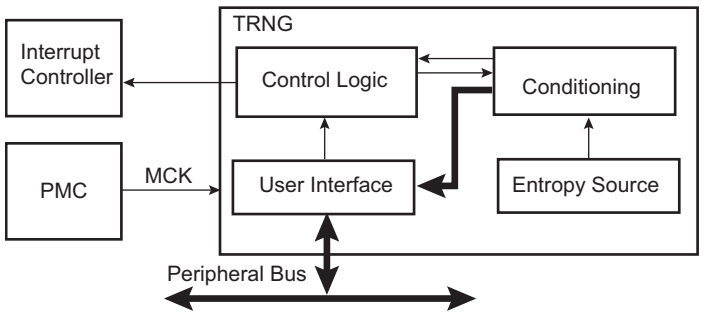
The TRNG is fully designed with digital cells, and under the specified operating conditions, external factors such as temperature, humidity, etc. affect TRNG ageing in the same manner as all other digital peripherals (CPU core, bus matrix, etc.) of the product.

42.2. Embedded Characteristics

- Passes *NIST Special Publication 800-22 Test Suite*
- Passes *Diehard Suite of Tests*
- Usable as Entropy Source for Seeding a NIST-approved DRNG (Deterministic RNG) as required by FIPS PUB 140-2 and 140-3
- Provides a 32-bit Random Number at Maximum 84 Clock Cycles
- Functional Safety Monitors and Reports:
  - Monitoring of internal sequencer abnormal states
  - Abnormal software access reports
  - Register write protection
  - Dedicated interrupt line for safety events
- Private Key Bus Interface to Transfer Cryptographic Keys Not Readable From Any Peripheral Nor From Software

42.3. Block Diagram

Figure 42-1. TRNG Block Diagram



42.4. Product Dependencies

42.4.1. Power Management

The TRNG interface may be clocked through the Power Management Controller (PMC), thus the programmer must first configure the PMC to enable the TRNG user interface clock. The user interface clock is independent from any clock that may be used in the entropy source logic circuitry. The source of entropy can be enabled before enabling the user interface clock.

© 2026 Microchip Technology Inc. and its subsidiaries

42.4.2. Interrupt Sources

The TRNG interface has one interrupt line used for standard functions and one interrupt line used to trigger any safety or security event that may occur. Both lines are connected to the Interrupt Controller. The interrupt line for standard functions is driven by TRNG_IMR and TRNG_ISR, whereas the interrupt line for safety/security functions is driven by TRNG_WPSR flags. If one of the flags WPVS, CGD, SEQE or SWE is set, the interrupt line is asserted. In order to handle interrupts, the Interrupt Controller must be programmed before configuring the TRNG.

42.5. Functional Description

As soon as the TRNG is enabled in the Control register (TRNG_CR), the generator provides one 32-bit random value at a maximum streaming rate of 84 clock cycles. Entropy rate increases at a lower frequency. It is possible to divide by 2 the streaming rate by configuring the Mode register (TRNG_MR) to achieve better entropy if the streaming rate does not require new data every 84 clock cycles. For a lower streaming rate, the software intervention is required to skip, on a regular basis, the data ready information reported in the Status register (TRNG_ISR).

A sequence of random values can be generated by the TRNG and a random value can be directly loaded through the private key bus into specific private key internal registers of the private key bus clients (for example, AES or other encryption unit). There is no possibility of reading these keys from the processor and software from system bus. This is done by writing the Private Key Bus Control register (TRNG_PKBCR) with the appropriate destination (KSLAVE) and length of the key to be generated (KLENGTH).

This random value transferred through the private key bus cannot be used for encrypted communications with remote equipment, but is useful while the system remains in Active mode to reinforce the security of data processed by the application running on the system and stored temporarily in external memories. The cryptography keys are never known to application software, thus they cannot be exchanged or provided to the external world in any case.

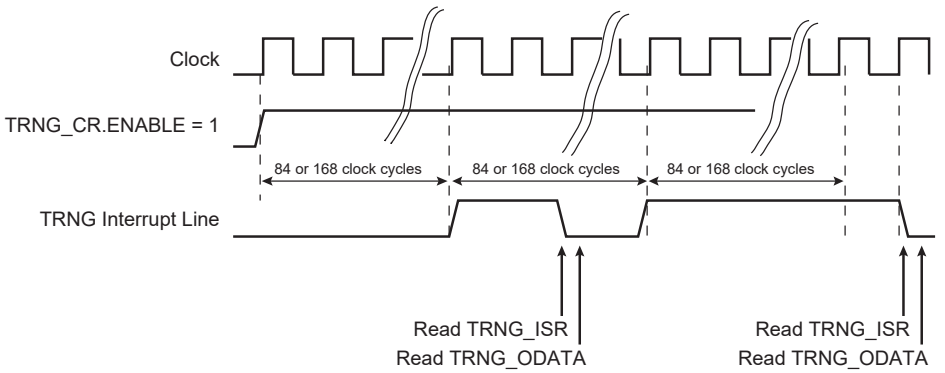
By writing a '1' to the HALFR bit in the Mode register (TRNG_MR), the random values are provided every 168 cycles instead of every 84 cycles. HALFR must be written to '1' when the TRNG peripheral clock frequency is above 100 MHz.

The TRNG interrupt line can be enabled in the Interrupt Enable register (TRNG_IER), and disabled in the Interrupt Disable register (TRNG_IDR). This interrupt is set when a new random value is available or when a transfer over the private key bus is complete and is cleared when the Status register (TRNG_ISR) is read. The flag TRNG_ISR.DATRDY is set when the random data is ready to be read out on the 32-bit Output Data register (TRNG_ODATA). The flag TRNG_ISR.EOTPKB is set when the transfer through the private key bus is complete.

Normal Operating Mode

The normal operating mode checks that the TRNG_ISR.DATRDY flag equals '1' before reading TRNG_ODATA when a 32-bit random value is required by the software application.

Figure 42-2. TRNG Data Generation Sequence



© 2026 Microchip Technology Inc. and its subsidiaries

Key Bus Operating Mode

After a write to KSLAVE and KLENGTH in TRNG_PKBCR, the software:

- waits for the end of transfer of the key indicated by the TRNG_ISR.EOTPKB flag being read at '1', optionally after a TRNG interrupt,
- checks for any key bus access violation in the selected private key bus destination client status register,
- uses the private key bus destination client or launches any other private key bus transfer.

Figure 42-3. TRNG Private Key Bus

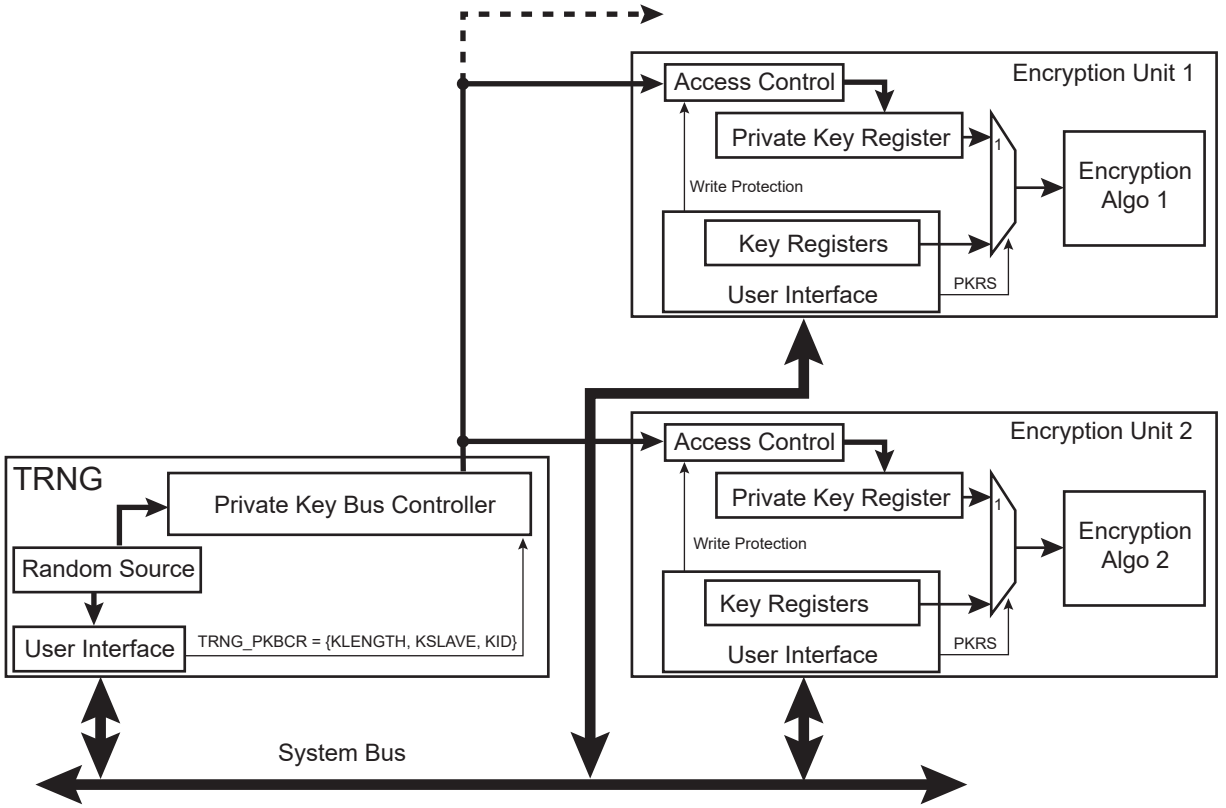
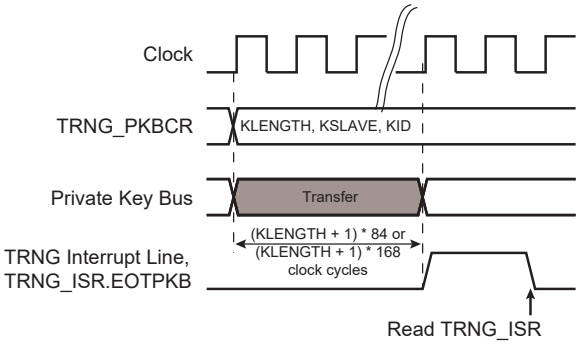


Figure 42-4. TRNG Private Key Bus Transfer



42.5.1. First Value Read after Power-up

After a power-up and the first configuration to enable the TRNG, the first data can be read as soon as the flag DATRDY is set in the Interrupt Status register (TRNG_ISR). However, randomness

© 2026 Microchip Technology Inc. and its subsidiaries

(entropy) of a sequence of first value read after a power-up sequence is correct only if the TRNG has been enabled for a significant period of time.

When the first value after power-up is a key factor for the application, it is recommended to wait for 5 ms before reading the first value and after the power-up followed by the initial enable of the TRNG.

42.5.2. Entropy

The TRNG provides a new random data at a maximum rate of peripheral clock divided by 84. However, entropy increases as the reading rate decreases.

42.5.3. Register Write Protection

To prevent any single software error from corrupting TRNG behavior, certain registers in the address space can be write-protected by setting the WPEN (Write Protection Enable), WPITEN (Write Protection Interrupt Enable), and/or WPCREN (Write Protection Control Enable) bits in the TRNG Write Protection Mode Register (TRNG_WPMR).

If a write access to the protected registers is detected, the WPVS flag in the TRNG Write Protection Status Register (TRNG_WPSR) is set and the field WPVSRC indicates the register in which the write access has been attempted.

The WPVS flag is automatically cleared by reading TRNG_WPSR.

The following register can be write-protected when WPEN is set:

- TRNG Mode Register

The following registers can be write-protected when WPITEN is set:

- TRNG Interrupt Enable Register
- TRNG Interrupt Disable Register

The following registers can be write-protected when WPCREN is set:

- TRNG Control Register
- TRNG Private Key Bus Control Register

42.5.4. Security and Functional Analysis and Reports

Several type of checks are performed when the TRNG is enabled.

The peripheral clock of the TRNG is monitored by specific circuitry to detect abnormal waveforms on the internal clock net that may affect the behavior of the TRNG. Corruption on the triggering edge of the clock or a pulse with a minimum duration may be identified. If the flag TRNG_WPSR.CGD is set, an abnormal condition occurred on the peripheral clock. This flag is not set under normal operating conditions.

The internal sequencer of the TRNG is also monitored and if an abnormal state is detected, the flag TRNG_WPSR.SEQE is set. This flag is not set under normal operating conditions.

The software accesses to the TRNG are monitored and if an incorrect access is performed, the flag TRNG_WPSR.SWE is set. The type of incorrect/abnormal software access is reported in the TRNG_WPSR.SWETYP field (see TRNG Write Protection Status Register for details). e.g., reading the TRNG_ODATA when the TRNG is disabled is an error, as well as reading the TRNG_ODATA, when the TRNG_ISR.DATRDY flag is cleared. TRNG_WPSR.ECLASS is an indicator reporting the criticality of the SWETYP report.

The flags CGD, SEQE, SWE and WPVS are automatically cleared when TRNG_WPSR is read.

If one of these flags is set, the flag TRNG_ISR.SECE is set and can trigger an interrupt if the TRNG_IMR.SECE bit is ‘1’. SECE is cleared by reading TRNG_ISR.

© 2026 Microchip Technology Inc. and its subsidiaries

42.6. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	TRNG_CR	31:24	WAKEY[23:16]							
		23:16	WAKEY[15:8]							
		15:8	WAKEY[7:0]							
		7:0								ENABLE
0x04	TRNG_MR	31:24								
		23:16								
		15:8								
		7:0								HALFR
0x08	TRNG_PKBCR	31:24	WAKEY[15:8]							
		23:16	WAKEY[7:0]							
		15:8	KLENGTH[7:0]							
		7:0			KSLAVE[1:0]					KID
0x0C ... 0x0F	Reserved									
0x10	TRNG_IER	31:24								
		23:16								
		15:8								
		7:0						EOTPKB	SECE	DATRDY
0x14	TRNG_IDR	31:24								
		23:16								
		15:8								
		7:0						EOTPKB	SECE	DATRDY
0x18	TRNG_IMR	31:24								
		23:16								
		15:8								
		7:0						EOTPKB	SECE	DATRDY
0x1C	TRNG_ISR	31:24								
		23:16								
		15:8								
		7:0						EOTPKB	SECE	DATRDY
0x20 ... 0x4F	Reserved									
0x50	TRNG_ODATA	31:24	ODATA[31:24]							
		23:16	ODATA[23:16]							
		15:8	ODATA[15:8]							
		7:0	ODATA[7:0]							
0x54 ... 0xE3	Reserved									
0xE4	TRNG_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0				FIRSTE		WPCREN	WPITEN	WPEN
0xE8	TRNG_WPSR	31:24	ECLASS				SWETYP[3:0]			
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0					SWE	SEQE	CGD	WPVS

© 2026 Microchip Technology Inc. and its subsidiaries

42.6.1. TRNG Control Register

Name: TRNG_CR
Offset: 0x00
Reset: –
Property: Write-only

This register can only be written if the WPCREN bit is cleared in the TRNG Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	WAKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	WAKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	WAKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
								ENABLE
Access								W
Reset								–

Bits 31:8 – WAKEY[23:0] Register Write Access Key

Value	Name	Description
0x524E47	PASSWD	Writing any other value in this field aborts the write operation.

Bit 0 – ENABLE Enable TRNG to Provide Random Values

Value	Description
0	Disables the TRNG if 0x524E47 (“RNG” in ASCII) is written in WAKEY field at the same time.
1	Enables the TRNG if 0x524E47 (“RNG” in ASCII) is written in WAKEY field at the same time.

© 2026 Microchip Technology Inc. and its subsidiaries

42.6.2. TRNG Mode Register

Name: TRNG_MR
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the TRNG Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								HALFR
Access								R/W
Reset								0

Bit 0 – HALFR Half Rate Enable

Value	Name	Description
0	DISABLED	Maximum stream rate provided (1 sample every 84 MCK clock cycles).
1	ENABLED	Half maximum stream rate provided if the peripheral clock frequency is above 100 MHz (1 sample every 168 MCK clock cycles).

© 2026 Microchip Technology Inc. and its subsidiaries

42.6.3. TRNG Private Key Bus Control Register

Name: TRNG_PKBCR
Offset: 0x08
Reset: –
Property: Write-only

This register can only be written if the WPCREN bit is cleared in the TRNG Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	WAKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	23	22	21	20	19	18	17	16
	WAKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	15	14	13	12	11	10	9	8
	KLENGTH[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–
Bit	7	6	5	4	3	2	1	0
			KSLAVE[1:0]					KID
Access			W	W				W
Reset			–	–				–

Bits 31:16 – WAKEY[15:0] Register Write Access Key

Value	Name	Description
0x524B	PASSWD	Writing any other value in this field aborts the write operation.

Bits 15:8 – KLENGTH[7:0] Key Length
Length-1 in 32-bit words of the key(s) to be directly loaded from the TRNG into the private key internal registers of the private key bus client KSLAVE.
Example: for one 64-bit key to be loaded, KLENGTH must be written to 1. For 128-bit keys, KLENGTH must be written to 3.

Bits 5:4 – KSLAVE[1:0] Key Bus Client
Private key bus client identifier for the destination encryption unit to be loaded from the TRNG.

Value	Name	Description
0	AES_ID	AES
1	AESB_ID	AESB
2	Reserved_ID	Reserved
3	Reserved_ID	Reserved

Bit 0 – KID Key ID (Must always be written to 0)

© 2026 Microchip Technology Inc. and its subsidiaries

42.6.4. TRNG Interrupt Enable Register

Name: TRNG_IER
Offset: 0x10
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the TRNG Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						EOTPKB	SECE	DATRDY
Access						W	W	W
Reset						–	–	–

Bit 2 – EOTPKB End Of Transfer on Private Key Bus Interrupt Enable

Bit 1 – SECE Security and/or Safety Event Interrupt Enable

Bit 0 – DATRDY Data Ready Interrupt Enable

© 2026 Microchip Technology Inc. and its subsidiaries

42.6.5. TRNG Interrupt Disable Register

Name: TRNG_IDR
Offset: 0x14
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the TRNG Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						EOTPKB	SECE	DATRDY
Access						W	W	W
Reset						–	–	–

Bit 2 – EOTPKB End Of Transfer on Private Key Bus Interrupt Disable

Bit 1 – SECE Security and/or Safety Event Interrupt Disable

Bit 0 – DATRDY Data Ready Interrupt Disable

© 2026 Microchip Technology Inc. and its subsidiaries

42.6.6. TRNG Interrupt Mask Register

Name: TRNG_IMR
Offset: 0x18
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						EOTPKB	SECE	DATRDY
Access						R	R	R
Reset						0	0	0

Bit 2 – EOTPKB End Of Transfer on Private Key Bus Interrupt Mask

Bit 1 – SECE Security and/or Safety Event Interrupt Mask

Bit 0 – DATRDY Data Ready Interrupt Mask

© 2026 Microchip Technology Inc. and its subsidiaries

42.6.7. TRNG Interrupt Status Register

Name: TRNG_ISR
Offset: 0x1C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						EOTPKB	SECE	DATRDY
Access						R	R	R
Reset						0	0	0

Bit 2 – EOTPKB End Of Transfer on Private Key Bus (cleared on read)

Value	Description
0	No private key bus transfer has ended since the last read of the Interrupt Status Register.
1	The private key bus transfer has ended.

Bit 1 – SECE Security and/or Safety Event (cleared on read)

Value	Description
0	No safety or security event occurred since the last read of the Interrupt Status Register.
1	One or more safety or security event occurred since the last read of TRNG_ISR. For details on the event, see TRNG Write Protection Status Register.

Bit 0 – DATRDY Data Ready (cleared on read)

Value	Description
0	Output data is not valid or TRNG is disabled.
1	New random value has been completed since the last read of TRNG_ISR.

© 2026 Microchip Technology Inc. and its subsidiaries

42.6.8. TRNG Output Data Register

Name: TRNG_ODATA
Offset: 0x50
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	ODATA[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ODATA[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ODATA[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ODATA[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ODATA[31:0] Output Data
The 32-bit Output Data register contains the 32-bit random data.

© 2026 Microchip Technology Inc. and its subsidiaries

42.6.9. TRNG Write Protection Mode Register

Name: TRNG_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
				FIRSTE		WPCREN	WPITEN	WPEN
Access				R/W		R/W	R/W	R/W
Reset				0		0	0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x524E47	PASSWD	Writing any other value in this field aborts the write operation of bits WPEN, WPITEN and WPCREN. Always reads as 0.

Bit 4 – FIRSTE First Error Report Enable

Value	Description
0	The last write protection violation source is reported in TRNG_WPSR.WPVSRC and the last software control error type is reported in TRNG_WPSR.SWETYP. The TRNG_ISR.SECE flag is set at the first error occurrence within a series.
1	Only the first write protection violation source is reported in TRNG_WPSR.WPVSRC and only the first software control error type is reported in TRNG_WPSR.SWETYP. The TRNG_ISR.SECE flag is set at the first error occurrence within a series.

Bit 2 – WPCREN Write Protection Control Enable

Value	Description
0	Disables the write protection on control register if WPKEY corresponds to 0x524E47 (“RNG” in ASCII).
1	Enables the write protection on control register if WPKEY corresponds to 0x524E47 (“RNG” in ASCII).

Bit 1 – WPITEN Write Protection Interrupt Enable

Value	Description
0	Disables the write protection on interrupt registers if WPKEY corresponds to 0x524E47 (“RNG” in ASCII).
1	Enables the write protection on interrupt registers if WPKEY corresponds to 0x524E47 (“RNG” in ASCII).

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 0 – WPEN Write Protection Enable

See Register Write Protection for the list of registers that can be write-protected.

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x524E47 (“RNG” in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x524E47 (“RNG” in ASCII).

© 2026 Microchip Technology Inc. and its subsidiaries

42.6.10. TRNG Write Protection Status Register

Name: TRNG_WPSR
Offset: 0xE8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	ECLASS				SWETYP[3:0]			
Access	R				R	R	R	R
Reset	0				0	0	0	0
Bit	23	22	21	20	19	18	17	16
	WPVSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
					SWE	SEQE	CGD	WPVS
Access					R	R	R	R
Reset					0	0	0	0

Bit 31 – ECLASS Software Error Class (cleared on read)

Value	Name	Description
0	WARNING	An abnormal access that does not affect system functionality.
1	ERROR	Reading TRNG_ODATA when TRNG is disabled or used for private key bus transfer does not provide a random value. Writing to the PKB_CTRL register while a private key bus transfer is ongoing does not launch a new private key bus transfer.

Bits 27:24 – SWETYP[3:0] Software Error Type (cleared on read)

Value	Name	Description
0	READ_WO	TRNG is enabled and a write-only register has been read (Warning).
1	WRITE_RO	TRNG is enabled and a write access has been performed on a read-only register (Warning).
2	UNDEF_RW	Access to an undefined address.
3	TRNG_DIS	The TRNG_ODATA register has been read when TRNG is disabled or used for private key bus transfer (Error).
4	PKB_BUSY	A write access to the PKB_CTRL register has been attempted during a private key bus transfer (Error).
5	LOCK_ERR	A write access to TRNG_WPMR has been attempted when one of the write protection bits is already locked, its corresponding lock control bit is set and the corresponding write protection bit is cleared, which looks like an unlock tentative (Warning).

Bits 23:8 – WPVSRC[15:0] Write Protection Violation Source (cleared on read)
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.
When WPVS=0 and SWE=1, WPVSRC reports the address of the incorrect software access. As soon as WPVS=1, WPVSRC returns the address of the write-protected violation.

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 3 – SWE Software Control Error (cleared on read)

Value	Description
0	No software error has occurred since the last read of TRNG_WPSR.
1	A software error has occurred since the last read of TRNG_WPSR. The field SWETYP details the type of software error; the associated incorrect software access is reported in the field WPVSRC (if WPVS=0).

Bit 2 – SEQE Internal Sequencer Error (cleared on read)

Value	Description
0	No peripheral internal sequencer error has occurred since the last read of TRNG_WPSR.
1	A peripheral internal sequencer error has occurred since the last read of TRNG_WPSR. This flag is set under abnormal operating conditions.

Bit 1 – CGD Clock Glitch Detected (cleared on read)

Value	Description
0	No clock glitch has occurred since the last read of TRNG_WPSR. Under normal operating conditions, this bit is always cleared.
1	A clock glitch has occurred since the last read of TRNG_WPSR. This flag is set in case of abnormal clock signal waveform (glitch).

Bit 0 – WPVS Write Protection Violation Status (cleared on read)

Value	Description
0	No write protection violation has occurred since the last read of TRNG_WPSR.
1	A write protection violation has occurred since the last read of TRNG_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

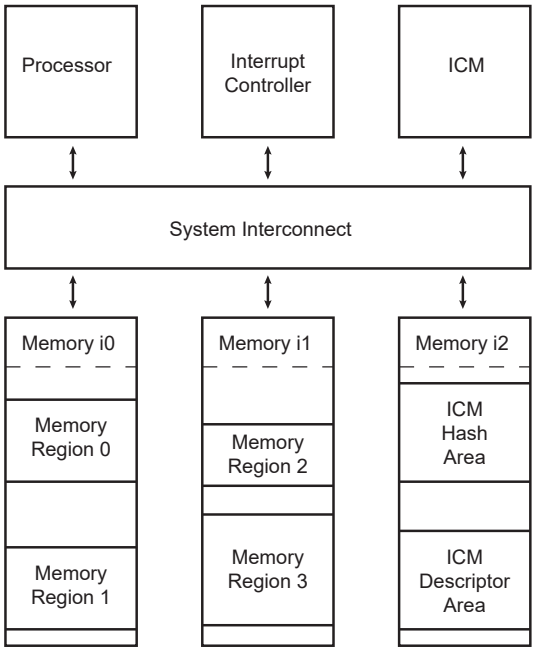
© 2026 Microchip Technology Inc. and its subsidiaries

43. Integrity Check Monitor (ICM)

43.1. Description

The Integrity Check Monitor (ICM) is a DMA controller that performs hash calculation over multiple memory regions through the use of transfer descriptors located in memory (ICM Descriptor Area). The Hash function is based on the Secure Hash Algorithm (SHA). The ICM integrates two modes of operation. The first one is used to hash a list of memory regions and save the digests to memory (ICM Hash Area). The second mode is an active monitoring of the memory. In that mode, the hash function is evaluated and compared to the digest located at a predefined memory address (ICM Hash Area). If a mismatch occurs, an interrupt is raised. See the figure below for an example of four-region monitoring. Hash and Descriptor areas are located in Memory instance i2, and the four regions are split in memory instances i0 and i1.

Figure 43-1. Four-region Monitoring Example



The ICM SHA engine is compliant with the American FIPS (Federal Information Processing Standard) Publication 180-2 specification.

The following terms are concise definitions of the ICM concepts used throughout this document:

- Region — A partition of instruction or data memory space.
- Region Descriptor — A data structure stored in memory, defining region attributes.
- Region Attributes — Region start address, region size, region SHA engine processing mode, Write Back or Compare function mode.
- Context Registers — A set of ICM non-memory-mapped, internal registers which are automatically loaded, containing the attributes of the region being processed.
- Main List — A list of region descriptors. Each element associates the start address of a region with a set of attributes.
- Secondary List — A linked list defined on a per region basis that describes the memory layout of the region (when the region is non-contiguous).
- Hash Area — predefined memory space where the region hash results (digest) are stored.

© 2026 Microchip Technology Inc. and its subsidiaries

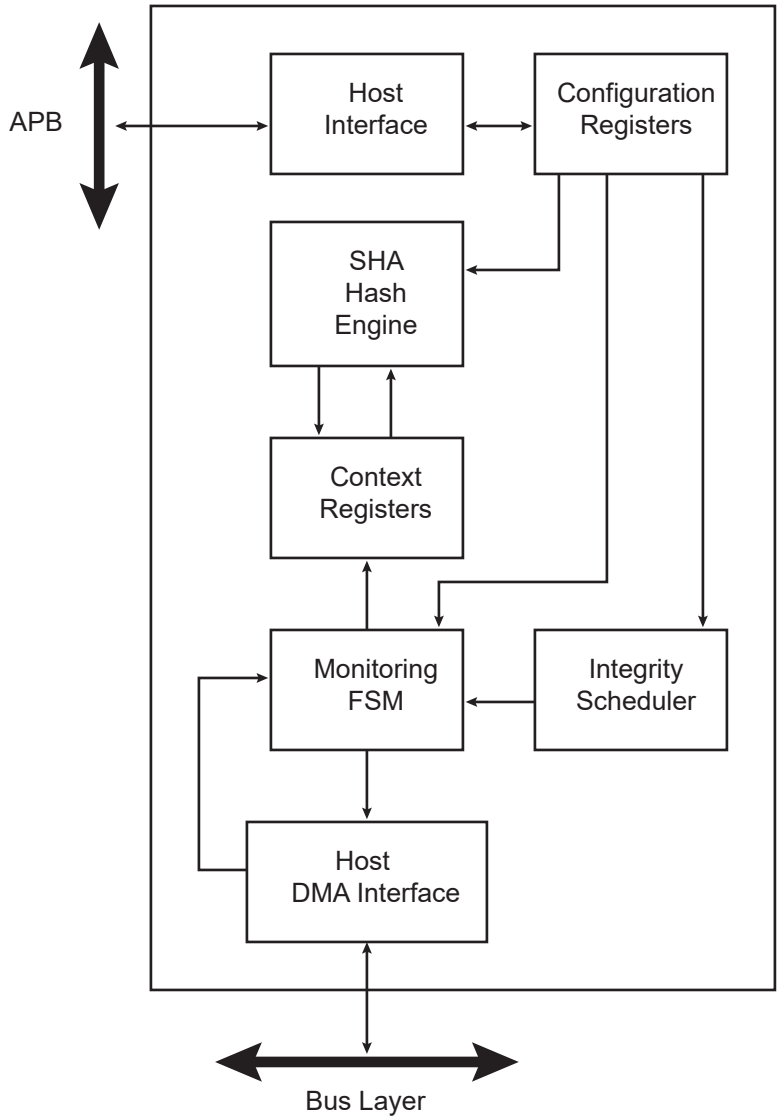
43.2. Embedded Characteristics

- Host DMA Interface
- Supports Monitoring of up to 4 Non-Contiguous Memory Regions
- Supports Block Gathering Using Linked Lists
- Supports Secure Hash Algorithm (SHA1, SHA224, SHA256)
- Compliant with FIPS Publication 180-2
- Configurable Processing Period:
  - When SHA1 algorithm is processed, the runtime period is either 85 or 209 clock cycles.
  - When SHA256 or SHA224 algorithm is processed, the runtime period is either 72 or 194 clock cycles.
- Programmable Bus Burden
- Two Separate Interrupt Lines: One Line Driven by Standard Functions and the Second Line Driven by Safety Checks

© 2026 Microchip Technology Inc. and its subsidiaries

43.3. Block Diagram

Figure 43-2. ICM Block Diagram



43.4. Product Dependencies

43.4.1. Power Management

The peripheral clock is not continuously provided to the ICM. The programmer must first enable the ICM clock in the Power Management Controller (PMC) before using the ICM.

43.4.2. Interrupt Sources

The ICM has one interrupt line used for standard functions and one interrupt line used to trigger any safety or security events that may occur. Both lines are connected to the Interrupt Controller.

Separate lines ease management of interrupt priorities related to safety/security checks.

The interrupt line for standard functions is driven by ICM_IMR and ICM_ISR, whereas the interrupt line for safety/security functions is driven by ICM_WPSR flags. If one of the flags in ICM_WPSR is set, the interrupt line is asserted. Handling the ICM interrupt requires programming the interrupt controller before configuring the ICM.

© 2026 Microchip Technology Inc. and its subsidiaries

43.5. Functional Description

43.5.1. Overview

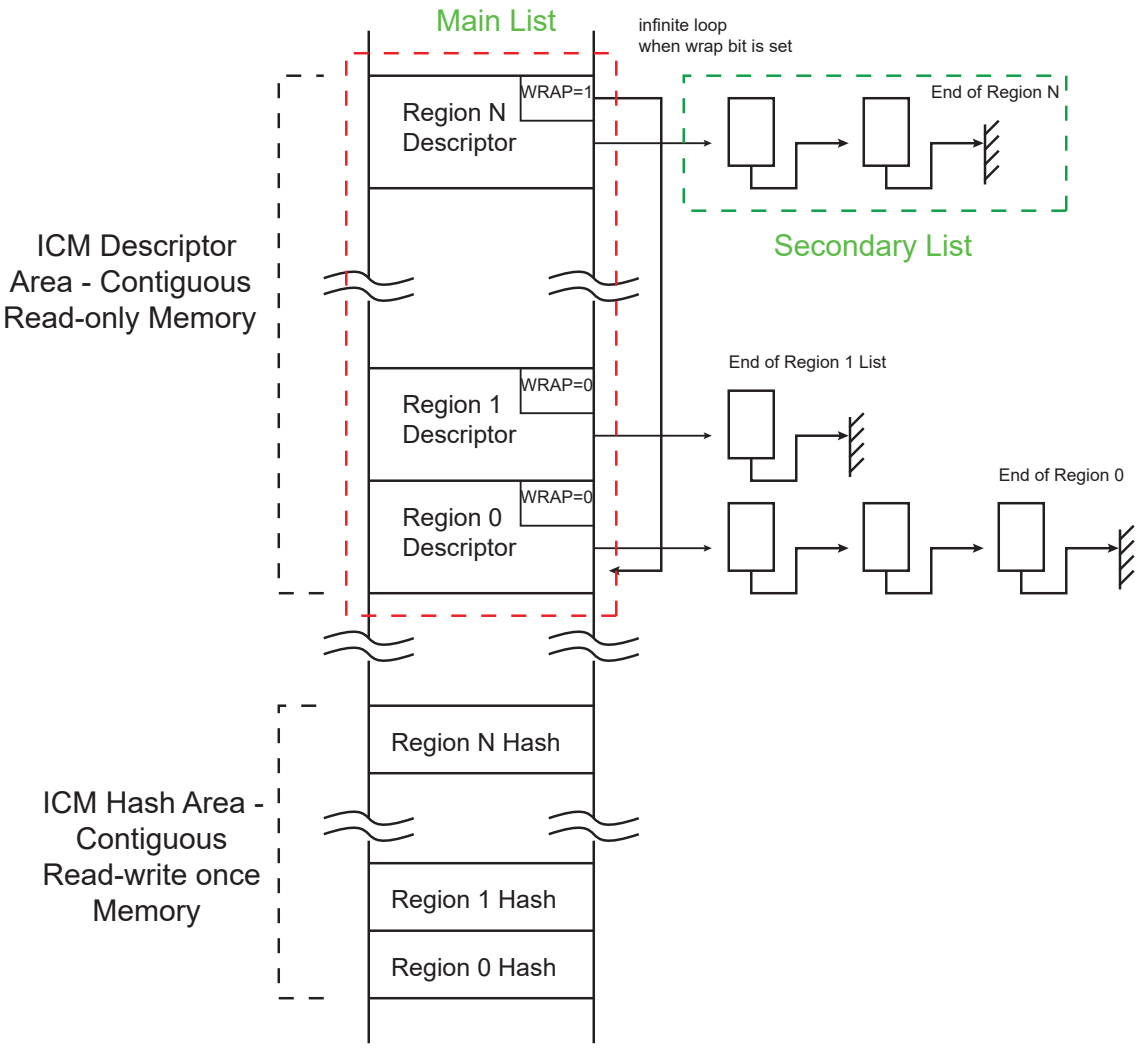
The Integrity Check Monitor (ICM) is a DMA controller that performs SHA-based memory hashing over memory regions. As shown in figure Integrity Check Monitor Block Diagram, it integrates a DMA interface, a Monitoring Finite State Machine (FSM), an integrity scheduler, a set of context registers, a SHA engine, an interface for configuration and status registers.

The ICM integrates a Secure Hash Algorithm engine (SHA). This engine requires a message padded according to FIPS180-2 specification when used as a SHA calculation unit only. Otherwise, if the ICM is used as integrated check for memory content, the padding is not mandatory. The SHA module produces an N-bit message digest each time a block is read and a processing period ends. N is 160 for SHA1, 224 for SHA224, 256 for SHA256.

When the ICM module is enabled, it sequentially retrieves a circular list of region descriptors from the memory (Main List described in figure ICM Region Descriptor and Hash Areas). Up to four regions may be monitored. Each region descriptor is composed of four words indicating the layout of the memory region (see figure Region Descriptor). It also contains the hashing engine configuration on a per-region basis. As soon as the descriptor is loaded from the memory and context registers are updated with the data structure, the hashing operation starts. A programmable number of blocks (see TRSIZE field of the ICM_RCTRL structure member) is transferred from the memory to the SHA engine. When the desired number of blocks have been transferred, the digest is either moved to memory (Write Back function) or compared with a digest reference located in the system memory (Compare function). If a digest mismatch occurs, an interrupt is triggered if unmasked. The ICM module passes through the region descriptor list until the end of the list marked by an end of list marker (WRAP or EOM bit in ICM_RCFG structure member set to one). To continuously monitor the list of regions, the WRAP bit must be set to one in the last data structure and EOM must be cleared.

© 2026 Microchip Technology Inc. and its subsidiaries

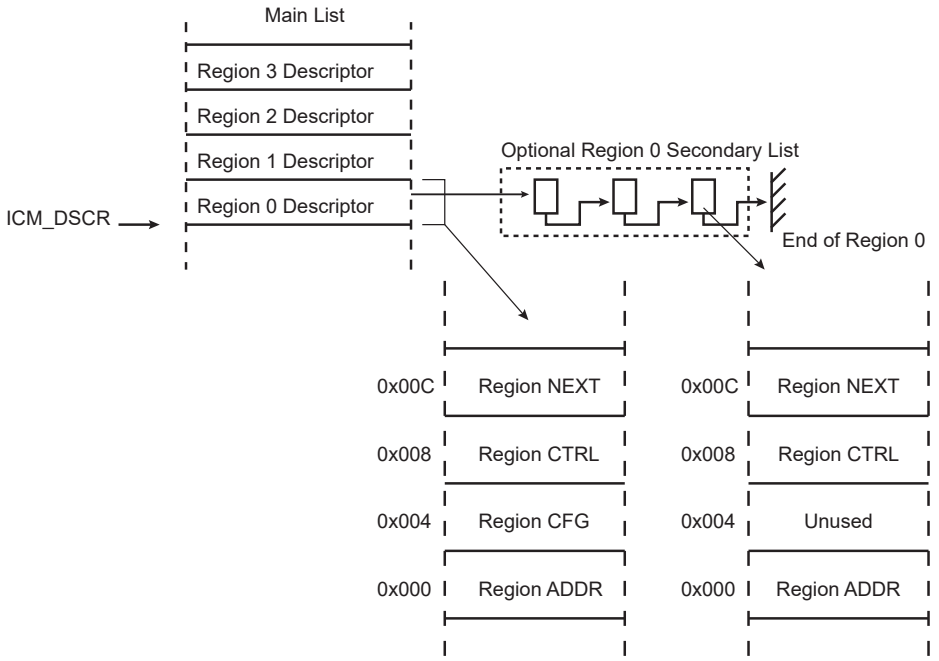
Figure 43-3. ICM Region Descriptor and Hash Areas



Each region descriptor supports gathering of data through the use of the Secondary List. Unlike the Main List, the Secondary List cannot modify the configuration attributes of the region. When the end of the Secondary List has been encountered, the ICM returns to the Main List. Memory integrity monitoring can be considered as a background service and the mandatory bandwidth shall be very limited. In order to limit the ICM memory bandwidth, use ICM_CFG.BBC to control the ICM memory load.

© 2026 Microchip Technology Inc. and its subsidiaries

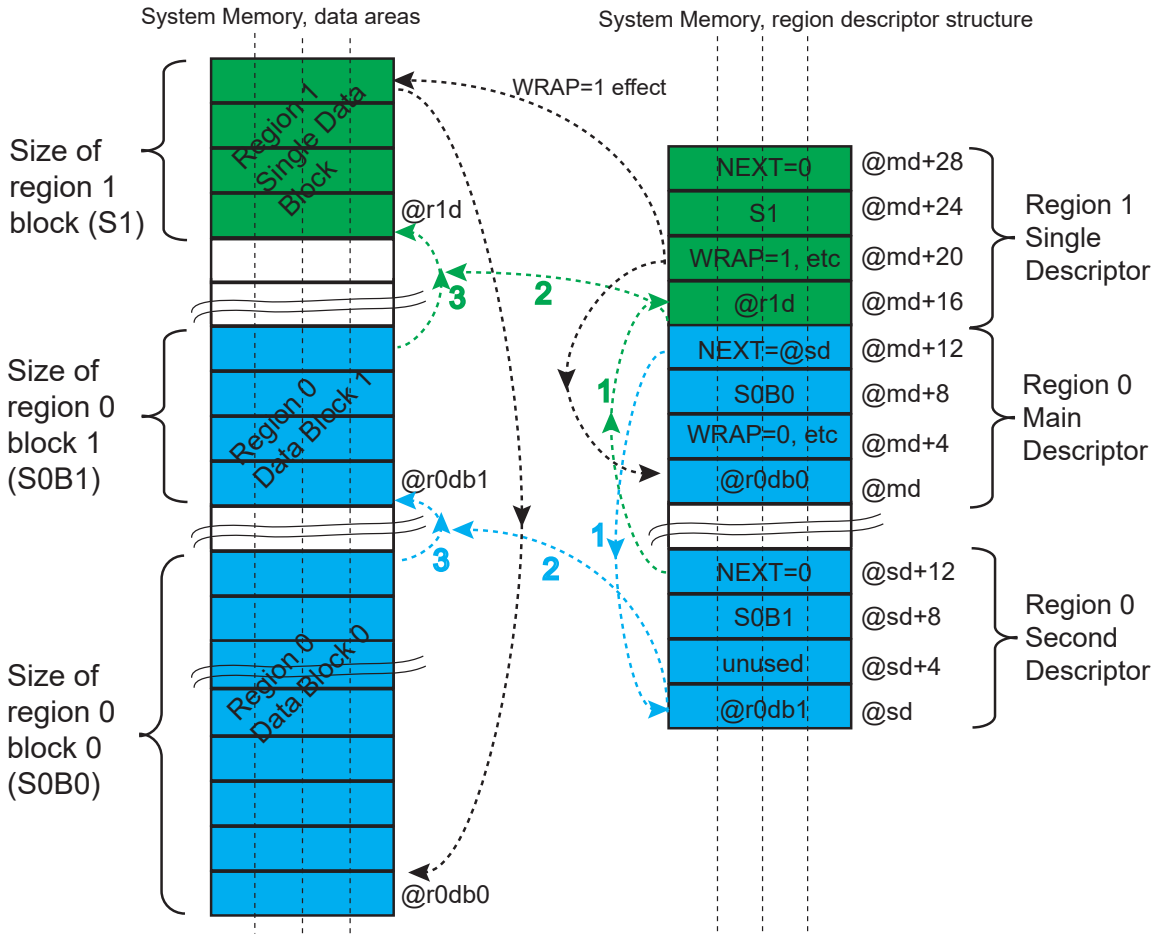
Figure 43-4. Region Descriptor



The figure below shows an example of the mandatory ICM settings required to monitor three memory data blocks of the system memory (defined as two regions) with one region being not contiguous (two separate areas) and one contiguous memory area. For each region, the SHA algorithm may be independently selected (different for each region). The wrap allows continuous monitoring.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 43-5. Example: Monitoring of 3 Memory Data Blocks (Defined as 2 Regions)



43.5.2. ICM Region Descriptor Structure

The ICM Region Descriptor Area is a contiguous area of system memory that the controller and the processor can access. When the ICM is activated, the controller performs a descriptor fetch operation at *(ICM_DSCR) address. If the Main List contains more than one descriptor (i.e., more than one region is to be monitored), the fetch address is *(ICM_DSCR) + (RID<<4) where RID is the region identifier.

Table 43-1. Region Descriptor Structure (Main List)

Offset	Structure Member	Name
ICM_DSCR+0x000+RID*(0x10)	ICM Region Start Address	ICM_RADDR
ICM_DSCR+0x004+RID*(0x10)	ICM Region Configuration	ICM_RCFG
ICM_DSCR+0x008+RID*(0x10)	ICM Region Control	ICM_RCTRL
ICM_DSCR+0x00C+RID*(0x10)	ICM Region Next Address	ICM_RNEXT

© 2026 Microchip Technology Inc. and its subsidiaries

43.5.2.1. ICM Region Start Address Structure Member

Name: ICM_RADDR
Property: Read/Write

Register offset is calculated as ICM_DSCR+0x000+RID*(0x10).

Bit	31	30	29	28	27	26	25	24
	RADDR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								

Bit	23	22	21	20	19	18	17	16
	RADDR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								

Bit	15	14	13	12	11	10	9	8
	RADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								

Bit	7	6	5	4	3	2	1	0
	RADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								

Bits 31:0 – RADDR[31:0] Region Start Address
This field indicates the first byte address of the region.

© 2026 Microchip Technology Inc. and its subsidiaries

43.5.2.2. ICM Region Configuration Structure Member

Name: ICM_RCFG
Property: Read/Write

Register offset is calculated as ICM_DSCR+0x004+RID*(0x10).

Bit	31	30	29	28	27	26	25	24
			MPROT[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		ALGO[2:0]				PROCDLY	SUIEN	ECIEN
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset								

Bit	7	6	5	4	3	2	1	0
	WCIEN	BEIEN	DMIEN	RHIEN		EOM	WRAP	CDWBN
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset								

Bits 29:24 – MPROT[5:0] Memory Region System Bus Protection
Indicates the value of the protection attribute propagated on the system bus when the ICM reads the memory region to be monitored.

Value	Description
0	The memory region system bus protection is in a user-access level region.
1	The memory region system bus protection is in a privileged-access level region.

Bits 14:12 – ALGO[2:0] SHA Algorithm
Values which are not listed in the table must be considered as “reserved”.

Value	Name	Description
0	SHA1	SHA1 algorithm processed
1	SHA256	SHA256 algorithm processed
4	SHA224	SHA224 algorithm processed

Bit 10 – PROCDLY Processing Delay
When SHA1 algorithm is processed, the runtime period is either 85 or 209 clock cycles.
When SHA256 or SHA224 algorithm is processed, the runtime period is either 72 or 194 clock cycles.

Value	Name	Description
0	SHORTEST	SHA processing runtime is the shortest one.
1	LONGEST	SHA processing runtime is the longest one.

Bit 9 – SUIEN Monitoring Status Updated Condition Interrupt (Default Enabled)

Value	Description
0	The ICM_ISR.RSU[i] flag is set when the corresponding descriptor is loaded from memory to ICM.
1	The ICM_ISR.RSU[i] flag remains cleared even if the setting condition is met.

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 8 – ECIEN End Bit Condition Interrupt (Default Enabled)

Value	Description
0	The ICM_ISR.REC[i] flag is set when the descriptor with the EOM bit set is processed.
1	The ICM_ISR.REC[i] flag remains cleared even if the setting condition is met.

Bit 7 – WCIEN Wrap Condition Interrupt Disable (Default Enabled)

Value	Description
0	The ICM_ISR.RWC[i] flag is set when the WRAP bit is set in a descriptor of the main list.
1	ICM_ISR.RWC[i] flag remains cleared even if the setting condition is met.

Bit 6 – BEIEN Bus Error Interrupt Disable (Default Enabled)

Value	Description
0	The flag is set when an error is reported on the system bus by the bus matrix.
1	The flag remains cleared even if the setting condition is met.

Bit 5 – DMIEN Digest Mismatch Interrupt Disable (Default Enabled)

Value	Description
0	The ICM_ISR.RBE[i] flag is set when the hash value just calculated from the processed region differs from expected hash value.
1	The ICM_ISR.RBE[i] flag remains cleared even if the setting condition is met.

Bit 4 – RHIEN Region Hash Completed Interrupt Disable (Default Enabled)

Value	Description
0	The ICM_ISR.RHC[i] flag is set when the field NEXT = 0 in a descriptor of the main or second list.
1	The ICM_ISR.RHC[i] flag remains cleared even if the setting condition is met.

Bit 2 – EOM End Of Monitoring

Value	Description
0	The current descriptor does not terminate the monitoring.
1	The current descriptor terminates the Main List. WRAP value has no effect.

Bit 1 – WRAP Wrap Command

Value	Description
0	The next region descriptor address loaded is the current region identifier descriptor address incremented by 0x10.
1	The next region descriptor address loaded is ICM_DSCR.

Bit 0 – CDWBN Compare Digest or Write Back Digest

Value	Description
0	The digest is written to the Hash area.
1	The digest value is compared to the digest stored in the Hash area.

© 2026 Microchip Technology Inc. and its subsidiaries

43.5.2.3. ICM Region Control Structure Member

Name: ICM_RCTRL
Property: Read/Write

Register offset is calculated as ICM_DSCR+0x008+RID*(0x10).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	TRSIZE[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								

Bit	7	6	5	4	3	2	1	0
	TRSIZE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								

Bits 15:0 – TRSIZE[15:0] Transfer Size for the Current Chunk of Data
ICM performs a transfer of (TRSIZE + 1) blocks of 512 bits.

© 2026 Microchip Technology Inc. and its subsidiaries

43.5.2.4. ICM Region Next Address Structure Member

Name: ICM_RNEXT
Property: Read/Write

Register offset is calculated as ICM_DSCR+0x00C+RID*(0x10).

Bit	31	30	29	28	27	26	25	24
	NEXT[29:22]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								

Bit	23	22	21	20	19	18	17	16
	NEXT[21:14]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								

Bit	15	14	13	12	11	10	9	8
	NEXT[13:6]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset								

Bit	7	6	5	4	3	2	1	0
	NEXT[5:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset								

Bits 31:2 – NEXT[29:0] Region Transfer Descriptor Next Address
When configured to 0, this field indicates that the current descriptor is the last descriptor of the Secondary List, otherwise it points at a new descriptor of the Secondary List.

43.5.3. ICM Hash Area

The ICM Hash Area is a contiguous area of system memory that the controller and the processor can access. The physical location is configured in the ICM hash area start address register. This address is a multiple of 128 bytes. If the CDWBN bit of the context register is cleared (i.e., Write Back activated), the ICM performs a digest write operation at the following starting location: *(ICM_HASH) + (RID<<5), where RID is the current region context identifier. If the CDWBN bit of the context register is set (i.e., Digest Comparison activated), the ICM performs a digest read operation at the same address.

43.5.3.1. Message Digest Example

Considering the following 512-bit message (example given in FIPS 180-2):

“61626380000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000018”

The message is written to memory in a Little Endian (LE) system architecture.

Table 43-2. 512-bit Message Memory Mapping

Memory Address	Address Offset / Byte Lane			
	0x3 / 31:24	0x2 / 23:16	0x1 / 15:8	0x0 / 7:0
0x000	80	63	62	61
0x004–0x038	00	00	00	00
0x03C	18	00	00	00

© 2026 Microchip Technology Inc. and its subsidiaries

The digest is stored at the memory location pointed at by the ICM_HASH pointer with a Region Offset.

Table 43-3. LE Resulting SHA-160 Message Digest Memory Mapping

Memory Address	Address Offset / Byte Lane			
	0x3 / 31:24	0x2 / 23:16	0x1 / 15:8	0x0 / 7:0
0x000	36	3e	99	a9
0x004	6a	81	06	47
0x008	71	25	3e	ba
0x00C	6c	c2	50	78
0x010	9d	d8	d0	9c

Table 43-4. Resulting SHA-224 Message Digest Memory Mapping

Memory Address	Address Offset / Byte Lane			
	0x3 / 31:24	0x2 / 23:16	0x1 / 15:8	0x0 / 7:0
0x000	22	7d	09	23
0x004	22	d8	05	34
0x008	77	a4	42	86
0x00C	b3	55	a2	bd
0x010	e4	bc	ad	2a
0x014	f7	b3	a0	bd
0x018	a7	9d	6c	e3

Table 43-5. Resulting SHA-256 Message Digest Memory Mapping

Memory Address	Address Offset / Byte Lane			
	0x3 / 31:24	0x2 / 23:16	0x1 / 15:8	0x0 / 7:0
0x000	bf	16	78	ba
0x004	ea	cf	01	8f
0x008	de	40	41	41
0x00C	23	22	ae	5d
0x010	a3	61	03	b0
0x014	9c	7a	17	96
0x018	61	ff	10	b4
0x01C	ad	15	00	f2

Considering the following 1024-bit message (example given in FIPS 180-2):

“6162638000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000018”

The message is written to memory in a Little Endian (LE) system architecture.

Table 43-6. 1024 bits Message Memory Mapping

Memory Address	Address Offset / Byte Lane			
	0x3 / 31:24	0x2 / 23:16	0x1 / 15:8	0x0 / 7:0
0x000	80	63	62	61
0x004–0x078	00	00	00	00
0x07C	18	00	00	00

© 2026 Microchip Technology Inc. and its subsidiaries

43.5.4. Using ICM as SHA Engine

The ICM can be configured to only calculate a SHA1, SHA224, SHA256 digest value.

43.5.4.1. Settings for Simple SHA Calculation

The start address of the system memory containing the data to hash must be configured in the transfer descriptor of the DMA embedded in the ICM.

The transfer descriptor is a system memory area integer multiple of 4 x 32-bit words and the start address of the descriptor must be configured in ICM_DSCR (the start address must be aligned on 64-bytes; six LSB must be cleared). If the data to hash is already padded according to SHA standards, only a single descriptor is required, and ICM_RCFG.EOM must be written to 1. If the data to hash does not contain a padding area, it is possible to define the padding area in another system memory location, the ICM can be configured to automatically jump from a memory area to another one by configuring the descriptor register ICM_RNEXT with a value that differs from 0. Configuring ICM_RNEXT.NEXT with the start address of the padding area forces the ICM to concatenate both areas, thus providing the SHA result from the start address of the hash area configured in ICM_HASH.

Whether the system memory is configured as a single or multiple data block area, ICM_RCFG.CDWBN and ICM_RCFG.WRAP must be cleared. The bits WBDIS, EOMDIS, SLBDIS must be cleared in ICM_CFG.

ICM_RCTRL.RHIEN and ICM_RCTRL.ECIEN must be written to 1. The flag RHC[i], i being the region index, is set (if RHIEN is set) when the hash result is available at address defined in ICM_HASH. The flag REC[i], i being the region index, is set (if ECIEN is set) when the hash result is available at the address defined in ICM_HASH.

An interrupt is generated if the bit RHC[i] is written to 1 in the ICM_IER (if RHC[i] is set in ICM_RCTRL of region i) or if the bit REC[i] is written to 1 in the ICM_IER (if REC[i] is set in ICM_RCTRL of region i).

43.5.4.2. Processing Period

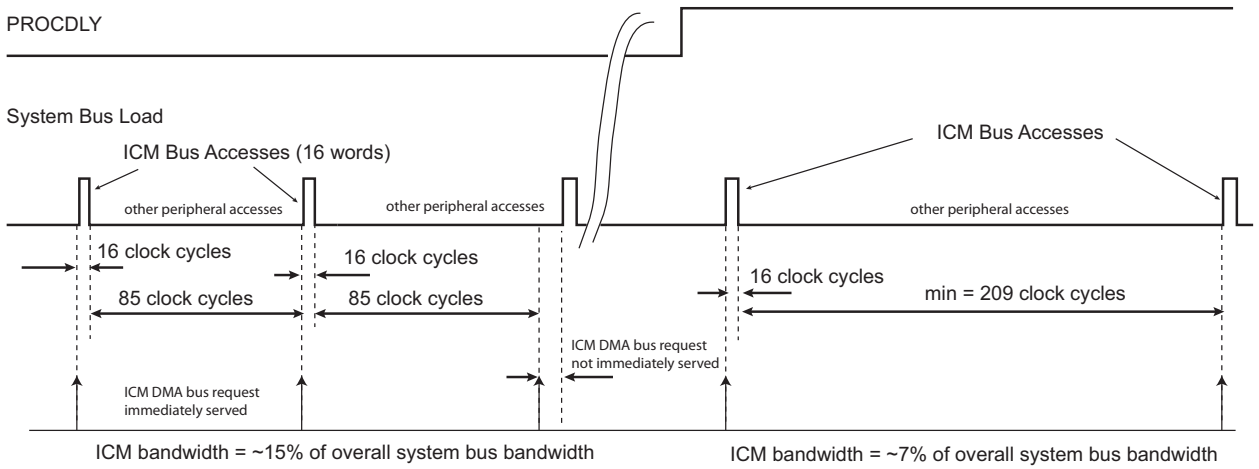
The ICM engine has a core (SHA) inherent processing period that may result, depending on the application, in a significant bandwidth usage at system bus level. In some applications, it may be important to keep as much bandwidth as possible for the other peripherals (e.g. CPU, DMA). The ICM SHA engine processing period can be configured to reduce the bandwidth required by writing ICM_RCFG.PROCDLY=1.

In SHA1 mode, the shortest processing period is 85 clock cycles + 2 clock cycles for start command synchronization (ICM_RCFG.PROCDLY=0). The longest period is 209 clock cycles + 2 clock cycles when ICM_RCFG.PROCDLY=1 (see the figure below).

In SHA256 or SHA224 mode, the shortest processing period is 72 clock cycles + 2 clock cycles for start command synchronization. The longest period is 194 clock cycles + 2 clock cycles.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 43-6. Bandwidth Usage in SHA1 Mode



43.5.5. ICM Automatic Monitoring Mode

ICM_CFG.ASCD is used to activate the ICM Automatic Monitoring mode. When ICM_CFG.ASCD is set and bits CDWBN and EOM in ICM.RCFG equal 0, the ICM performs the following actions:

1. The ICM passes through the Main List once to calculate the message digest of the monitored area.
2. When WRAP = 1 in ICM_RCFG, the ICM begins monitoring. CDWBN in ICM_RCFG is now automatically set and EOM is cleared. These bits have no effect during the monitoring period that ends when EOM is set.

43.5.6. Programming the ICM

Table 43-7. Region Attributes

Transfer Type		Main List	ICM_RCFG			ICM_RNEXT	Comments
			CDWBN	WRAP	EOM	NEXT	
Single Region	Contiguous list of blocks Digest written to memory Monitoring disabled	1 item	0	0	1	0	The Main List contains only one descriptor. The Secondary List is empty for that descriptor. The digest is computed and saved to memory.
	Non-contiguous list of blocks Digest written to memory Monitoring disabled	1 item	0	0	1	Secondary List address of the current region identifier	The Main List contains only one descriptor. The Secondary List describes the layout of the non-contiguous region.
	Contiguous list of blocks Digest comparison enabled Monitoring enabled	1 item	1	1	0	0	When the hash computation is terminated, the digest is compared with the one saved in memory.

© 2026 Microchip Technology Inc. and its subsidiaries

Table 43-7. Region Attributes (continued)

Transfer Type		Main List	ICM_RCFG			ICM_RNEXT	Comments
			CDWBN	WRAP	EOM	NEXT	
Multiple Regions	Contiguous list of blocks Digest written to memory Monitoring disabled	More than one item	0	0	1 for the last, 0 otherwise	0	ICM passes through the list once.
	Contiguous list of blocks Digest comparison is enabled Monitoring is enabled	More than one item	1	1 for the last, 0 otherwise	0	0	ICM performs active monitoring of the regions. If a mismatch occurs, an interrupt is raised.
	Non-contiguous list of blocks Digest is written to memory Monitoring is disabled	More than one item	0	0	1	Secondary List address	ICM performs hashing and saves digests to the Hash area.
	Non-contiguous list of blocks Digest comparison is enabled Monitoring is enabled	More than one item	1	1	0	Secondary List address	ICM performs data gathering on a per region basis.

43.5.7. Security Features

When an undefined register access occurs, the URAD bit in the Interrupt Status Register (ICM_ISR) is set if unmasked. Its source is then reported in the Undefined Access Status Register (ICM_UASR). Only the first undefined register access is available through the ICM_UASR.URAT field.

Several kinds of unspecified register accesses can occur:

- Unspecified structure member set to one detected when the descriptor is loaded
- Configuration register (ICM_CFG) modified during active monitoring
- Descriptor register (ICM_DSCR) modified during active monitoring
- Hash register (ICM_HASH) modified during active monitoring
- Write-only register read access

The URAD bit and the URAT field can only be reset by writing a 1 to the ICM_CTRL.SWRST bit.

© 2026 Microchip Technology Inc. and its subsidiaries

43.6. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	ICM_CFG	31:24			DAPROT[5:0]					
		23:16			HAPROT[5:0]					
		15:8	UALGO[2:0]			UIHASH			DUALBUFF	ASCD
		7:0	BBC[3:0]					SLBDIS	EOMDIS	WBDIS
0x04	ICM_CTRL	31:24								
		23:16								
		15:8	RMEN[3:0]				RMDIS[3:0]			
		7:0	REHASH[3:0]					SWRST	DISABLE	ENABLE
0x08	ICM_SR	31:24								
		23:16								
		15:8	RMDIS[3:0]				RAWRMDIS[3:0]			
		7:0								ENABLE
0x0C ... 0x0F	Reserved									
0x10	ICM_IER	31:24								URAD
		23:16	RSU[3:0]				REC[3:0]			
		15:8	RWC[3:0]				RBE[3:0]			
		7:0	RDM[3:0]				RHC[3:0]			
0x14	ICM_IDR	31:24								URAD
		23:16	RSU[3:0]				REC[3:0]			
		15:8	RWC[3:0]				RBE[3:0]			
		7:0	RDM[3:0]				RHC[3:0]			
0x18	ICM_IMR	31:24								URAD
		23:16	RSU[3:0]				REC[3:0]			
		15:8	RWC[3:0]				RBE[3:0]			
		7:0	RDM[3:0]				RHC[3:0]			
0x1C	ICM_ISR	31:24								URAD
		23:16	RSU[3:0]				REC[3:0]			
		15:8	RWC[3:0]				RBE[3:0]			
		7:0	RDM[3:0]				RHC[3:0]			
0x20	ICM_UASR	31:24								
		23:16								
		15:8								
		7:0						URAT[2:0]		
0x24 ... 0x2F	Reserved									
0x30	ICM_DSCR	31:24	DASA[25:18]							
		23:16	DASA[17:10]							
		15:8	DASA[9:2]							
		7:0	DASA[1:0]							
0x34	ICM_HASH	31:24	HASA[24:17]							
		23:16	HASA[16:9]							
		15:8	HASA[8:1]							
		7:0	HASA[0]							
0x38	ICM_UIHVAL0	31:24	VAL[31:24]							
		23:16	VAL[23:16]							
		15:8	VAL[15:8]							
		7:0	VAL[7:0]							
0x3C	ICM_UIHVAL1	31:24	VAL[31:24]							
		23:16	VAL[23:16]							
		15:8	VAL[15:8]							
		7:0	VAL[7:0]							
0x40	ICM_UIHVAL2	31:24	VAL[31:24]							
		23:16	VAL[23:16]							
		15:8	VAL[15:8]							
		7:0	VAL[7:0]							

© 2026 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x44	ICM_UIHVAL3	31:24	VAL[31:24]							
		23:16	VAL[23:16]							
		15:8	VAL[15:8]							
		7:0	VAL[7:0]							
0x48	ICM_UIHVAL4	31:24	VAL[31:24]							
		23:16	VAL[23:16]							
		15:8	VAL[15:8]							
		7:0	VAL[7:0]							
0x4C	ICM_UIHVAL5	31:24	VAL[31:24]							
		23:16	VAL[23:16]							
		15:8	VAL[15:8]							
		7:0	VAL[7:0]							
0x50	ICM_UIHVAL6	31:24	VAL[31:24]							
		23:16	VAL[23:16]							
		15:8	VAL[15:8]							
		7:0	VAL[7:0]							
0x54	ICM_UIHVAL7	31:24	VAL[31:24]							
		23:16	VAL[23:16]							
		15:8	VAL[15:8]							
		7:0	VAL[7:0]							

© 2026 Microchip Technology Inc. and its subsidiaries

43.6.1. ICM Configuration Register

Name: ICM_CFG
Offset: 0x00
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
			DAPROT[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
			HAPROT[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	UALGO[2:0]			UIHASH			DUALBUFF	ASCD
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	0	0	0			0	0

Bit	7	6	5	4	3	2	1	0
	BBC[3:0]					SLBDIS	EOMDIS	WBDIS
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bits 29:24 – DAPROT[5:0] Region Descriptor Area Protection
Indicates the value of the protection attribute propagated on the system bus when the ICM accesses the Descriptor Area.

Value	Description
0	The region descriptor area is in a user-access level region.
1	The region descriptor area is in a privileged-access level region.

Bits 21:16 – HAPROT[5:0] Region Hash Area Protection
Indicates the value of the protection attribute propagated on the system bus when the ICM accesses the Hash Area.

Value	Description
0	The region hash area is in a user-access level region.
1	The region hash area is in a privileged-access level region.

Bits 15:13 – UALGO[2:0] User SHA Algorithm

Value	Name	Description
0	SHA1	SHA1 algorithm processed
1	SHA256	SHA256 algorithm processed
4	SHA224	SHA224 algorithm processed

Bit 12 – UIHASH User Initial Hash Value

Value	Description
0	The secure hash standard provides the initial hash value.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
1	The initial hash value is programmable. Field UALGO provides the SHA algorithm. The ALGO field of the ICM_RCFG structure member has no effect.

Bit 9 – DUALBUFF Dual Input Buffer

Value	Description
0	Dual Input Buffer mode is disabled.
1	Dual Input Buffer mode is enabled (better performances, higher bandwidth required on system bus).

Bit 8 – ASCD Automatic Switch To Compare Digest

Value	Description
0	Automatic monitoring mode is disabled.
1	The ICM passes through the Main List once to calculate the message digest of the monitored area. When WRAP = 1 in ICM_RCFG, the ICM begins monitoring.

Bits 7:4 – BBC[3:0] Bus Burden Control

This field is used to control the burden of the ICM system bus. The number of system clock cycles between the end of the current processing and the next block transfer is set to 2^{BBC}. Up to 32,768 cycles can be inserted.

Bit 2 – SLBDIS Secondary List Branching Disable

Value	Description
0	Branching to the Secondary List is permitted.
1	Branching to the Secondary List is forbidden. The NEXT field of the ICM_RNEXT structure member has no effect and is always considered as zero.

Bit 1 – EOMDIS End of Monitoring Disable

Value	Description
0	End of Monitoring is permitted.
1	End of Monitoring is forbidden. The EOM bit of the ICM_RCFG structure member has no effect.

Bit 0 – WBDIS Write Back Disable

When ASCD is set, WBDIS has no effect.

Value	Description
0	Write Back operations are permitted.
1	Write Back operations are forbidden. Context register CDWBN bit is internally set to one and cannot be modified by a linked list element. ICM_RCFG.CDWBN has no effect.

© 2026 Microchip Technology Inc. and its subsidiaries

43.6.2. ICM Control Register

Name: ICM_CTRL
Offset: 0x04
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	RMEN[3:0]				RMDIS[3:0]			
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	REHASH[3:0]					SWRST	DISABLE	ENABLE
Access	W	W	W	W		W	W	W
Reset	–	–	–	–		–	–	–

Bits 15:12 – RMEN[3:0] Region Monitoring Enable
Monitoring is activated by default.

Value	Description
0	No effect
1	When bit RMEN[i] is set to one, the monitoring of region with identifier i is activated.

Bits 11:8 – RMDIS[3:0] Region Monitoring Disable

Value	Description
0	No effect
1	When bit RMDIS[i] is set to one, the monitoring of region with identifier i is disabled.

Bits 7:4 – REHASH[3:0] Recompute Internal Hash

Value	Description
0	No effect
1	When REHASH[i] is set to one, Region i digest is re-computed. This bit is only available when region monitoring is disabled.

Bit 2 – SWRST Software Reset

Value	Description
0	No effect
1	Resets the ICM.

Bit 1 – DISABLE ICM Disable Register

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	No effect
1	The ICM is disabled. If a region is active, this region is terminated.

Bit 0 – ENABLE ICM Enable

Value	Description
0	No effect
1	When set to one, the ICM is activated.

© 2026 Microchip Technology Inc. and its subsidiaries

43.6.3. ICM Status Register

Name: ICM_SR
Offset: 0x08
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	RMDIS[3:0]				RAWRMDIS[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								ENABLE
Access								R
Reset								0

Bits 15:12 – RMDIS[3:0] Region Monitoring Disabled Status

Value	Description
0	Region i is being monitored (occurs after integrity check value has been calculated and written to Hash area).
1	Region i monitoring is not being monitored.

Bits 11:8 – RAWRMDIS[3:0] Region Monitoring Disabled Raw Status

Value	Description
0	Region i monitoring has been activated by writing a 1 in RMEN[i] of ICM_CTRL.
1	Region i monitoring has been deactivated by writing a 1 in RMDIS[i] of ICM_CTRL.

Bit 0 – ENABLE ICM Enable Register

Value	Description
0	ICM is disabled.
1	ICM is activated.

© 2026 Microchip Technology Inc. and its subsidiaries

43.6.4. ICM Interrupt Enable Register

Name: ICM_IER
Offset: 0x10
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
								URAD
Access								W
Reset								–

Bit	23	22	21	20	19	18	17	16
	RSU[3:0]				REC[3:0]			
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	RWC[3:0]				RBE[3:0]			
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	RDM[3:0]				RHC[3:0]			
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 24 – URAD Undefined Register Access Detection Interrupt Enable

Value	Description
0	No effect.
1	The Undefined Register Access interrupt is enabled.

Bits 23:20 – RSU[3:0] Region Status Updated Interrupt Disable

Value	Description
0	No effect.
1	When RSU[i] is set to one, the region i Status Updated interrupt is enabled.

Bits 19:16 – REC[3:0] Region End bit Condition Detected Interrupt Enable

Value	Description
0	No effect.
1	When REC[i] is set to one, the region i End bit Condition interrupt is enabled.

Bits 15:12 – RWC[3:0] Region Wrap Condition detected Interrupt Enable

Value	Description
0	No effect.
1	When RWC[i] is set to one, the Region i Wrap Condition interrupt is enabled.

Bits 11:8 – RBE[3:0] Region Bus Error Interrupt Enable

Value	Description
0	No effect.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
1	When RBE[i] is set to one, the Region i Bus Error interrupt is enabled.

Bits 7:4 – RDM[3:0] Region Digest Mismatch Interrupt Enable

Value	Description
0	No effect.
1	When RDM[i] is set to one, the Region i Digest Mismatch interrupt is enabled.

Bits 3:0 – RHC[3:0] Region Hash Completed Interrupt Enable

Value	Description
0	No effect.
1	When RHC[i] is set to one, the Region i Hash Completed interrupt is enabled.

© 2026 Microchip Technology Inc. and its subsidiaries

43.6.5. ICM Interrupt Disable Register

Name: ICM_IDR
Offset: 0x14
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
								URAD
Access								W
Reset								–

Bit	23	22	21	20	19	18	17	16
	RSU[3:0]				REC[3:0]			
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	RWC[3:0]				RBE[3:0]			
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	RDM[3:0]				RHC[3:0]			
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 24 – URAD Undefined Register Access Detection Interrupt Disable

Value	Description
0	No effect.
1	Undefined Register Access Detection interrupt is disabled.

Bits 23:20 – RSU[3:0] Region Status Updated Interrupt Disable

Value	Description
0	No effect.
1	When RSU[i] is set to one, the region i Status Updated interrupt is disabled.

Bits 19:16 – REC[3:0] Region End bit Condition detected Interrupt Disable

Value	Description
0	No effect.
1	When REC[i] is set to one, the region i End bit Condition interrupt is disabled.

Bits 15:12 – RWC[3:0] Region Wrap Condition Detected Interrupt Disable

Value	Description
0	No effect.
1	When RWC[i] is set to one, the Region i Wrap Condition interrupt is disabled.

Bits 11:8 – RBE[3:0] Region Bus Error Interrupt Disable

Value	Description
0	No effect.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
1	When RBE[i] is set to one, the Region i Bus Error interrupt is disabled.

Bits 7:4 – RDM[3:0] Region Digest Mismatch Interrupt Disable

Value	Description
0	No effect.
1	When RDM[i] is set to one, the Region i Digest Mismatch interrupt is disabled.

Bits 3:0 – RHC[3:0] Region Hash Completed Interrupt Disable

Value	Description
0	No effect.
1	When RHC[i] is set to one, the Region i Hash Completed interrupt is disabled.

© 2026 Microchip Technology Inc. and its subsidiaries

43.6.6. ICM Interrupt Mask Register

Name: ICM_IMR
Offset: 0x18
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
								URAD
Access								R
Reset								0

Bit	23	22	21	20	19	18	17	16
	RSU[3:0]				REC[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RWC[3:0]				RBE[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RDM[3:0]				RHC[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 24 – URAD Undefined Register Access Detection Interrupt Mask

Value	Description
0	Interrupt is disabled
1	Interrupt is enabled.

Bits 23:20 – RSU[3:0] Region Status Updated Interrupt Mask

Value	Description
0	When RSU[i] is set to zero, the interrupt is disabled for region i.
1	When RSU[i] is set to one, the interrupt is enabled for region i.

Bits 19:16 – REC[3:0] Region End Bit Condition Detected Interrupt Mask

Value	Description
0	When REC[i] is set to zero, the interrupt is disabled for region i.
1	When REC[i] is set to one, the interrupt is enabled for region i.

Bits 15:12 – RWC[3:0] Region Wrap Condition Detected Interrupt Mask

Value	Description
0	When RWC[i] is set to zero, the interrupt is disabled for region i.
1	When RWC[i] is set to one, the interrupt is enabled for region i.

Bits 11:8 – RBE[3:0] Region Bus Error Interrupt Mask

Value	Description
0	When RBE[i] is set to zero, the interrupt is disabled for region i.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
1	When RBE[i] is set to one, the interrupt is enabled for region i.

Bits 7:4 – RDM[3:0] Region Digest Mismatch Interrupt Mask

Value	Description
0	When RDM[i] is set to zero, the interrupt is disabled for region i.
1	When RDM[i] is set to one, the interrupt is enabled for region i.

Bits 3:0 – RHC[3:0] Region Hash Completed Interrupt Mask

Value	Description
0	When RHC[i] is set to zero, the interrupt is disabled for region i.
1	When RHC[i] is set to one, the interrupt is enabled for region i.

© 2026 Microchip Technology Inc. and its subsidiaries

43.6.7. ICM Interrupt Status Register

Name: ICM_ISR
Offset: 0x1C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
								URAD
Access								R
Reset								0

Bit	23	22	21	20	19	18	17	16
	RSU[3:0]				REC[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RWC[3:0]				RBE[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RDM[3:0]				RHC[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 24 – URAD Undefined Register Access Detection Status
The URAD bit is only reset by the SWRST bit in ICM_CTRL.
The URAT field in ICM_UASR indicates the unspecified access type.

Value	Description
0	No undefined register access has been detected since the last SWRST.
1	At least one undefined register access has been detected since the last SWRST.

Bits 23:20 – RSU[3:0] Region Status Updated Detected
When RSU[i] is set, it indicates that a region status updated condition has been detected.

Bits 19:16 – REC[3:0] Region End Bit Condition Detected
When REC[i] is set, it indicates that an end bit condition has been detected.

Bits 15:12 – RWC[3:0] Region Wrap Condition Detected
When RWC[i] is set, it indicates that a wrap condition has been detected.

Bits 11:8 – RBE[3:0] Region Bus Error
When RBE[i] is set, it indicates that a bus error has been detected while hashing memory region i.

Bits 7:4 – RDM[3:0] Region Digest Mismatch
When RDM[i] is set, it indicates that there is a digest comparison mismatch between the hash value of the region with identifier i and the reference value located in the Hash Area.

Bits 3:0 – RHC[3:0] Region Hash Completed
When RHC[i] is set, it indicates that the ICM has completed the region with identifier i.

© 2026 Microchip Technology Inc. and its subsidiaries

43.6.8. ICM Undefined Access Status Register

Name: ICM_UASR
Offset: 0x20
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						URAT[2:0]		
Access						R	R	R
Reset						0	0	0

Bits 2:0 – URAT[2:0] Undefined Register Access Trace
Only the first Undefined Register Access Trace is available through the URAT field.
The URAT field is only reset by the SWRST bit in the ICM_CTRL register.

Value	Name	Description
0	UNSPEC_STRUCT_MEMBER	Unspecified structure member set to one detected when the descriptor is loaded.
1	ICM_CFG_MODIFIED	ICM_CFG modified during active monitoring.
2	ICM_DSCR_MODIFIED	ICM_DSCR modified during active monitoring.
3	ICM_HASH_MODIFIED	ICM_HASH modified during active monitoring
4	READ_ACCESS	Write-only register read access

© 2026 Microchip Technology Inc. and its subsidiaries

43.6.9. ICM Descriptor Area Start Address Register

Name: ICM_DSCR
Offset: 0x30
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	DASA[25:18]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DASA[17:10]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DASA[9:2]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DASA[1:0]							
Access	R/W	R/W						
Reset	0	0						

Bits 31:6 – DASA[25:0] Descriptor Area Start Address
The start address is a multiple of the total size of the data structure (64 bytes).

© 2026 Microchip Technology Inc. and its subsidiaries

43.6.10. ICM Hash Area Start Address Register

Name: ICM_HASH
Offset: 0x34
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	HASA[24:17]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	HASA[16:9]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	HASA[8:1]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	HASA[0]							
Access	R/W							
Reset	0							

Bits 31:7 – HASA[24:0] Hash Area Start Address
This field points at the Hash memory location. The address must be a multiple of 128 bytes.

© 2026 Microchip Technology Inc. and its subsidiaries

43.6.11. ICM User Initial Hash Value Register

Name: ICM_UIHVALx
Offset: 0x38 + x*0x04 [x=0..7]
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	VAL[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	VAL[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	VAL[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	VAL[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – VAL[31:0] Initial Hash Value
When ICM_CFG.UIHASH is set, the Initial Hash Value is user-programmable.
To meet the desired standard, use the following example values.
For ICM_UIHVAL0 field:

Example	Comment
0x67452301	SHA1 algorithm
0xC1059ED8	SHA224 algorithm
0x6A09E667	SHA256 algorithm

For ICM_UIHVAL1 field:

Example	Comment
0xEFCDAB89	SHA1 algorithm
0x367CD507	SHA224 algorithm
0xBB67AE85	SHA256 algorithm

For ICM_UIHVAL2 field:

Example	Comment
0x98BADCFE	SHA1 algorithm
0x3070DD17	SHA224 algorithm
0x3C6EF372	SHA256 algorithm

For ICM_UIHVAL3 field:

Example	Comment
0x10325476	SHA1 algorithm
0xF70E5939	SHA224 algorithm

© 2026 Microchip Technology Inc. and its subsidiaries

VAL (continued)	
Example	**Comment**
0xA54FF53A	SHA256 algorithm

For ICM_UIHVAL4 field:

Example	Comment
0xC3D2E1F0	SHA1 algorithm
0xFFC00B31	SHA224 algorithm
0x510E527F	SHA256 algorithm

For ICM_UIHVAL5 field:

Example	Comment
0x68581511	SHA224 algorithm
0x9B05688C	SHA256 algorithm

For ICM_UIHVAL6 field:

Example	Comment
0x64F98FA7	SHA224 algorithm
0x1F83D9AB	SHA256 algorithm

For ICM_UIHVAL7 field:

Example	Comment
0xBEFA4FA4	SHA224 algorithm
0x5BE0CD19	SHA256 algorithm

Example of Initial Value for SHA-1 Algorithm

Register Address	Address Offset / Byte Lane			
	0x3 / 31:24	0x2 / 23:16	0x1 / 15:8	0x0 / 7:0
0x000 ICM_UIHVAL0	01	23	45	67
0x004 ICM_UIHVAL1	89	ab	cd	ef
0x008 ICM_UIHVAL2	fe	dc	ba	98
0x00C ICM_UIHVAL3	76	54	32	10
0x010 ICM_UIHVAL4	f0	e1	d2	c3

© 2026 Microchip Technology Inc. and its subsidiaries

44. Classical Public Key Cryptography Controller (CPKCC)

44.1. Description

The Classical Public Key Cryptography Controller (CPKCC) processes public key cryptography algorithm calculus in both GF(p) and GF(2^n) fields. The ROMed CPKCL, the Classical Public Key Cryptography Library, is the library built on the top of the CPKCC.

The Classical Public Key Cryptography Library includes complete implementation of the following public key cryptography algorithms:

- RSA, DSA
  - Modular exponentiation with CRT up to 7168 bits
  - Modular exponentiation without CRT up to 5376 bits
  - Prime generation
  - Utilities: GCD/Modular Inverse, Divide, Modular reduction, Multiply, etc.
- Elliptic Curves
  - ECDSA GF(p) up to 1504 bits
  - ECDSA GF(2^n) up to 1440 bits
  - Point multiply
  - Point add/doubling
  - Elliptic curves in GF(p) or GF(2^n)
  - Choice of the curves parameters so compatibility with NIST curves or others.
  - Deterministic Random Number Generation (DRNG ANSI X9.31) for DSA

44.2. Product Dependencies

44.2.1. Power Management

The CPKCC is not continuously clocked. The CPCKCC interface is clocked through the Power Management Controller (PMC).

44.2.2. Interrupt Sources

The CPKCC has an interrupt line connected to the Nested Vector Interrupt Controller (NVIC). Handling interrupts requires programming the NVIC before configuring the CPKCC.

44.3. Functional Description

The CPKCC macrocell is managed by the CPKCL library available in the ROM memory. The user interface of the CPKCC is not described in this chapter.

The usage description of the CPKCC and its associated library is provided in a separate document. Contact a Microchip Sales Representative for further details.

© 2026 Microchip Technology Inc. and its subsidiaries

45. Multi Channel Serial Peripheral Interface (MCSPI)

45.1. Description

The Multi Channel Serial Peripheral Interface (MCSPI) circuit is a full-featured SPI module with specific enhancements to ease interfacing with external devices such as MCP3910 24-bit ADC.

A client device is selected when the host asserts its NSS signal. If multiple client devices exist, the host generates a separate client select signal for each client (NPCS).

The SPI system consists of two data lines and two control lines:

- Host Out Client In (MOSI)—This data line supplies the output data from the host shifted into the input(s) of the client(s).
- Host In Client Out (MISO)—This data line supplies the output data from a client to the input of the host. There may be no more than one client transmitting data during any particular transfer.
- Serial Clock (SPCK)—This control line is driven by the host and regulates the flow of the data bits. The host can transmit data at a variety of baud rates; there is one SPCK pulse for each bit that is transmitted.
- Client Select (NSS)—This control line allows clients to be turned on and off by hardware.

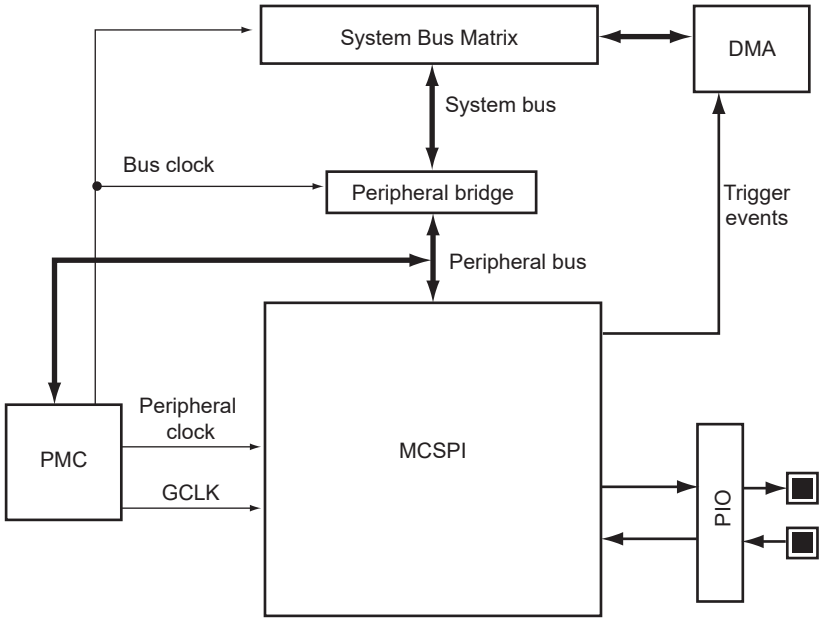
45.2. Embedded Characteristics

- Host or Client Serial Peripheral Bus Interface
  - 8-bit to 16-bit programmable data length per chip select
  - Programmable phase and polarity per chip select
  - Programmable transfer delay between consecutive transfers and delay before MCSPI clock per chip select
  - Programmable delay between chip selects
  - Selectable mode fault detection
- Two-Pin Mode Support For Easy Link with External MCP3910 ADC and CRC Field Support
- Host Mode can Drive SPCK up to Peripheral Clock
- 8-data Transmit and Receive FIFOs
- Host Mode Bit Rate can be Independent of the Processor/Peripheral Clock
- Client Mode Operates on SPCK, Asynchronously with Core and Bus Clock
- Four Chip Selects with External Decoder Support Allow Communication with up to 15 Peripherals
- Communication with Serial External Devices Supported
  - Serial memories, such as DataFlash and 3-wire EEPROMs
  - Serial peripherals, such as ADCs, DACs, LCD controllers, CAN controllers and sensors
  - External coprocessors
- Connection to PDC Channel Capabilities, Optimizing Data Transfers
  - One channel for the receiver
  - One channel for the transmitter
- Register Write Protection

© 2026 Microchip Technology Inc. and its subsidiaries

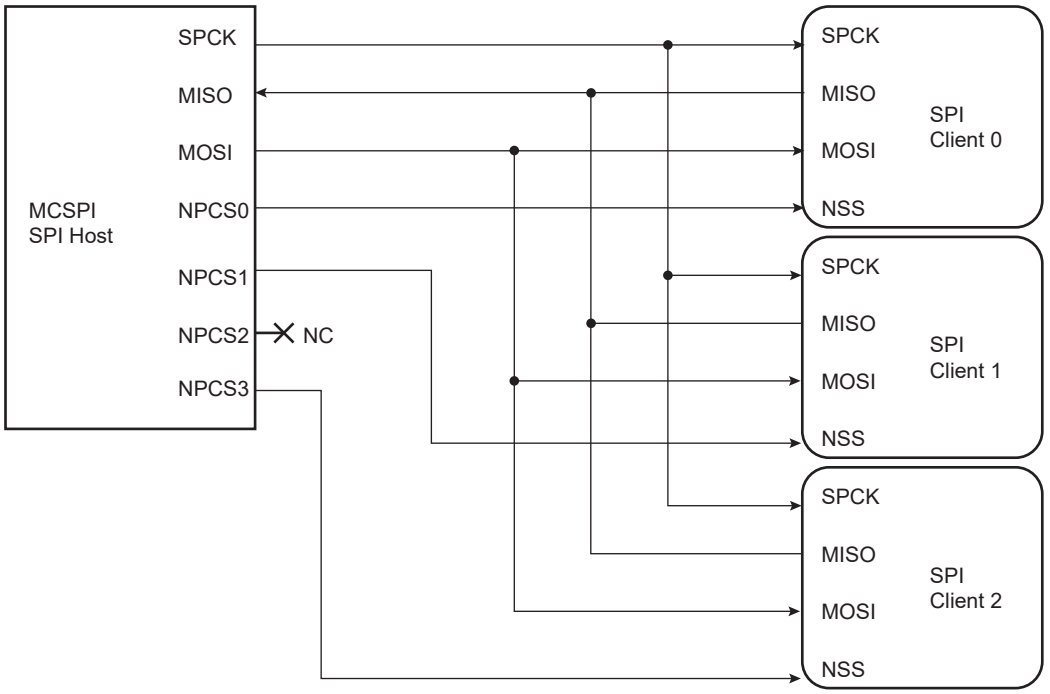
45.3. Block Diagram

Figure 45-1. MCSPI Block Diagram



45.4. Application Block Diagram

Figure 45-2. MCSPI Application Block Diagram: Single Host/Multiple Client Implementation



© 2026 Microchip Technology Inc. and its subsidiaries

45.5. Signal Description

Table 45-1. Signal Description

Pin Name	Pin Description	Type	
		Host	Client
MISO	Host In Client Out	Input	Output
MOSI0–MOSI3	Host Out Client In	Output	Input
SPCK	Serial Clock	Output	Input
NPCS1–NPCS3	Peripheral Chip Selects	Output	Unused
NPCS0/NSS	Peripheral Chip Select/Client Select	Output	Input

45.6. Product Dependencies

45.6.1. I/O Lines

The pins used for interfacing the compliant external devices can be multiplexed with PIO lines. The programmer must first program the PIO controllers to assign the MCSPI pins to their peripheral functions.

45.6.2. Power Management

The MCSPI can be clocked through the Power Management Controller (PMC), thus the programmer must first configure the PMC to enable the MCSPI clock.

45.6.3. Interrupt

The MCSPI has an interrupt line connected to the interrupt controller. Handling the MCSPI interrupt requires programming the interrupt controller before configuring the MCSPI.

45.6.4. Peripheral DMA Controller (PDC)

The MCSPI can be used in conjunction with the PDC in order to reduce processor overhead. For a full description of the PDC, refer to the relevant section.

45.7. Functional Description

45.7.1. Modes of Operation

The MCSPI operates in Host mode or in Client mode.

- The MCSPI operates in Host mode by setting the Host/Client Mode (MSTR) bit in the MCSPI Mode register (MCSPI_MR):
  - Pins NPCS0 to NPCS3 are all configured as outputs.
  - The SPCK pin is driven.
  - The MISO line is wired on the receiver input.
  - The MOSI line is driven as an output by the transmitter.
- The MCSPI operates in Client mode if the MCSPI_MR.MSTR bit is written to ‘0’:
  - The MISO line is driven by the transmitter output.
  - The MOSI line is wired on the receiver input.
  - The SPCK pin is driven by the transmitter to synchronize the receiver.
  - The NPCS0 pin becomes an input, and is used as a client select signal (NSS).
  - The NPCS1 to NPCS3 pins are not driven and can be used for other purposes.

The data transfers are identically programmable for both modes of operation. The baud rate generator is activated only in Host mode.

© 2026 Microchip Technology Inc. and its subsidiaries

45.7.2. Data Transfer

Four combinations of polarity and phase are available for data transfers. The clock polarity is programmed with the Clock Polarity (CPOL) bit in the MCSPI Chip Select registers (MCSPI_CSRx). The clock phase is programmed with the Clock Phase (NCPHA) bit in MCSPI_CSRx. These two parameters determine the edges of the clock signal on which data is driven and sampled. Each of the two parameters has two possible states, resulting in four possible combinations that are incompatible with one another. Consequently, a host/client pair must use the same parameter pair values to communicate. If multiple clients are connected and require different configurations, the host must reconfigure itself each time it needs to communicate with a different client.

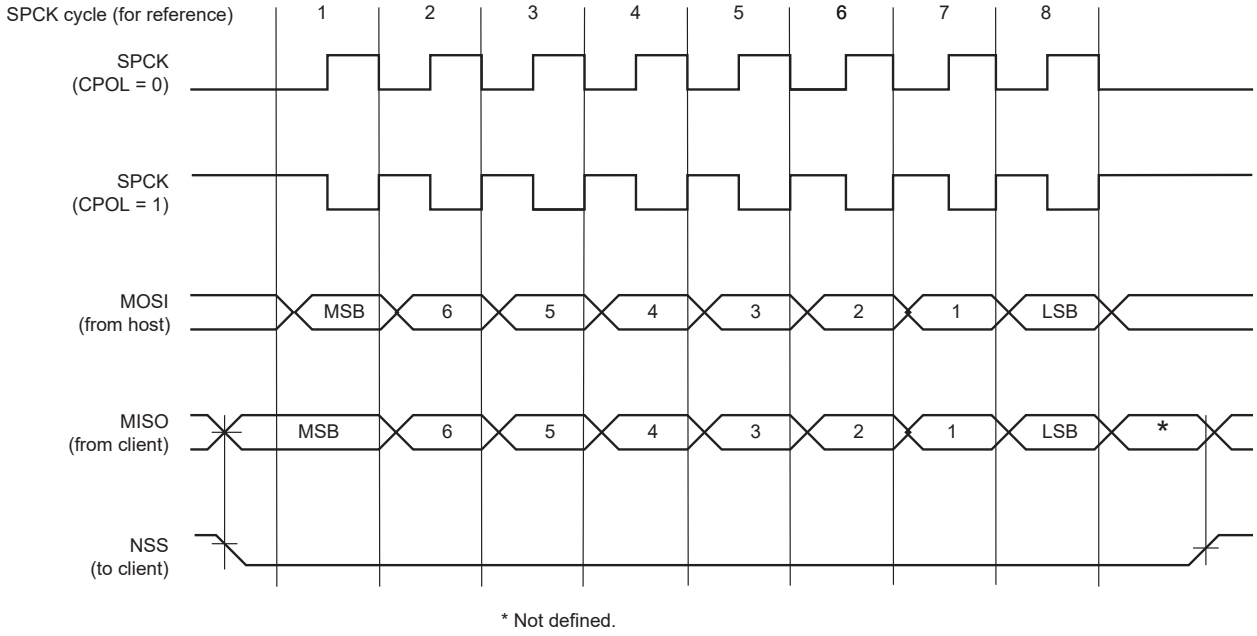
The following table shows the four modes and corresponding parameter settings.

Table 45-2. MCSPI Bus Protocol Modes

MCSPI Mode	CPOL	NCPHA	Shift SPCK Edge	Capture SPCK Edge	SPCK Inactive Level
0	0	1	Falling	Rising	Low
1	0	0	Rising	Falling	Low
2	1	1	Rising	Falling	High
3	1	0	Falling	Rising	High

Figure 45-3 and Figure 45-4 show examples of data transfers.

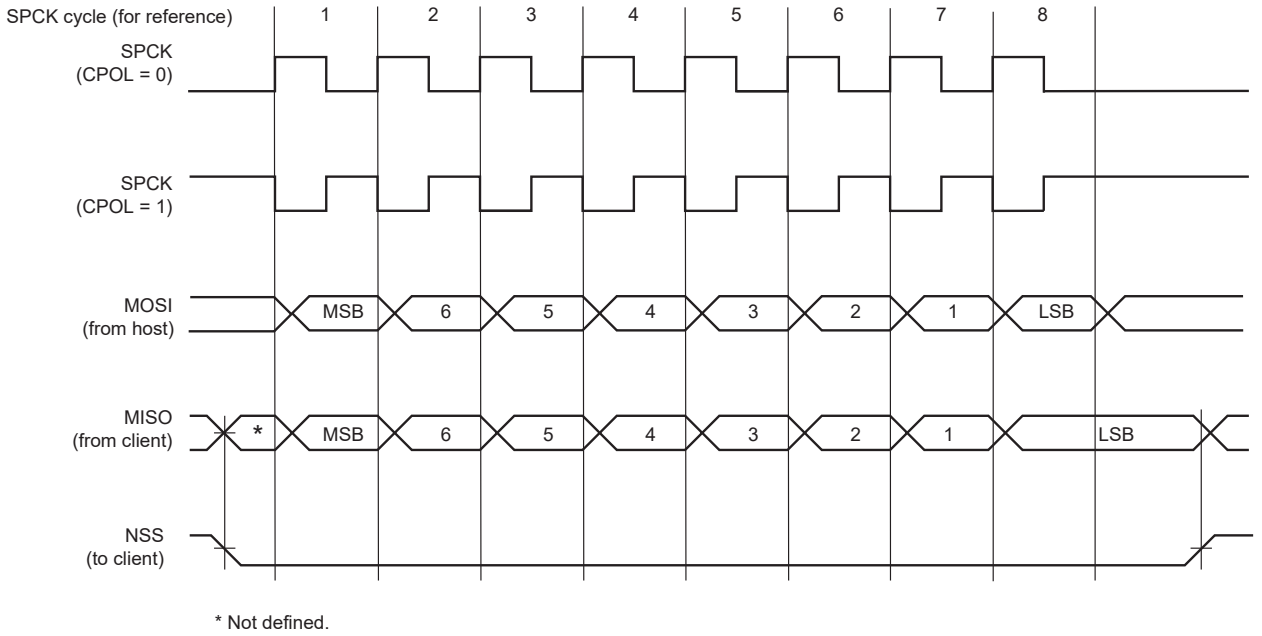
Figure 45-3. MCSPI Transfer Format (NCPHA = 1, 8 bits per transfer)



* Not defined.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 45-4. MCSPI Transfer Format (NCPHA = 0, 8 bits per transfer)



* Not defined.

45.7.3. Host Mode Operations

When configured in Host mode, the MCSPI operates on the clock generated by the internal programmable baud rate generator. It fully controls the data transfers to and from the client(s) connected to the SPI bus. The MCSPI drives the chip select line to the client and the serial clock signal (SPCK).

The MCSPI features two holding registers, the Transmit Data register (MCSPI_TDR) and the Receive Data register (MCSPI_RDR), and a single shift register. The holding registers maintain the data flow at a constant rate.

After enabling the MCSPI, a data transfer starts when the processor writes to MCSPI_TDR. The written data is immediately transferred into the internal shift register and the transfer on the MCSPI bus starts. While the data in the shift register is shifted on the MOSI line, the MISO line is sampled and shifted into the shift register. Data cannot be loaded in MCSPI_RDR without transmitting data. If there is no data to transmit, dummy data can be used (MCSPI_TDR filled with ones). If the Wait for Data Read Before Transfer (WDRBT) bit in MCSPI_MR is set, transmission can occur only if MCSPI_RDR has been read. If Receiving mode is not required, for example when communicating with a client receiver only (such as an LCD), the receive status flags in the MCSPI Status register (MCSPI_SR) can be discarded.

Before writing MCSPI_TDR, MCSPI_MR.PCS must be set in order to select a client.

If new data is written in MCSPI_TDR during the transfer, it is kept in MCSPI_TDR until the current transfer is completed. Then, the received data is transferred from the shift register to MCSPI_RDR, the data in MCSPI_TDR is loaded in the shift register and a new transfer starts.

As soon as MCSPI_TDR is written, the Transmit Data Register Empty (TDRE) flag in MCSPI_SR is cleared. When the data written in MCSPI_TDR is loaded into the shift register, MCSPI_SR.TDRE is set. The TDRE flag is used to trigger the Transmit PDC channel.

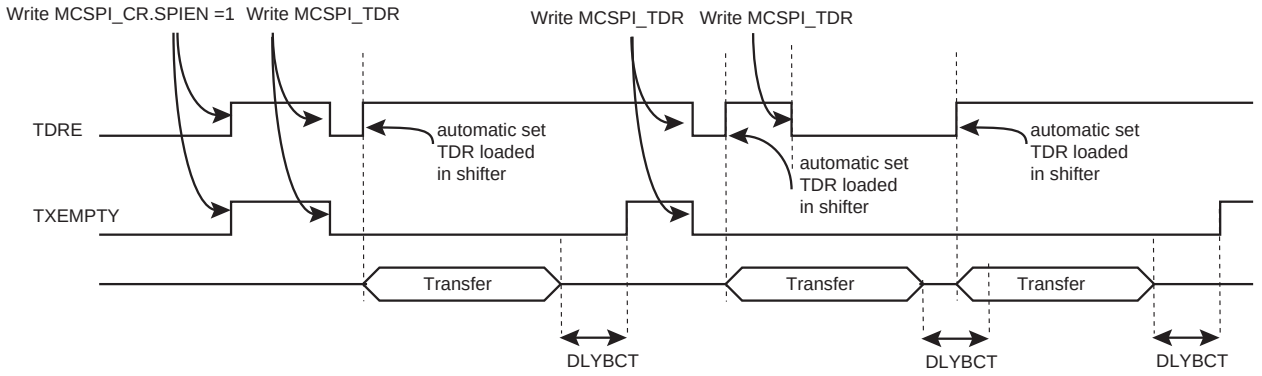
See the figure below.

The end of transfer is indicated by the Transmission Registers Empty (TXEMPTY) flag in MCSPI_SR. If a transfer delay (DLYBCT) is greater than 0 for the last transfer, TXEMPTY is set after the completion of this delay. The peripheral clock can be switched off at this time.

© 2026 Microchip Technology Inc. and its subsidiaries

Note: When the MCSPI is enabled, the TDRE and TXEMPTY flags are set.

Figure 45-5. TDRE and TXEMPTY Flag Behavior



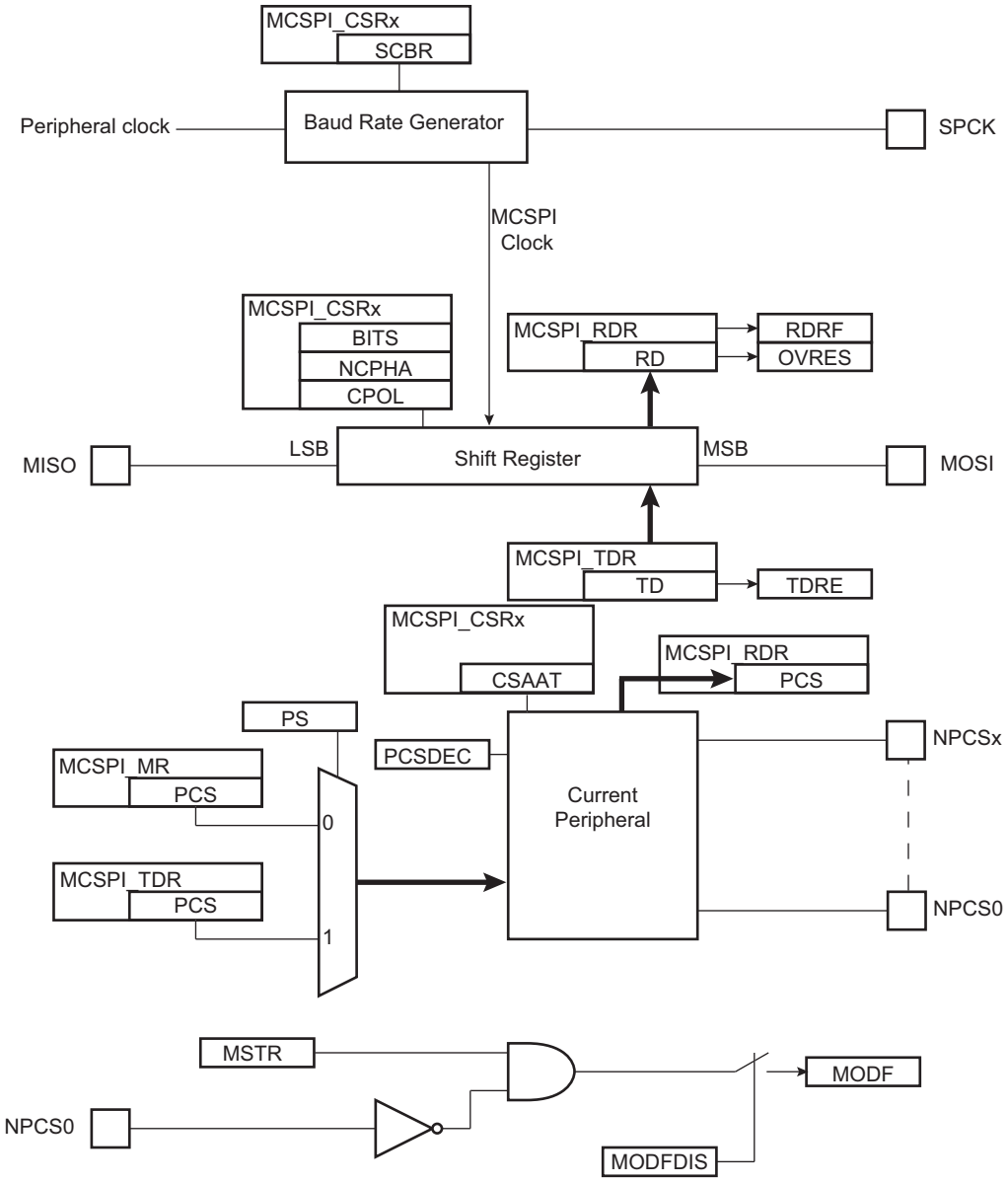
The transfer of received data from the internal shift register to MCSPI_RDR is indicated by the Receive Data Register Full (RDRF) bit in MCSPI_SR. When the received data is read, MCSPI_SR.RDRF is cleared.

If MCSPI_RDR was not read before new data is received, the Overrun Error (OVRES) flag in MCSPI_SR is set. As long as this flag is set, data is loaded in MCSPI_RDR. The user has to read MCSPI_SR to clear OVRES.

Figure 45-6 shows a block diagram of the MCSPI when operating in Host mode. Figure 45-7 shows a flow chart describing how transfers are handled.

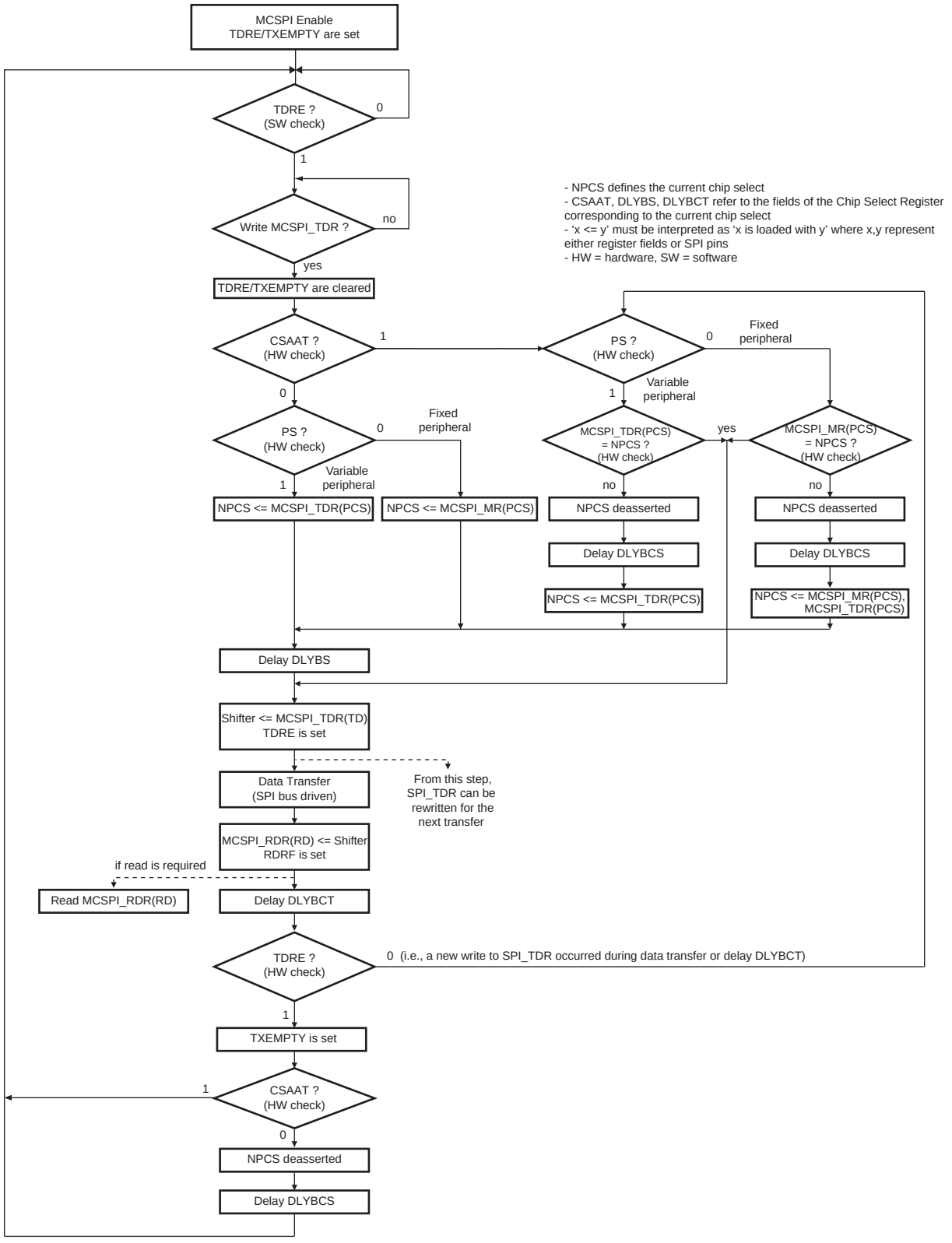
© 2026 Microchip Technology Inc. and its subsidiaries

Figure 45-6. MCSPI Host Mode Block Diagram



© 2026 Microchip Technology Inc. and its subsidiaries

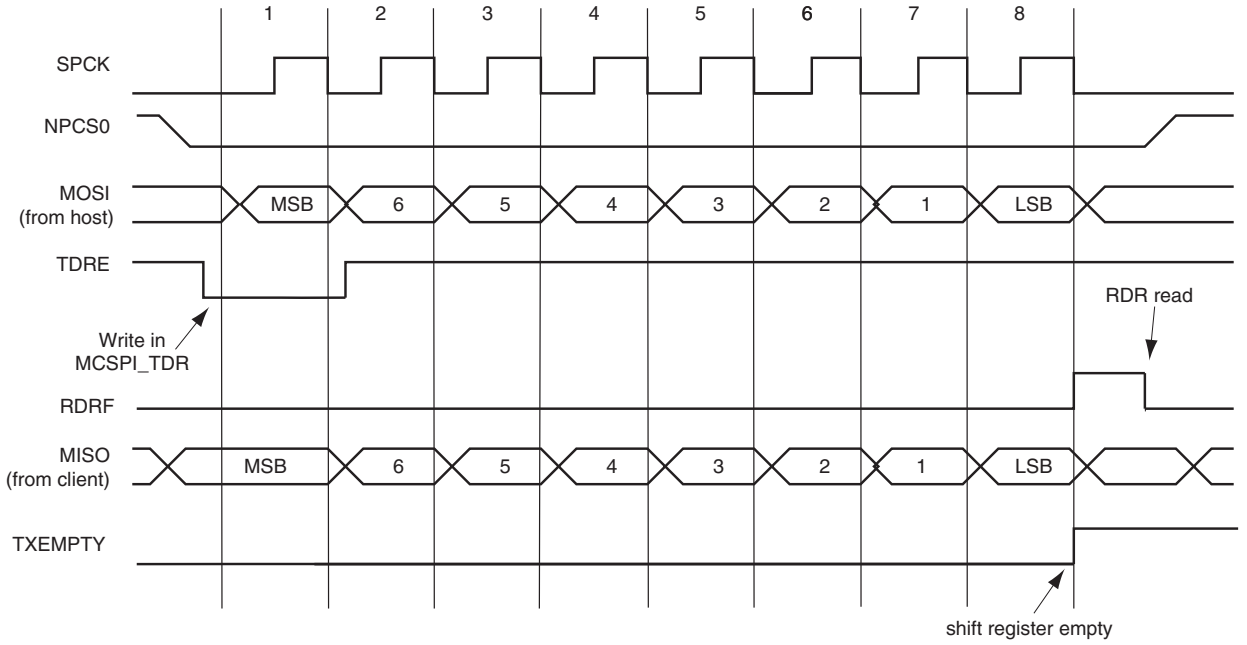
Figure 45-7. MCSPI Host Mode Flow Diagram



The figure below shows the behavior of the TDRE, RDRF and TXEMPTY status flags within MCSPI_SR during an 8-bit data transfer in Fixed mode without the PDCinvolved.

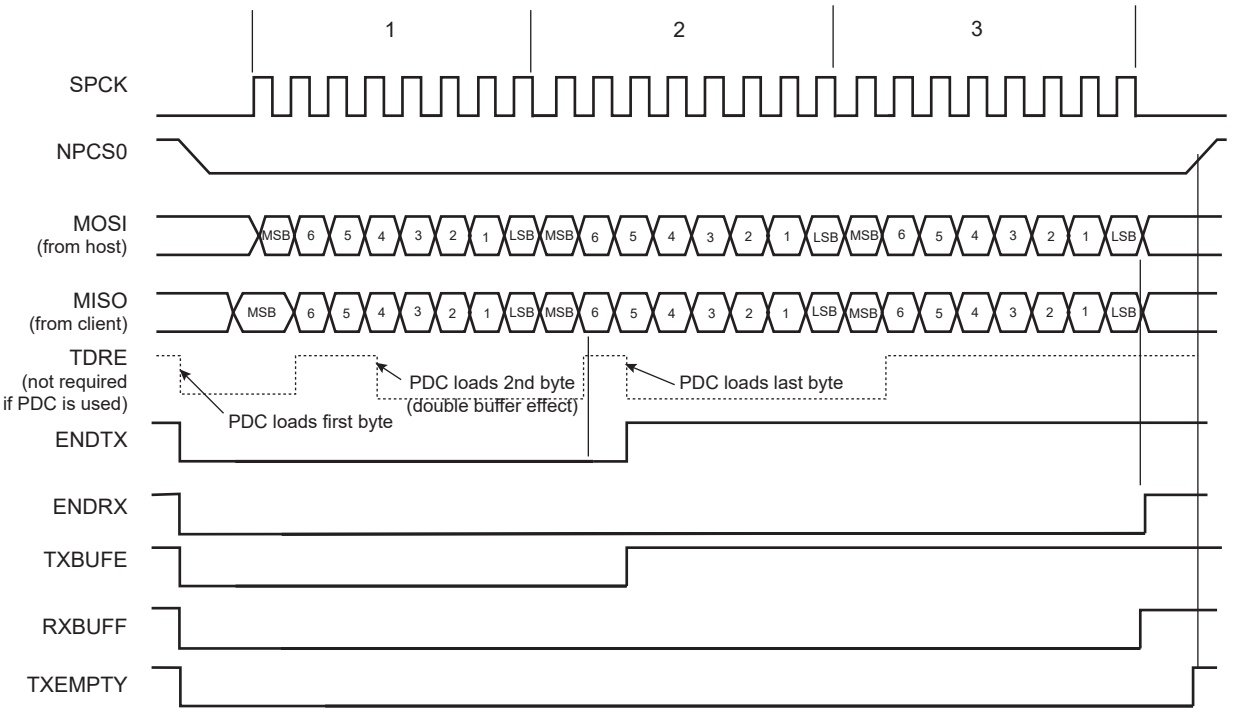
© 2026 Microchip Technology Inc. and its subsidiaries

Figure 45-8. Status Register Flag Behavior



The figure below shows the behavior of Transmission Register Empty (TXEMPTY), End of RX Buffer (ENDRX), End of TX Buffer (ENDTX), RX Buffer Full (RXBUFF) and TX Buffer Empty (TXBUFE) status flags within MCSPI_SR during an 8-bit data transfer in Fixed mode with the PDC involved. The PDC is programmed to transfer and receive three units of data. The next pointer and counter are not used. Flags RDRF and TDRE are not shown because they are managed by the PDC when using the PDC.

Figure 45-9. PDC Status Register Flag Behavior



© 2026 Microchip Technology Inc. and its subsidiaries

45.7.3.1. Clock Generation

The MCSPI baud rate clock is generated by dividing the peripheral clock by a value between 1 and 255.

If the Serial Clock Bit Rate (SCBR) field in MCSPI_CSRx is programmed to 1, the operating baud rate is peripheral clock (refer to the section “Electrical Characteristics” for the SPCK maximum frequency). Triggering a transfer while SCBR is at 0 can lead to unpredictable results.

At reset, SCBR=0 and the user has to program it to a valid value before performing the first transfer.

The divisor can be defined independently for each chip select, as it has to be programmed in SCBR. This allows the MCSPI to automatically adapt the baud rate for each interfaced peripheral without reprogramming.

If GCLK is selected as the source clock (MCSPI_MR.BRSRCCLK = 1), the bit rate is independent of the processor/bus clock and MCSPI_CSRx.SCBR must set to 2 or higher. Thus, the processor clock can be changed while the MCSPI is enabled. The processor clock frequency changes must be performed only by programming PMC_MCKR.PRES (refer to the section “Power Management Controller” (PMC)). Any other method to modify the processor/bus clock frequency (PLL multiplier, etc.) is forbidden when the MCSPI is enabled.

The peripheral clock frequency must be at least three times higher than GCLK frequency. SPCK clock jitter is minimized when the ratio between peripheral clock and GCLK is the greatest.

45.7.3.2. Transfer Delays

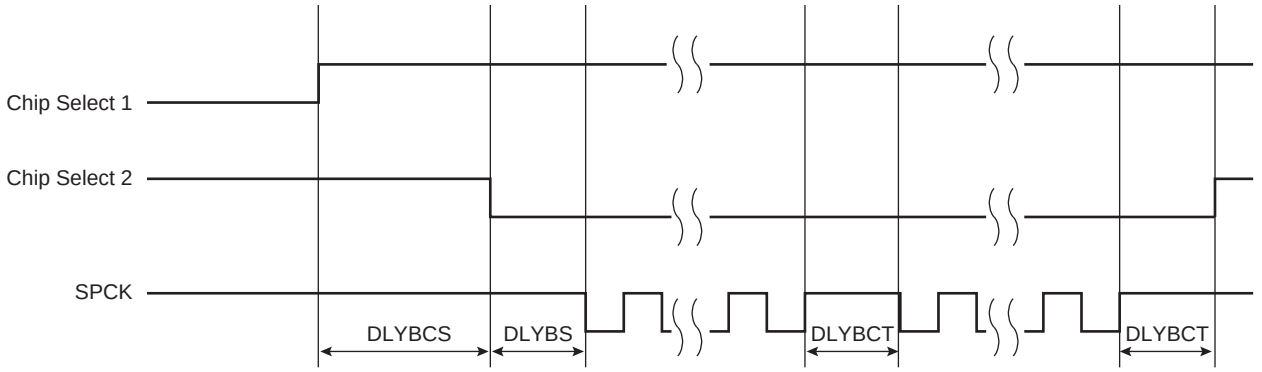
Figure 45-10 shows a chip select transfer change and consecutive transfers on the same chip select. Three delays can be programmed to modify the transfer waveforms:

- Delay between the chip selects—programmable only once for all chip selects by writing the Delay Between Chip Selects (DLYBCS) field in MCSPI_MR. The MCSPI client device deactivation delay is managed through DLYBCS. If there is only one MCSPI client device connected to the host, DLYBCS does not need to be configured. If several client devices are connected to a host, DLYBCS must be configured depending on the highest deactivation delay. Refer to details on the MCSPI client device in the section “Electrical Characteristics”.
- Delay before SPCK—independently programmable for each chip select by writing the Delay Before SPCK (DLYBS) field in MCSPI_CSRx. The MCSPI client device activation delay is managed through DLYBS. Refer to details on the MCSPI client device in the section “Electrical Characteristics” to define DLYBS.
- Delay between consecutive transfers—independently programmable for each chip select by writing the Delay Between Consecutive Transfers (DLYBCT) field in MCSPI_CSRx. The time required by the MCSPI client device to process received data is managed through DLYBCT. This time depends on the MCSPI client system activity.

These delays allow the MCSPI to be adapted to the interfaced peripherals and their speed and bus release time.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 45-10. Programmable Delays



45.7.3.3. Peripheral Selection

The serial peripherals are selected through the assertion of the NPCS0 to NPCS3 signals. By default, all NPCS signals are high before and after each transfer.

- Fixed Peripheral Select Mode: MCSPI exchanges data with only one peripheral.
  Fixed Peripheral Select mode is enabled by clearing MCSPI_MR.PS. In this case, the current peripheral is defined by MCSPI_MR.PCS. MCSPI_TDR.PCS has no effect.
- Variable Peripheral Select Mode: Data can be exchanged with more than one peripheral without having to reprogram MCSPI_MR.PCS.
  Variable Peripheral Select mode is enabled by setting MCSPI_MR.PS. MCSPI_TDR.PCS is used to select the current peripheral. This means that the peripheral selection can be defined for each new data. The value must be written in a single access to MCSPI_TDR in the following format:

  [xxxxxxx(7-bit) + LASTXFER(1-bit)[1]+ xxxx(4-bit) + PCS (4-bit) + TD (8- to 16-bit data)]

  with LASTXFER (Last Transfer) at 0 or 1 depending on the Chip Select Active After Transfer (CSAAT) bit, and PCS (Peripheral Chip Select) equal to the chip select to assert, as defined in MCSPI_TDR.

  Note:

  a. Optional

  For details on CSAAT, LASTXFER and CSNAAT, see Peripheral Deselection with PDC.

  If LASTXFER is used, the command must be issued after writing the last character. Instead of LASTXFER, the user can use the SPIDIS command. After the end of the PDC transfer, it is necessary to wait for the TXEMPTY flag and then write SPIDIS into the MCSPI Control register (MCSPI_CR). This does not change the configuration register values. The NPCS is disabled after the last character transfer. Then, another PDC transfer can be started if the MCSPI Enable (SPIEN) bit in MCSPI_CR was previously written.

45.7.3.4. MCSPI Peripheral DMA Controller (PDC)

In both Fixed and Variable Peripheral Select modes, the Peripheral DMA Controller (PDC) can be used to reduce processor overhead.

The fixed peripheral selection allows buffer transfers with a single peripheral. Using the PDC is an optimal means, as the size of the data transfer between the memory and the MCSPI is either 8 bits or 16 bits. However, if the peripheral selection is modified, MCSPI_MR must be reprogrammed.

The variable peripheral selection allows buffer transfers with multiple peripherals without reprogramming MCSPI_MR. Data written in MCSPI_TDR is 32 bits wide and defines the real data to be transmitted and the destination peripheral. Using the PDC in this mode requires 32-bit wide buffers, with the data in the LSBs and the PCS and LASTXFER fields in the MSBs. However, the MCSPI still controls the number of bits (8 to 16) to be transferred through MISO and MOSI lines with the Chip Select registers (MCSPI_CSRx). This is not the optimal means in terms of memory size for the

© 2026 Microchip Technology Inc. and its subsidiaries

buffers, but it provides a very effective means to exchange data with several peripherals without any processor intervention.

45.7.3.4.1. Transfer Size

Depending on the data size to transmit, from 8 to 16 bits, the PDC manages automatically the type of pointer size it has to point to. The PDC performs the following transfer, depending on the mode and number of bits per data.

- Fixed mode:
  - 8-bit data:
    1-byte transfer, PDC pointer address = address + 1 byte,
    PDC counter = counter - 1
  - 9-bit to 16-bit data:
    2-byte transfer. n-bit data transfer with don't care data (MSB) filled with 0's,
    PDC pointer address = address + 2 bytes,
    PDC counter = counter - 1
- Variable mode:
  - In Variable mode, PDC pointer address = address + 4 bytes and PDC counter = counter - 1 for 8 to 16-bit transfer size.
  - When using the PDC, the TDRE and RDRF flags are handled by the PDC. The user's application does not have to check these bits. Only End of RX Buffer (ENDRX), End of TX Buffer (ENDTX), Buffer Full (RXBUFF), TX Buffer Empty (TXBUFE) are significant. For further details about the Peripheral DMA Controller and user interface, refer to the "Peripheral DMA Controller (PDC)" section.

45.7.3.5. Peripheral Chip Select Decoding

The user can program the MCSPI to operate with up to 15 client peripherals by decoding the four chip select lines, NPCS0 to NPCS3 with an external decoder/demultiplexer (see the figure below). This can be enabled by setting the Chip Select Decode (PCSDEC) bit in MCSPI_MR.

When operating without decoding, the MCSPI makes sure that in any case only one chip select line is activated, i.e., one NPCS line driven low at a time. If two bits are defined low in a PCS field, only the lowest numbered chip select is driven low.

When operating with decoding, the MCSPI directly outputs the value defined by the PCS field on the NPCS lines of either MCSPI_MR or MCSPI_TDR (depending on PS).

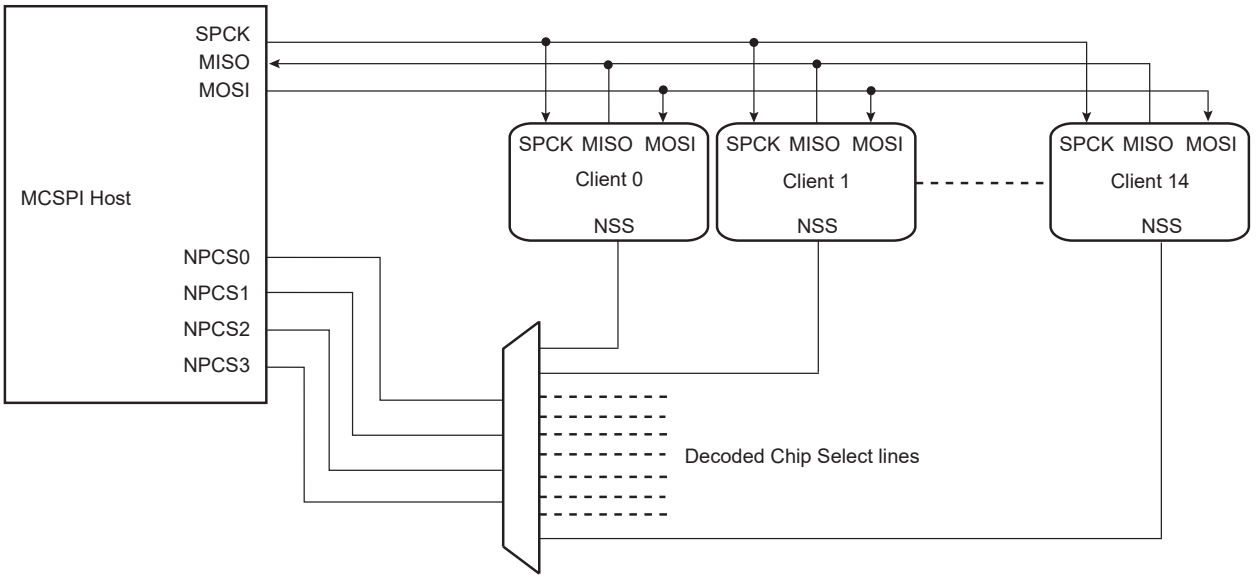
As the MCSPI sets a default value of 0xF on the chip select lines (i.e., all chip select lines at 1) when not processing any transfer, only 15 peripherals can be decoded.

The MCSPI has four chip select registers (MCSPI_CSR0...MCSPI_CSR3). As a result, when external decoding is activated, each NPCS chip select defines the characteristics of up to four peripherals. As an example, MCSPI_CRS0 defines the characteristics of the externally decoded peripherals 0 to 3, corresponding to the PCS values 0x0 to 0x3. Consequently, the user has to make sure to connect compatible peripherals on the decoded chip select lines 0 to 3, 4 to 7, 8 to 11 and 12 to 14. The following figure shows this type of implementation.

If the MCSPI_CSRx.CSAAT bit is used, with or without the PDC, the Mode Fault detection for NPCS0 line must be disabled. This is not required for all other chip select lines since mode fault detection is only on NPCS0.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 45-11. Chip Select Decoding Application Block Diagram: Single Host/Multiple Client Implementation



45.7.3.6. Peripheral Deselection without PDC

During a transfer of more than one unit of data on a chip select without the PDC, MCSPI_TDR is loaded by the processor, the TDRE flag rises as soon as the content of MCSPI_TDR is transferred into the internal shift register. When this flag is detected high, MCSPI_TDR can be reloaded. If this reload by the processor occurs before the end of the current transfer and if the next transfer is performed on the same chip select as the current transfer, the chip select is not deasserted between the two transfers. But depending on the application software handling the MCSPI status register flags (by interrupt or polling method) or servicing other interrupts or other tasks, the processor may not reload MCSPI_TDR in time to keep the chip select active (low). A null DLYBCT value (delay between consecutive transfers) in MCSPI_CSR, gives even less time for the processor to reload MCSPI_TDR. With some MCSPI client peripherals, if the chip select line must remain active (low) during a full set of transfers, communication errors can occur.

To facilitate interfacing with such devices, the chip select registers [MCSPI_CSR0...MCSPI_CSR3] can be programmed with the MCSPI_CSRx.CSAAT bit at 1. This allows the chip select lines to remain in their current state (low = active) until a transfer to another chip select is required. Even if MCSPI_TDR is not reloaded, the chip select remains active. To deassert the chip select line at the end of the transfer, the MCSPI_SR.LASTXFER bit must be set after writing the last data to transmit into MCSPI_TDR.

45.7.3.7. Peripheral Deselection with PDC

PDC provides faster reloads of MCSPI_TDR compared to software. However, depending on the system activity, it is not guaranteed that MCSPI_TDR is written with the next data before the end of the current transfer. Consequently, data can be lost by deassertion of the NPCS line for MCSPI client peripherals requiring the chip select line to remain active between two transfers. The only way to guarantee a safe transfer in this case is the use of the CSAAT and LASTXFER bits.

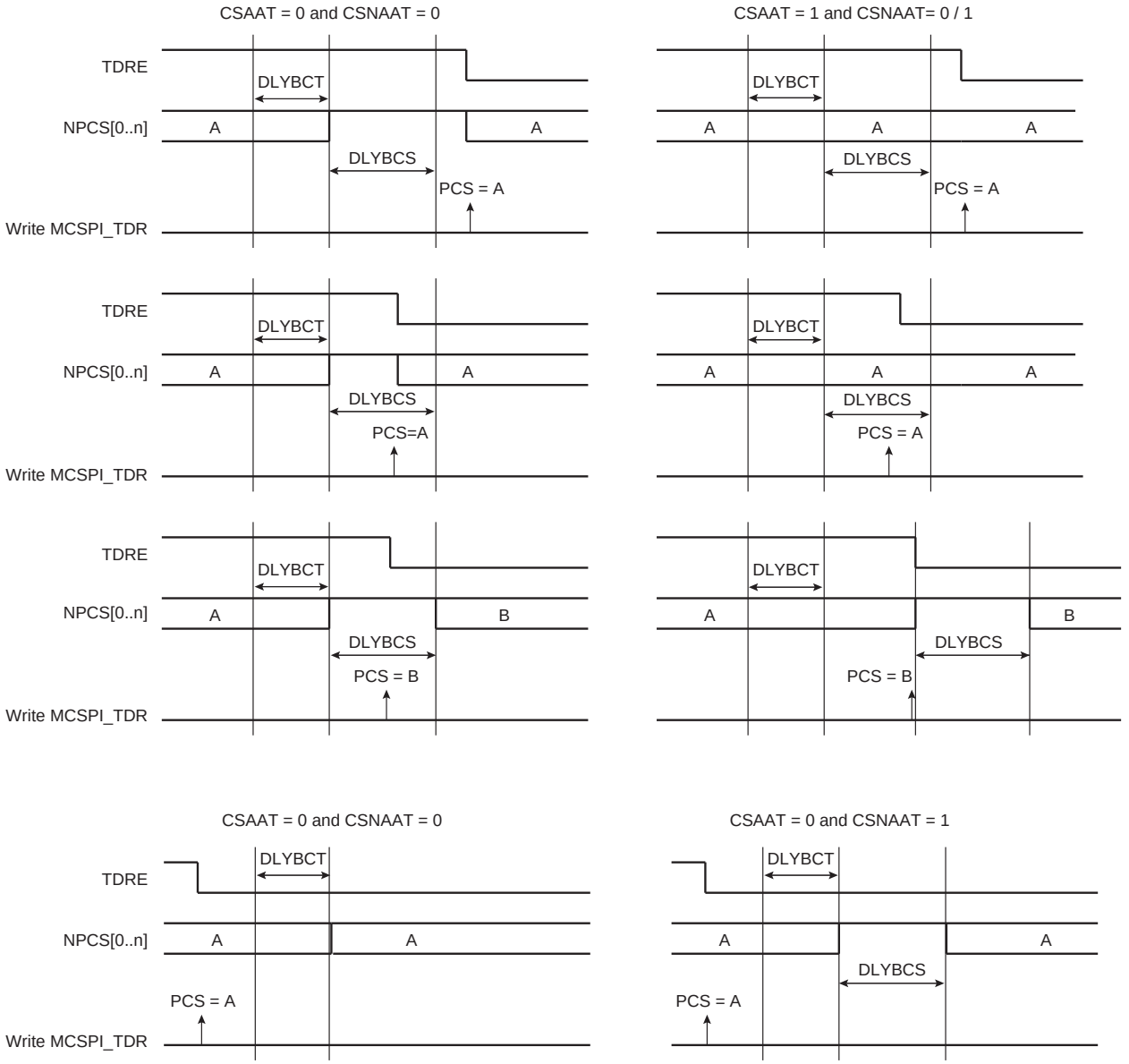
When the CSAAT bit is configured to 0, the NPCS does not rise in all cases between two transfers on the same peripheral. During a transfer on a chip select, the TDRE flag rises as soon as the content of MCSPI_TDR is transferred into the internal shift register. When this flag is detected, MCSPI_TDR can be reloaded. If this reload occurs before the end of the current transfer and if the next transfer is performed on the same chip select as the current transfer, the chip select is not deasserted between the two transfers. This can lead to difficulties to interface with some serial peripherals requiring the chip select to be deasserted after each transfer. To facilitate interfacing with such devices,

© 2026 Microchip Technology Inc. and its subsidiaries

MCSPI_CSR can be programmed with the Chip Select Not Active After Transfer (CSNAAT) bit at 1. This allows the chip select lines to be deasserted systematically during a time “DLYBCS” (the value of the CSNAAT bit is processed only if the CSAAT bit is configured to 0 for the same chip select).

The following figure shows different peripheral deselection cases and the effect of the CSAAT and CSNAAT bits.

Figure 45-12. Peripheral Deselection



45.7.3.8. Mode Fault Detection

The MCSPI has the capability to operate in multihost environment. Consequently, the NPCS0/NSS line must be monitored. If one of the hosts on the SPI bus is currently transmitting, the NPCS0/NSS line is low and the MCSPI must not transmit any data. A mode fault is detected when the MCSPI is programmed in Host mode and a low level is driven by an external host on the NPCS0/NSS signal. In multihost environment, NPCS0, MOSI, MISO and SPCK pins must be configured in open drain (through the PIO controller). When a mode fault is detected, the Mode Fault Error (MODF) bit in MCSPI_SR is set until MCSPI_SR is read and the MCSPI is automatically disabled until it is reenabled by setting the MCSPI_CR.SPIEN bit.

© 2026 Microchip Technology Inc. and its subsidiaries

By default, the mode fault detection is enabled. The user can disable it by setting the Mode Fault Detection (MODFDIS) bit in MCSPI_MR.

45.7.4. MCSPI Client Mode

When operating in Client mode, the MCSPI processes data bits on the clock provided on the MCSPI clock pin (SPCK).

The MCSPI waits until NSS goes active before receiving the serial clock from an external host. When NSS falls, the clock is validated and the data is loaded in MCSPI_RDR depending on the configuration of the Bits Per Transfer (BITS) field in MCSPI_CSR0. These bits are processed following a phase and a polarity defined respectively by the NCPHA and CPOL bits in MCSPI_CSR0. Note that the fields BITS, CPOL and NCPHA of the other chip select registers (MCSPI_CSR1...MCSPI_CSR3) have no effect when the MCSPI is programmed in Client mode.

The bits are shifted out on the MISO line and sampled on the MOSI line.

Note: For more information on MCSPI_CSRx.BITS, see Note in MCSPI_CSRx.

When all bits are processed, the received data is transferred to MCSPI_RDR and the RDRF bit rises. If MCSPI_RDR has not been read before new data is received, the Overrun Error Status (OVRES) bit in MCSPI_SR is set. As long as this flag is set, data is loaded in MCSPI_RDR. The user must read MCSPI_SR to clear the OVRES bit.

When a transfer starts, the data shifted out is the data present in the internal shift register. If no data has been written in MCSPI_TDR, the last data received is transferred. If no data has been received since the last reset, all bits are transmitted low, as the internal shift register resets to 0.

When a first data is written in MCSPI_TDR, it is transferred immediately in the internal shift register and the TDRE flag rises. If new data is written, it remains in MCSPI_TDR until a transfer occurs, i.e., NSS falls and there is a valid clock on the SPCK pin. When the transfer occurs, the last data written in MCSPI_TDR is transferred to the internal shift register and the TDRE flag rises. This enables frequent updates of critical variables with single transfers.

Then, new data is loaded in the internal shift register from MCSPI_TDR. If no character is ready to be transmitted, i.e., no character has been written in MCSPI_TDR since the last load from MCSPI_TDR to the internal shift register, MCSPI_TDR is retransmitted. In this case the Underrun Error Status Flag (UNDES) is set in MCSPI_SR.

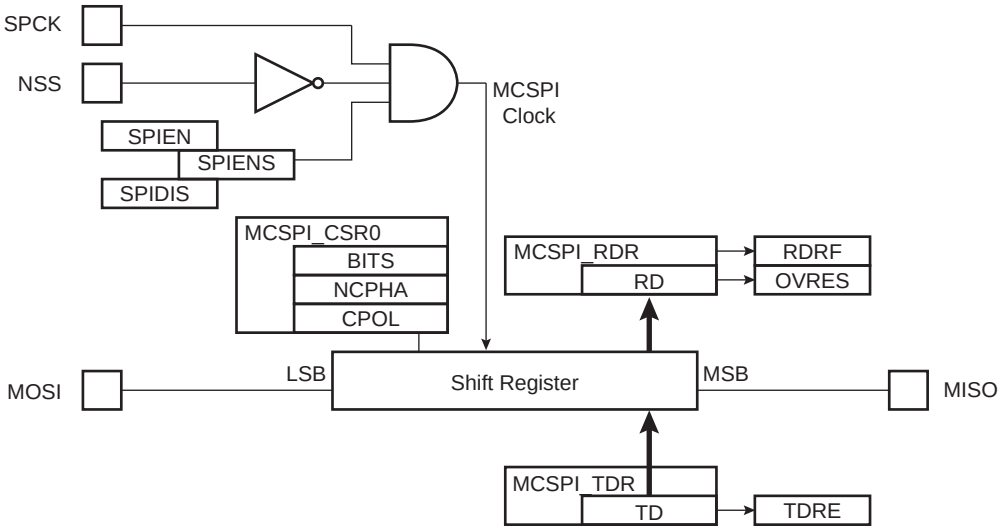
If NSS rises between two characters, it must be kept high for two MCK clock periods or more and the next SPCK capture edge must not occur less than four MCK periods after NSS rises.

In Client mode, if the NSS line rises and the received character length does not match the configuration defined in MCSPI_CSR0.BITS, the Client Frame Error (SFERR) flag is set in MCSPI_SR.

The following figure shows a block diagram of the MCSPI when operating in Client mode.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 45-13. MCSPI Client Mode Functional Block Diagram



45.7.5. CRC Generation and Checking

The MCSPI offers the possibility to compute and check a CRC while receiving a frame. To enable the CRC check, the CRC Enable (CRCEN) bit must be set to '1' in MCSPI_MR.

The CRC register (MCSPI_CRCR) configures the frame length using the Frame Length (FRL) field and the CRC size using the CRC size (CRCS) bit. It is possible to define a frame header length with the Frame Header Length (FRHL) field, and a frame header can be included or excluded from CRC calculation depending on the Frame Header Excluded (FHE) bit configuration. In case of continuous read frames, the Continuous Read Mode (CRM) bit defines if the frame header is sent only at the beginning of the first frame or at every frame. The CRC Error (CRCERR) flag in MCSPI_SR indicates any failed CRC check.

The CRC checksum uses the 16-bit CRC-16 ANSI polynomial as defined in the IEEE 802.3 standard: x16 + x15 + x2 + 1. This polynomial can also be noted as 0x8005.

It is possible to define a frame header length with the Frame Header Length (FRHL) field, and a frame header can be included or excluded from CRC calculation depending on the Frame Header Excluded (FHE) bit configuration. In case of continuous read frames, the Continuous Read Mode (CRM) bit defines if the frame header is sent only at the beginning of the first frame or at every frame. The CRC Error (CRCERR) flag in MCSPI_SR indicates any failed CRC check.

It is possible to configure the MCSPI to receive or not CRC and/or Header as a classic data configuring MCSPI_CRCR.DCRX and MCSPI_CRCR.DHRX. When CRC and/or Header is configured not to be received as data, upon receiving the Header and/or CRC, the RDRF flag does not rise and MCSPI_RDR is not updated with the received value.

45.7.6. Two-Pin Mode

45.7.6.1. MCP3910 Protocol Overview

In Two-Pin mode, the MCSPI interfaces to analog front-end ADC devices such as MCP3910. The MCP3910 device data output (SDO) is the host for the MOSI line, although the MCSPI generates the SPCK clock. MCSPI_CSRx.SCBR defines the frequency of the SPCK output clock. The MISO line is not used.

In Two-Pin mode, only two lines are used for communication, SPCK and MOSI.

In Two-Pin mode, MCSPI_CSR0.CPOL and MCSPI_CSR0.NCPHA must be set to '0'.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 45-14. MCP3910 Basic Connection to MCSPI

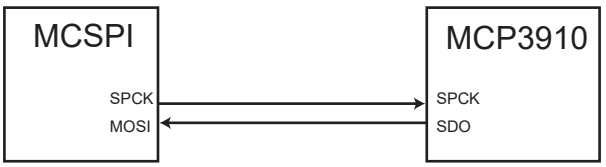
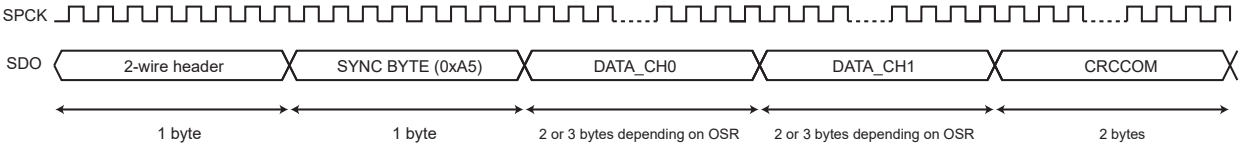
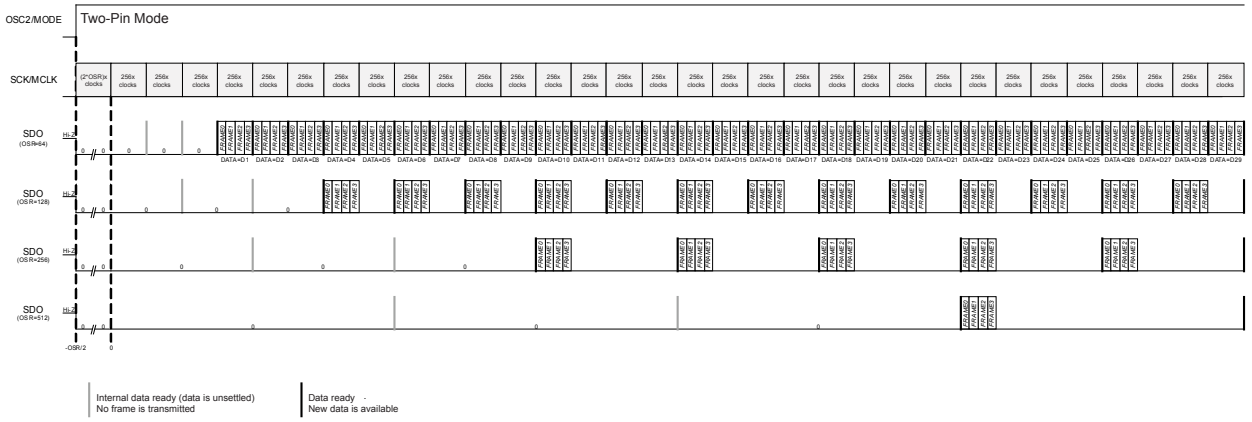


Figure 45-15. Two-Pin MCP3910 Frame



SPI Two-Pin mode connects with up to four two-pin devices. Each MCP3910 product sends four times the same frame. See the following figure.

Figure 45-16. MCP3910 Communication Protocol



45.7.6.2. Two-Pin Mode Configuration

The Two-Pin mode is enabled when MCSPI_MR.TPMEN=1 and MCSPI_MR.MSTR=0.

When Two-Pin mode is enabled, SPI Mode 0 must be configured (MCSPI_CSR0.CPOL=0 and MCSPI_CSR0.NCPHA=0) and the data length must be 8 bits (MCSPI_CSRx.BITS=0).

For proper CRC calculation and checking, MCSPI_CRCR.FHE must be set to '0', MCSPI_CRCR.CRM to '0', MCSPI_CRCR.FRHL to '1' and MCSPI_CRCR.CRCS to '0'. MCSPI_CRCR.FRL must be set according to the value used for MCSPI_TPMR.OSR (refer to “Two-Wire Serial Interface Description” in the data sheet “MCP3910” available on www.microchip.com for more details on frame length and OSR).

See CRC Generation and Checking for details on CRC check configuration.

To obtain synchronization on the synchronization byte (SYNC BYTE), the Comparison mode must be enabled (MCSPI_MR.CMPMODE=1) and the comparison value must be configured to SYNC BYTE (MCSPI_CMPR.VAL1/2= 0xA5). The clock is generated on SPCK when MCSPI_CR.SPIEN is set.

Once the MCSPI is configured in Two-Pin mode and Comparison mode is set, SYNC BYTE is monitored to start receiving a frame. Upon SYNC BYTE reception, the previously received header byte is stored in the Two-Pin Header register (MCSPI_TPHR) and the SYNC BYTE is stored in MCSPI_RDR (the RDRF flag indicates when the data is available). Each frame byte received is written in MCSPI_RDR until reception of the CRCCOM end field (CRC is received as a data).

© 2026 Microchip Technology Inc. and its subsidiaries

The Oversampling Rate (OSR) field in the Two-Pin Mode register (MCSPI_TPMR) defines the OSR frame configuration of the MCP3910, and thus the frame length.

Note: It is mandatory to configure MCSPI_CRCR when Two-Pin mode is enabled.

45.7.6.3. Connection Example

When more than one MCP3910 device is connected to the MCSPI, the SDO data lines from the MCP3910 products must be multiplexed prior to driving the MCSPI MOSI line.

The MCSPI offers two multiplexing capabilities (external or internal).

- External Multiplexing mode

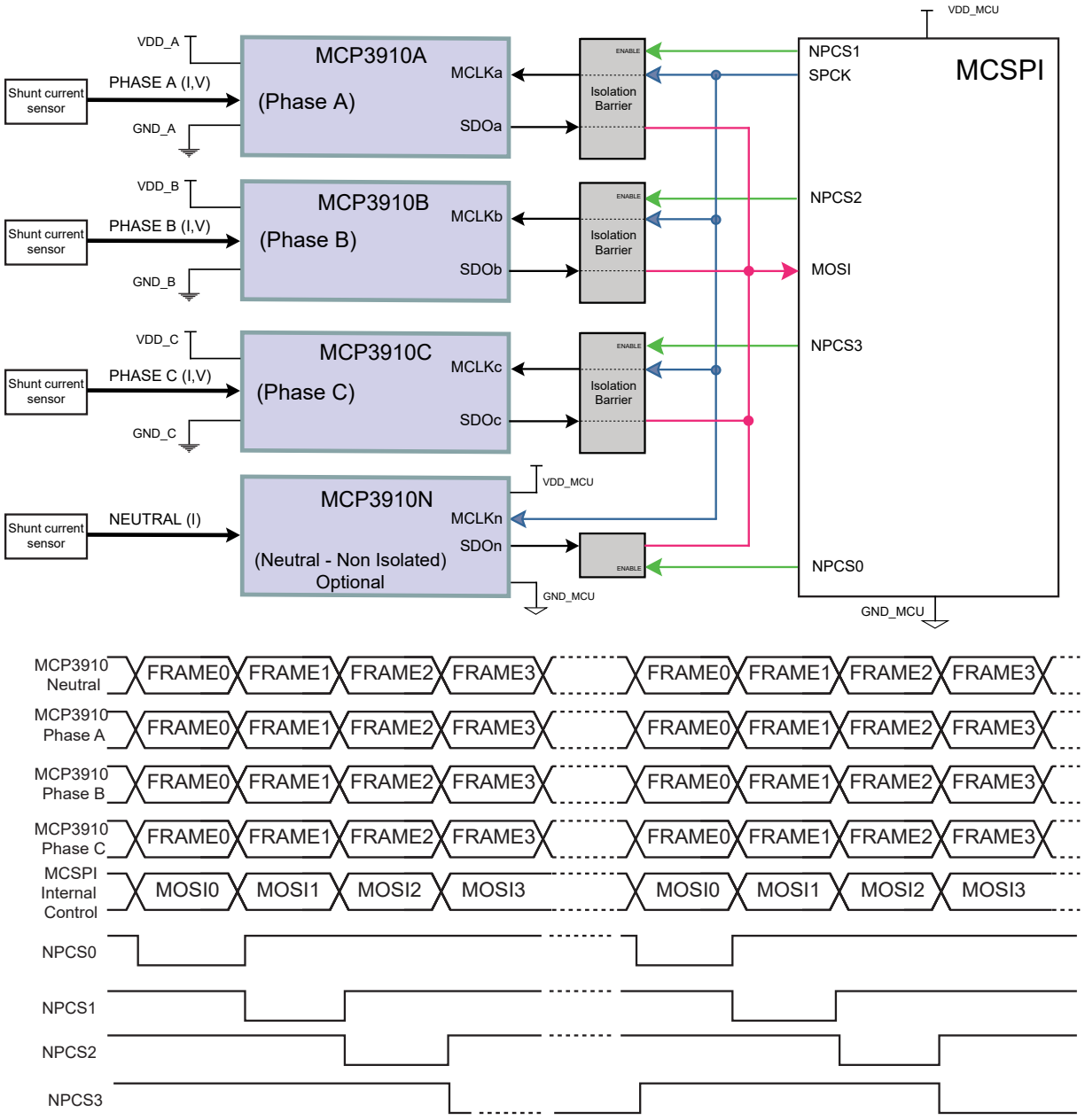
Multiplexing can be performed by successively enabling each SDO line from the MCP3910 with external discrete components (see the figure below).

In this mode, the four NPCS lines are activated successively after each frame reception.

The External Multiplexing mode is enabled by writing a 1 in the Chip Select Mode (CSM) bit in MCSPI_TPMR. The polarity of chip select output NPCS is defined by configuring the Chip Select Inversion Enable (CSIE) bit in MCSPI_MR.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 45-17. External Multiplexing Mode Example



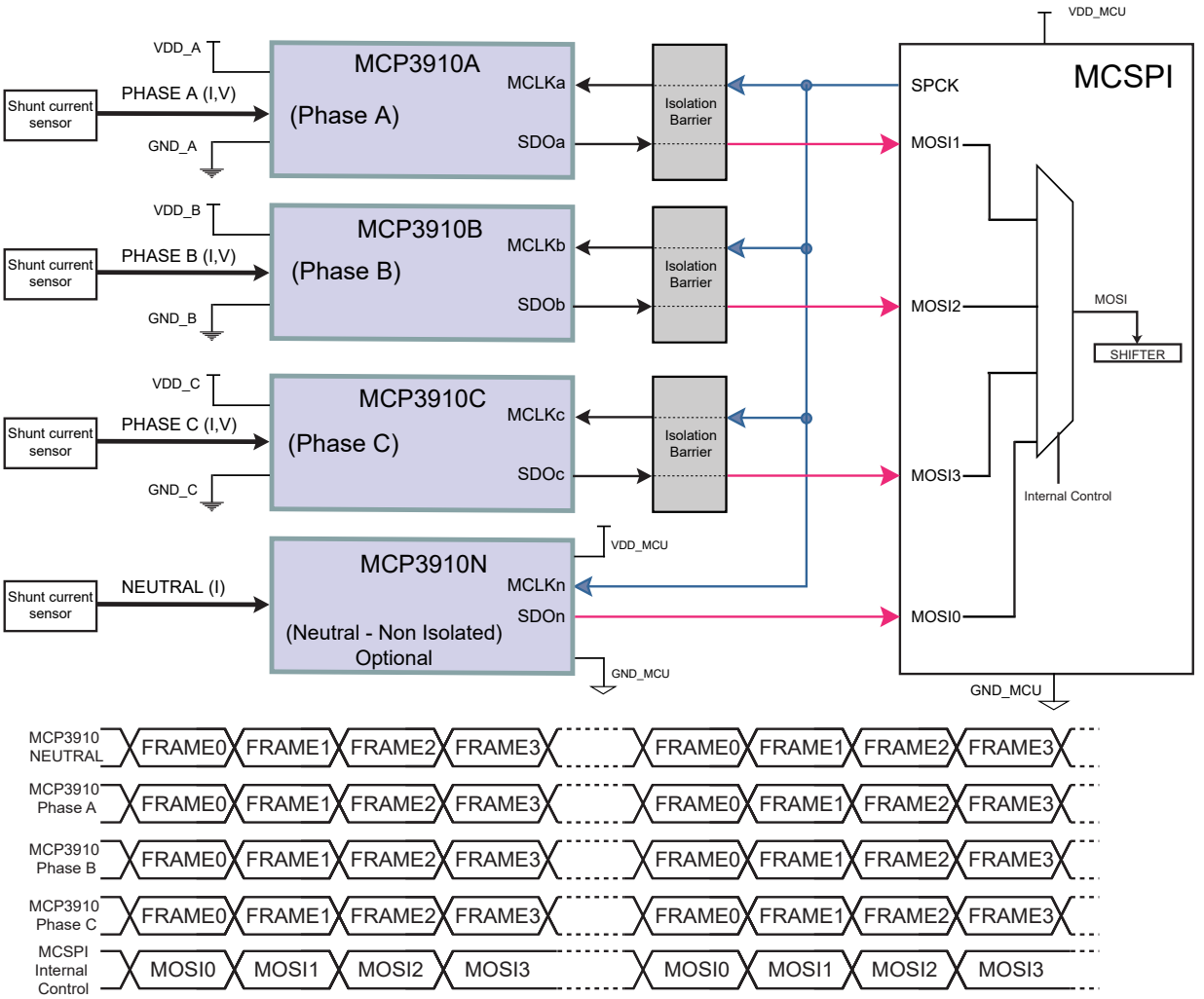
- Example of Internal Multiplexing mode

Multiplexing is performed within the MCSPI. Each MOSI line is read successively.

To enable the Internal Multiplexing mode, the Multiple Input Line (MIL) bit must be set to ‘1’ in MCSPI_TPMR. The polarity of the MOSI lines can be defined by configuring the MOSI Inversion Enable (MOSIIE) bit in MCSPI_MR.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 45-18. Internal Multiplexing Mode Example



45.7.7. MCSPI Comparison Function on Received Character

The comparison is only relevant for MCSPI Client mode (MCSPI_MR.MSTR=0).

In Active mode, the Comparison Status (CMP) flag in MCSPI_SR is raised. It is set when the received character matches the conditions programmed in the MCSPI Comparison register (MCSPI_CMPR). The CMP flag is set as soon as MCSPI_RDR is loaded with the newly received character. The CMP flag is cleared by reading MCSPI_SR.

MCSPI_CMPR (see MCSPI_CMPR) can be programmed to provide different comparison methods. These are listed below:

- If VAL1 equals VAL2, then the comparison is performed on a single value and the flag is set to 1 if the received character equals VAL1.
- If VAL1 is strictly lower than VAL2, then any value between VAL1 and VAL2 sets the CMP flag.
- If VAL1 is strictly higher than VAL2, then the CMP flag is set to 1 if any received character equals VAL1 or VAL2.

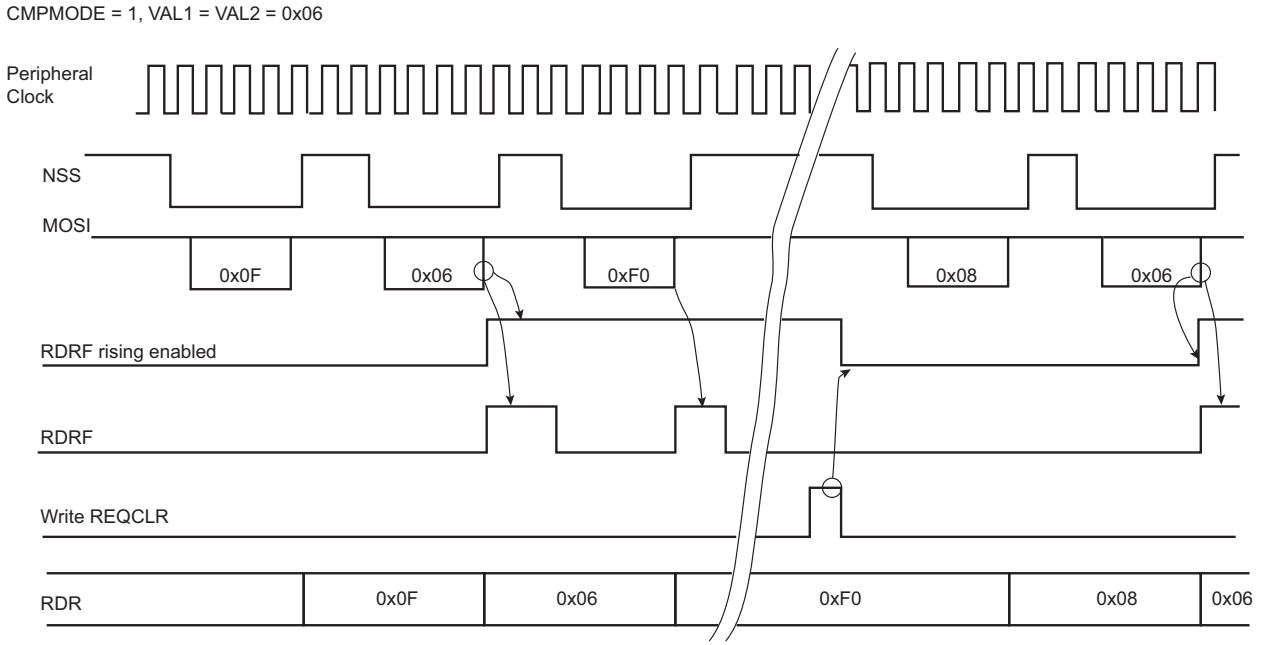
When MCSPI_MR.CMPMODE is cleared, all received data is loaded in MCSPI_RDR and the CMP flag provides the status of the comparison result.

© 2026 Microchip Technology Inc. and its subsidiaries

By setting MCSPI_MR.CMPMODE, the comparison result triggers the start of MCSPI_RDR loading (see the figure below). The trigger condition exists as soon as the received character value matches the conditions defined by VAL1 and VAL2 in MCSPI_CMPR. The comparison trigger event is restarted by setting MCSPI_CR.REQCLR .

The value programmed in the VAL1 and VAL2 fields must not exceed the maximum value of the received character (see MCSPI_CSR0.BITS).

Figure 45-19. Receive Data Register Management



45.7.8. FIFOs

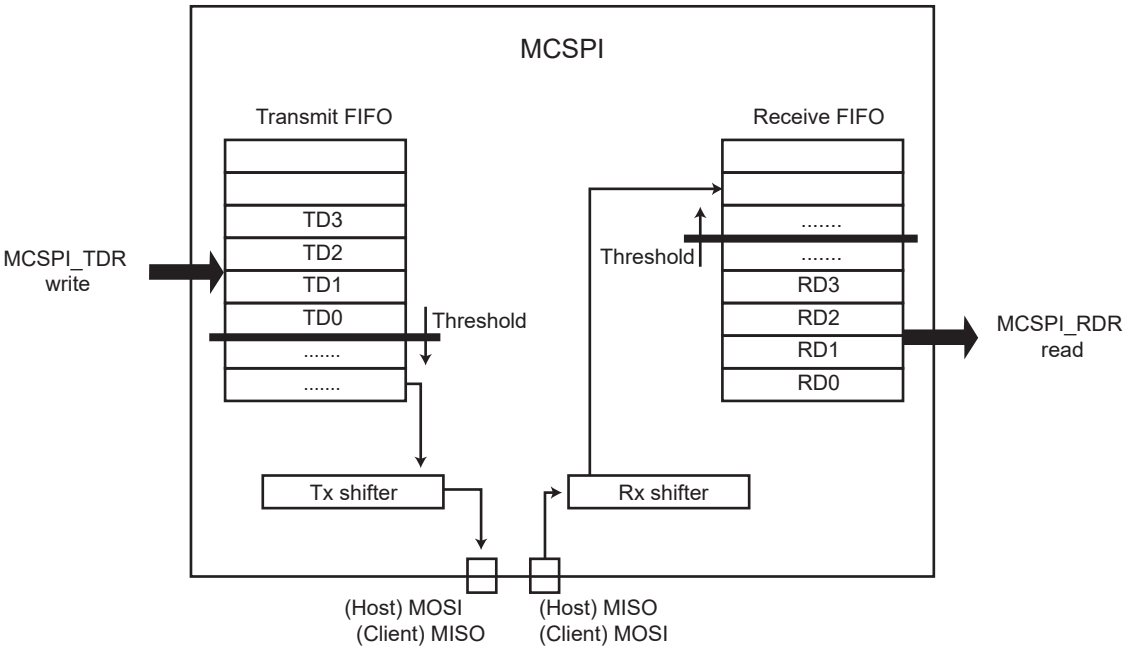
The MCSPI includes two FIFOs which can be enabled/disabled using MCSPI_CR.FIFOEN/FIFODIS. The MCSPI must be disabled (MCSPI_CR.SPIDIS) before enabling or disabling the MCSPI FIFOs.

Writing MCSPI_CR.FIFOEN to ‘1’ enables a 8-data Transmit FIFO and a 8-data Receive FIFO.

It is possible to write or to read single or multiple data in the same access to MCSPI_TDR/RDR. See Single Data Mode and Multiple Data Mode.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 45-20. MCSPI FIFO Block Diagram



45.7.8.1. Sending Data with FIFO Enabled

When the Transmit FIFO is enabled, write access to MCSPI_TDR loads the Transmit FIFO.

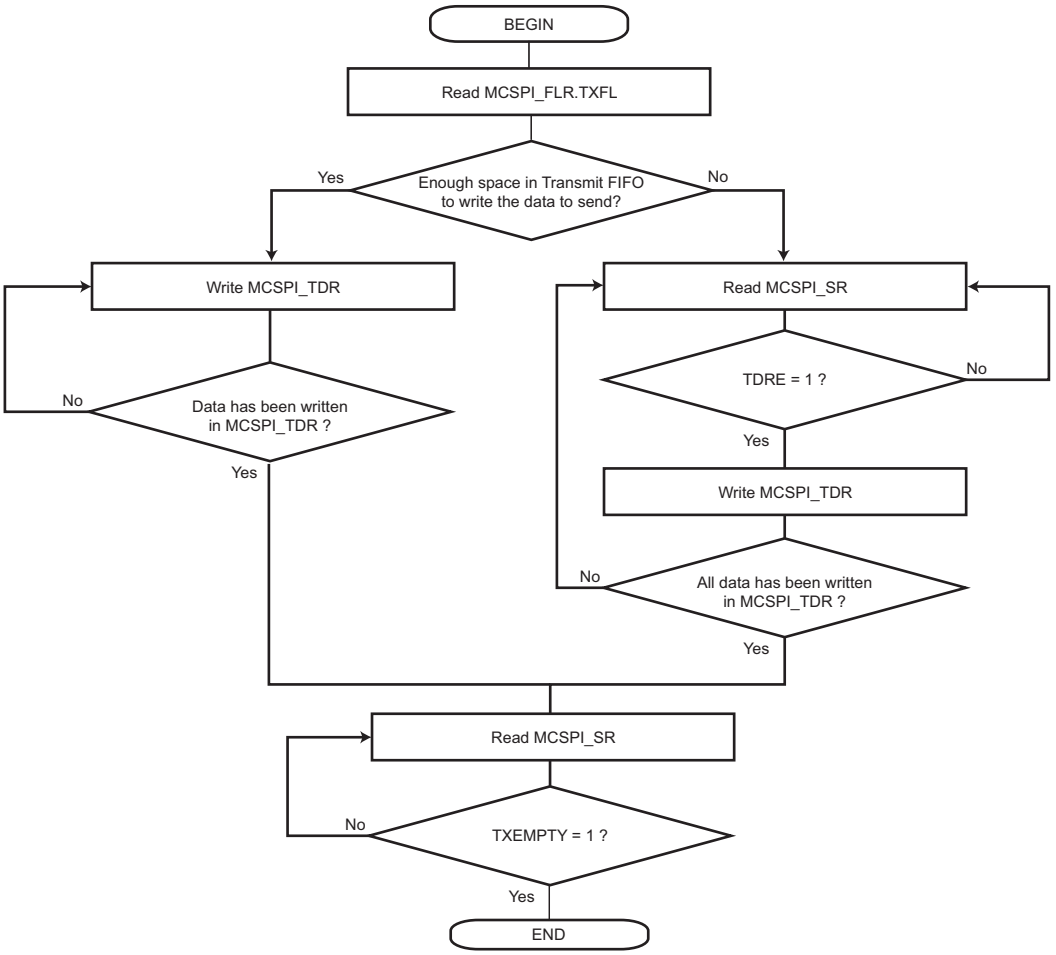
The FIFO level is provided in the Transmit FIFO (TXFL) Level field in the FIFO Level register (MCSPI_FLR). If the FIFO can accept the number of data to be transmitted, there is no need to monitor MCSPI_SR.TDRE and the data can be successively written in MCSPI_TDR.

If the FIFO cannot accept the data due to insufficient space, wait for the TDRE flag to be set before writing the data in MCSPI_TDR.

When the space in the FIFO allows only a portion of the data to be written, the TDRE flag must be monitored before writing the remaining data.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 45-21. Sending Data with FIFO



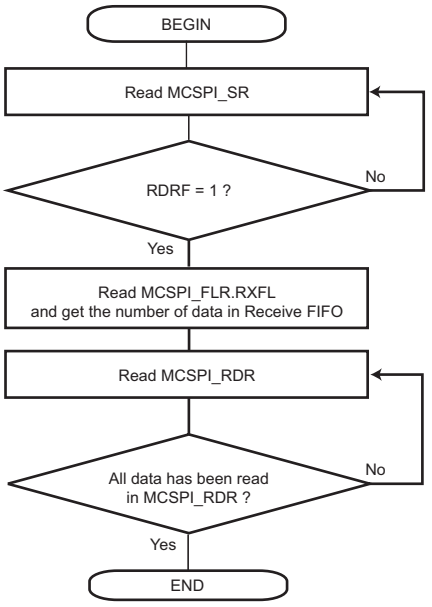
45.7.8.2. Receiving Data with FIFO Enabled

When the Receive FIFO is enabled, MCSPI_RDR access reads the FIFO.

When data are present in the Receive FIFO (RDRF flag set to '1'), the exact number of data can be checked with MCSPI_FLR.RXFL. All the data can be read successively in MCSPI_RDR without checking the RDRF flag between each access.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 45-22. Receiving Data with FIFO



45.7.8.3. Clearing/Flushing FIFOs

Each FIFO can be cleared/flushed using MCSPI_CR.TXFCLR/RXFCLR.

45.7.8.4. TXEMPTY, TDRE and RDRF Behavior

The MCSPI_SR.TXEMPTY, MCSPI_SR.TDRE and MCSPI_SR.RDRF flags display a specific behavior when FIFOs are enabled.

The TXEMPTY flag is cleared as long as there are characters in the Transmit FIFO or in the internal shift register. TXEMPTY is set when there are no characters in the Transmit FIFO and in the internal shift register.

TDRE indicates if a data can be written in the Transmit FIFO. Thus the TDRE flag is set as long as the Transmit FIFO can accept new data. See Figure 45-23.

RDRF indicates if an unread data is present in the Receive FIFO. Thus the RDRF flag is set as soon as one unread data is in the Receive FIFO. See Figure 45-24.

TDRE and RDRF behavior can be modified using the Transmit Data Register Empty Mode (TXRDYM) and Receive Data Register Empty Mode (RXRDYM) fields in the MCSPI FIFO Mode register (MCSPI_FMR) to reduce the number of accesses to MCSPI_TDR/RDR. However, for some configurations, the following constraints apply:

- When the Variable Peripheral Select mode is used (MCSPI_MR.PS=1), MCSPI_FMR.TXRDYM/RXRDYM must be cleared.
- In Host mode (MCSPI_MR.MSTR=1), MCSPI_FMR.RXRDYM must be cleared.

As an example, in Host mode, the Transmit FIFO can be loaded with multiple data in the same access by configuring TXRDYM>0.

See MCSPI_FMR for the FIFO configuration.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 45-23. TDRE in Single Data Mode and TXRDYM=0

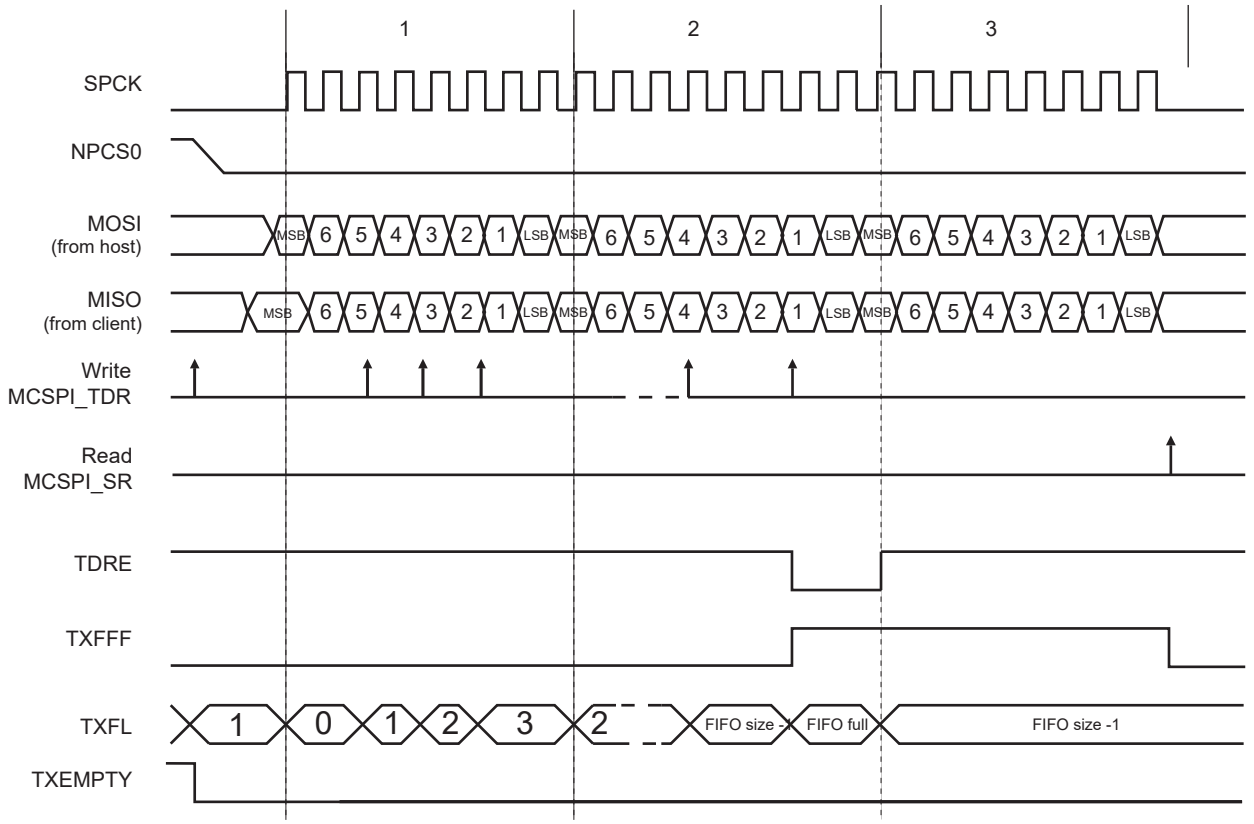
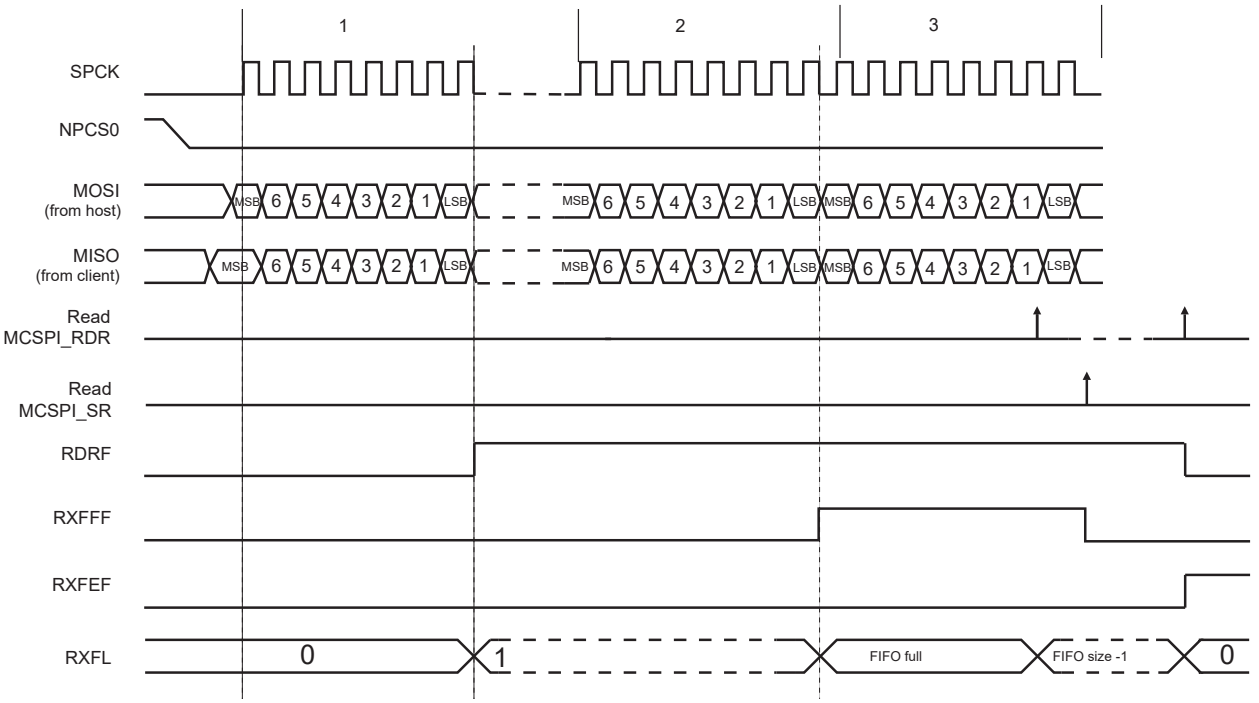


Figure 45-24. RDRF in Single Data Mode and RXRDYM=0



© 2026 Microchip Technology Inc. and its subsidiaries

45.7.8.5. Single Data Mode

In Single Data mode, only one data is written every time MCSPI_TDR is accessed, and only one data is read every time MCSPI_RDR is accessed.

When MCSPI_FMR.TXRDYM = 0, the Transmit FIFO operates in Single Data mode.

When MCSPI_FMR.RXRDYM = 0, the Receive FIFO operates in Single Data mode.

If Host mode is used (MCSPI_MR.MSTR=1), the Receive FIFO must operate in Single Data mode.

If Variable Peripheral Select mode is used (MCSPI_MR.PS=1), the Transmit FIFO must operate in Single Data mode.

See MCSPI_RDR and MCSPI_TDR.

45.7.8.5.1. PDC

When FIFOs operate in Single Data mode, the PDC transfer type must be configured either in bytes, halfwords or words depending on the values of the Peripheral Select (PS) bit in MCSPI_MR and the MCSPI_CSRx.BITS field.

The same conditions for transfer type apply when FIFOs are disabled.

45.7.8.6. Multiple Data Mode

Multiple Data mode minimizes the number of accesses by concatenating the data to send/read in one access.

When MCSPI_FMR.TXRDYM > 0, the Transmit FIFO operates in Multiple Data mode.

When MCSPI_FMR.RXRDYM > 0, the Receive FIFO operates in Multiple Data mode.

Multiple data can be read from the Receive FIFO only in Client mode (MCSPI_MR.MSTR=0).

The Transmit FIFO can be loaded with multiple data in the same access by configuring TXRDYM>0 and when MCSPI_MR.PS=0.

In Multiple Data mode, up to two data can be written in one MCSPI_TDR write access. It is also possible to read up to four data in one MCSPI_RDR access if MCSPI_CSRx.BITS is configured to '0' (8-bit data size) and up to two data if MCSPI_CSRx.BITS is configured to a value other than '0' (more than 8-bit data size).

The number of data to write/read is defined by the size of the register access. If the access is a byte-size register access, only one data is written/read. If the access is a halfword size register access, then up to two data are read and only one data is written. Lastly, if the access is a word-size register access, then up to four data are read and up to two data are written.

Written/read data are always right-aligned, as described in MCSPI_RDR (FIFO_MULTI_DATA_8), MCSPI_RDR (FIFO_MULTI_DATA_16) and MCSPI_TDR (FIFO_MULTI_DATA).

As an example, if the Transmit FIFO is empty and there are six data to send, either of the following write accesses may be performed:

- six MCSPI_TDR-byte write accesses
- three MCSPI_TDR-halfword write accesses

With a Receive FIFO containing six data, any of the following read accesses may be performed:

- six MCSPI_RDR-byte read accesses
- three MCSPI_RDR-halfword read accesses
- one MCSPI_RDR-word read access and one MCSPI_RDR-halfword read access

45.7.8.6.1. TDRE and RDRF Configuration

In Multiple Data mode, it is possible to write one or more data in the same MCSPI_TDR/MCSPI_RDR access. The TDRE flag indicates if one or more data can be written in the FIFO depending on the configuration of MCSPI_FMR.TXRDYM/RXRDYM.

© 2026 Microchip Technology Inc. and its subsidiaries

As an example, if two data are written each time in MCSPI_TDR, it is useful to configure the TXRDYM field to the value '1' so that the TDRE flag is at '1' only when at least two data can be written in the Transmit FIFO.

Similarly, if four data are read each time in MCSPI_RDR, it is useful to configure the RXRDYM field to the value '2' so that the RDRF flag is at '1' only when at least four unread data are in the Receive FIFO.

45.7.8.6.2. PDC

It is mandatory to configure PDC channel size (byte, halfword or word) according to FLEX_MCSPI_FMR.TXRDYM/RXRDYM configuration. See Multiple Data Mode for constraints.

45.7.8.7. FIFO Pointer Error

A FIFO overflow is reported in MCSPI_SR.

If the Transmit FIFO is full and a write access is performed on MCSPI_TDR, it generates a Transmit FIFO pointer error and sets MCSPI_SR.TXFPTEF.

In Multiple Data mode, if the number of data written in MCSPI_TDR (according to the register access size) is greater than the free space in the Transmit FIFO, a Transmit FIFO pointer error is generated and MCSPI_SR.TXFPTEF is set.

A FIFO underflow is reported in MCSPI_SR.

In Multiple Data mode, if the number of data read in MCSPI_RDR (according to the register access size) is greater than the number of unread data in the Receive FIFO, a Receive FIFO pointer error is generated and MCSPI_SR.RXFPTEF is set.

No pointer error occurs if the FIFO state/level is checked before writing/reading in MCSPI_TDR/MCSPI_RDR. The FIFO state/level can be checked either with TXRDY, RXRDY, TXFL or RXFL. When a pointer error occurs, other FIFO flags may not behave as expected; their states should be ignored.

If a pointer error occurs, a software reset must be performed using MCSPI_CR.SWRST (configuration will be lost).

45.7.8.8. FIFO Thresholds

Each Transmit and Receive FIFO includes a threshold feature used to set a flag and an interrupt when a FIFO threshold is crossed. Thresholds are defined as a number of data in the FIFO, and the FIFO state (TXFL or RXFL) represents the number of data currently in the FIFO.

The Transmit FIFO threshold can be set using MCSPI_FMR.TXFTHRES. Each time the Transmit FIFO goes from the 'above threshold' to the 'equal or below threshold' state, MCSPI_SR.TXFTHF is set. The application is warned that the Transmit FIFO has reached the defined threshold and that it can be reloaded.

The Receive FIFO threshold can be set using MCSPI_FMR.RXFTHRES. Each time the Receive FIFO goes from the 'below threshold' to the 'equal or above threshold' state, MCSPI_SR.RXFTHF is set. The application is warned that the Receive FIFO has reached the defined threshold and that it can be read to prevent an underflow.

The TXFTHF and RXFTHF flags can be configured to generate an interrupt using the Interrupt Enable register (MCSPI_IER) and the Interrupt Disable register (MCSPI_IDR).

45.7.8.9. FIFO Flags

FIFOs come with a set of flags which can be configured to generate interrupts through MCSPI_IER and MCSPI_IDR.

FIFO flag states can be read in MCSPI_SR. They are cleared when MCSPI_SR is read.

45.7.9. Register Write Protection

To prevent any single software error from corrupting MCSPI behavior, certain registers in the address space can be write-protected in the MCSPI Write Protection Mode Register (MCSPI_WPMR).

© 2026 Microchip Technology Inc. and its subsidiaries

If a write access to a write-protected register is detected, the Write Protection Violation Status (WPVS) bit in the MCSPI Write Protection Status Register (MCSPI_WPSR) is set and the Write Protection Violation Source (WPVSRC) field indicates the register in which the write access has been attempted.

The WPVS bit is automatically cleared after reading MCSPI_WPSR.

The following registers are write-protected when WPEN is set in MCSPI_WPMR:

- MCSPI Mode Register
- MCSPI Chip Select Register
- MCSPI CRC Register
- MCSPI Two-Pin Mode Register

The following register is write-protected when WPCREN is set in MCSPI_WPMR:

- MCSPI Control Register

The following registers are write-protected when WPITEN is set in MCSPI_WPMR:

- MCSPI Interrupt Enable Register
- MCSPI Interrupt Disable Register

© 2026 Microchip Technology Inc. and its subsidiaries

45.8. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	MCSPI_CR	31:24	FIFODIS	FIFOEN						LASTXFER
		23:16							RXFCLR	TXFCLR
		15:8				REQCLR				
		7:0	SWRST						SPIDIS	SPIEN
0x04	MCSPI_MR	31:24	DLYBCS[7:0]							
		23:16					PCS[3:0]			
		15:8	MOSIIE	CSIE	TPMEN	CMPMODE			TMCSMUX	LSBHALF
		7:0	LLB	CRCEN	WDRBT	MODFDIS	BRSRCCLK	PCSDEC	PS	MSTR
0x08	MCSPI_RDR	31:24								
		23:16					PCS[3:0]			
		15:8	RD[15:8]							
		7:0	RD[7:0]							
0x08	MCSPI_RDR (FIFO_MULTI_DATA_8)	31:24	RD3[7:0]							
		23:16	RD2[7:0]							
		15:8	RD1[7:0]							
		7:0	RD0[7:0]							
0x08	MCSPI_RDR (FIFO_MULTI_DATA_16)	31:24	RD1[15:8]							
		23:16	RD1[7:0]							
		15:8	RD0[15:8]							
		7:0	RD0[7:0]							
0x0C	MCSPI_TDR	31:24								LASTXFER
		23:16					PCS[3:0]			
		15:8	TD[15:8]							
		7:0	TD[7:0]							
0x0C	MCSPI_TDR (FIFO_MULTI_DATA)	31:24	TD1[15:8]							
		23:16	TD1[7:0]							
		15:8	TD0[15:8]							
		7:0	TD0[7:0]							
0x10	MCSPI_SR	31:24	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
		23:16								SPIENS
		15:8			CRCERR	SFERR	CMP	UNDES	TXEMPTY	NSSR
		7:0	TXBUFE	RXBUFF	ENDTX	ENDRX	OVRES	MODF	TDRE	RDRF
0x14	MCSPI_IER	31:24	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
		23:16								
		15:8			CRCERR		CMP	UNDES	TXEMPTY	NSSR
		7:0	TXBUFE	RXBUFF	ENDTX	ENDRX	OVRES	MODF	TDRE	RDRF
0x18	MCSPI_IDR	31:24	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
		23:16								
		15:8			CRCERR		CMP	UNDES	TXEMPTY	NSSR
		7:0	TXBUFE	RXBUFF	ENDTX	ENDRX	OVRES	MODF	TDRE	RDRF
0x1C	MCSPI_IMR	31:24	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
		23:16								
		15:8			CRCERR		CMP	UNDES	TXEMPTY	NSSR
		7:0	TXBUFE	RXBUFF	ENDTX	ENDRX	OVRES	MODF	TDRE	RDRF
0x20 ... 0x2F	Reserved									
0x30	MCSPI_CSR0	31:24	DLYBCT[7:0]							
		23:16	DLYBS[7:0]							
		15:8	SCBR[7:0]							
		7:0	BITS[3:0]				CSAAT	CSNAAT	NCPHA	CPOL
0x34	MCSPI_CSR1	31:24	DLYBCT[7:0]							
		23:16	DLYBS[7:0]							
		15:8	SCBR[7:0]							
		7:0	BITS[3:0]				CSAAT	CSNAAT	NCPHA	CPOL
0x38	MCSPI_CSR2	31:24	DLYBCT[7:0]							
		23:16	DLYBS[7:0]							
		15:8	SCBR[7:0]							
		7:0	BITS[3:0]				CSAAT	CSNAAT	NCPHA	CPOL

© 2026 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x3C	MCSPI_CSR3	31:24	DLYBCT[7:0]							
		23:16	DLYBS[7:0]							
		15:8	SCBR[7:0]							
		7:0	BITS[3:0]				CSAAT	CSNAAT	NCPHA	CPOL
0x40	MCSPI_FMR	31:24			RXFTHRES[5:0]					
		23:16			TXFTHRES[5:0]					
		15:8								
		7:0			RXRDYM[1:0]				TXRDYM[1:0]	
0x44	MCSPI_FLR	31:24								
		23:16			RXFL[5:0]					
		15:8								
		7:0			TXFL[5:0]					
0x48	MCSPI_CMPR	31:24	VAL2[15:8]							
		23:16	VAL2[7:0]							
		15:8	VAL1[15:8]							
		7:0	VAL1[7:0]							
0x4C	MCSPI_CRCR	31:24							FHE	CRM
		23:16	FRHL[3:0]							CRCS
		15:8								
		7:0	FRL[7:0]							
0x50	MCSPI_TPMR	31:24								
		23:16								
		15:8								
		7:0					OSR[1:0]		MIL	CSM
0x54	MCSPI_TPHR	31:24								
		23:16								
		15:8								
		7:0		OSR[1:0]		GAIN[1:0]		BOOST	CNT[1:0]	
0x58 ... 0xE3	Reserved									
0xE4	MCSPI_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0						WPCREN	WPITEN	WPEN
0xE8	MCSPI_WPSR	31:24								
		23:16								
		15:8	WPVSRC[7:0]							
		7:0								WPVS
0xEC ... 0xFF	Reserved									
0x0100 ... 0x124	Reserved for PDC registers	31:24								
		23:16								
		15:8								
		7:0								

© 2026 Microchip Technology Inc. and its subsidiaries

45.8.1. MCSPI Control Register

Name: MCSPI_CR
Offset: 0x00
Reset: –
Property: Write-only

This register can only be written if the WPCREN bit is cleared in the MCSPI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	FIFODIS	FIFOEN						LASTXFER
Access	W	W						W
Reset	–	–						–

Bit	23	22	21	20	19	18	17	16
							RXFCLR	TXFCLR
Access							W	W
Reset							–	–

Bit	15	14	13	12	11	10	9	8
				REQCLR				
Access				W				
Reset				–				

Bit	7	6	5	4	3	2	1	0
	SWRST						SPIDIS	SPIEN
Access	W						W	W
Reset	–						–	–

Bit 31 – FIFODIS FIFO Disable

Value	Description
0	No effect.
1	Disables the Transmit and Receive FIFOs.

Bit 30 – FIFOEN FIFO Enable

Value	Description
0	No effect.
1	Enables the Transmit and Receive FIFOs.

Bit 24 – LASTXFER Last Transfer
See Peripheral Selection for more details.

Value	Description
0	No effect.
1	The current NPCS is deasserted after the character written in TD has been transferred. When MCSPI_CSRx.CSAAT is set, the communication with the current serial peripheral can be closed by raising the corresponding NPCS line as soon as TD transfer is completed.

Bit 17 – RXFCLR Receive FIFO Clear

Value	Description
0	No effect.
1	Empties the Receive FIFO.

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 16 – TXFCLR Transmit FIFO Clear

Value	Description
0	No effect.
1	Empties the Transmit FIFO.

Bit 12 – REQCLR Request to Clear the Comparison Trigger

Value	Description
0	No effect.
1	Restarts the comparison trigger to enable MCSPI_RDR loading.

Bit 7 – SWRST MCSPI Software Reset

The MCSPI is in Client mode after software reset.
PDC channels are not affected by software reset.

Value	Description
0	No effect.
1	Resets the MCSPI. A software-triggered hardware reset of the MCSPI interface is performed.

Bit 1 – SPIDIS MCSPI Disable

All pins are set in Input mode after completion of the transmission in progress, if any.
If a transfer is in progress when SPIDIS is set, the MCSPI completes the transmission of the shifter register and does not start any new transfer, even if MCSPI_THR is loaded.
Note: If both SPIEN and SPIDIS are equal to one when MCSPI_CR is written, the MCSPI is disabled.

Value	Description
0	No effect.
1	Disables the MCSPI.

Bit 0 – SPIEN MCSPI Enable

Value	Description
0	No effect.
1	Enables the MCSPI to transfer and receive data.

© 2026 Microchip Technology Inc. and its subsidiaries

45.8.2. MCSPI Mode Register

Name: MCSPI_MR
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the MCSPI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	DLYBCS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
					PCS[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	MOSIIE	CSIE	TPMEN	CMPMODE			TMCSMUX	LSBHALF
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	0	0	0			0	0

Bit	7	6	5	4	3	2	1	0
	LLB	CRCEN	WDRBT	MODFDIS	BRSRCCLK	PCSDEC	PS	MSTR
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – DLYBCS[7:0] Delay Between Chip Selects
This field defines the delay between the inactivation and the activation of NPCS. The DLYBCS time guarantees nonoverlapping chip selects and solves bus contentions in case of peripherals having long data float times.
If DLYBCS is lower than 6, six peripheral clock periods are inserted by default.
Otherwise, the following equations determine the delay:
If BRSRCCLK = 0:

$$\text{Delay Between Chip Selects} = \frac{\text{DLYBCS}}{\text{f}_{\text{peripheral clock}}}$$

If BRSRCCLK = 1:

$$\text{Delay Between Chip Selects} = \frac{\text{DLYBCS}}{\text{f}_{\text{GCLK}}}$$

Bits 19:16 – PCS[3:0] Peripheral Chip Select
This field is only used if fixed peripheral select is active (PS = 0).
If MCSPI_MR.PCSDEC = 0:
PCS = xxx0 NPCS[3:0] = 1110
PCS = xx01 NPCS[3:0] = 1101
PCS = x011 NPCS[3:0] = 1011
PCS = 0111 NPCS[3:0] = 0111
PCS = 1111 forbidden (no peripheral is selected)
(x = don't care)
If MCSPI_MR.PCSDEC = 1:
NPCS[3:0] output signals = PCS.

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 15 – MOSIIE MOSI Inversion Enable

Value	Description
0	MOSI inputs are used “as is” by the IP.
1	MOSI inputs are internally inverted before being used by the IP.

Bit 14 – CSIE Chip Select Inversion Enable

Value	Description
0	Chip select NPCS IOs use the MCSPI standard “active low” scheme.
1	Chip select NPCS IOs use an “active high” scheme

Bit 13 – TPMEN Two-Pin Mode Enable

Value	Description
0	Two-Wire mode is disabled.
1	Two-Wire mode is enabled.

Bit 12 – CMPMODE Comparison Mode

Value	Name	Description
0	FLAG_ONLY	Any character is received and comparison function drives CMP flag.
1	START_CONDITION	Comparison condition must be met to start reception of all incoming characters until REQCLR is set.

Bit 9 – TMCSMUX Two-pin MOSI Chip Select External Multiplexing Mode

Value	Name	Description
0	DISABLED	Multiplexing of external MOSI lines is not required via Chip Select lines. When MCSPI_MR.TPMEN=0, MCSPI_TWMR.MIL=1 or MCSPI_TWMR.CSM=0, TMCSMUX must be written to 0.
1	ENABLED	Enables external multiplexing of MOSI lines via Chip Select lines. When MCSPI_MR.TPMEN=1, TMCSMUX must be written to 1 if MCSPI_TWMR.MIL=0 and MCSPI_TWMR.CSM=1.

Bit 8 – LSBHALF LSB Timing Selection

Value	Description
0	To be used only if SPI client LSB timing is 100% compliant with SPI standard (LSB duration is a full bit time). This value provides the best margin for SPI client response delay (less than 1 SPCK clock cycle).
1	To be selected if the SPI client LSB timing does not behave as the SPI standard (not triggered by NPCS deassertion in mode); the client response delay is limited to less than 1/2 SPCK cycle.

Bit 7 – LLB Local Loopback Enable

LLB controls the local loopback on the data shift register for testing in Host mode only (MISO is internally connected on MOSI).

Value	Description
0	Local loopback path disabled.
1	Local loopback path enabled.

Bit 6 – CRCEN CRC Enable

Value	Description
0	CRC calculation is disabled.
1	CRC calculation is enabled. The BITS field in MCSPI_CSRx registers must be at value ‘0’.

Bit 5 – WDRBT Wait for Data Read Before Transfer

Value	Description
0	No Effect. In Host mode, a transfer can be initiated regardless of MCSPI_RDR state.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
1	In Host mode, a transfer can start only if MCSPI_RDR is empty, i.e., does not contain any unread data. This mode prevents overrun error in reception.

Bit 4 – MODFDIS Mode Fault Detection

Value	Description
0	Mode fault detection enabled
1	Mode fault detection disabled

Bit 3 – BRSRCCLK Bit Rate Source Clock

0 (PERIPH_CLK): The peripheral clock is the source clock for the bit rate generation.
1 (GCLK): PMC GCLK is the source clock for the bit rate generation, thus the bit rate can be independent of the core/peripheral clock.
Note: If bit BRSRCCLK = 1, the MCSPI_CSRx.SCBR field must be programmed with a value greater than 1.

Bit 2 – PCSDEC Chip Select Decode

When PCSDEC = 1, up to 15 chip select signals can be generated with the four NPCS lines using an external 4-bit to 16-bit decoder. The chip select registers define the characteristics of the 15 chip selects, with the following rules:
MCSPI_CSR0 defines peripheral chip select signals 0 to 3.
MCSPI_CSR1 defines peripheral chip select signals 4 to 7.
MCSPI_CSR2 defines peripheral chip select signals 8 to 11.
MCSPI_CSR3 defines peripheral chip select signals 12 to 14.

Value	Description
0	The chip select lines are directly connected to a peripheral device.
1	The four NPCS chip select lines are connected to a 4-bit to 16-bit decoder.

Bit 1 – PS Peripheral Select

Value	Description
0	Fixed Peripheral Select
1	Variable Peripheral Select

Bit 0 – MSTR Host/Client Mode

Value	Description
0	MCSPI is in Client mode.
1	MCSPI is in Host mode.

© 2026 Microchip Technology Inc. and its subsidiaries

45.8.3. MCSPI Receive Data Register

Name: MCSPI_RDR
Offset: 0x08
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					PCS[3:0]			
Access					R	R	R	R
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RD[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RD[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 19:16 – PCS[3:0] Peripheral Chip Select
In Host mode only, these bits indicate the value on the NPCS pins at the end of a transfer. Otherwise, these bits are read as zero.
Note: When using Variable Peripheral Select mode (PS = 1 in MCSPI_MR), it is mandatory to set the MCSPI_MR.WDRBT bit if the PCS field must be processed in MCSPI_RDR.

Bits 15:0 – RD[15:0] Receive Data
Data received by the MCSPI Interface is stored in this register in a right-justified format. Unused bits are read as zero.

© 2026 Microchip Technology Inc. and its subsidiaries

45.8.4. MCSPI Receive Data Register (FIFO Multiple Data, 8-bit)

Name: MCSPI_RDR (FIFO_MULTI_DATA_8)
Offset: 0x08
Reset: 0x00000000
Property: Read-only

If FIFO is enabled (MCSPI_CR.FIFOEN bit), see Multiple Data Mode.

Bit	31	30	29	28	27	26	25	24
	RD3[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RD2[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RD1[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RD0[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0:7, 8:15, 16:23, 24:31 – RDx Receive Data
First unread data in the Receive FIFO. Data received by the MCSPI interface is stored in this register in a right-justified format. Unused bits are read as zero.

© 2026 Microchip Technology Inc. and its subsidiaries

45.8.5. MCSPI Receive Data Register (FIFO Multiple Data, 16-bit)

Name: MCSPI_RDR (FIFO_MULTI_DATA_16)
Offset: 0x08
Reset: 0x00000000
Property: Read-only

If FIFO is enabled (MCSPI_CR.FIFOEN bit), see Multiple Data Mode.

Bit	31	30	29	28	27	26	25	24
	RD1[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RD1[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RD0[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RD0[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0:15, 16:31 – RDx Receive Data
First unread data in the Receive FIFO. Data received by the MCSPI interface is stored in this register in a right-justified format. Unused bits are read as zero.

© 2026 Microchip Technology Inc. and its subsidiaries

45.8.6. MCSPI Transmit Data Register

Name: MCSPI_TDR
Offset: 0x0C
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
								LASTXFER
Access								W
Reset								–

Bit	23	22	21	20	19	18	17	16
					PCS[3:0]			
Access					W	W	W	W
Reset					–	–	–	–

Bit	15	14	13	12	11	10	9	8
	TD[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	TD[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 24 – LASTXFER Last Transfer
This field is only used if variable peripheral select is active (MCSPI_MR.PS = 1).

Value	Description
0	No effect
1	The current NPCS is deasserted after the transfer of the character written in TD. When MCSPI_CSRx.CSAAT is set, the communication with the current serial peripheral can be closed by raising the corresponding NPCS line as soon as TD transfer is completed.

Bits 19:16 – PCS[3:0] Peripheral Chip Select
This field is only used if variable peripheral select is active (MCSPI_MR.PS = 1).
If MCSPI_MR.PCSDEC = 0:
PCS = xxx0 NPCS[3:0] = 1110
PCS = xx01 NPCS[3:0] = 1101
PCS = x011 NPCS[3:0] = 1011
PCS = 0111 NPCS[3:0] = 0111
PCS = 1111 forbidden (no peripheral is selected)
(x = don't care)
If MCSPI_MR.PCSDEC = 1:
NPCS[3:0] output signals = PCS

Bits 15:0 – TD[15:0] Transmit Data
Data to be transmitted by the MCSPI interface is stored in this register. Information to be transmitted must be written to this register in a right-justified format.

© 2026 Microchip Technology Inc. and its subsidiaries

45.8.7. MCSPI Transmit Data Register (FIFO Multiple Data, 8- to 16-bit)

Name: MCSPI_TDR (FIFO_MULTI_DATA)
Offset: 0x0C
Reset: –
Property: Write-only

If FIFO is enabled (MCSPI_CR.FIFOEN bit), see Multiple Data Mode.

Bit	31	30	29	28	27	26	25	24
	TD1[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	TD1[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	TD0[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	TD0[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0:15, 16:31, 32:47, 48:63 – TDx Transmit Data
Next data to write in the Transmit FIFO. Information to be transmitted must be written to this register in a right-justified format.

© 2026 Microchip Technology Inc. and its subsidiaries

45.8.8. MCSPI Status Register

Name: MCSPI_SR
Offset: 0x10
Reset: 0x000000F0
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
								SPIENS
Access								R
Reset								0

Bit	15	14	13	12	11	10	9	8
			CRCERR	SFERR	CMP	UNDES	TXEMPTY	NSSR
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TXBUFE	RXBUFF	ENDTX	ENDRX	OVRES	MODF	TDRE	RDRF
Access	R	R	R	R	R	R	R	R
Reset	1	1	1	1	0	0	0	0

Bit 31 – RXFPTEF Receive FIFO Pointer Error Flag
See FIFO Pointer Error for details.

Value	Description
0	No Receive FIFO pointer occurred
1	Receive FIFO pointer error occurred. Receiver must be reset.

Bit 30 – TXFPTEF Transmit FIFO Pointer Error Flag
See FIFO Pointer Error for details.

Value	Description
0	No Transmit FIFO pointer occurred
1	Transmit FIFO pointer error occurred. Transceiver must be reset

Bit 29 – RXFTHF Receive FIFO Threshold Flag

Value	Description
0	Number of unread data in Receive FIFO is below RXFTHRES threshold or RXFTH flag has been cleared.
1	Number of unread data in Receive FIFO has reached RXFTHRES threshold (coming from “below threshold” state to “equal or above threshold” state).

Bit 28 – RXFFF Receive FIFO Full Flag

Value	Description
0	Receive FIFO is not empty or RXFE flag has been cleared.
1	Receive FIFO has become full (coming from “not full” state to “full” state).

Bit 27 – RXFEF Receive FIFO Empty Flag

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	Receive FIFO is not empty or RXFE flag has been cleared.
1	Receive FIFO has become empty (coming from “not empty” state to “empty” state).

Bit 26 – TXFTHF Transmit FIFO Threshold Flag (cleared on read)

Value	Description
0	Number of data in Transmit FIFO is above TXFTHRES threshold.
1	Number of data in Transmit FIFO has reached TXFTHRES threshold since the last read of MCSPI_SR.

Bit 25 – TXFFF Transmit FIFO Full Flag (cleared on read)

Value	Description
0	Transmit FIFO is not full or TXFF flag has been cleared.
1	Transmit FIFO has been filled since the last read of MCSPI_SR.

Bit 24 – TXFEF Transmit FIFO Empty Flag (cleared on read)

Value	Description
0	Transmit FIFO is not empty.
1	Transmit FIFO has been emptied since the last read of MCSPI_SR.

Bit 16 – SPIENS MCSPI Enable Status

Value	Description
0	MCSPI is disabled.
1	MCSPI is enabled.

Bit 13 – CRCERR CRC Error (cleared on read)

Value	Description
0	CRC calculation is disabled or no received frame contains CRC error since the last read of MCSPI_SR.
1	Since the last read of MCSPI_SR, a received frame contains a CRC error.

Bit 12 – SFERR Client Frame Error (cleared on read)

Value	Description
0	There is no frame error detected for a client access since the last read of MCSPI_SR.
1	In Client mode, the chip select raised while the transfer of the character defined in MCSPI_CSR0.BITS was not complete.

Bit 11 – CMP Comparison Status (cleared on read)

Value	Description
0	No received character matched the comparison criteria programmed in the VAL1 and VAL2 fields in MCSPI_CMPR since the last read of MCSPI_SR.
1	A received character matched the comparison criteria since the last read of MCSPI_SR.

Bit 10 – UNDES Underrun Error Status (Client mode only) (cleared on read)

Value	Description
0	No underrun has been detected since the last read of MCSPI_SR.
1	A transfer starts whereas no data has been loaded in MCSPI_TDR.

Bit 9 – TXEMPTY Transmission Registers Empty (cleared by writing MCSPI_TDR)

Value	Description
0	As soon as data is written in MCSPI_TDR.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
1	MCSPI_TDR and internal shift register are empty. If a transfer delay has been defined, TXEMPTY is set after the end of this delay.

Bit 8 – NSSR NSS Rising (cleared on read)

Value	Description
0	No rising edge detected on NSS pin since the last read of MCSPI_SR.
1	A rising edge occurred on NSS pin since the last read of MCSPI_SR.

Bit 7 – TXBUFE TX Buffer Empty (cleared by writing MCSPI_TCR or MCSPI_TNCR)
Note: MCSPI_TCR and MCSPI_TNCR are PDC registers.

Value	Description
0	MCSPI_TCR or MCSPI_TNCR has a value other than 0.
1	Both MCSPI_TCR and MCSPI_TNCR have a value of 0.

Bit 6 – RXBUFF RX Buffer Full (cleared by writing MCSPI_RCR or MCSPI_RNCR)
Note: MCSPI_RCR and MCSPI_RNCR are PDC registers.

Value	Description
0	MCSPI_RCR or MCSPI_RNCR has a value other than 0.
1	Both MCSPI_RCR and MCSPI_RNCR have a value of 0.

Bit 5 – ENDTX End of TX Buffer (cleared by writing MCSPI_TCR or MCSPI_TNCR)
Note: MCSPI_TCR and MCSPI_TNCR are PDC registers.

Value	Description
0	The Transmit Counter register has not reached 0 since the last write in MCSPI_TCR or MCSPI_TNCR.
1	The Transmit Counter register has reached 0 since the last write in MCSPI_TCR or MCSPI_TNCR.

Bit 4 – ENDRX End of RX Buffer (cleared by writing MCSPI_RCR or MCSPI_RNCR)
Note: MCSPI_RCR and MCSPI_RNCR are PDC registers.

Value	Description
0	The Receive Counter register has not reached 0 since the last write in MCSPI_RCR or MCSPI_RNCR.
1	The Receive Counter register has reached 0 since the last write in MCSPI_RCR or MCSPI_RNCR.

Bit 3 – OVRES Overrun Error Status (cleared on read)
An overrun occurs when MCSPI_RDR is loaded at least twice from the internal shift register since the last read of MCSPI_RDR.

Value	Description
0	No overrun has been detected since the last read of MCSPI_SR.
1	An overrun has occurred since the last read of MCSPI_SR.

Bit 2 – MODF Mode Fault Error (cleared on read)

Value	Description
0	No mode fault has been detected since the last read of MCSPI_SR.
1	A mode fault occurred since the last read of MCSPI_SR.

Bit 1 – TDRE Transmit Data Register Empty (cleared by writing MCSPI_TDR)
When FIFOs are disabled:
0: Data has been written to MCSPI_TDR and not yet transferred to the internal shift register.
1: The last data written in MCSPI_TDR has been transferred to the internal shift register.
TDRE is cleared when the MCSPI is disabled or at reset. Enabling the MCSPI sets the TDRE flag.
When FIFOs are enabled:

© 2026 Microchip Technology Inc. and its subsidiaries

0: Transmit FIFO is full and cannot accept more data.
1: Transmit FIFO is not full; one or more data can be written according to TXRDYM field configuration.
TDRE behavior with FIFO enabled is illustrated in TXEMPTY, TDRE and RDRF Behavior.

Bit 0 – RDRF Receive Data Register Full (cleared by reading MCSPI_RDR)
When FIFOs are disabled:
0: No data has been received since the last read of MCSPI_RDR.
1: Data has been received and the received data has been transferred from the internal shift register to MCSPI_RDR since the last read of MCSPI_RDR.
When FIFOs are enabled:
0: Receive FIFO is empty; no data to read.
1: At least one unread data is in the Receive FIFO.
RDRF behavior with FIFO enabled is illustrated in TXEMPTY, TDRE and RDRF Behavior.

© 2026 Microchip Technology Inc. and its subsidiaries

45.8.9. MCSPI Interrupt Enable Register

Name: MCSPI_IER
Offset: 0x14
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the MCSPI Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			CRCERR		CMP	UNDES	TXEMPTY	NSSR
Access			W		W	W	W	W
Reset			–		–	–	–	–

Bit	7	6	5	4	3	2	1	0
	TXBUFE	RXBUFF	ENDTX	ENDRX	OVRES	MODF	TDRE	RDRF
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 31 – RXFPTEF RXFPTEF Interrupt Enable

Bit 30 – TXFPTEF TXFPTEF Interrupt Enable

Bit 29 – RXFTHF RXFTHF Interrupt Enable

Bit 28 – RXFFF RXFFF Interrupt Enable

Bit 27 – RXFEF RXFEF Interrupt Enable

Bit 26 – TXFTHF TXFTHF Interrupt Enable

Bit 25 – TXFFF TXFFF Interrupt Enable

Bit 24 – TXFEF TXFEF Interrupt Enable

Bit 13 – CRCERR CRC Error Interrupt Enable

Bit 11 – CMP Comparison Interrupt Enable

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 10 – UNDES Underrun Error Interrupt Enable

Bit 9 – TXEMPTY Transmission Registers Empty Enable

Bit 8 – NSSR NSS Rising Interrupt Enable

Bit 7 – TXBUFE Transmit Buffer Empty Interrupt Enable

Bit 6 – RXBUFF Receive Buffer Full Interrupt Enable

Bit 5 – ENDTX End of Transmit Buffer Interrupt Enable

Bit 4 – ENDRX End of Receive Buffer Interrupt Enable

Bit 3 – OVRES Overrun Error Interrupt Enable

Bit 2 – MODF Mode Fault Error Interrupt Enable

Bit 1 – TDRE MCSPI Transmit Data Register Empty Interrupt Enable

Bit 0 – RDRF Receive Data Register Full Interrupt Enable

© 2026 Microchip Technology Inc. and its subsidiaries

45.8.10. MCSPI Interrupt Disable Register

Name: MCSPI_IDR
Offset: 0x18
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the MCSPI Write Protection Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			CRCERR		CMP	UNDES	TXEMPTY	NSSR
Access			W		W	W	W	W
Reset			–		–	–	–	–

Bit	7	6	5	4	3	2	1	0
	TXBUFE	RXBUFF	ENDTX	ENDRX	OVRES	MODF	TDRE	RDRF
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 31 – RXFPTEF RXFPTEF Interrupt Disable

Bit 30 – TXFPTEF TXFPTEF Interrupt Disable

Bit 29 – RXFTHF RXFTHF Interrupt Disable

Bit 28 – RXFFF RXFFF Interrupt Disable

Bit 27 – RXFEF RXFEF Interrupt Disable

Bit 26 – TXFTHF TXFTHF Interrupt Disable

Bit 25 – TXFFF TXFFF Interrupt Disable

Bit 24 – TXFEF TXFEF Interrupt Disable

Bit 13 – CRCERR CRC Error Interrupt Disable

Bit 11 – CMP Comparison Interrupt Disable

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 10 – UNDES Underrun Error Interrupt Disable

Bit 9 – TXEMPTY Transmission Registers Empty Disable

Bit 8 – NSSR NSS Rising Interrupt Disable

Bit 7 – TXBUFE Transmit Buffer Empty Interrupt Disable

Bit 6 – RXBUFF Receive Buffer Full Interrupt Disable

Bit 5 – ENDTX End of Transmit Buffer Interrupt Disable

Bit 4 – ENDRX End of Receive Buffer Interrupt Disable

Bit 3 – OVRES Overrun Error Interrupt Disable

Bit 2 – MODF Mode Fault Error Interrupt Disable

Bit 1 – TDRE MCSPI Transmit Data Register Empty Interrupt Disable

Bit 0 – RDRF Receive Data Register Full Interrupt Disable

© 2026 Microchip Technology Inc. and its subsidiaries

45.8.11. MCSPI Interrupt Mask Register

Name: MCSPI_IMR
Offset: 0x1C
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
	RXFPTEF	TXFPTEF	RXFTHF	RXFFF	RXFEF	TXFTHF	TXFFF	TXFEF
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			CRCERR		CMP	UNDES	TXEMPTY	NSSR
Access			R		R	R	R	R
Reset			0		0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TXBUFE	RXBUFF	ENDTX	ENDRX	OVRES	MODF	TDRE	RDRF
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 31 – RXFPTEF RXFPTEF Interrupt Mask

Bit 30 – TXFPTEF TXFPTEF Interrupt Mask

Bit 29 – RXFTHF RXFTHF Interrupt Mask

Bit 28 – RXFFF RXFFF Interrupt Mask

Bit 27 – RXFEF RXFEF Interrupt Mask

Bit 26 – TXFTHF TXFTHF Interrupt Mask

Bit 25 – TXFFF TXFFF Interrupt Mask

Bit 24 – TXFEF TXFEF Interrupt Mask

Bit 13 – CRCERR CRC Error Interrupt Mask

Bit 11 – CMP Comparison Interrupt Mask

Bit 10 – UNDES Underrun Error Interrupt Mask

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 9 – TXEMPTY Transmission Registers Empty Mask

Bit 8 – NSSR NSS Rising Interrupt Mask

Bit 7 – TXBUFE Transmit Buffer Empty Interrupt Mask

Bit 6 – RXBUFF Receive Buffer Full Interrupt Mask

Bit 5 – ENDTX End of Transmit Buffer Interrupt Mask

Bit 4 – ENDRX End of Receive Buffer Interrupt Mask

Bit 3 – OVRES Overrun Error Interrupt Mask

Bit 2 – MODF Mode Fault Error Interrupt Mask

Bit 1 – TDRE MCSPI Transmit Data Register Empty Interrupt Mask

Bit 0 – RDRF Receive Data Register Full Interrupt Mask

© 2026 Microchip Technology Inc. and its subsidiaries

45.8.12. MCSPI Chip Select Register

Name: MCSPI_CSRx
Offset: 0x30 + x*0x04 [x=0..3]
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the MCSPI Write Protection Mode Register.

Note: MCSPI_CSRx must be written even if default reset values will be used. The BITS field is not updated with the translated value unless the register is written.

Bit	31	30	29	28	27	26	25	24
	DLYBCT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DLYBS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SCBR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BITS[3:0]				CSAAT	CSNAAT	NCPHA	CPOL
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – DLYBCT[7:0] Delay Between Consecutive Transfers
This field defines the delay between two consecutive transfers with the same peripheral without removing the chip select. The delay is always inserted after each transfer and before removing the chip select if needed.
When DLYBCT = 0, no delay between consecutive transfers is inserted and the clock keeps its duty cycle over the character transfers.
Otherwise, the following equations determine the delay:
If MCSPI_MR.BRSRCCLK = 0: DLYBCT = Delay Between Consecutive Transfers × $f_{peripheral\ clock}$ / 32
If MCSPI_MR.BRSRCCLK = 1: DLYBCT = Delay Between Consecutive Transfers × f_{GCLK} / 32

Bits 23:16 – DLYBS[7:0] Delay Before SPCK
This field defines the delay from NPCS falling edge (activation) to the first valid SPCK transition.
When DLYBS = 0, the delay is half the SPCK clock period.
Otherwise, the following equations determine the delay:
If MCSPI_MR.BRSRCCLK = 0: DLYBS = Delay Before SPCK × $f_{peripheral\ clock}$
If MCSPI_MR.BRSRCCLK = 1: DLYBS = Delay Before SPCK × f_{GCLK}

Bits 15:8 – SCBR[7:0] Serial Clock Bit Rate
In Host mode, the MCSPI Interface uses a modulus counter to derive the SPCK bit rate from the clock defined by the MCSPI_MR.BRSRCCLK bit. The bit rate is selected by writing a value from 1 to 255 in the SCBR field. The following equations determine the SPCK bit rate:
If MCSPI_MR.BRSRCCLK = 0: SCBR = $f_{peripheral\ clock}$ / SPCK Bit Rate

© 2026 Microchip Technology Inc. and its subsidiaries

If MCSPI_MR.BRSRCCLK = 1: SCBR = f_{GCLK} / SPCK Bit Rate
Programming the SCBR field to 0 is forbidden. Triggering a transfer while SCBR is at 0 can lead to unpredictable results.
If MCSPI_MR.BRSRCCLK = 1, SCBR must be programmed with a value greater than 1.
At reset, SCBR is 0 and the user has to program it at a valid value before performing the first transfer.
Note: If one of the MCSPI_CSRx.SCBR fields is set to 1, the other MCSPI_CSRx.SCBR fields must be set to 1 as well, if they are used to transfer data. If they are not used to transfer data, they can be set to any value.

Bits 7:4 – BITS[3:0] Bits Per Transfer
See Note above.
The BITS field determines the number of data bits transferred. Reserved values should not be used.

Value	Name	Description
0	8_BIT	8 bits for transfer
1	9_BIT	9 bits for transfer
2	10_BIT	10 bits for transfer
3	11_BIT	11 bits for transfer
4	12_BIT	12 bits for transfer
5	13_BIT	13 bits for transfer
6	14_BIT	14 bits for transfer
7	15_BIT	15 bits for transfer
8	16_BIT	16 bits for transfer
9	–	Reserved
10	–	Reserved
11	–	Reserved
12	–	Reserved
13	–	Reserved
14	–	Reserved
15	–	Reserved

Bit 3 – CSAAT Chip Select Active After Transfer

Value	Description
0	The peripheral chip select line rises as soon as the last transfer is achieved.
1	The peripheral chip select line does not rise after the last transfer is achieved. It remains active until a new transfer is requested on a different chip select.

Bit 2 – CSNAAT Chip Select Not Active After Transfer (ignored if CSAAT = 1)

Value	Description
0	The peripheral chip select line does not rise between two transfers if MCSPI_TDR is reloaded before the end of the first transfer and if the two transfers occur on the same chip select.
1	The peripheral chip select line rises systematically after each transfer performed on the same client. It remains inactive after the end of transfer for a minimal duration of: If MCSPI_MR.BRSRCCLK = 0: $\frac{DLYBCS}{f_{peripheral\ clock}}$ If MCSPI_MR.BRSRCCLK = 1: $\frac{DLYBCS}{f_{GCLK}}$ If field DLYBCS is lower than 6, a minimum of six periods is introduced.

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 1 – NCPHA Clock Phase

NCPHA determines which edge of SPCK causes data to change and which edge causes data to be captured. NCPHA is used with CPOL to produce the required clock/data relationship between host and client devices.

Value	Description
0	Data is changed on the leading edge of SPCK and captured on the following edge of SPCK.
1	Data is captured on the leading edge of SPCK and changed on the following edge of SPCK.

Bit 0 – CPOL Clock Polarity

CPOL is used to determine the inactive state value of the serial clock (SPCK). It is used with NCPHA to produce the required clock/data relationship between host and client devices.

Value	Description
0	The inactive state value of SPCK is logic level zero.
1	The inactive state value of SPCK is logic level one.

© 2026 Microchip Technology Inc. and its subsidiaries

45.8.13. MCSPI FIFO Mode Register

Name: MCSPI_FMR
Offset: 0x40
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
			RXFTHRES[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
			TXFTHRES[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			RXRDYM[1:0]				TXRDYM[1:0]	
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bits 29:24 – RXFTHRES[5:0] Receive FIFO Threshold

Value	Description
0-8	Defines the Receive FIFO threshold value (number of data). MCSPI_SR.RXFTH will be set when the Receive FIFO goes from “below” threshold state to “equal or above” threshold state.

Bits 21:16 – TXFTHRES[5:0] Transmit FIFO Threshold

Value	Description
0-8	Defines the Transmit FIFO threshold value (number of data). MCSPI_SR.TXFTH will be set when the Transmit FIFO goes from “above” threshold state to “equal or below” threshold state.

Bits 5:4 – RXRDYM[1:0] Receive Data Register Full Mode
If FIFOs are enabled, the MCSPI_SR.RDRF flag behaves as follows:

Value	Name	Description
0	ONE_DATA	RDRF will be at level ‘1’ when at least one unread data is in the Receive FIFO.
1	TWO_DATA	RDRF will be at level ‘1’ when at least two unread data are in the Receive FIFO. Cannot be used when MCSPI_MR.MSTR =1, or if MCSPI_MR.PS =1.
2	FOUR_DATA	RDRF will be at level ‘1’ when at least four unread data are in the Receive FIFO. Cannot be used when MCSPI_CSRx.BITS is greater than 0, or if MCSPI_MR.MSTR = 1, or if MCSPI_MR.PS = 1.

Bits 1:0 – TXRDYM[1:0] Transmit Data Register Empty Mode
If FIFOs are enabled, the MCSPI_SR.TDRE flag behaves as follows:

Value	Name	Description
0	ONE_DATA	TDRE will be at level ‘1’ when at least one data can be written in the Transmit FIFO.
1	TWO_DATA	TDRE will be at level ‘1’ when at least two data can be written in the Transmit FIFO. Cannot be used if MCSPI_MR.PS =1.

© 2026 Microchip Technology Inc. and its subsidiaries

45.8.14. MCSPI FIFO Level Register

Name: MCSPI_FLR
Offset: 0x44
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			RXFL[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			TXFL[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bits 21:16 – RXFL[5:0] Receive FIFO Level

Value	Description
0	There is no unread data in the Receive FIFO.
1-8	Indicates the number of unread data in the Receive FIFO.

Bits 5:0 – TXFL[5:0] Transmit FIFO Level

Value	Description
0	There is no data in the Transmit FIFO.
1-8	Indicates the number of data in the Transmit FIFO.

© 2026 Microchip Technology Inc. and its subsidiaries

45.8.15. MCSPI Comparison Register

Name: MCSPI_CMPR
Offset: 0x48
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the MCSPI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	VAL2[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	VAL2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	VAL1[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	VAL1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – VAL2[15:0] Second Comparison Value for Received Character

Value	Description
0–65535	The received character must be lower than or equal to VAL2 and higher than or equal to VAL1 to set the MCSPI_SR.CMP flag.

Bits 15:0 – VAL1[15:0] First Comparison Value for Received Character

Value	Description
0–65535	The received character must be higher than or equal to VAL1 and lower than or equal to VAL2 to set the MCSPI_SR.CMP flag.

© 2026 Microchip Technology Inc. and its subsidiaries

45.8.16. MCSPI CRC Register

Name: MCSPI_CRCR
Offset: 0x4C
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the MCSPI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
							FHE	CRM
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
	FRHL[3:0]							CRCS
Access	R/W	R/W	R/W	R/W				R/W
Reset	0	0	0	0				0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	FRL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 25 – FHE Frame Header Excluded

Value	Description
0	The frame header is included in the CRC calculation.
1	The frame header is excluded from the CRC calculation.

Bit 24 – CRM Continuous Read Mode

Value	Description
0	A header is sent every frame in case of contiguous frames (without de-asserting the corresponding NPCS).
1	A header is sent only on the first frame in case of contiguous frames (without de-asserting the corresponding NPCS).

Bits 23:20 – FRHL[3:0] Frame Header Length
If MCSPI_MR.TPMEN= 0, this value is the length of the frame header, in bytes.
If MCSPI_MR.TPMEN= 1, this value is the length of the frame header minus 1, in bytes.

Bit 16 – CRCS CRC Size

Value	Name	Description
0	16B_CRC	CRC size is 16 bits.
1	32B_CRC	CRC size is 32 bits.

Bits 7:0 – FRL[7:0] Frame Length
If MCSPI_MR.TPMEN= 0, this value is the length of the frame (header and CRC included), in bytes.

© 2026 Microchip Technology Inc. and its subsidiaries

If MCSPI_MR.TPMEN= 1, this value is the length in bytes of the frame starting from the SYNC field (included) with CRC included.

© 2026 Microchip Technology Inc. and its subsidiaries

45.8.17. MCSPI Two-Pin Mode Register

Name: MCSPI_TPMR
Offset: 0x50
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the MCSPI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					OSR[1:0]		MIL	CSM
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 3:2 – OSR[1:0] Oversampling Rate
Defines the MCP3910 OSR setting used.

Bit 1 – MIL Multiple Input Lines

Value	Description
0	The MCSPI manages a single MOSI line (data multiplexing must be performed externally). When a single MCP3910 is driven, CSM and MIL must be written to 0.
1	The MCSPI manages up to four MOSI lines and data multiplexing is done internally.

Bit 0 – CSM Chip Select Mode

Value	Description
0	Chip select is not driven. When a single MCP3910 is driven, CSM and MIL must be written to 0.
1	Chip select is driven and can be used to control enable pin of external device. Depending of the MCSPI_MR.PCSDEC bit, an external decoder can be used to minimize the number of IOs used.

© 2026 Microchip Technology Inc. and its subsidiaries

45.8.18. MCSPI Two-Pin Header Register

Name: MCSPI_TPHR
Offset: 0x54
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		OSR[1:0]		GAIN[1:0]		BOOST	CNT[1:0]	
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bits 6:5 – OSR[1:0] Oversampling Rate
Oversampling rate setting

Bits 4:3 – GAIN[1:0] Gain
Gain setting

Bit 2 – BOOST Current Boost
Current boost setting

Bits 1:0 – CNT[1:0] Frame Counter
Frame counter value

© 2026 Microchip Technology Inc. and its subsidiaries

45.8.19. MCSPI Write Protection Mode Register

Name: MCSPI_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
						WPCREN	WPITEN	WPEN
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key
See Register Write Protection for the list of registers that can be write-protected.

Value	Name	Description
0x535049	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0.

Bit 2 – WPCREN Write Protection Control Register Enable

Value	Description
0	Disables the write protection on the Control register if WPKEY corresponds to 0x535049.
1	Enables the write protection on the Control register if WPKEY corresponds to 0x535049.

Bit 1 – WPITEN Write Protection Interrupt Enable

Value	Description
0	Disables the write protection on Interrupt registers if WPKEY corresponds to 0x535049.
1	Enables the write protection on Interrupt registers if WPKEY corresponds to 0x535049.

Bit 0 – WPEN Write Protection Enable

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x535049 (“SPI” in ASCII)
1	Enables the write protection if WPKEY corresponds to 0x535049 (“SPI” in ASCII)

© 2026 Microchip Technology Inc. and its subsidiaries

45.8.20. MCSPI Write Protection Status Register

Name: MCSPI_WPSR
Offset: 0xE8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
								WPVS
Access								R
Reset								0

Bits 15:8 – WPVSRC[7:0] Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protection Violation Status

Value	Description
0	No write protection violation has occurred since the last read of MCSPI_WPSR.
1	A write protection violation has occurred since the last read of MCSPI_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

© 2026 Microchip Technology Inc. and its subsidiaries

46. Pulse Width Modulation Controller (PWM)

46.1. Description

The Pulse Width Modulation Controller (PWM) generates output pulses on 3 channels independently according to parameters defined per channel. Each channel controls two complementary square output waveforms. Characteristics of the output waveforms such as period, duty-cycle, polarity and dead-times (also called dead-bands or non-overlapping times) are configured through the user interface. Each channel selects and uses one of the clocks provided by the clock generator. The clock generator provides several clocks resulting from the division of the PWM peripheral clock. External triggers can modify the output values in real time.

All accesses to the PWM are made through registers mapped on the peripheral bus. All channels integrate a double buffering system in order to prevent an unexpected output waveform while modifying the period, the spread spectrum, the duty-cycle or the dead-times.

Channels can be linked together as synchronous channels to be able to update their duty-cycle or dead-times at the same time.

The update of duty-cycles of synchronous channels can be performed by the Peripheral DMA Controller channel which offers buffer transfer without processor Intervention.

The PWM includes a spread-spectrum counter to allow a constantly varying period (only for Channel 0). This counter may be useful to minimize electromagnetic interference or to reduce the acoustic noise of a PWM driven motor.

The PWM provides 8 independent comparison units capable of comparing a programmed value to the counter of the synchronous channels (counter of channel 0). These comparisons are intended to generate software interrupts, to trigger pulses on the 1 independent event lines (in order to synchronize ADC conversions with a lot of flexibility independently of the PWM outputs) and to trigger Peripheral DMA Controller transfer requests.

PWM outputs can be overridden synchronously or asynchronously to their channel counter.

The PWM provides a fault protection mechanism with 8 fault inputs, capable to detect a fault condition and to override the PWM outputs asynchronously (outputs forced to ‘0’, ‘1’ or Hi-Z).

For safety usage, some configuration registers are write-protected.

46.2. Embedded Characteristics

- 3 Channels
- Common Clock Generator Providing Thirteen Different Clocks
  - A Modulo n counter providing eleven clocks
  - Two independent linear dividers working on Modulo n counter outputs
- Independent Channels
  - Independent 24-bit counter for each channel
  - Independent complementary outputs with 12-bit dead-time generator (also called dead-band or non-overlapping time) for each channel
  - Independent Push-Pull mode for each channel
  - Independent enable-disable command for each channel
  - Independent clock selection for each channel
  - Independent period, duty-cycle and dead-time for each channel
  - Independent double buffering of period, duty-cycle and dead-times for each channel

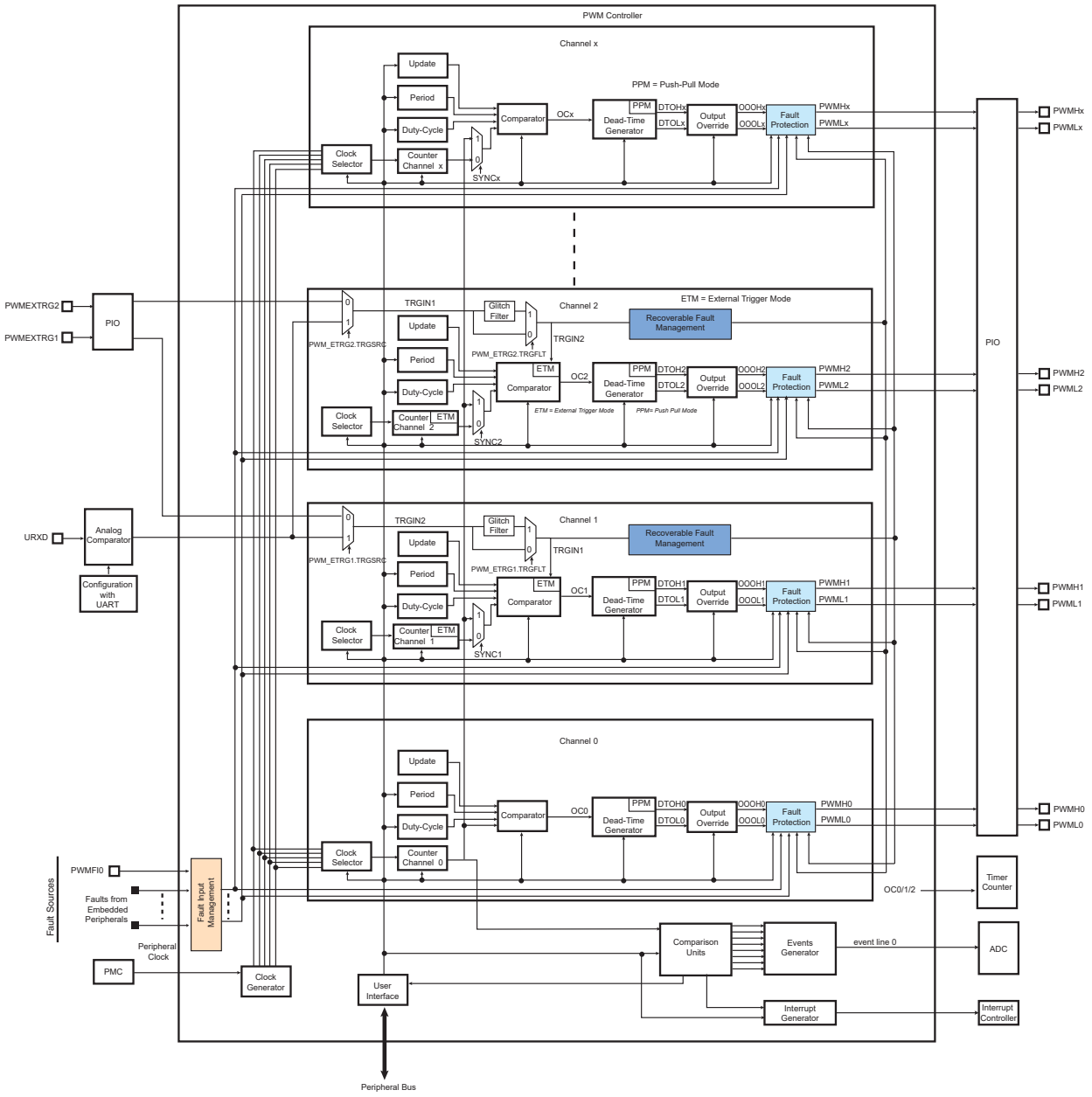
© 2026 Microchip Technology Inc. and its subsidiaries

  - Independent programmable selection of the output waveform polarity for each channel, with double buffering
  - Independent programmable center- or left-aligned output waveform for each channel
  - Independent output override for each channel
  - Independent interrupt for each channel, at each period for left-aligned or center-aligned configuration
  - Independent update time selection of double buffering registers (polarity, duty cycle) for each channel, at each period for left-aligned or center-aligned configuration
- External Trigger Input Management (e.g., for DC/DC or Lighting Control)
  - External PWM Reset mode
  - External PWM Start mode
  - Cycle-by-cycle duty cycle mode
  - Leading-edge blanking
- One 2-bit Gray Up/Down Channels for Stepper Motor Control
- Spread Spectrum Counter to Allow a Constantly Varying Duty Cycle (only for Channel 0)
- Synchronous Channel Mode
  - Synchronous channels share the same counter
  - Mode to update the synchronous channels registers after a programmable number of periods
  - Synchronous channels support connection of one Peripheral DMA Controller channel offers buffer transfer without processor intervention to update duty-cycle registers
- 1 Independent Event Lines Intended to Synchronize ADC Conversions
  - Programmable delay for event lines to delay ADC measurements
- 8 Comparison Units Intended to Generate Interrupts, Pulses on Event Lines and Peripheral DMA Controller Transfer Requests
- 8 Programmable Fault Inputs Providing Asynchronous Protection of PWM Outputs
  - One driven by the user through PIO inputs
  - Driven by the PMC when crystal oscillator clock fails
  - Driven by the ADC Controller through configurable comparison function
  - Driven by the Timer/Counter through configurable comparison function
- Register Write Protection

© 2026 Microchip Technology Inc. and its subsidiaries

46.3. Block Diagram

Figure 46-1. PWM Controller Block Diagram



Note: For a more detailed illustration of the fault protection circuitry, refer to Fault Protection.

46.4. I/O Lines Description

Each channel outputs two complementary external I/O lines.

Table 46-1. I/O Lines Description

Name	Description	Type
PWMHx	PWM Waveform Output High for channel x	Output
PWMLx	PWM Waveform Output Low for channel x	Output
PWMFIx	PWM Fault Input x	Input

© 2026 Microchip Technology Inc. and its subsidiaries

Table 46-1. I/O Lines Description (continued)

Name	Description	Type
PWMEXTRGy	PWM Trigger Input y	Input

46.5. Product Dependencies

46.5.1. I/O Lines

The pins used for interfacing the PWM are multiplexed with PIO lines. The programmer must first program the PIO controller to assign the desired PWM pins to their peripheral function. If I/O lines of the PWM are not used by the application, they can be used for other purposes by the PIO controller.

All of the PWM outputs may or may not be enabled. If an application requires only four channels, then only four PIO lines are assigned to PWM outputs.

46.5.2. Power Management

The PWM is not continuously clocked. The programmer must first enable the PWM clock in the Power Management Controller (PMC) before using the PWM. However, if the application does not require PWM operations, the PWM clock can be stopped when not needed and be restarted later. In this case, the PWM will resume its operations where it left off.

46.5.3. Interrupt Sources

The PWM interrupt line is connected on one of the internal sources of the Interrupt Controller. Using the PWM interrupt requires the Interrupt Controller to be programmed first.

46.5.4. Fault Inputs

The PWM has the fault inputs connected to the different modules. Refer to the implementation of these modules within the product for detailed information about the fault generation procedure. The PWM receives faults from:

- PIO inputs
- the PMC
- the ADC controller
- Timer/Counters

Table 46-2. Fault Inputs

Fault Generator	External PWM Fault Input Number	Polarity Level(1)	Fault Input ID
PD1 or PD12	PWMFI0	User-defined	0
Main OSC (PMC)	–	To be configured to 1	2
ADC	–	To be configured to 1	3
TC3	–	To be configured to 1	4
TC0	–	To be configured to 1	5
TC1	–	To be configured to 1	6
TC2	–	To be configured to 1	7

Note:

1. FPOL field in PWM_FMR.

© 2026 Microchip Technology Inc. and its subsidiaries

46.5.5. External Trigger Inputs

Table 46-3. External Trigger Inputs

IO Pin Name	External Trigger Pin Name	Polarity Level
PD1 or PD12	PWMEXTRG1	1
PD18	PWMEXTRG2	1

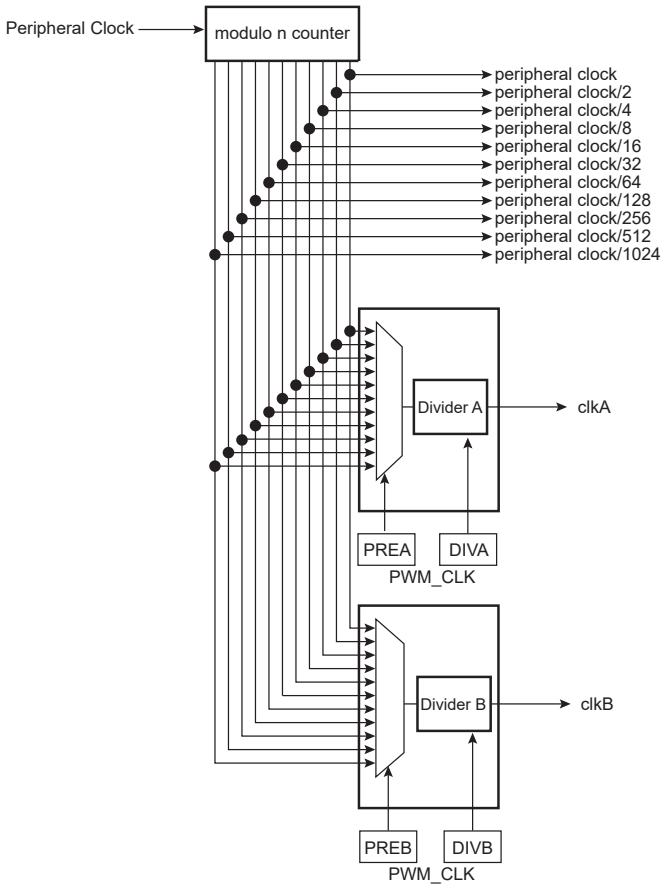
46.6. Functional Description

The PWM controller is primarily composed of a clock generator module and 3 channels.

- Clocked by the peripheral clock, the clock generator module provides 13 clocks.
- Each channel can independently choose one of the clock generator outputs.
- Each channel generates an output waveform with attributes that can be defined independently for each channel through the user interface registers.

46.6.1. PWM Clock Generator

Figure 46-2. Functional View of the Clock Generator Block Diagram



The PWM peripheral clock is divided in the clock generator module to provide different clocks available for all channels. Each channel can independently select one of the divided clocks.

The clock generator is divided into different blocks:

- a modulo n counter which provides 11 clocks:
  - $f_{peripheral\ clock}$

© 2026 Microchip Technology Inc. and its subsidiaries

- $f_{peripheral\ clock}/2$
- $f_{peripheral\ clock}/4$
- $f_{peripheral\ clock}/8$
- $f_{peripheral\ clock}/16$
- $f_{peripheral\ clock}/32$
- $f_{peripheral\ clock}/64$
- $f_{peripheral\ clock}/128$
- $f_{peripheral\ clock}/256$
- $f_{peripheral\ clock}/512$
- $f_{peripheral\ clock}/1024$

• two linear dividers (1, 1/2, 1/3, ... 1/255) that provide two separate clocks: clkA and clkB

Each linear divider can independently divide one of the clocks of the modulo n counter. The selection of the clock to be divided is made according to the PREA (PREB) field of the PWM Clock register (PWM_CLK). The resulting clock clkA (clkB) is the clock selected divided by DIVA (DIVB) field value.

After a reset of the PWM controller, DIVA (DIVB) and PREA (PREB) are set to '0'. This implies that after reset clkA (clkB) are turned off.

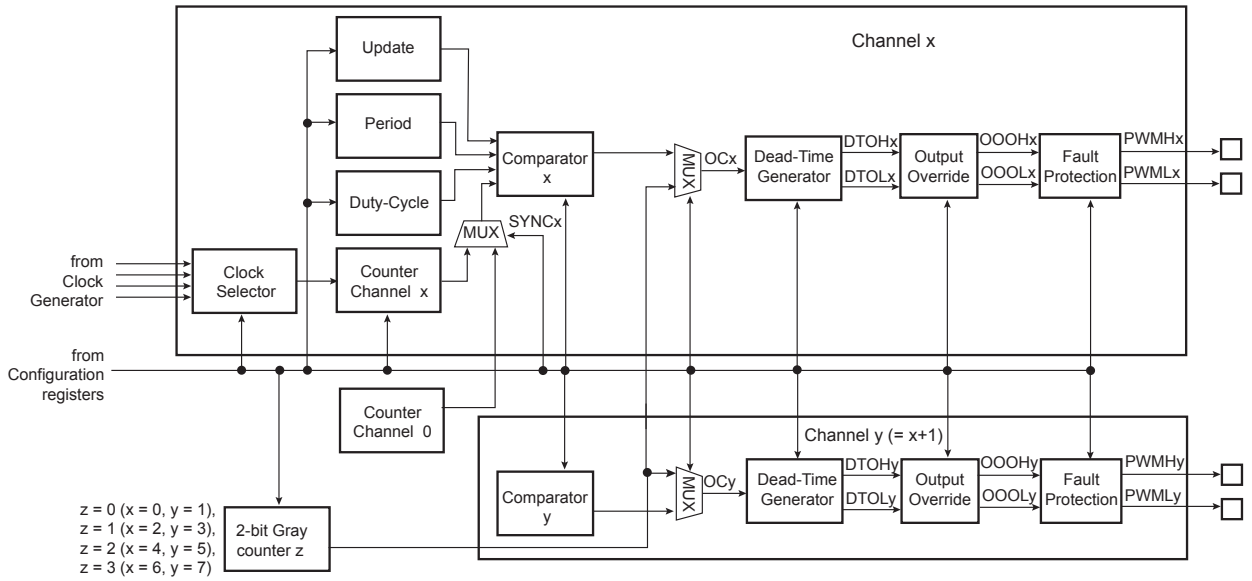
At reset, all clocks provided by the modulo n counter are turned off except the peripheral clock. This situation is also true when the PWM peripheral clock is turned off through the Power Management Controller.

CAUTION Before using the PWM controller, the programmer must first enable the peripheral clock in the Power Management Controller (PMC).

46.6.2. PWM Channel

46.6.2.1. Channel Block Diagram

Figure 46-3. Functional View of the Channel Block Diagram



© 2026 Microchip Technology Inc. and its subsidiaries

Each of the 3 channels is composed of six blocks:

- A clock selector which selects one of the clocks provided by the clock generator (described in PWM Clock Generator).
- A counter clocked by the output of the clock selector. This counter is incremented or decremented according to the channel configuration and comparators matches. The size of the counter is 24 bits.
- A comparator used to compute the OCx output waveform according to the counter value and the configuration. The counter value can be the one of the channel counter or the one of the channel 0 counter according to SYNCx bit in the PWM Sync Channels Mode Register (PWM_SCM).
- A 2-bit configurable Gray counter enables the stepper motor driver. One Gray counter drives 2 channels.
- A dead-time generator providing two complementary outputs (DTOHx/DTOLx) which allows to drive external power control switches safely.
- An output override block that can force the two complementary outputs to a programmed value (OOOHx/OOOLx).
- An asynchronous fault protection mechanism that has the highest priority to override the two complementary outputs (PWMHx/PWMLx) in case of fault detection (outputs forced to '0', '1' or Hi-Z).

46.6.2.2. Comparator

The comparator continuously compares its counter value with the channel period defined by CPRD in the PWM Channel Period Register (PWM_CPRDx) and the duty-cycle defined by CDTY in the PWM Channel Duty Cycle Register (PWM_CDTYx) to generate an output signal OCx accordingly.

The different properties of the waveform of the output OCx are:

- the clock selection. The channel counter is clocked by one of the clocks provided by the clock generator described in the previous section. This channel parameter is defined in the CPRE field of the PWM Channel Mode Register (PWM_CMRx). This field is reset at '0'.
- the waveform period. This channel parameter is defined in the CPRD field of the PWM_CPRDx register.
  If the waveform is left-aligned, then the output waveform period depends on the counter source clock and can be calculated:
  By using the PWM peripheral clock divided by a given prescaler value "X" (where X = 2^{PREA} is 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, or 1024). The resulting period formula is:

  $$\frac{(X \times \mathrm{CPRD})}{f_{\text{peripheral clock}}}$$

  By using the PWM peripheral clock divided by a given prescaler value "X" (see above) and by either the DIVA or the DIVB divider. The formula becomes, respectively:

  $$\frac{(X \times \mathrm{CPRD} \times \mathrm{DIVA})}{f_{\text{peripheral clock}}} \text{ or } \frac{(X \times \mathrm{CPRD} \times \mathrm{DIVB})}{f_{\text{peripheral clock}}}$$

  If the waveform is center-aligned, then the output waveform period depends on the counter source clock and can be calculated:
  By using the PWM peripheral clock divided by a given prescaler value "X" (where X = 2^{PREA} is 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, or 1024). The resulting period formula is:

  $$\frac{(2 \times X \times \mathrm{CPRD})}{f_{\text{peripheral clock}}}$$

© 2026 Microchip Technology Inc. and its subsidiaries

By using the PWM peripheral clock divided by a given prescaler value “X” (see above) and by either the DIVA or the DIVB divider. The formula becomes, respectively:

$$\frac{(2 \times X \times \text{CPRD} \times \text{DIVA})}{f_{\text{peripheral clock}}} \text{ or}$$

$$\frac{(2 \times X \times \text{CPRD} \times \text{DIVB})}{f_{\text{peripheral clock}}}$$

- the waveform duty-cycle. This channel parameter is defined in the CDTY field of the PWM_CDTYx register.
  If the waveform is left-aligned, then:

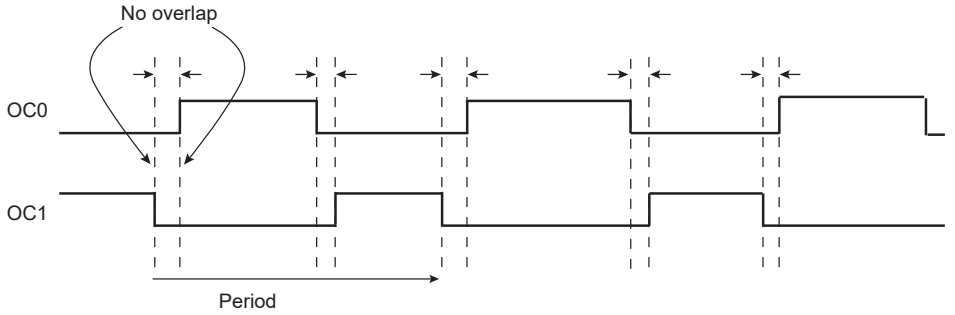
  $$\text{duty cycle} = \frac{\text{PWM\_period} - ((1/\text{clkN}) \times \text{CDTY})}{\text{PWM\_period}}$$

  If the waveform is center-aligned, then:

  $$\text{duty cycle} = \frac{((\text{PWM\_period}/2) - 1) - ((1/\text{clkN}) \times \text{CDTY})}{\text{PWM\_period}/2}$$

- the waveform polarity. At the beginning of the period, the signal can be at high or low level. This property is defined in the CPOL bit of PWM_CMRx. By default, the signal starts by a low level. The DPOLI bit in PWM_CMRx defines the PWM polarity when the channel is disabled (CHIDx = 0 in PWM_SR). For more details, see the figure *Waveform Properties.*
  - DPOLI = 0: PWM polarity when the channel is disabled is the same as the one defined for the beginning of the PWM period.
  - DPOLI = 1: PWM polarity when the channel is disabled is inverted compared to the one defined for the beginning of the PWM period.
- the waveform alignment. The output waveform can be left- or center-aligned. Center-aligned waveforms can be used to generate non-overlapped waveforms. This property is defined in the CALG bit of PWM_CMRx. The default mode is left-aligned.

Figure 46-4. Non-Overlapped Center-Aligned Waveforms



Note: See the figure Figure 46-5 for a detailed description of center-aligned waveforms.

When center-aligned, the channel counter increases up to CPRD and decreases down to 0. This ends the period.

When left-aligned, the channel counter increases up to CPRD and is reset. This ends the period.

Thus, for the same CPRD value, the period for a center-aligned channel is twice the period for a left-aligned channel.

Waveforms are fixed at 0 when:

© 2026 Microchip Technology Inc. and its subsidiaries

- CDTY = CPRD and CPOL = 0 (Note that if TRGMODE = MODE3, the PWM waveform switches to 1 at the external trigger event (see Cycle-By-Cycle Duty Mode)).
- CDTY = 0 and CPOL = 1

Waveforms are fixed at 1 (once the channel is enabled) when:

- CDTY = 0 and CPOL = 0
- CDTY = CPRD and CPOL = 1 (Note that if TRGMODE = MODE3, the PWM waveform switches to 0 at the external trigger event (see Cycle-By-Cycle Duty Mode)).

The waveform polarity must be set before enabling the channel. This immediately affects the channel output level.

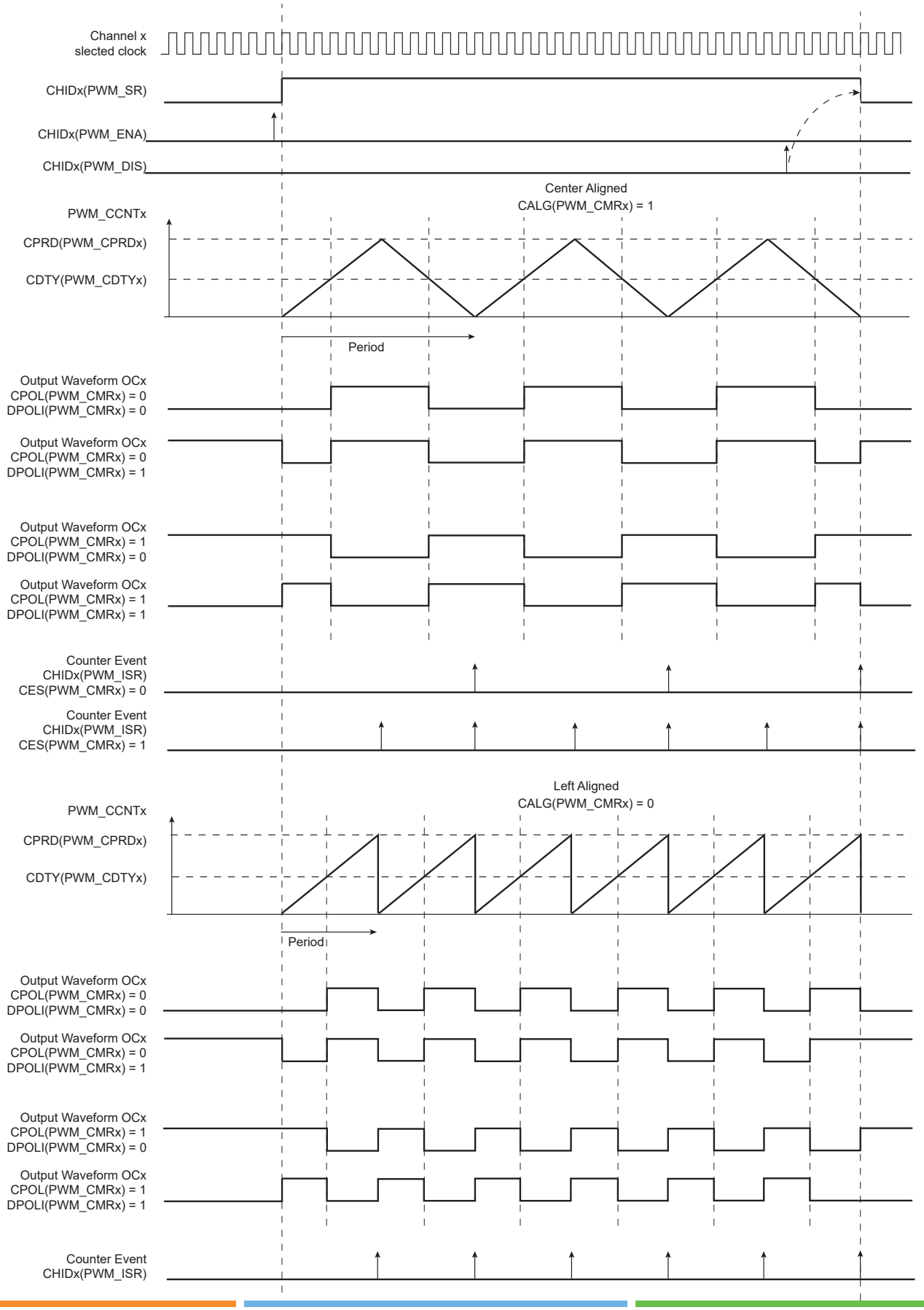
Modifying CPOL in PWM Channel Mode Register while the channel is enabled can lead to an unexpected behavior of the device being driven by PWM.

In addition to generating the output signals OCx, the comparator generates interrupts depending on the counter value. When the output waveform is left-aligned, the interrupt occurs at the end of the counter period. When the output waveform is center-aligned, the bit CES of PWM_CMRx defines when the channel counter interrupt occurs. If CES is set to '0', the interrupt occurs at the end of the counter period. If CES is set to '1', the interrupt occurs at the end of the counter period and at half of the counter period.

The figure below illustrates the counter interrupts depending on the configuration.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 46-5. Waveform Properties



© 2026 Microchip Technology Inc. and its subsidiaries

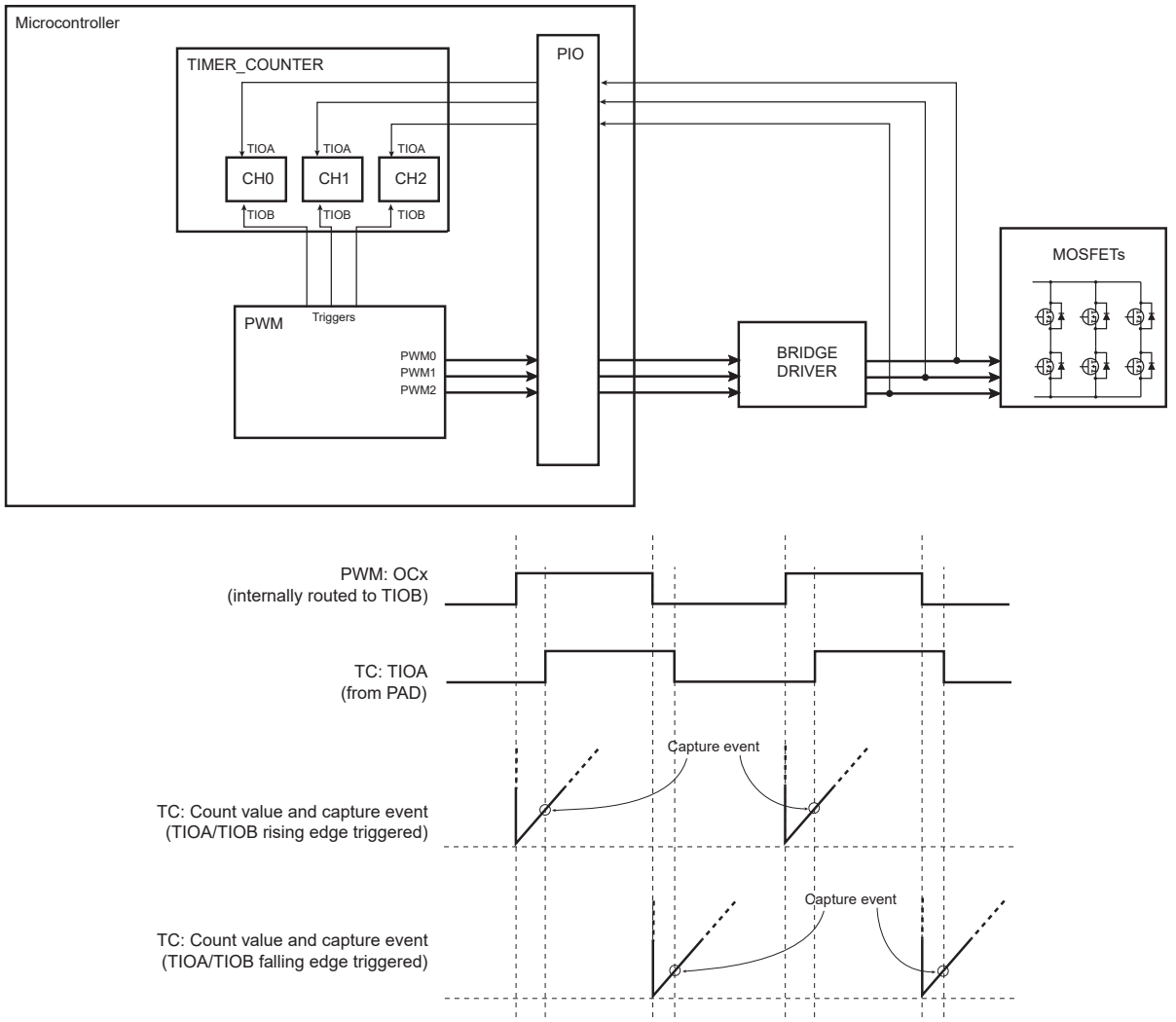
46.6.2.3. Trigger Selection for Timer Counter

The PWM controller can be used as a trigger source for the Timer Counter (TC) to achieve the two application examples described below.

46.6.2.3.1. Delay Measurement

To measure the delay between the channel x comparator output (OCx) and the feedback from the bridge driver of the MOSFETs (see the figure below), the bit TCTS in the PWM Channel Mode Register must be at 0. This defines the comparator output of the channel x as the TC trigger source. The TIOB trigger (TC internal input) is used to start the TC; the TIOA input (from PAD) is used to capture the delay.

Figure 46-6. Triggering the TC: Delay Measurement

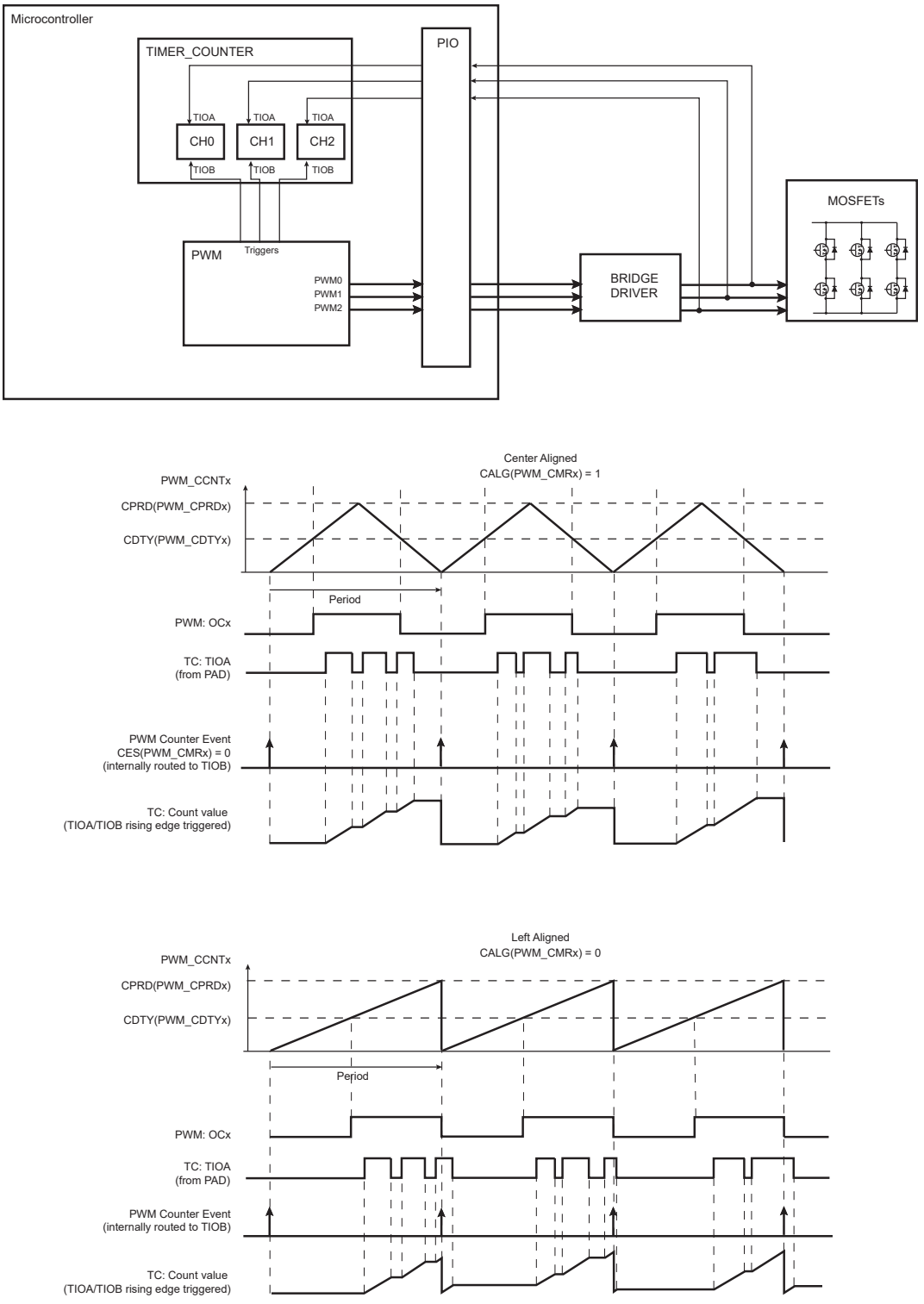


46.6.2.3.2. Cumulated ON Time Measurement

To measure the cumulated “ON” time of MOSFETs (see the figure below), the bit TCTS of the PWM Channel Mode Register must be set to 1 to define the counter event (see the figure *Waveform Properties*) as the Timer Counter trigger source.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 46-7. Triggering the TC: Cumulated “ON” Time Measurement



46.6.2.4. 2-bit Gray Up/Down Counter for Stepper Motor

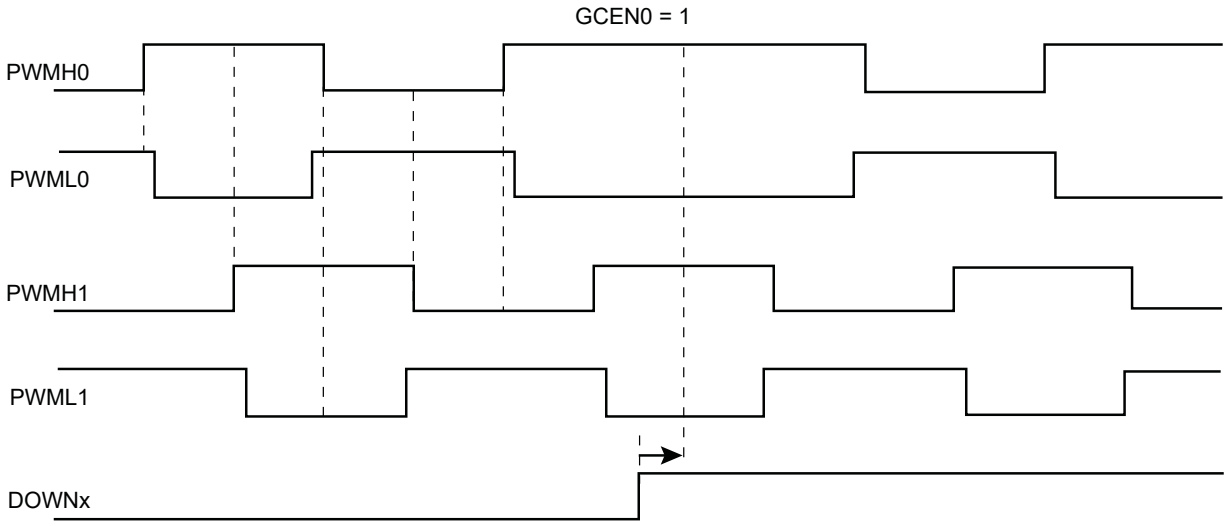
A pair of channels may provide a 2-bit Gray count waveform on two outputs. Dead-time generator and other downstream logic can be configured on these channels.

Up or Down Count mode can be configured on-the-fly by means of PWM_SMMR configuration registers.

© 2026 Microchip Technology Inc. and its subsidiaries

When GCEN0 is set to ‘1’, channels 0 and 1 outputs are driven with a Gray counter.

Figure 46-8. 2-bit Gray Up/Down Counter



46.6.2.5. Dead-Time Generator

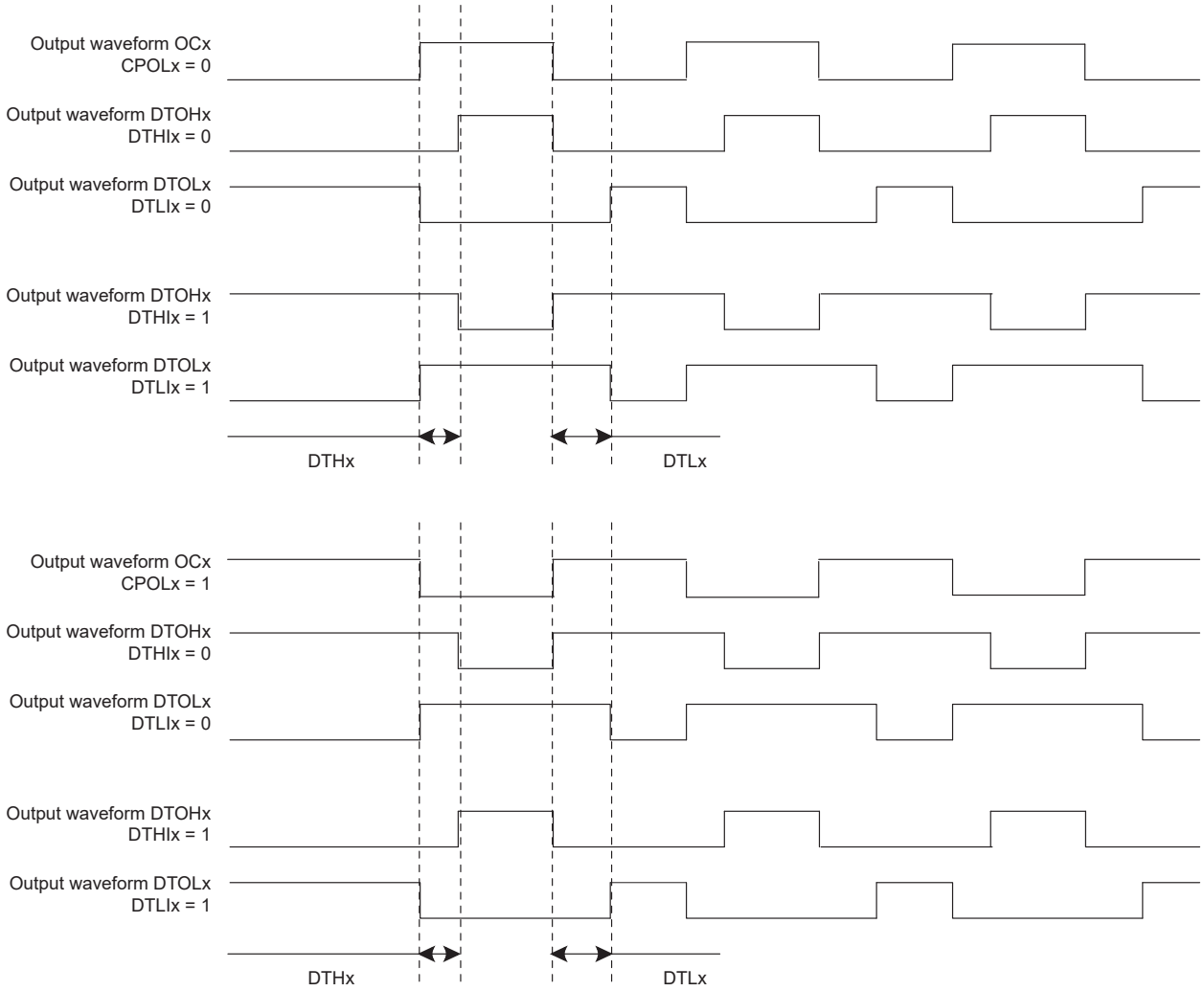
The dead-time generator uses the comparator output OCx to provide the two complementary outputs DTOHx and DTOLx, which allows the PWM macrocell to drive external power control switches safely. When the dead-time generator is enabled by setting the bit DTE to 1 or 0 in the PWM Channel Mode Register (PWM_CMRx), dead-times (also called dead-bands or non-overlapping times) are inserted between the edges of the two complementary outputs DTOHx and DTOLx. Note that enabling or disabling the dead-time generator is allowed only if the channel is disabled.

The dead-time is adjustable by the PWM Channel Dead Time Register (PWM_DTx). Each output of the dead-time generator can be adjusted separately by DTH and DTL. The dead-time values can be updated synchronously to the PWM period by using the PWM Channel Dead Time Update Register (PWM_DTUPDx).

The dead-time is based on a specific counter which uses the same selected clock that feeds the channel counter of the comparator. Depending on the edge and the configuration of the dead-time, DTOHx and DTOLx are delayed until the counter has reached the value defined by DTH or DTL. An inverted configuration bit (DTHI and DTLI bit in PWM_CMRx) is provided for each output to invert the dead-time outputs. The following figure shows the waveform of the dead-time generator.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 46-9. Complementary Output Waveforms

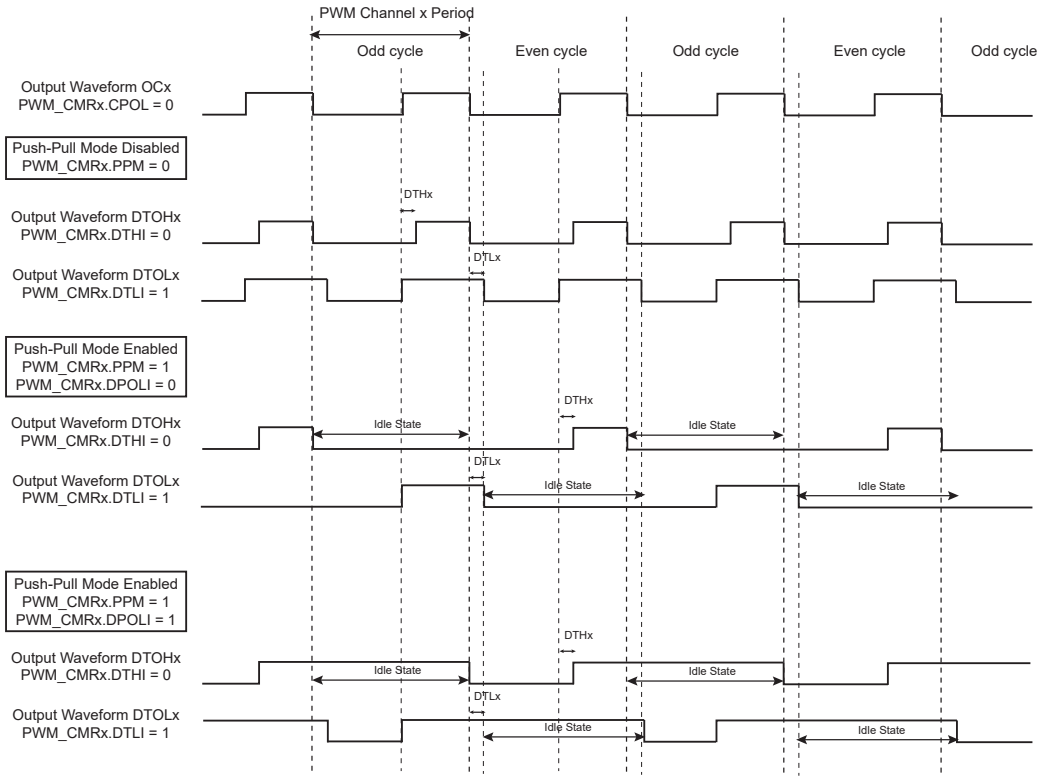


46.6.2.5.1. PWM Push-Pull Mode

When a PWM channel is configured in Push-Pull mode, the dead-time generator output is managed alternately on each PWM cycle. The polarity of the PWM line during the idle state of the Push-Pull mode is defined by the DPOLI bit in the PWM Channel Mode Register (PWM_CMRx). The Push-Pull mode can be enabled separately on each channel by writing a one to bit PPM in the PWM Channel Mode Register.

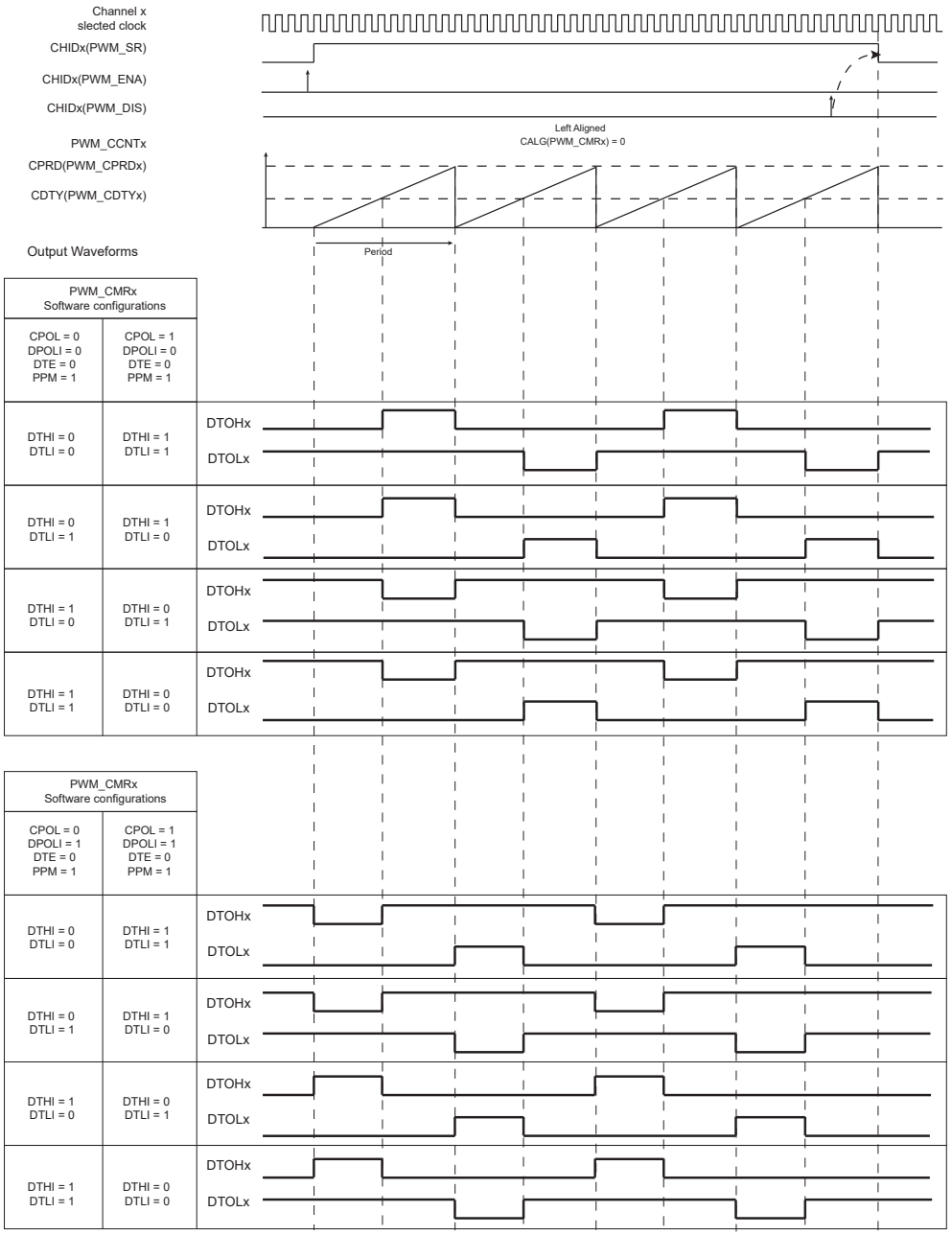
© 2026 Microchip Technology Inc. and its subsidiaries

Figure 46-10. PWM Push-Pull Mode



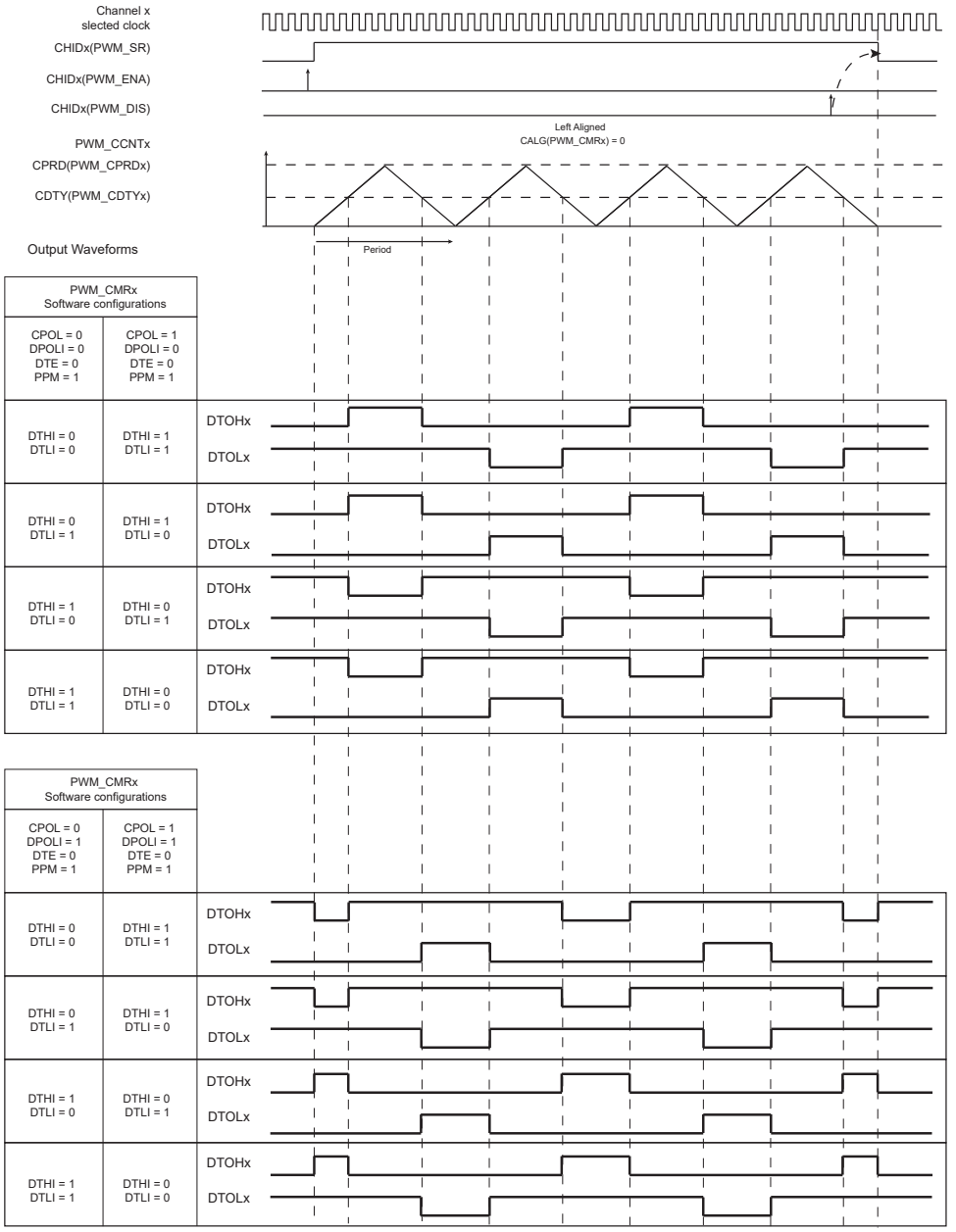
© 2026 Microchip Technology Inc. and its subsidiaries

Figure 46-11. PWM Push-Pull Waveforms: Left-Aligned Mode



© 2026 Microchip Technology Inc. and its subsidiaries

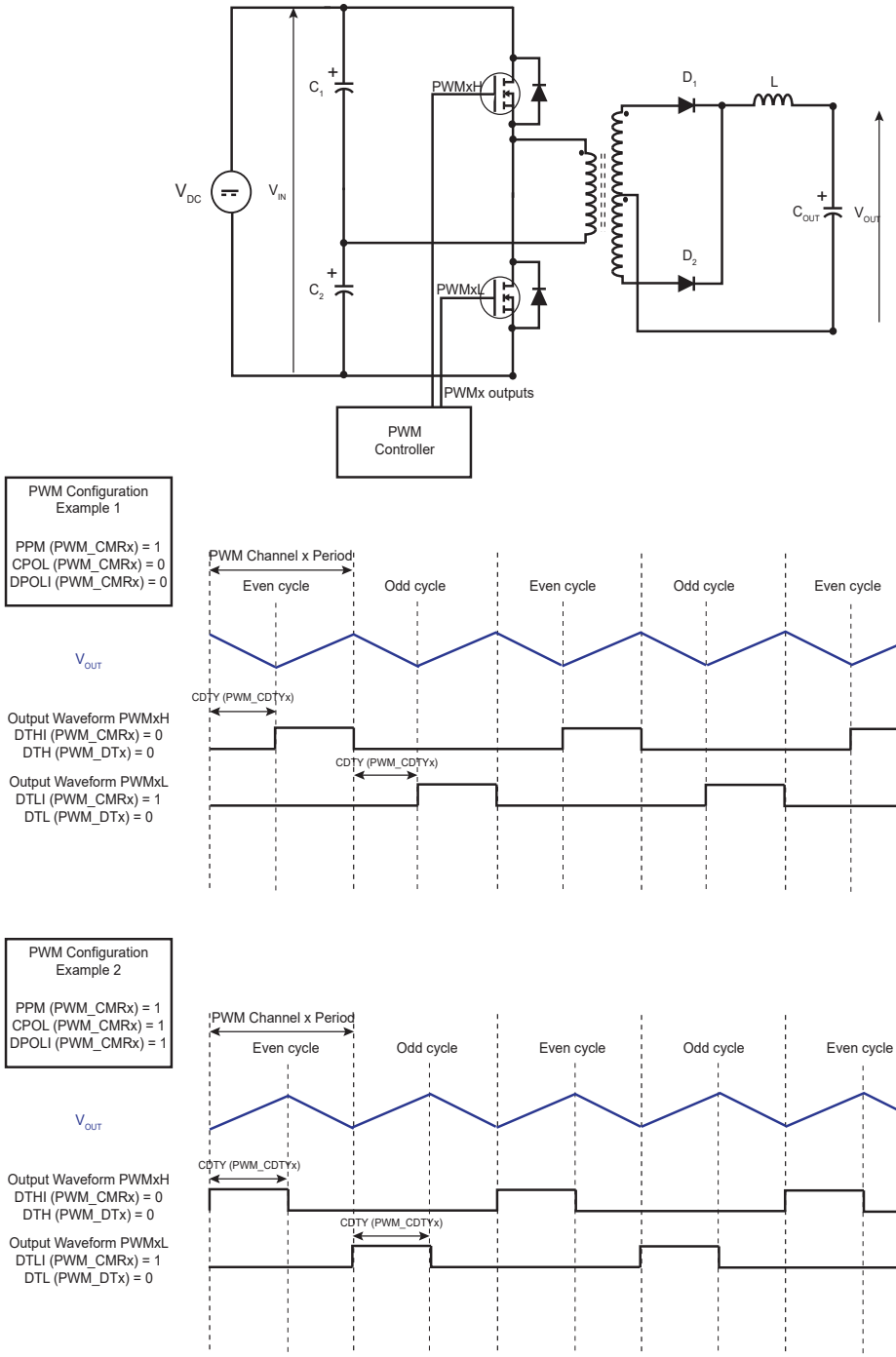
Figure 46-12. PWM Push-Pull Waveforms: Center-Aligned Mode



The PWM Push-Pull mode can be useful in transformer-based power converters, such as a half-bridge converter. The Push-Pull mode prevents the transformer core from being saturated by any direct current.

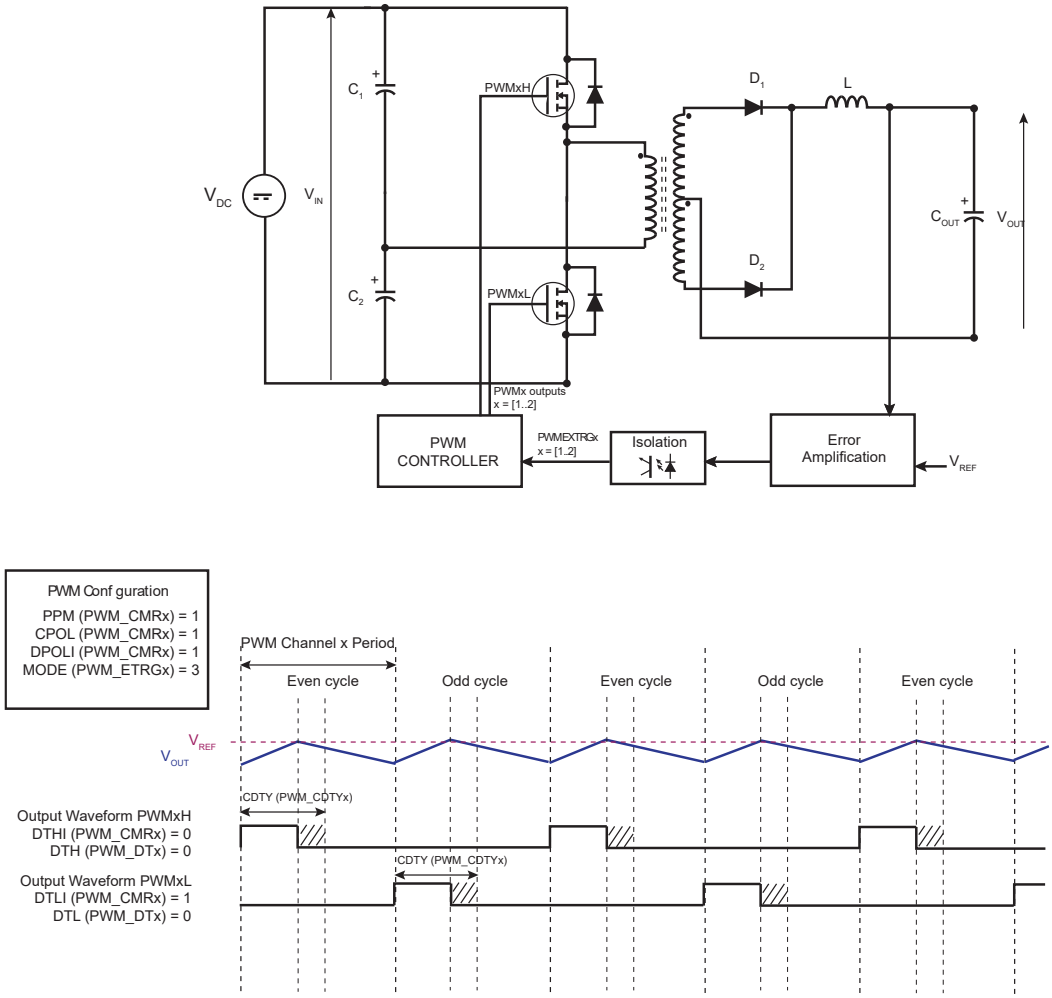
© 2026 Microchip Technology Inc. and its subsidiaries

Figure 46-13. Half-Bridge Converter Application: No Feedback Regulation



© 2026 Microchip Technology Inc. and its subsidiaries

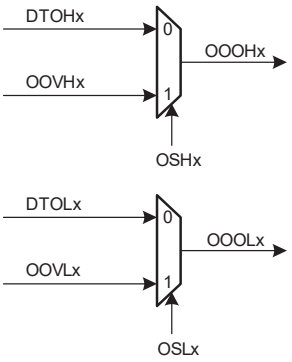
Figure 46-14. Half-Bridge Converter Application: Feedback Regulation



46.6.2.6. Output Override

The two complementary outputs DTOHx and DTOLx of the dead-time generator can be forced to a value defined by the software.

Figure 46-15. Override Output Selection



The fields OSHx and OSLx in the PWM Output Selection Register (PWM_OS) allow the outputs of the dead-time generator DTOHx and DTOLx to be overridden by the value defined in the fields OOVHx and OOVLx in the PWM Output Override Value Register (PWM_OOV).

© 2026 Microchip Technology Inc. and its subsidiaries

The set registers PWM Output Selection Set Register (PWM_OSS) and PWM Output Selection Set Update Register (PWM_OSSUPD) enable the override of the outputs of a channel regardless of other channels. In the same way, the clear registers PWM Output Selection Clear Register (PWM_OSC) and PWM Output Selection Clear Update Register (PWM_OSCUPD) disable the override of the outputs of a channel regardless of other channels.

By using buffer registers PWM_OSSUPD and PWM_OSCUPD, the output selection of PWM outputs is done synchronously to the channel counter, at the beginning of the next PWM period.

By using registers PWM_OSS and PWM_OSC, the output selection of PWM outputs is done asynchronously to the channel counter, as soon as the register is written.

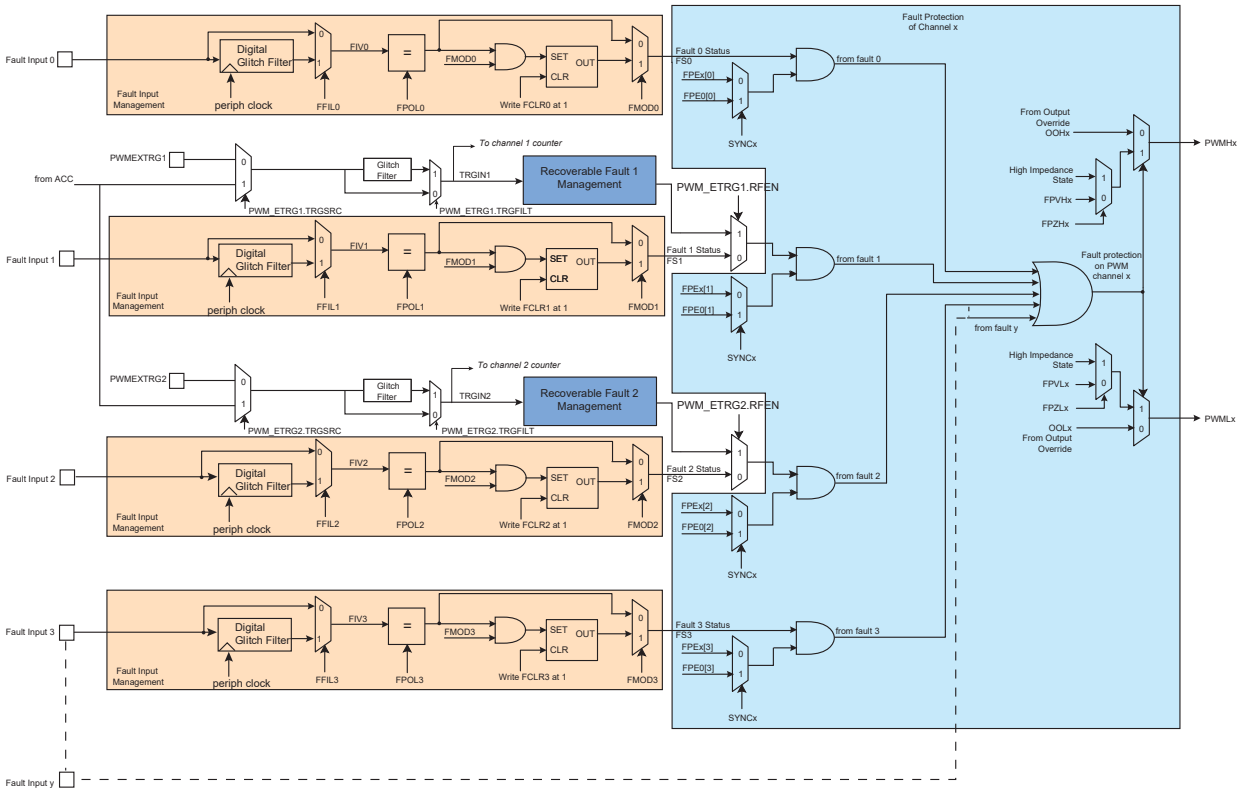
The value of the current output selection can be read in PWM_OS.

While overriding PWM outputs, the channel counters continue to run, only the PWM outputs are forced to user defined values.

46.6.2.7. Fault Protection

8 inputs provide fault protection which can force any of the PWM output pairs to a programmable value. This mechanism has priority over output overriding.

Figure 46-16. Fault Protection



The polarity level of the fault inputs is configured by the FPOL field in the PWM Fault Mode Register (PWM_FMR). For fault inputs coming from internal peripherals such as ADC or Timer Counter, the polarity level must be FPOL = 1. For fault inputs coming from external GPIO pins the polarity level depends on the user's implementation.

The configuration of the Fault Activation mode (FMOD field in PWMC_FMR) depends on the peripheral generating the fault. If the corresponding peripheral does not have “Fault Clear” management, then the FMOD configuration to use must be FMOD = 1, to avoid spurious fault

© 2026 Microchip Technology Inc. and its subsidiaries

detection. Refer to the corresponding peripheral documentation for details on handling fault generation.

Fault inputs may or may not be glitch-filtered depending on the FFIL field in PWM_FMR. When the filter is activated, glitches on fault inputs with a width inferior to the PWM peripheral clock period are rejected.

A fault becomes active as soon as its corresponding fault input has a transition to the programmed polarity level. If the corresponding bit FMOD is set to '0' in PWM_FMR, the fault remains active as long as the fault input is at this polarity level. If the corresponding FMOD field is set to '1', the fault remains active until the fault input is no longer at this polarity level and until it is cleared by writing the corresponding bit FCLR in the PWM Fault Clear Register (PWM_FCR). In the PWM Fault Status Register (PWM_FSR), the field FIV indicates the current level of the fault inputs and the field FIS indicates whether a fault is currently active.

Each fault can be taken into account or not by the fault protection mechanism in each channel. To be taken into account in the channel x, the fault y must be enabled by the bit FPEx[y] in the PWM Fault Protection Enable register (PWM_FPE). However, synchronous channels (see Synchronous Channels) do not use their own fault enable bits, but those of the channel 0 (bits FPE0[y]).

The fault protection on a channel is triggered when this channel is enabled and when any one of the faults that are enabled for this channel is active. It can be triggered even if the PWM peripheral clock is not running but only by a fault input that is not glitch-filtered.

When the fault protection is triggered on a channel, the fault protection mechanism resets the counter of this channel and forces the channel outputs to the values defined by the fields FPVHx and FPVLx in the PWM Fault Protection Value Register 1 (PWM_FPV) and fields FPZHx/FPZLx in the PWM Fault Protection Value Register 2, as shown in the table below. The output forcing is made asynchronously to the channel counter.

Table 46-4. Forcing Values of PWM Outputs by Fault Protection

FPZH/Lx	FPVH/Lx	Forcing Value of PWMH/Lx
0	0	0
0	1	1
1	–	High impedance state (Hi-Z)

CAUTION

- To prevent any unexpected activation of the status flag FSy in PWM_FSR, the FMODy bit can be set to '1' only if the FPOLy bit has been previously configured to its final value.
- To prevent any unexpected activation of the Fault Protection on the channel x, the bit FPEx[y] can be set to '1' only if the FPOLy bit has been previously configured to its final value.

If a comparison unit is enabled (see PWM Comparison Units) and if a fault is triggered in the channel 0, then the comparison cannot match.

As soon as the fault protection is triggered on a channel, an interrupt (different from the interrupt generated at the end of the PWM period) can be generated but only if it is enabled and not masked. The interrupt is reset by reading the interrupt status register, even if the fault which has caused the trigger of the fault protection is kept active.

46.6.2.7.1. Recoverable Fault

The PWM provides a Recoverable Fault mode on fault 1 and 2 (see figure Fault Protection).

The recoverable fault signal is an internal signal generated as soon as an external trigger event occurs (see PWM External Trigger Mode).

© 2026 Microchip Technology Inc. and its subsidiaries

When the fault 1 or 2 is defined as a recoverable fault, the corresponding fault input pin is ignored and bits FFIL1/2, FMOD1/2 and FFIL1/2 are not taken into account.

The fault 1 is managed as a recoverable fault by the PWMEXTRG1 input trigger when PWM_ETRG1.RFEN = 1, PWM_ENA.CHID1 = 1, and PWM_ETRG1.TRGMODE ≠ 0.

The fault 2 is managed as a recoverable fault by the PWMEXTRG2 input trigger when PWM_ETRG2.RFEN = 1, PWM_ENA.CHID2 = 1, and PWM_ETRG2.TRGMODE ≠ 0.

Recoverable fault 1 and 2 can be taken into account by all channels by enabling the bit FPEx[1/2] in the PWM Fault Protection Enable registers (PWM_FPEx). However the synchronous channels (see Synchronous Channels) do not use their own fault enable bits, but those of the channel 0 (bits FPE0[1/2]).

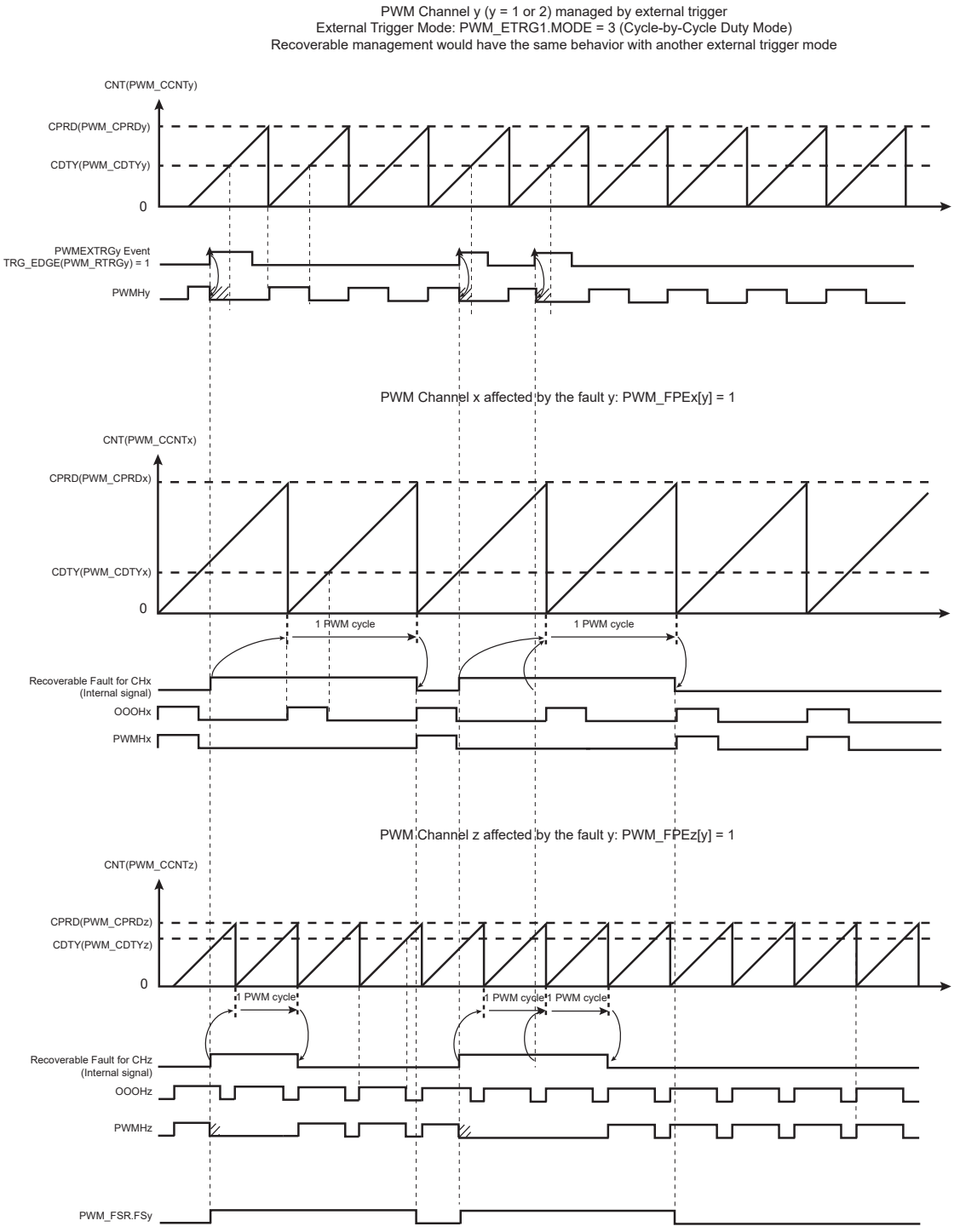
When a recoverable fault is triggered (according to the PWM_ETRGx.TRGMODE setting), the PWM counter of the affected channels is not cleared (unlike in the classic fault protection mechanism) but the channel outputs are forced to the values defined by the fields FPVHx and FPVLx in the PWM Fault Protection Value Register 1 (PWM_FPV), as per table *Forcing Values of PWM Outputs by Fault Protection*. The output forcing is made asynchronously to the channel counter and lasts from the recoverable fault occurrence to the end of the next PWM cycle (if the recoverable fault is no longer present) (see the figure below).

The recoverable fault does not trigger an interrupt. The Fault Status FSy (with y = 1 or 2) is not reported in the PWM Fault Status Register when the fault y is a recoverable fault.

See Cycle-By-Cycle Duty Mode: LED String Control for an application case associating the Recoverable Fault mode with the External Trigger mode.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 46-17. Recoverable Fault Management



46.6.2.8. Spread Spectrum Counter

The PWM macrocell includes a spread spectrum counter allowing the generation of a constantly varying duty cycle on the output PWM waveform (only for the channel 0). This feature may be useful to minimize electromagnetic interference or to reduce the acoustic noise of a PWM driven motor.

This is achieved by varying the effective period in a range defined by a spread spectrum value which is programmed by the field SPRD in the PWM Spread Spectrum Register (PWM_SSPR). The

© 2026 Microchip Technology Inc. and its subsidiaries

effective period of the output waveform is the value of the spread spectrum counter added to the programmed waveform period CPRD in the PWM Channel Period Register (PWM_CPRD0).

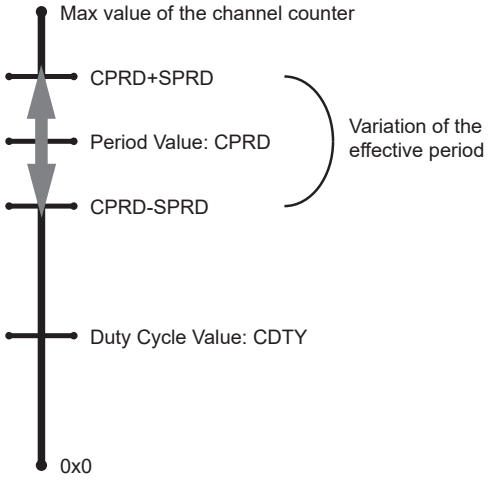
It will cause the effective period to vary from CPRD-SPRD to CPRD+SPRD. This leads to a constantly varying duty cycle on the PWM output waveform because the duty cycle value programmed is unchanged.

The value of the spread spectrum counter can change in two ways depending on the bit SPRDM in PWM_SSPR.

If SPRDM = 0, the Triangular mode is selected. The spread spectrum counter starts to count from -SPRD when the channel 0 is enabled or after reset and counts upwards at each period of the channel counter. When it reaches SPRD, it restarts to count from -SPRD again.

If SPRDM = 1, the Random mode is selected. A new random value is assigned to the spread spectrum counter at each period of the channel counter. This random value is between -SPRD and +SPRD and is uniformly distributed.

Figure 46-18. Spread Spectrum Counter



46.6.2.9. Synchronous Channels

Some channels can be linked together as synchronous channels. They have the same source clock, the same period, the same alignment and are started together. In this way, their counters are synchronized together.

The synchronous channels are defined by the SYNCx bits in the PWM Sync Channels Mode Register (PWM_SCM). Only one group of synchronous channels is allowed.

When a channel is defined as a synchronous channel, the channel 0 is also automatically defined as a synchronous channel. This is because the channel 0 counter configuration is used by all the synchronous channels.

If a channel x is defined as a synchronous channel, the fields/bits for the channel 0 are used instead of those of channel x:

- CPRE in PWM_CMR0 instead of CPRE in PWM_CMRx (same source clock)
- CPRD in PWM_CPRD0 instead of CPRD in PWM_CPRDx (same period)
- CALG in PWM_CMR0 instead of CALG in PWM_CMRx (same alignment)

Modifying the fields CPRE, CPRD and CALG of for channels with index greater than 0 has no effect on output waveforms.

Because counters of synchronous channels must start at the same time, they are all enabled together by enabling the channel 0 (by the CHID0 bit in PWM_ENA register). In the same way, they

© 2026 Microchip Technology Inc. and its subsidiaries

are all disabled together by disabling channel 0 (by the CHID0 bit in PWM_DIS register). However, a synchronous channel x different from channel 0 can be enabled or disabled independently from others (by the CHIDx bit in PWM_ENA and PWM_DIS registers).

Defining a channel as a synchronous channel while it is an asynchronous channel (by writing the bit SYNCx to '1' while it was at '0') is allowed only if the channel is disabled at this time (CHIDx = 0 in PWM_SR). In the same way, defining a channel as an asynchronous channel while it is a synchronous channel (by writing the SYNCx bit to '0' while it was '1') is allowed only if the channel is disabled at this time.

The UPDM field (Update Mode) in the PWM_SCM register selects one of the three methods to update the registers of the synchronous channels:

- Method 1 (UPDM = 0): The period value, the duty-cycle values and the dead-time values must be written by the processor in their respective update registers (respectively PWM_CPRDUPDx, PWM_CDTYUPDx and PWM_DTUPDx).The update is triggered at the next PWM period as soon as the bit UPDULOCK in the PWM Sync Channels Update Control Register (PWM_SCUC) is set to '1'.
- Method 2 (UPDM = 1): The period value, the duty-cycle values, the dead-time values and the update period value must be written by the processor in their respective update registers (respectively PWM_CPRDUPDx, PWM_CDTYUPDx and PWM_DTUPD). The update of the period value and of the dead-time values is triggered at the next PWM period as soon as the bit UPDULOCK in the PWM_SCUC register is set to '1'. The update of the duty-cycle values and the update period value is triggered automatically after an update period defined by the field UPR in the PWM Sync Channels Update Period Register (PWM_SCUP).
- Method 3 (UPDM = 2): Same as Method 2 apart from the fact that the duty-cycle values of ALL synchronous channels are written by the Peripheral DMA Controller. The user can choose to synchronize the Peripheral DMA Controller transfer request with a comparison match (see Section 7.3 "PWM Comparison Units"), by the fields PTRM and PTRCS in the PWM_SCM register.

Table 46-5. Summary of the Update of Registers of Synchronous Channels

Register	UPDM = 0	UPDM = 1	UPDM = 2
Period Value (PWM_CPRDUPDx)	Write by the processor		
	Update is triggered at the next PWM period as soon as the bit UPDULOCK is set to '1'		
Dead-Time Values (PWM_DTUPDx)	Write by the processor		
	Update is triggered at the next PWM period as soon as the bit UPDULOCK is set to '1'		
Duty-Cycle Values (PWM_CDTYUPDx)	Write by the processor	Write by the processor	Write by the Peripheral DMA Controller
	Update is triggered at the next PWM period as soon as the bit UPDULOCK is set to '1'	Update is triggered at the next PWM period as soon as the update period counter has reached the value UPR	
Update Period Value (PWM_SCUPUPD)	Not applicable	Write by the processor	
	Not applicable	Update is triggered at the next PWM period as soon as the update period counter has reached the value UPR	

46.6.2.9.1. Method 1: Manual write of duty-cycle values and manual trigger of the update

In this mode, the update of the period value, the duty-cycle values and the dead-time values must be done by writing in their respective update registers with the processor (respectively PWM_CPRDUPDx, PWM_CDTYUPDx and PWM_DTUPDx).

To trigger the update, the user must use the bit UPDULOCK in the PWM_SCUC register which allows to update synchronously (at the same PWM period) the synchronous channels:

- If the bit UPDULOCK is set to '1', the update is done at the next PWM period of the synchronous channels.
- If the UPDULOCK bit is not set to '1', the update is locked and cannot be performed.

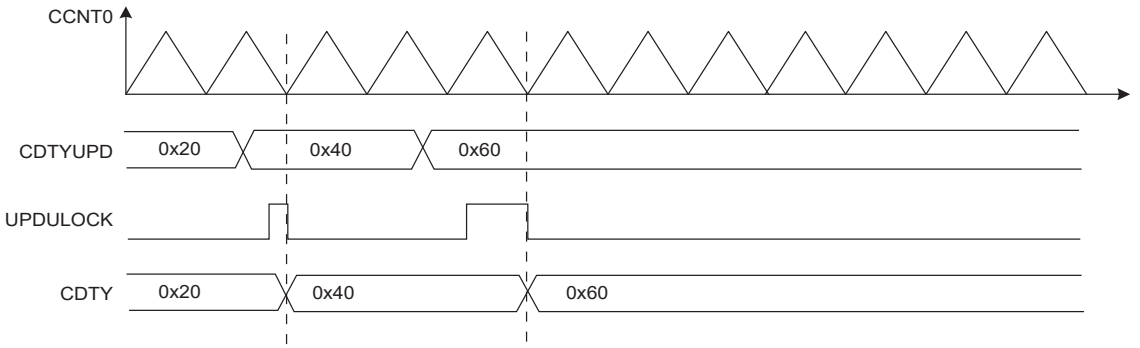
© 2026 Microchip Technology Inc. and its subsidiaries

After writing the UPDULOCK bit to '1', it is held at this value until the update occurs, then it is read 0.

Sequence for Method 1:

1. Select the manual write of duty-cycle values and the manual update by setting the UPDM field to '0' in the PWM_SCM register.
2. Define the synchronous channels by the SYNCx bits in the PWM_SCM register.
3. Enable the synchronous channels by writing CHID0 in the PWM_ENA register.
4. If an update of the period value and/or the duty-cycle values and/or the dead-time values is required, write registers that need to be updated (PWM_CPRDUPDx, PWM_CDTYUPDx and PWM_DTUPDx).
5. Set UPDULOCK to '1' in PWM_SCUC.
6. The update of the registers will occur at the beginning of the next PWM period. When the UPDULOCK bit is reset, go to Step 4. for new values.

Figure 46-19. Method 1 (UPDM = 0)



46.6.2.9.2. Method 2: Manual write of duty-cycle values and automatic trigger of the update

In this mode, the update of the period value, the duty-cycle values, the dead-time values and the update period value must be done by writing in their respective update registers with the processor (respectively PWM_CPRDUPDx, PWM_CDTYUPDx, PWM_DTUPDx and PWM_SCUPUPD).

To trigger the update of the period value and the dead-time values, the user must use the bit UPDULOCK in the PWM_SCUC register, which updates synchronously (at the same PWM period) the synchronous channels:

- If the bit UPDULOCK is set to '1', the update is done at the next PWM period of the synchronous channels.
- If the UPDULOCK bit is not set to '1', the update is locked and cannot be performed.

After writing the UPDULOCK bit to '1', it is held at this value until the update occurs, then it is read 0.

The update of the duty-cycle values and the update period is triggered automatically after an update period.

To configure the automatic update, the user must define a value for the update period by the UPR field in the PWM_SCUP register. The PWM controller waits UPR+1 period of synchronous channels before updating automatically the duty values and the update period value.

The status of the duty-cycle value write is reported in the PWM Interrupt Status Register 2 (PWM_ISR2) by the following flags:

- WRDY: this flag is set to '1' when the PWM Controller is ready to receive new duty-cycle values and a new update period value. It is reset to '0' when the PWM_ISR2 register is read.

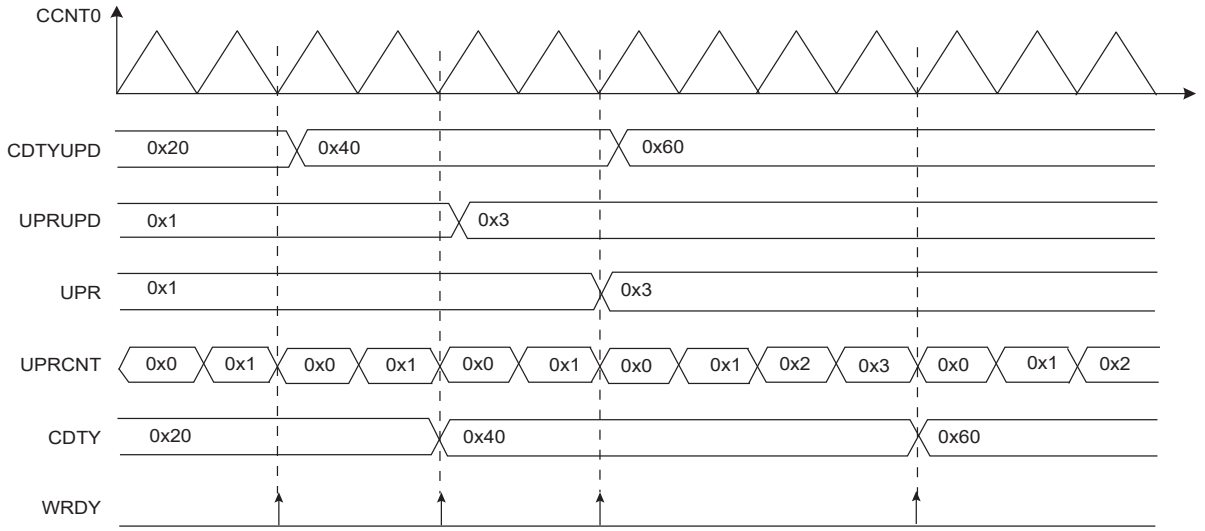
Depending on the interrupt mask in the PWM Interrupt Mask Register 2 (PWM_IMR2), an interrupt can be generated by these flags.

© 2026 Microchip Technology Inc. and its subsidiaries

Sequence for Method 2:

1. Select the manual write of duty-cycle values and the automatic update by setting the field UPDM to '1' in the PWM_SCM register
2. Define the synchronous channels by the bits SYNCx in the PWM_SCM register.
3. Define the update period by the field UPR in the PWM_SCUP register.
4. Enable the synchronous channels by writing CHID0 in the PWM_ENA register.
5. If an update of the period value and/or of the dead-time values is required, write registers that need to be updated (PWM_CPRDUPDx, PWM_DTUPDx), else go to Step 8.
6. Set UPDULOCK to '1' in PWM_SCUC.
7. The update of these registers will occur at the beginning of the next PWM period. At this moment the bit UPDULOCK is reset, go to Step 5. for new values.
8. If an update of the duty-cycle values and/or the update period is required, check first that write of new update values is possible by polling the flag WRDY (or by waiting for the corresponding interrupt) in PWM_ISR2.
9. Write registers that need to be updated (PWM_CDTYUPDx, PWM_SCUPUPD).
10. The update of these registers will occur at the next PWM period of the synchronous channels when the Update Period is elapsed. Go to Step 8. for new values.

Figure 46-20. Method 2 (UPDM = 1)



46.6.2.9.3. Method 3: Automatic write of duty-cycle values and automatic trigger of the update

In this mode, the update of the duty cycle values is made automatically by the Peripheral DMA Controller. The update of the period value, the dead-time values and the update period value must be done by writing in their respective update registers with the processor (respectively PWM_CPRDUPDx, PWM_DTUPDx and PWM_SCUPUPD).

To trigger the update of the period value and the dead-time values, the user must use the bit UPDULOCK which allows to update synchronously (at the same PWM period) the synchronous channels:

- If the bit UPDULOCK is set to '1', the update is done at the next PWM period of the synchronous channels.
- If the UPDULOCK bit is not set to '1', the update is locked and cannot be performed.

After writing the UPDULOCK bit to '1', it is held at this value until the update occurs, then it is read 0.

© 2026 Microchip Technology Inc. and its subsidiaries

The update of the duty-cycle values and the update period value is triggered automatically after an update period.

To configure the automatic update, the user must define a value for the Update Period by the field UPR in the PWM_SCUP register. The PWM controller waits UPR+1 periods of synchronous channels before updating automatically the duty values and the update period value.

Using the Peripheral DMA Controller removes processor overhead by reducing its intervention during the transfer. This significantly reduces the number of clock cycles required for a data transfer, which improves microcontroller performance.

The Peripheral DMA Controller must write the duty-cycle values in the synchronous channels index order. For example if the channels 0, 1 and 3 are synchronous channels, the Peripheral DMA Controller must write the duty-cycle of the channel 0 first, then the duty-cycle of the channel 1, and finally the duty-cycle of the channel 3.

The status of the Peripheral DMA Controller transfer is reported in PWM_ISR2 by the following flags:

- WRDY: this flag is set to '1' when the PWM Controller is ready to receive new duty-cycle values and a new update period value. It is reset to '0' when PWM_ISR2 is read. The user can choose to synchronize the WRDY flag and the Peripheral DMA Controller transfer request with a comparison match (see PWM Comparison Units), by the fields PTRM and PTRCS in the PWM_SCM register.
- ENDTX : this flag is set to '1' when a PDC transfer is completed
- TXBUFE : this flag is set to '1' when the PDC buffer is empty (no pending PDC transfers)
- UNRE: this flag is set to '1' when the update period defined by the UPR field has elapsed while the whole data has not been written by the Peripheral DMA Controller. It is reset to '0' when PWM_ISR2 is read.

Depending on the interrupt mask in PWM_IMR2, an interrupt can be generated by these flags.

Sequence for Method 3:

1. Select the automatic write of duty-cycle values and automatic update by setting the field UPDM to 2 in the PWM_SCM register.
2. Define the synchronous channels by the bits SYNCx in the PWM_SCM register.
3. Define the update period by the field UPR in the PWM_SCUP register.
4. Define when the WRDY flag and the corresponding Peripheral DMA Controller transfer request must be set in the update period by the PTRM bit and the PTRCS field in the PWM_SCM register (at the end of the update period or when a comparison matches).
5. Define the Peripheral DMA Controller transfer settings for the duty-cycle values and enable it in the Peripheral DMA Controller registers
6. Enable the synchronous channels by writing CHID0 in the PWM_ENA register.
7. If an update of the period value and/or of the dead-time values is required, write registers that need to be updated (PWM_CPRDUPDx, PWM_DTUPDx), else go to Step 10.
8. Set UPDULOCK to '1' in PWM_SCUC.
9. The update of these registers will occur at the beginning of the next PWM period. At this moment the bit UPDULOCK is reset, go to Step 7. for new values.
10. If an update of the update period value is required, check first that write of a new update value is possible by polling the flag WRDY (or by waiting for the corresponding interrupt) in PWM_ISR2, else go to Step 13.
11. Write the register that needs to be updated (PWM_SCUPUPD).
12. The update of this register will occur at the next PWM period of the synchronous channels when the Update Period is elapsed. Go to Step 10 for new values.

© 2026 Microchip Technology Inc. and its subsidiaries

13. Check the end of the PDC transfer by the flag ENDTX. If the transfer has ended, define a new PDC transfer in the PDC registers for new duty-cycle values. Go to Step 5.

Figure 46-21. Method 3 (UPDM = 2 and PTRM = 0)

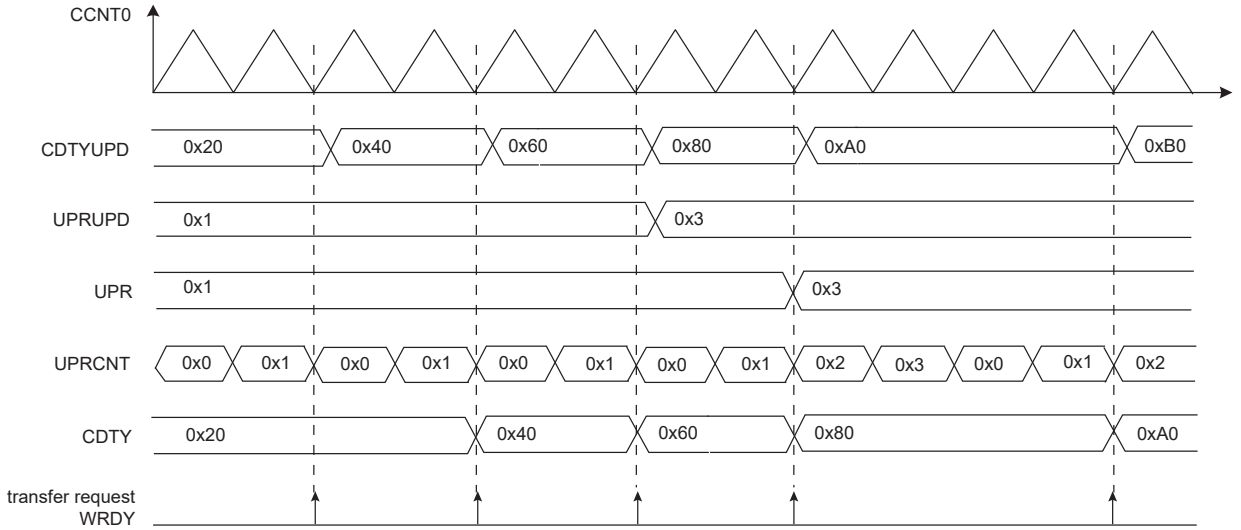
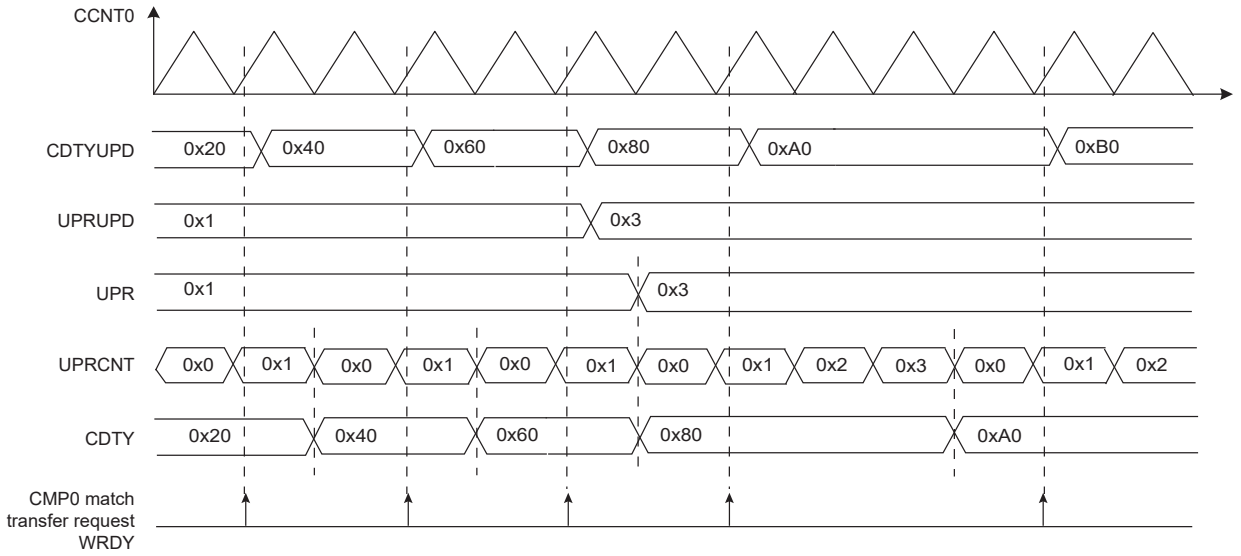


Figure 46-22. Method 3 (UPDM = 2 and PTRM = 1 and PTRCS = 0)



46.6.2.10. Update Time for Double-Buffering Registers

All channels integrate a double-buffering system in order to prevent an unexpected output waveform while modifying the period, the spread spectrum value, the polarity, the duty-cycle, the dead-times, the output override, and the synchronous channels update period.

This double-buffering system comprises the following update registers:

- PWM Sync Channels Update Period Update Register
- PWM Output Selection Set Update Register
- PWM Output Selection Clear Update Register
- PWM Spread Spectrum Update Register
- PWM Channel Duty Cycle Update Register

© 2026 Microchip Technology Inc. and its subsidiaries

- PWM Channel Period Update Register
- PWM Channel Dead Time Update Register
- PWM Channel Mode Update Register

When one of these update registers is written to, the write is stored, but the values are updated only at the next PWM period border. In Left-aligned mode (CALG = 0), the update occurs when the channel counter reaches the period value CPRD. In Center-aligned mode, the update occurs when the channel counter value is decremented and reaches the 0 value.

In Center-aligned mode, it is possible to trigger the update of the polarity and the duty-cycle at the next half period border. This mode concerns the following update registers:

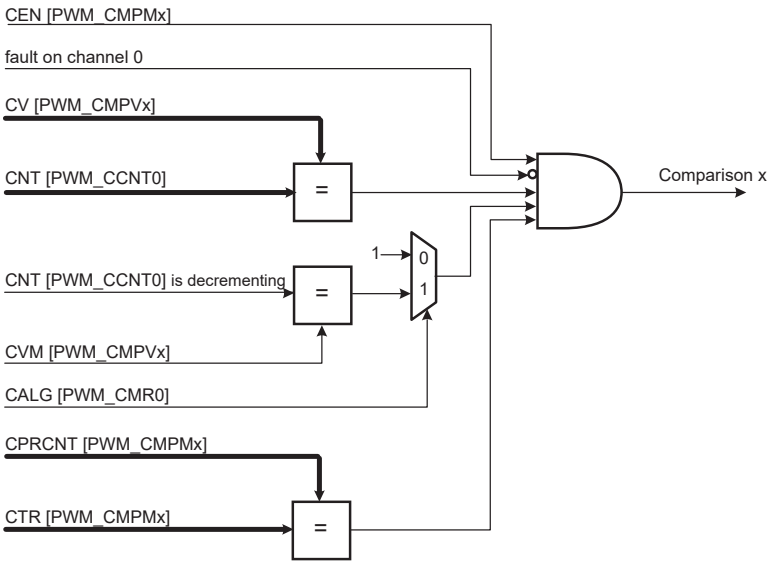
- PWM Channel Duty Cycle Update Register
- PWM Channel Mode Update Register

The update occurs at the first half period following the write of the update register (either when the channel counter value is incrementing and reaches the period value CPRD, or when the channel counter value is decrementing and reaches the 0 value). To activate this mode, the user must write a one to the bit UPDS in the PWM Channel Mode Register.

46.6.3. PWM Comparison Units

The PWM provides 8 independent comparison units able to compare a programmed value with the current value of the channel 0 counter (which is the channel counter of all synchronous channels, "Synchronous Channels"). These comparisons are intended to generate pulses on the event lines (used to synchronize ADC, see PWM Event Lines), to generate software interrupts and to trigger Peripheral DMA Controller transfer requests for the synchronous channels (see Method 3: Automatic write of duty-cycle values and automatic trigger of the update).

Figure 46-23. Comparison Unit Block Diagram



The comparison x matches when it is enabled by the bit CEN in the PWM Comparison x Mode Register (PWM_CMPMx for the comparison x) and when the counter of the channel 0 reaches the comparison value defined by the field CV in PWM Comparison x Value Register (PWM_CMPVx for the comparison x). If the counter of the channel 0 is center-aligned (CALG = 1 in PWM Channel Mode Register), the bit CVM in PWM_CMPVx defines if the comparison is made when the counter is counting up or counting down (in Left-alignment mode CALG = 0, this bit is useless).

© 2026 Microchip Technology Inc. and its subsidiaries

If a fault is active on the channel 0, the comparison is disabled and cannot match (see Fault Protection).

The user can define the periodicity of the comparison x by the fields CTR and CPR in PWM_CMPMx. The comparison is performed periodically once every CPR+1 periods of the counter of the channel 0, when the value of the comparison period counter CPRCNT in PWM_CMPMx reaches the value defined by CTR. CPR is the maximum value of the comparison period counter CPRCNT. If CPR = CTR = 0, the comparison is performed at each period of the counter of the channel 0.

The comparison x configuration can be modified while the channel 0 is enabled by using the PWM Comparison x Mode Update Register (PWM_CMPMUPDx registers for the comparison x). In the same way, the comparison x value can be modified while the channel 0 is enabled by using the PWM Comparison x Value Update Register (PWM_CMPVUPDx registers for the comparison x).

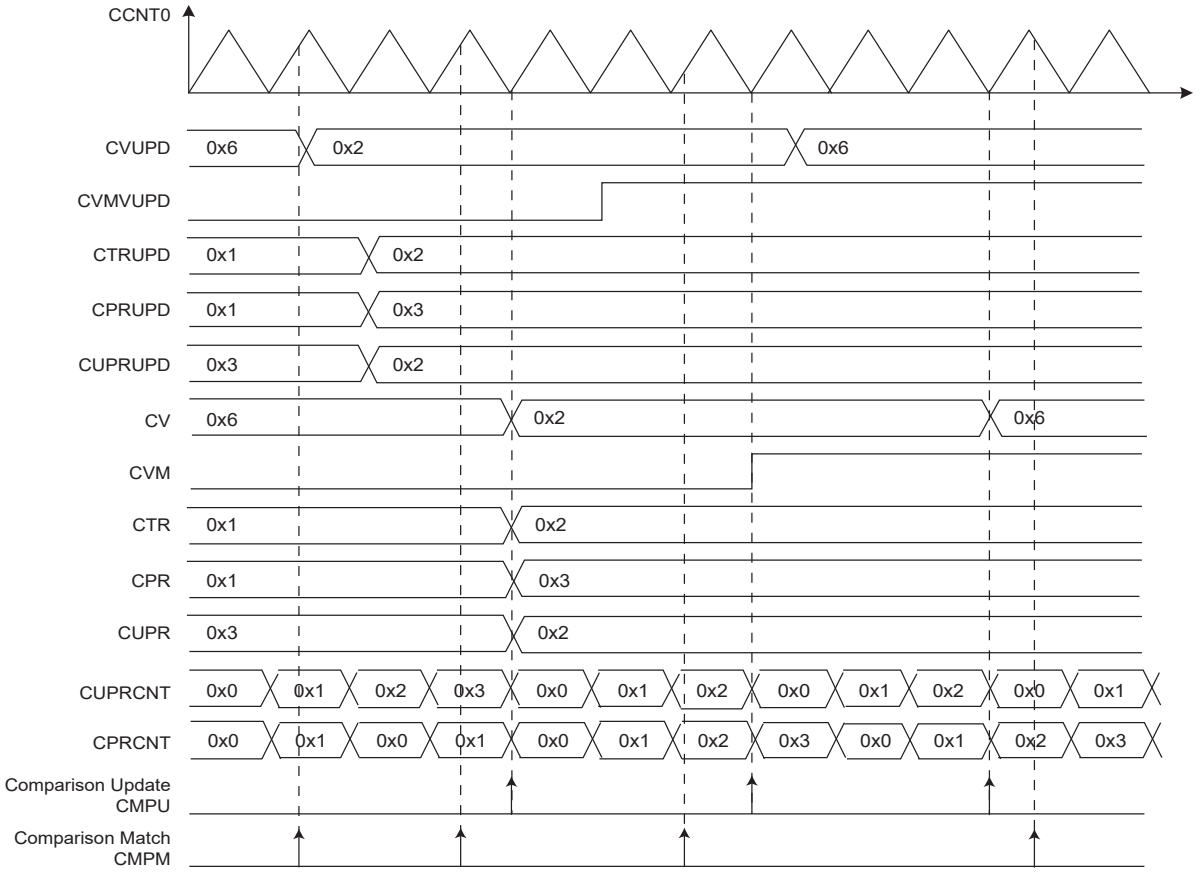
The update of the comparison x configuration and the comparison x value is triggered periodically after the comparison x update period. It is defined by the field CUPR in PWM_CMPMx. The comparison unit has an update period counter independent from the period counter to trigger this update. When the value of the comparison update period counter CUPRCNT (in PWM_CMPMx) reaches the value defined by CUPR, the update is triggered. The comparison x update period CUPR itself can be updated while the channel 0 is enabled by using the PWM_CMPMUPDx register.

⚠ CAUTION The write of PWM_CMPVUPDx must be followed by a write of PWM_CMPMUPDx.

The comparison match and the comparison update can be source of an interrupt, but only if it is enabled and not masked. These interrupts can be enabled by the PWM Interrupt Enable Register 2 and disabled by the PWM Interrupt Disable Register 2. The comparison match interrupt and the comparison update interrupt are reset by reading the PWM Interrupt Status Register 2.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 46-24. Comparison Waveform



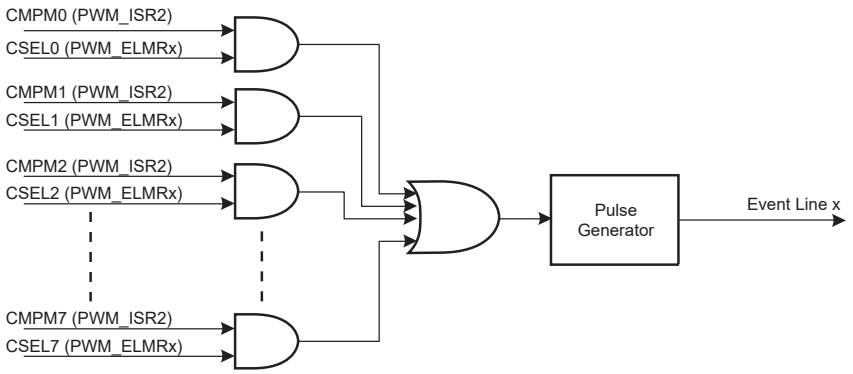
46.6.4. PWM Event Lines

The PWM provides 1 independent event lines intended to trigger actions in other peripherals (e.g., for the Analog-to-Digital Converter (ADC)).

A pulse (one cycle of the peripheral clock) is generated on an event line, when at least one of the selected comparisons is matching. The comparisons can be selected or unselected independently by the CSEL bits in the PWM Event Line x Register (PWM_ELMRx for the Event Line x).

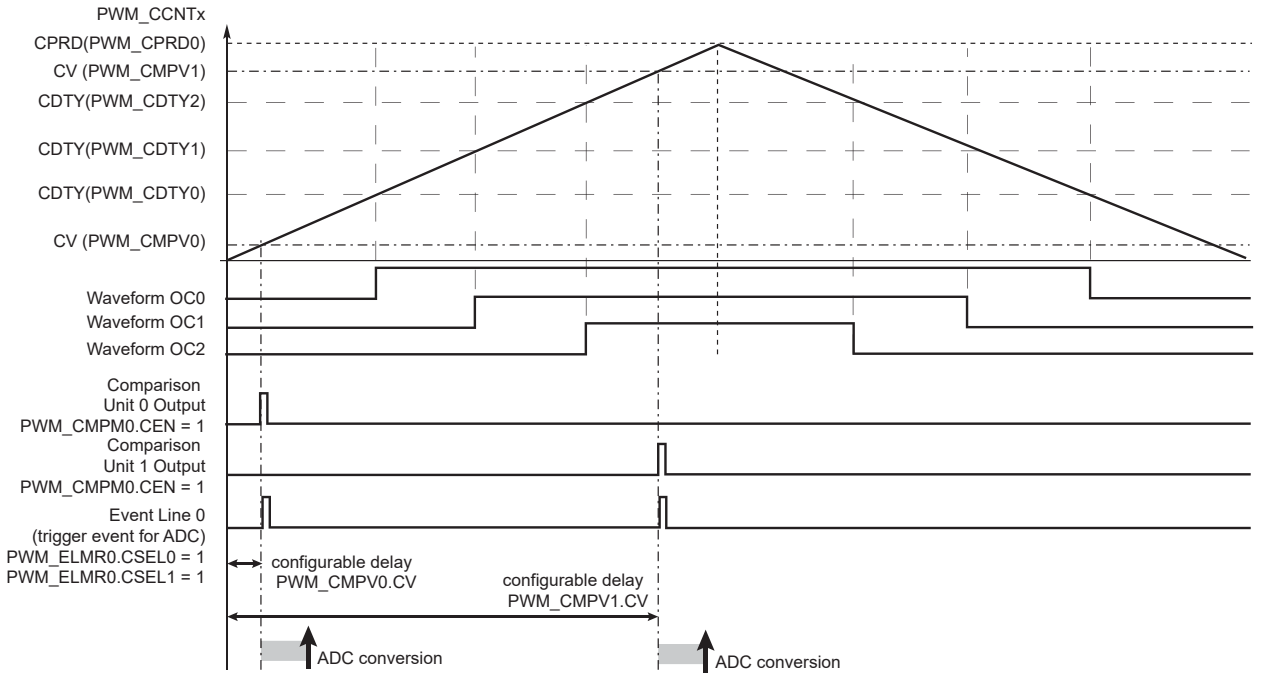
An example of event generation is provided in the figure Event Line Generation Waveform (Example).

Figure 46-25. Event Line Block Diagram



© 2026 Microchip Technology Inc. and its subsidiaries

Figure 46-26. Event Line Generation Waveform (Example)



46.6.5. Debug Mode

When debugging an application using breakpoints, the core is stopped in Debug mode when the breakpoint is hit. PWM registers are no longer updated by the application and thus PWML and/or PWMH outputs maintain the states or waveforms provided just before entering Debug mode. Depending on the application target, this state may not be valid for an undetermined period, or may have an undesired effect on devices driven by the PWM while the system is in Debug mode.

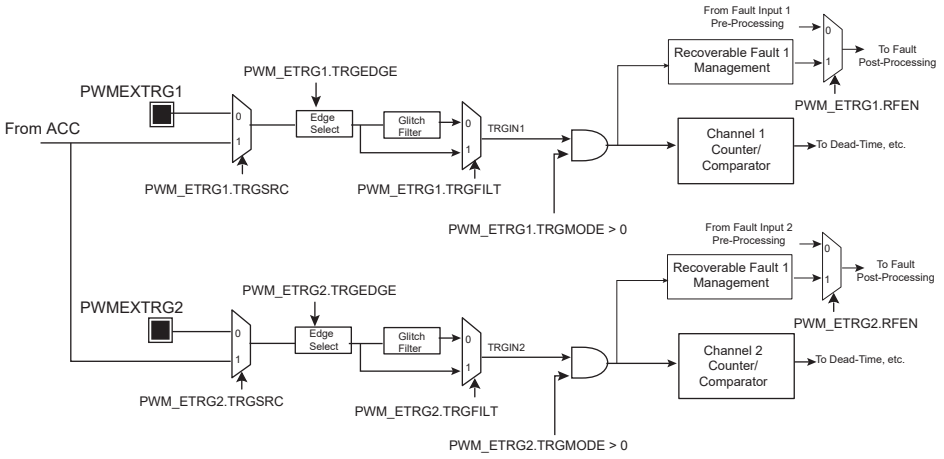
When entering Debug mode, the PWM outputs can be automatically and immediately forced to a predefined state if the bit OUTMODE=1 in the Debug register (PWM_DEBUG) (see PWM_DEBUG). The predefined state is defined in the Fault Protection Value registers (PWM_FPVx) (see PWM_FPV1 and PWM_FPV2).

46.6.6. PWM External Trigger Mode

The PWM channels 1 and 2 can be configured to use an external trigger for generating specific PWM signals to provide functions such as DC/DC converters, etc.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 46-27. External Trigger Mode Block Diagram



The external trigger source can be selected through the bit TRGSRC of the PWM External Trigger Register (see the table below).

Table 46-6. External Event Source Selection

Channel	Trigger Source Selection	Trigger Source
1	PWM_ETRG1.TRGSRC = 0	From PWMEXTRG1 input
	PWM_ETRG1.TRGSRC = 1	From Analog Comparator Controller
2	PWM_ETRG2.TRGSRC = 0	From PWMEXTRG2 input
	PWM_ETRG2.TRGSRC = 1	From Analog Comparator Controller

Each external trigger source can be filtered by writing a one to PWM_ETRGx.TRGFILT.

When the external trigger event is detected, the internal counter of the PWM channel can be modified when conditions are met, depending on the value of the PWM_ETRGx.TRGMODE field.

Each time an external trigger event is detected, the corresponding PWM channel counter value is stored in PWM_ETRGx.MAXCNT if it is greater than the previously stored value. Reading PWM_ETRGx clears the MAXCNT value.

To adapt to different use cases, three different external trigger mode modes (ETM) are available for channels 1 and 2 depending on the value of the PWM_ETRGx.TRGMODE field:

- TRGMODE = 1: External PWM Reset Mode (see External PWM Reset Mode)
- TRGMODE = 2: External PWM Start Mode (see External PWM Start Mode)
- TRGMODE = 3: Cycle-By-Cycle Duty Mode (see Cycle-By-Cycle Duty Mode)

The ETM feature is disabled when PWM_ETRGx.TRGMODE = 0.

The ETM mode can be associated with the recoverable fault mode (PWM_ETRGx.RFEN=1) to manage specific use case (see Cycle-By-Cycle Duty Mode: LED String Control).

The use cases described in the figures External PWM Start Mode: Buck DC/DC Converter and Cycle-By-Cycle Duty Mode are managed by PWM_ETRGx.RFEN=0.

The ETM must be enabled only if the corresponding channel is left-aligned (CALG = 0 in PWM Channel Mode Register of channel 1 or 2) and not managed as a synchronous channel (SYNCx = 0 in PWM Sync Channels Mode Register where x = 1 or 2). Programming the channel to be center-aligned or synchronous while TRGMODE is not 0 is forbidden.

46.6.6.1. External PWM Reset Mode

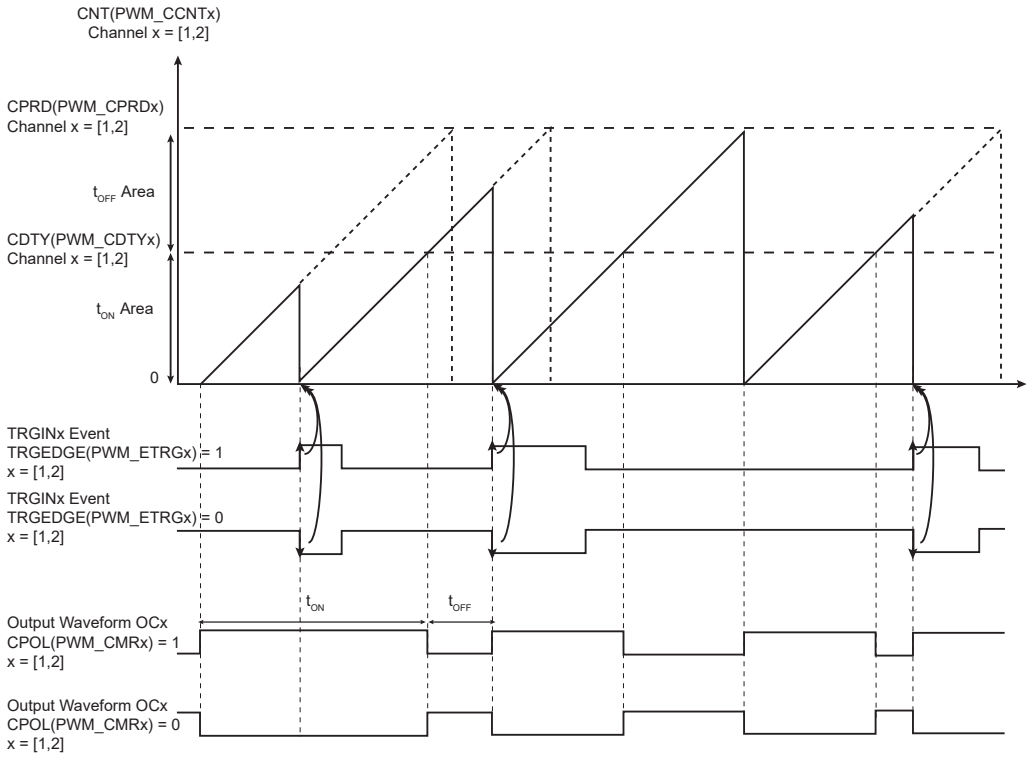
External PWM Reset mode is selected by programming TRGMODE = 1 in the PWM_ETRGx register.

© 2026 Microchip Technology Inc. and its subsidiaries

In this mode, when an edge is detected on the PWMEXTRGx input, the internal PWM counter is cleared and a new PWM cycle is restarted. The edge polarity can be selected by programming the TRGEDGE bit in the PWM_ETRGx register. If no trigger event is detected when the internal channel counter has reached the CPRD value in the PWM Channel Period Register, the internal counter is cleared and a new PWM cycle starts.

Note that this mode does not ensure a constant t_{ON} or t_{OFF} time.

Figure 46-28. External PWM Reset Mode



46.6.6.1.1. Application Example

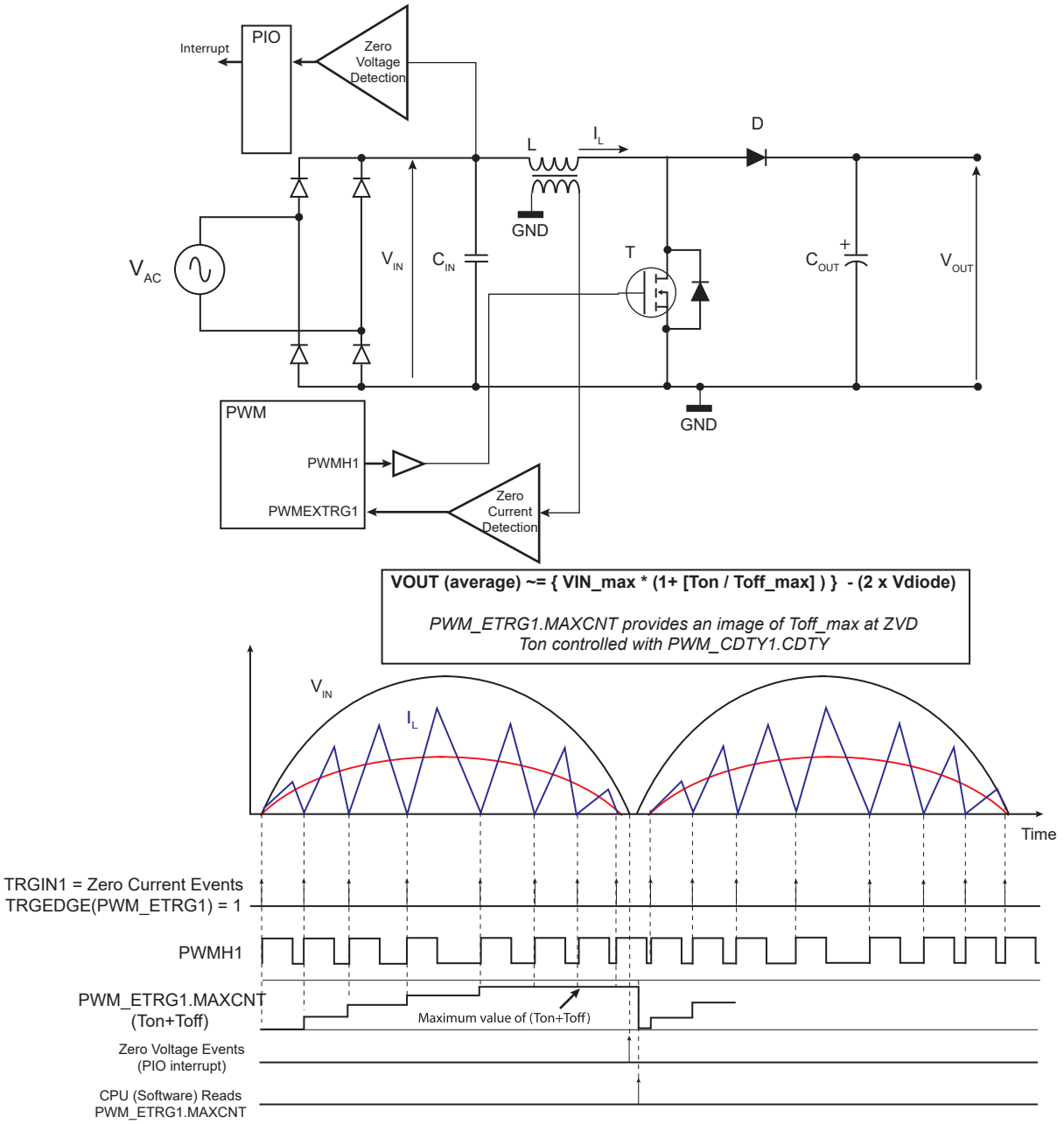
The external PWM Reset mode can be used in power factor correction applications.

In the example below, the external trigger input is the PWMEXTRG1 (therefore the PWM channel used for regulation is the channel 1). The PWM channel 1 period (CPRD in the PWM Channel Period Register of the channel 1) must be programmed so that the TRGIN1 event always triggers before the PWM channel 1 period elapses.

In the figure below, an external circuit (not shown) is required to sense the inductor current I_L. The internal PWM counter of the channel 1 is cleared when the inductor current falls below a specific threshold (I_{REF}). This starts a new PWM period and increases the inductor current.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 46-29. External PWM Reset Mode: Power Factor Correction Application



46.6.6.2. External PWM Start Mode

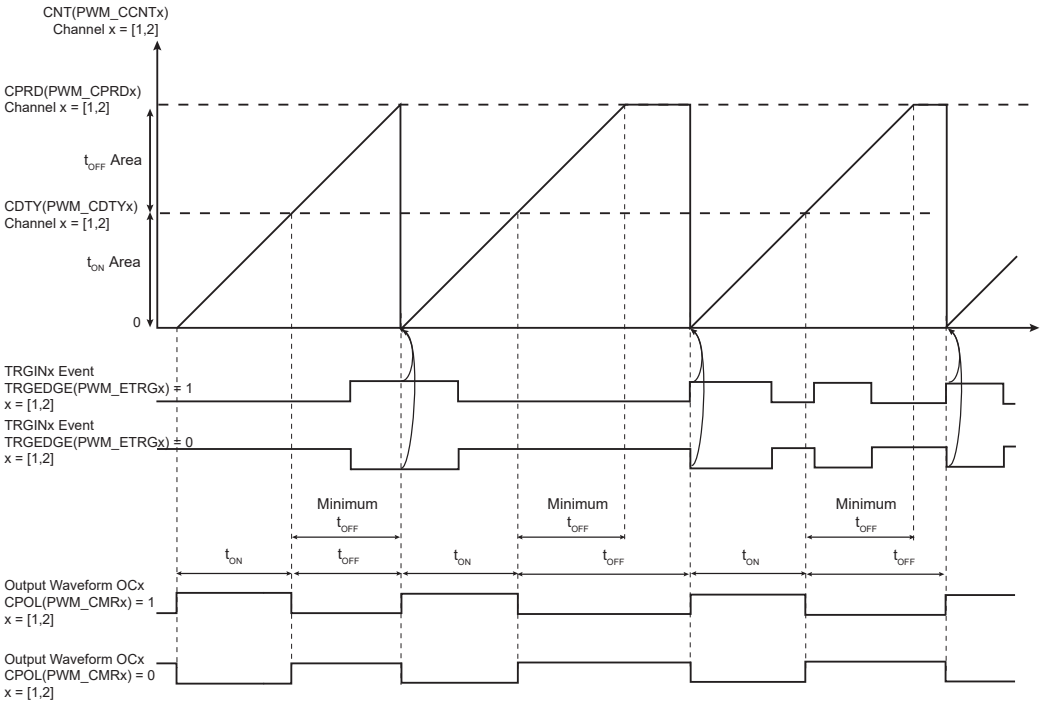
External PWM Start mode is selected by programming TRGMODE = 2 in the PWM_ETRGx register.

In this mode, the internal PWM counter can only be reset once it has reached the CPRD value in the PWM Channel Period Register and when the correct level is detected on the corresponding external trigger input. Both conditions have to be met to start a new PWM period. The active detection level is defined by the bit TRGEDGE of the PWM_ETRGx register.

Note that this mode ensures a constant t_{ON} time and a minimum t_{OFF} time.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 46-30. External PWM Start Mode



46.6.6.2.1. Application Example

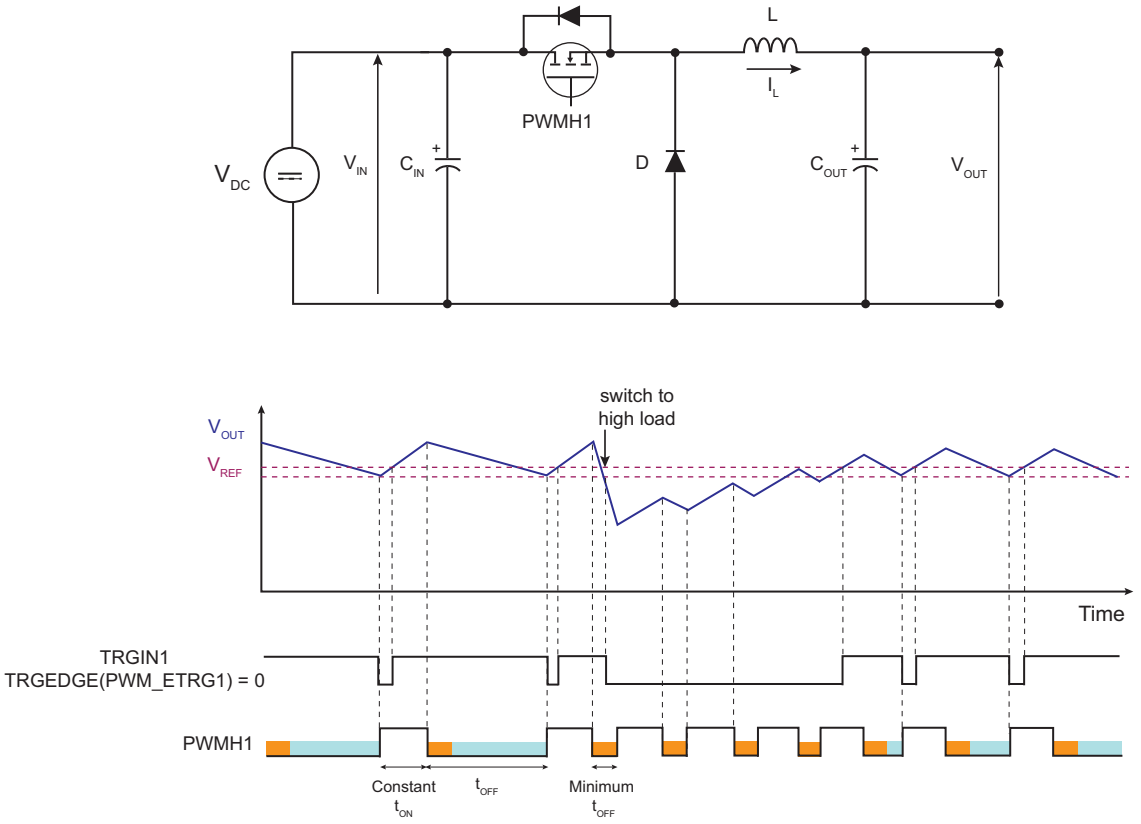
The external PWM Start mode generates a modulated frequency PWM signal with a constant active level duration (t_{ON}) and a minimum inactive level duration (minimum t_{OFF}).

The t_{ON} time is defined by the CDTY value in the PWM Channel Duty Cycle Register. The minimum t_{OFF} time is defined by CDTY - CPRD (PWM Channel Period Register). This mode can be useful in Buck DC/DC Converter applications.

When the output voltage V_{OUT} is above a specific threshold (Vref), the PWM inactive level is maintained as long as V_{OUT} remains above this threshold. If V_{OUT} is below this specific threshold, this mode ensures a minimum t_{OFF} time required for MOSFET driving (see the figure below).

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 46-31. External PWM Start Mode: Buck DC/DC Converter



46.6.6.3. Cycle-By-Cycle Duty Mode

Cycle-by-cycle duty mode is selected by programming TRGMODE = 3 in PWM_ETRGx.

In this mode, the PWM frequency is constant and is defined by the CPRD value in the PWM Channel Period Register.

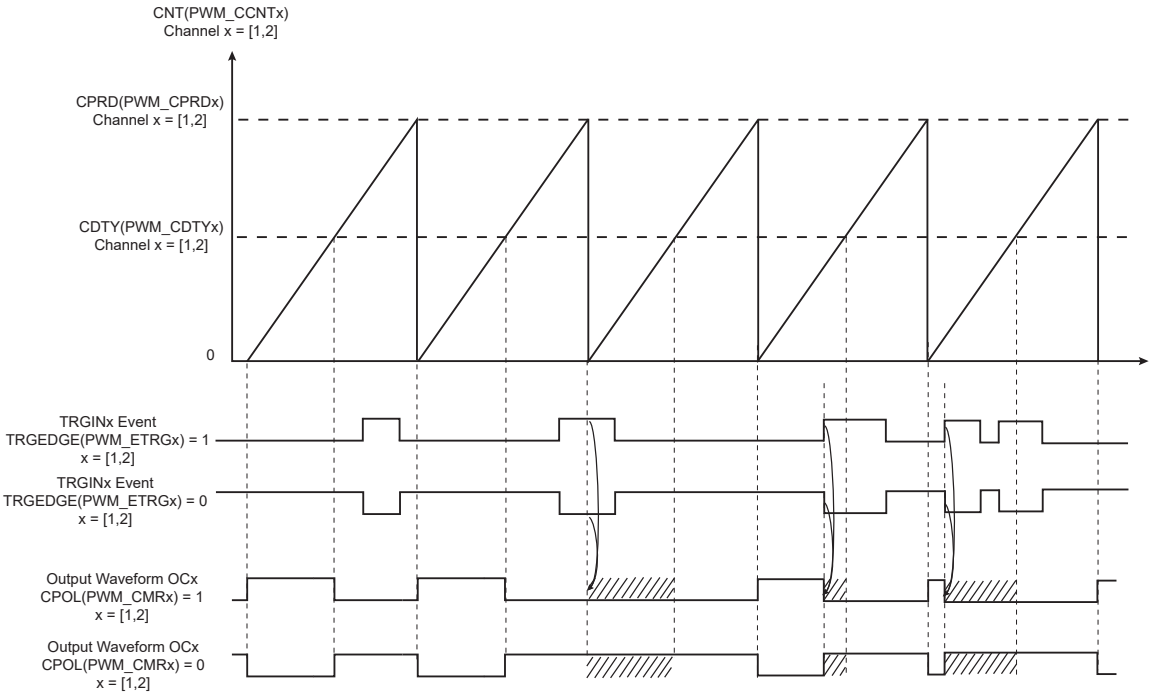
An external trigger event has no effect on the PWM output if it occurs while the internal PWM counter value is above the CDTY value of the PWM Channel Duty Cycle Register.

If the internal PWM counter value is below the value of CDTY of the PWM Channel Duty Cycle Register, an external trigger event makes the PWM output inactive.

The external trigger event can be detected on rising or falling edge according to the TRGEDGE bit in PWM_ETRGx.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 46-32. Cycle-By-Cycle Duty Mode



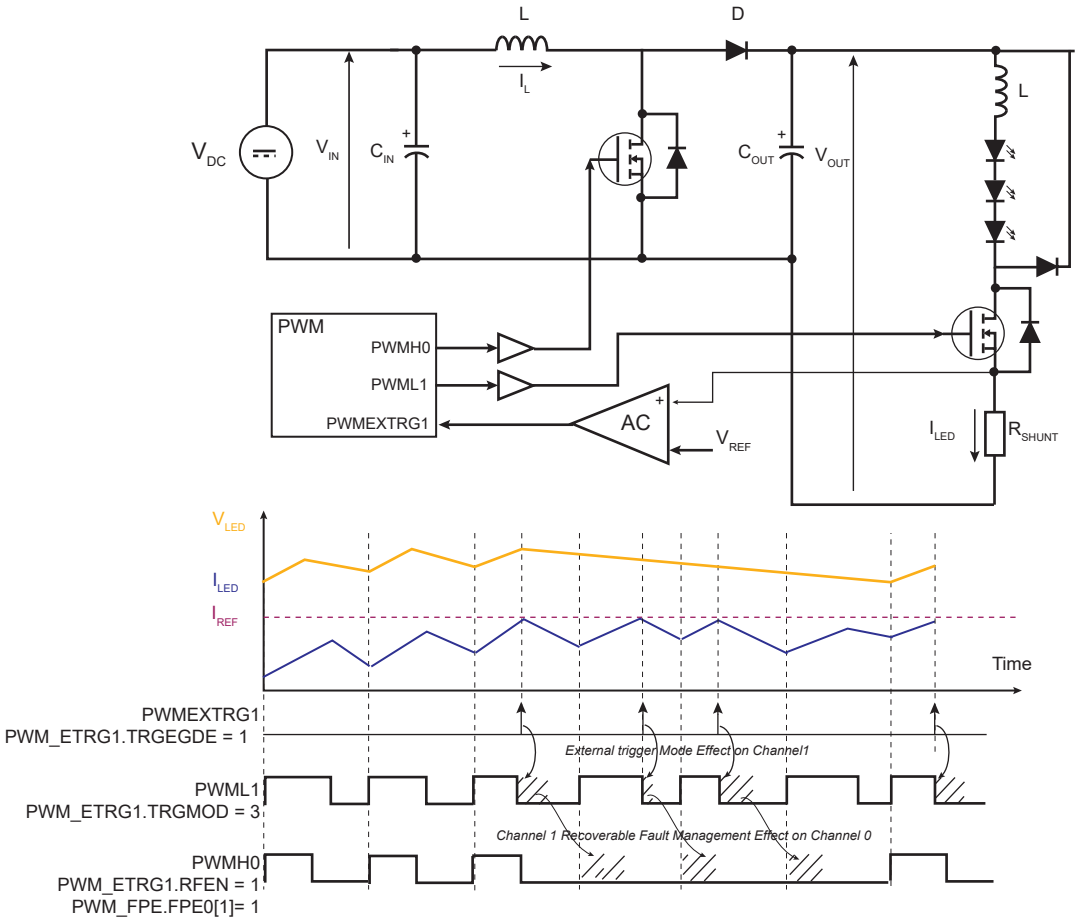
46.6.6.3.1. Application Example

The figure below illustrates an application example of the Cycle-by-cycle Duty mode.

In an LED string control circuit, Cycle-by-cycle Duty mode can be used to automatically limit the current in the LED string. This use case requires the recoverable fault mode to be enabled on channel 1 and associated fault signal must be enabled on channel 0.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 46-33. Cycle-By-Cycle Duty Mode: LED String Control



46.6.6.4. Leading-Edge Blanking (LEB)

PWM channels 1 and 2 support leading-edge blanking. Leading-edge blanking masks the external trigger input when a transient occurs on the corresponding PWM output. It masks potential spurious external events due to power transistor switching.

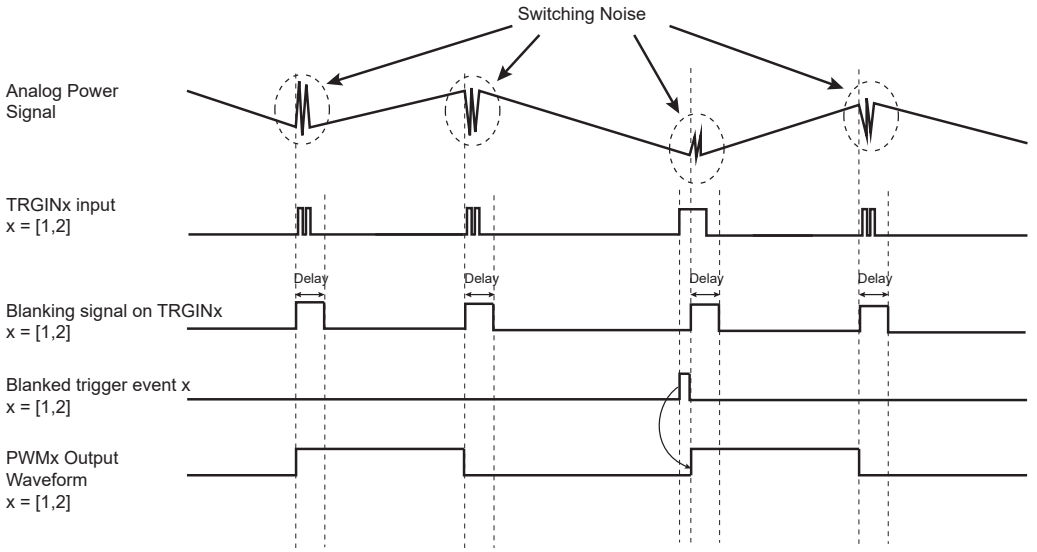
The blanking delay on each external trigger input is configured by programming the LEBDELAYx in the PWM Leading-Edge Blanking Register.

The LEB can be enabled on both the rising and the falling edges for the PWMH and PWML outputs through the bits PWMLFEN, PWMLREN, PWMHFEN, PWMHREN.

Any event on the PWMEXTRGx input which occurs during the blanking time is ignored.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 46-34. Leading-Edge Blanking



46.6.7. PWM Controller Operations

46.6.7.1. Initialization

Before enabling the channels, they must be configured by the software application as described below:

- Unlock User Interface by writing the WPCMD field in PWM_WPCR.
- Configuration of the clock generator (DIVA, PREA, DIVB, PREB in the PWM_CLK register if required).
- Selection of the clock for each channel (CPRE field in PWM_CMRx)
- Configuration of the waveform alignment for each channel (CALG field in PWM_CMRx)
- Selection of the counter event selection (if CALG = 1) for each channel (CES field in PWM_CMRx)
- Configuration of the output waveform polarity for each channel (CPOL bit in PWM_CMRx)
- Configuration of the period for each channel (CPRD in the PWM_CPRDx register). Writing in PWM_CPRDx register is possible while the channel is disabled. After validation of the channel, the user must use PWM_CPRDUPDx register to update PWM_CPRDx as explained below.
- Configuration of the duty-cycle for each channel (CDTY in the PWM_CDTYx register). Writing in PWM_CDTYx register is possible while the channel is disabled. After validation of the channel, the user must use PWM_CDTYUPDx register to update PWM_CDTYx as explained below.
- Configuration of the dead-time generator for each channel (DTH and DTL in PWM_DTx) if enabled (DTE bit in PWM_CMRx). Writing in the PWM_DTx register is possible while the channel is disabled. After validation of the channel, the user must use PWM_DTUPDx register to update PWM_DTx
- Selection of the synchronous channels (SYNCx in the PWM_SCM register)
- Selection of the moment when the WRDY flag and the corresponding Peripheral DMA Controller transfer request are set (PTRM and PTRCS in the PWM_SCM register)
- Configuration of the Update mode (UPDM in PWM_SCM register)
- Configuration of the update period (UPR in PWM_SCUP register) if needed
- Configuration of the comparisons (PWM_CMPVx and PWM_CMPMx)
- Configuration of the event lines (PWM_ELMRx)

© 2026 Microchip Technology Inc. and its subsidiaries

- Configuration of the fault inputs polarity (FPOL in PWM_FMR)
- Configuration of the fault protection (FMOD and FFIL in PWM_FMR, PWM_FPV and PWM_FPE1)
- Enable of the interrupts (writing CHIDx and FCHIDx in PWM_IER1, and writing WRDY, ENDTX, TXBUFE, UNRE, CMPMx and CMPUx in PWM_IER2)
- Enable of the PWM channels (writing CHIDx in the PWM_ENA register)

46.6.7.2. Source Clock Selection Criteria

The large number of source clocks can make selection difficult. The relationship between the value in the PWM Channel Period Register (PWM_CPRDx) and the PWM Channel Duty Cycle Register (PWM_CDTYx) helps the user select the appropriate clock. The event number written in the Period Register gives the PWM accuracy. The Duty-Cycle quantum cannot be lower than 1/CPRDx value. The higher the value of PWM_CPRDx, the greater the PWM accuracy.

For example, if the user sets 15 (in decimal) in PWM_CPRDx, the user is able to set a value from between 1 up to 14 in PWM_CDTYx. The resulting duty-cycle quantum cannot be lower than 1/15 of the PWM period.

46.6.7.3. Changing the Duty-Cycle, the Period and the Dead-Times

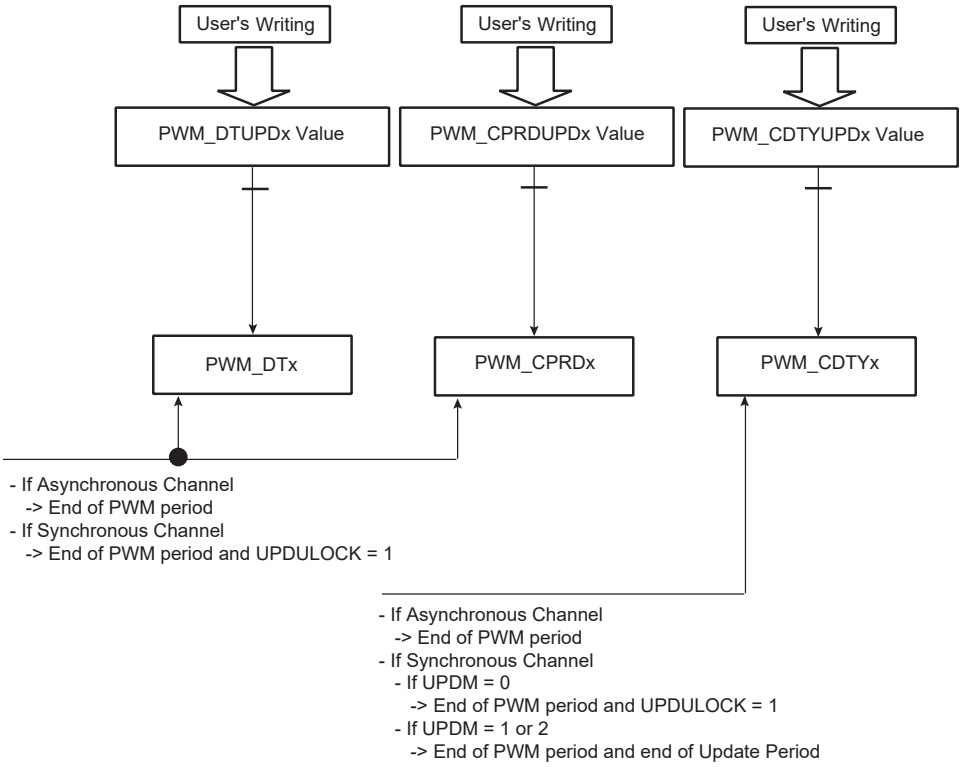
It is possible to modulate the output waveform duty-cycle, period and dead-times.

To prevent unexpected output waveform, the user must use the PWM Channel Duty Cycle Update Register (PWM_CDTYUPDx), the PWM Channel Period Update Register (PWM_CPRDUPDx) and the PWM Channel Dead Time Update Register (PWM_DTUPDx) to change waveform parameters while the channel is still enabled.

- If the channel is an asynchronous channel (SYNCx = 0 in PWM Sync Channels Mode Register (PWM_SCM)), these registers hold the new period, duty-cycle and dead-times values until the end of the current PWM period and update the values for the next period.
- If the channel is a synchronous channel and update method 0 is selected (SYNCx = 1 and UPDM = 0 in PWM_SCM register), these registers hold the new period, duty-cycle and dead-times values until the bit UPDULOCK is written at '1' (in PWM Sync Channels Update Control Register (PWM_SCUC)) and the end of the current PWM period, then update the values for the next period.
- If the channel is a synchronous channel and update method 1 or 2 is selected (SYNCx = 1 and UPDM = 1 or 2 in PWM_SCM register):
  - registers PWM_CPRDUPDx and PWM_DTUPDx hold the new period and dead-times values until the bit UPDULOCK is written at '1' (in PWM_SCUC) and the end of the current PWM period, then update the values for the next period.
  - register PWM_CDTYUPDx holds the new duty-cycle value until the end of the update period of synchronous channels (when UPRCNT is equal to UPR in PWM Sync Channels Update Period Register (PWM_SCUP)) and the end of the current PWM period, then updates the value for the next period.
    Note: If the update registers PWM_CDTYUPDx, PWM_CPRDUPDx and PWM_DTUPDx are written several times between two updates, only the last written value is taken into account.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 46-35. Synchronized Period, Duty-Cycle and Dead-Time Update



46.6.7.4. Changing the Update Period of Synchronous Channels

It is possible to change the update period of synchronous channels while they are enabled. See Method 2: Manual write of duty-cycle values and automatic trigger of the update and Method 3: Automatic write of duty-cycle values and automatic trigger of the update .

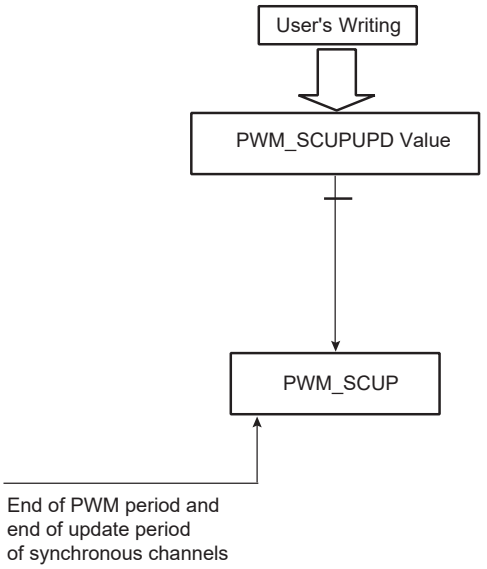
To prevent an unexpected update of the synchronous channels registers, the user must use the PWM Sync Channels Update Period Update Register (PWM_SCUPUPD) to change the update period of synchronous channels while they are still enabled. This register holds the new value until the end of the update period of synchronous channels (when UPRCNT is equal to UPR in PWM_SCUP) and the end of the current PWM period, then updates the value for the next period.

Notes:

1. If the update register PWM_SCUPUPD is written several times between two updates, only the last written value is taken into account.
2. Changing the update period does make sense only if there is one or more synchronous channels and if the update method 1 or 2 is selected (UPDM = 1 or 2 in PWM Sync Channels Mode Register).

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 46-36. Synchronized Update of Update Period Value of Synchronous Channels



46.6.7.5. Changing the Comparison Value and the Comparison Configuration

It is possible to change the comparison values and the comparison configurations while the channel 0 is enabled (see PWM Comparison Units).

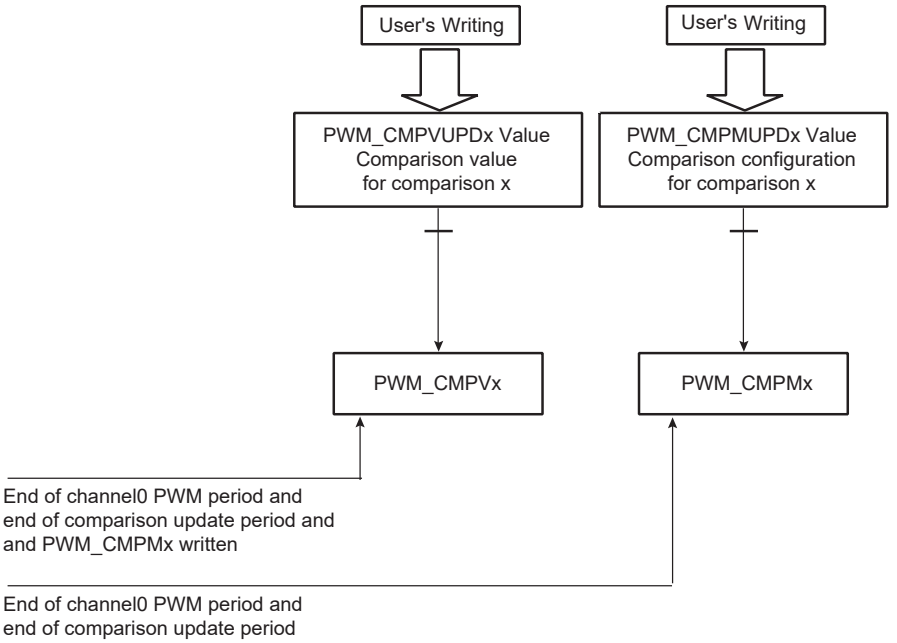
To prevent unexpected comparison match, the user must use the PWM Comparison x Value Update Register (PWM_CMPVUPDx) and the PWM Comparison x Mode Update Register (PWM_CMPMUPDx) to change, respectively, the comparison values and the comparison configurations while the channel 0 is still enabled. These registers hold the new values until the end of the comparison update period (when CUPRCNT is equal to CUPR in PWM Comparison x Mode Register (PWM_CMPMx) and the end of the current PWM period, then update the values for the next period.

CAUTION The write of the register PWM_CMPVUPDx must be followed by a write of the register PWM_CMPMUPDx.

Note: If the update registers PWM_CMPVUPDx and PWM_CMPMUPDx are written several times between two updates, only the last written value are taken into account.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 46-37. Synchronized Update of Comparison Values and Configurations



46.6.7.6. Interrupt Sources

Depending on the interrupt mask in PWM_IMR1 and PWM_IMR2, an interrupt can be generated at the end of the corresponding channel period (CHIDx in the PWM Interrupt Status Register 1 (PWM_ISR1)), after a fault event (FCHIDx in PWM_ISR1), after a comparison match (CMPMx in PWM_ISR2), after a comparison update (CMPUx in PWM_ISR2) or according to the Transfer mode of the synchronous channels (WRDY, ENDTX, TXBUFE and UNRE in PWM_ISR2).

If the interrupt is generated by the flags CHIDx or FCHIDx, the interrupt remains active until a read operation in PWM_ISR1 occurs.

If the interrupt is generated by the flags WRDY or UNRE or CMPMx or CMPUx, the interrupt remains active until a read operation in PWM_ISR2 occurs.

A channel interrupt is enabled by setting the corresponding bit in PWM_IER1 and PWM_IER2. A channel interrupt is disabled by setting the corresponding bit in PWM_IDR1 and PWM_IDR2.

46.6.8. Register Write Protection

To prevent any single software error that may corrupt PWM behavior, the registers listed below can be write-protected by writing the field WPCMD in the PWM Write Protection Control Register (PWM_WPCR). They are divided into six groups:

- Register group 0:
  - PWM Clock Register
- Register group 1:
  - PWM Disable Register
  - PWM Interrupt Enable Register 1
  - PWM Interrupt Disable Register 1
  - PWM Interrupt Enable Register 2
  - PWM Interrupt Disable Register 2
- Register group 2:
  - PWM Sync Channels Mode Register

© 2026 Microchip Technology Inc. and its subsidiaries

  - PWM Channel Mode Register
  - PWM Stepper Motor Mode Register
  - PWM Fault Protection Value Register 2
  - PWM Leading-Edge Blanking Register
  - PWM Channel Mode Update Register
- Register group 3:
  - PWM Spread Spectrum Register
  - PWM Spread Spectrum Update Register
  - PWM Channel Period Register
  - PWM Channel Period Update Register
- Register group 4:
  - PWM Channel Dead Time Register
  - PWM Channel Dead Time Update Register
- Register group 5:
  - PWM Fault Mode Register
  - PWM Fault Protection Value Register 1

There are two types of write protection:

- SW write protection—can be enabled or disabled by software
- HW write protection—can be enabled by software but only disabled by a hardware reset of the PWM controller

Both types of write protection can be applied independently to a particular register group by means of the WPCMD and WPRGx fields in PWM_WPCR. If at least one type of write protection is active, the register group is write-protected. The value of field WPCMD defines the action to be performed:

- 0: Disables SW write protection of the register groups of which the bit WPRGx is at '1'
- 1: Enables SW write protection of the register groups of which the bit WPRGx is at '1'
- 2: Enables HW write protection of the register groups of which the bit WPRGx is at '1'

At any time, the user can determine whether SW or HW write protection is active in a particular register group by the fields WPSWS and WPHWS in the PWM Write Protection Status Register (PWM_WPSR).

If a write access to a write-protected register is detected, the WPVS flag in PWM_WPSR is set and the field WPVSRC indicates the register in which the write access has been attempted.

The WPVS and WPVSRC fields are automatically cleared after reading PWM_WPSR.

© 2026 Microchip Technology Inc. and its subsidiaries

46.7. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	PWM_CLK	31:24					PREB[3:0]			
		23:16	DIVB[7:0]							
		15:8					PREA[3:0]			
		7:0	DIVA[7:0]							
0x04	PWM_ENA	31:24								
		23:16								
		15:8								
		7:0						CHID2	CHID1	CHID0
0x08	PWM_DIS	31:24								
		23:16								
		15:8								
		7:0						CHID2	CHID1	CHID0
0x0C	PWM_SR	31:24								
		23:16								
		15:8								
		7:0						CHID2	CHID1	CHID0
0x10	PWM_IER1	31:24								
		23:16						FCHID2	FCHID1	FCHID0
		15:8								
		7:0						CHID2	CHID1	CHID0
0x14	PWM_IDR1	31:24								
		23:16						FCHID2	FCHID1	FCHID0
		15:8								
		7:0						CHID2	CHID1	CHID0
0x18	PWM_IMR1	31:24								
		23:16						FCHID2	FCHID1	FCHID0
		15:8								
		7:0						CHID2	CHID1	CHID0
0x1C	PWM_ISR1	31:24								
		23:16						FCHID2	FCHID1	FCHID0
		15:8								
		7:0						CHID2	CHID1	CHID0
0x20	PWM_SCM	31:24								
		23:16		PTRCS[2:0]		PTRM			UPDM[1:0]	
		15:8								
		7:0						SYNC2	SYNC1	SYNC0
0x24 ... 0x27	Reserved									
0x28	PWM_SCUC	31:24								
		23:16								
		15:8								
		7:0								UPDULOCK
0x2C	PWM_SCUP	31:24								
		23:16								
		15:8								
		7:0		UPRCNT[3:0]			UPR[3:0]			
0x30	PWM_SCUPUPD	31:24								
		23:16								
		15:8								
		7:0					UPRUPD[3:0]			
0x34	PWM_IER2	31:24								
		23:16	CMPU7	CMPU6	CMPU5	CMPU4	CMPU3	CMPU2	CMPU1	CMPU0
		15:8	CMPM7	CMPM6	CMPM5	CMPM4	CMPM3	CMPM2	CMPM1	CMPM0
		7:0					UNRE	TXBUFE	ENDTX	WRDY
0x38	PWM_IDR2	31:24								
		23:16	CMPU7	CMPU6	CMPU5	CMPU4	CMPU3	CMPU2	CMPU1	CMPU0
		15:8	CMPM7	CMPM6	CMPM5	CMPM4	CMPM3	CMPM2	CMPM1	CMPM0
		7:0					UNRE	TXBUFE	ENDTX	WRDY

© 2026 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x3C	PWM_IMR2	31:24								
		23:16	CMPU7	CMPU6	CMPU5	CMPU4	CMPU3	CMPU2	CMPU1	CMPU0
		15:8	CMPM7	CMPM6	CMPM5	CMPM4	CMPM3	CMPM2	CMPM1	CMPM0
		7:0					UNRE	TXBUFE	ENDTX	WRDY
0x40	PWM_ISR2	31:24								
		23:16	CMPU7	CMPU6	CMPU5	CMPU4	CMPU3	CMPU2	CMPU1	CMPU0
		15:8	CMPM7	CMPM6	CMPM5	CMPM4	CMPM3	CMPM2	CMPM1	CMPM0
		7:0					UNRE	TXBUFE	ENDTX	WRDY
0x44	PWM_OOV	31:24								
		23:16						OOVL2	OOVL1	OOVL0
		15:8								
		7:0						OOVH2	OOVH1	OOVH0
0x48	PWM_OS	31:24								
		23:16						OSL2	OSL1	OSL0
		15:8								
		7:0						OSH2	OSH1	OSH0
0x4C	PWM_OSS	31:24								
		23:16						OSSL2	OSSL1	OSSL0
		15:8								
		7:0						OSSH2	OSSH1	OSSH0
0x50	PWM_OSC	31:24								
		23:16						OSCL2	OSCL1	OSCL0
		15:8								
		7:0						OSCH2	OSCH1	OSCH0
0x54	PWM_OSSUPD	31:24								
		23:16						OSSUPL2	OSSUPL1	OSSUPL0
		15:8								
		7:0						OSSUPH2	OSSUPH1	OSSUPH0
0x58	PWM_OSCUPD	31:24								
		23:16						OSCUPL2	OSCUPL1	OSCUPL0
		15:8								
		7:0						OSCUPH2	OSCUPH1	OSCUPH0
0x5C	PWM_FMR	31:24								
		23:16	FFIL[7:0]							
		15:8	FMOD[7:0]							
		7:0	FPOL[7:0]							
0x60	PWM_FSR	31:24								
		23:16								
		15:8	FS[7:0]							
		7:0	FIV[7:0]							
0x64	PWM_FCR	31:24								
		23:16								
		15:8								
		7:0	FCLR[7:0]							
0x68	PWM_FPV1	31:24								
		23:16						FPVL2	FPVL1	FPVL0
		15:8								
		7:0						FPVH2	FPVH1	FPVH0
0x6C	PWM_FPE	31:24								
		23:16	FPE2[7:0]							
		15:8	FPE1[7:0]							
		7:0	FPE0[7:0]							
0x70 ... 0x7B	Reserved									
0x7C	PWM_ELMR0	31:24								
		23:16								
		15:8								
		7:0	CSEL7	CSEL6	CSEL5	CSEL4	CSEL3	CSEL2	CSEL1	CSEL0

© 2026 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x80 ... 0x9F	Reserved									
0xA0	PWM_SSPR	31:24								SPRDM
		23:16	SPRD[23:16]							
		15:8	SPRD[15:8]							
		7:0	SPRD[7:0]							
0xA4	PWM_SSPUP	31:24								
		23:16	SPRDUP[23:16]							
		15:8	SPRDUP[15:8]							
		7:0	SPRDUP[7:0]							
0xA8 ... 0xAB	Reserved									
0xAC	PWM_DEBUG	31:24								
		23:16								
		15:8								
		7:0								OUTMODE
0xB0	PWM_SMMR	31:24								
		23:16								DOWN0
		15:8								
		7:0								GCEN0
0xB4 ... 0xBF	Reserved									
0xC0	PWM_FPV2	31:24								
		23:16						FPZL2	FPZL1	FPZL0
		15:8								
		7:0						FPZH2	FPZH1	FPZH0
0xC4 ... 0xE3	Reserved									
0xE4	PWM_WPCR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0	WPRG5	WPRG4	WPRG3	WPRG2	WPRG1	WPRG0	WPCMD[1:0]	
0xE8	PWM_WPSR	31:24	WPVSRC[15:8]							
		23:16	WPVSRC[7:0]							
		15:8			WPHWS5	WPHWS4	WPHWS3	WPHWS2	WPHWS1	WPHWS0
		7:0	WPVS		WPSWS5	WPSWS4	WPSWS3	WPSWS2	WPSWS1	WPSWS0
0xEC ... 0x012F	Reserved									
0x0130	PWM_CMPV0	31:24								CVM
		23:16	CV[23:16]							
		15:8	CV[15:8]							
		7:0	CV[7:0]							
0x0134	PWM_CMPVUPD0	31:24								CVMUPD
		23:16	CVUPD[23:16]							
		15:8	CVUPD[15:8]							
		7:0	CVUPD[7:0]							
0x0138	PWM_CMPM0	31:24								
		23:16	CUPRCNT[3:0]				CUPR[3:0]			
		15:8	CPRCNT[3:0]				CPR[3:0]			
		7:0	CTR[3:0]							CEN
0x013C	PWM_CMPMUPD0	31:24								
		23:16					CUPRUPD[3:0]			
		15:8					CPRUPD[3:0]			
		7:0	CTRUPD[3:0]							CENUPD

© 2026 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0140	PWM_CMPV1	31:24								CVM
		23:16	CV[23:16]							
		15:8	CV[15:8]							
		7:0	CV[7:0]							
0x0144	PWM_CMPVUPD1	31:24								CVMUPD
		23:16	CVUPD[23:16]							
		15:8	CVUPD[15:8]							
		7:0	CVUPD[7:0]							
0x0148	PWM_CMPM1	31:24								
		23:16	CUPRCNT[3:0]				CUPR[3:0]			
		15:8	CPRCNT[3:0]				CPR[3:0]			
		7:0	CTR[3:0]							CEN
0x014C	PWM_CMPMUPD1	31:24								
		23:16					CUPRUPD[3:0]			
		15:8					CPRUPD[3:0]			
		7:0	CTRUPD[3:0]							CENUPD
0x0150	PWM_CMPV2	31:24								CVM
		23:16	CV[23:16]							
		15:8	CV[15:8]							
		7:0	CV[7:0]							
0x0154	PWM_CMPVUPD2	31:24								CVMUPD
		23:16	CVUPD[23:16]							
		15:8	CVUPD[15:8]							
		7:0	CVUPD[7:0]							
0x0158	PWM_CMPM2	31:24								
		23:16	CUPRCNT[3:0]				CUPR[3:0]			
		15:8	CPRCNT[3:0]				CPR[3:0]			
		7:0	CTR[3:0]							CEN
0x015C	PWM_CMPMUPD2	31:24								
		23:16					CUPRUPD[3:0]			
		15:8					CPRUPD[3:0]			
		7:0	CTRUPD[3:0]							CENUPD
0x0160	PWM_CMPV3	31:24								CVM
		23:16	CV[23:16]							
		15:8	CV[15:8]							
		7:0	CV[7:0]							
0x0164	PWM_CMPVUPD3	31:24								CVMUPD
		23:16	CVUPD[23:16]							
		15:8	CVUPD[15:8]							
		7:0	CVUPD[7:0]							
0x0168	PWM_CMPM3	31:24								
		23:16	CUPRCNT[3:0]				CUPR[3:0]			
		15:8	CPRCNT[3:0]				CPR[3:0]			
		7:0	CTR[3:0]							CEN
0x016C	PWM_CMPMUPD3	31:24								
		23:16					CUPRUPD[3:0]			
		15:8					CPRUPD[3:0]			
		7:0	CTRUPD[3:0]							CENUPD
0x0170	PWM_CMPV4	31:24								CVM
		23:16	CV[23:16]							
		15:8	CV[15:8]							
		7:0	CV[7:0]							
0x0174	PWM_CMPVUPD4	31:24								CVMUPD
		23:16	CVUPD[23:16]							
		15:8	CVUPD[15:8]							
		7:0	CVUPD[7:0]							
0x0178	PWM_CMPM4	31:24								
		23:16	CUPRCNT[3:0]				CUPR[3:0]			
		15:8	CPRCNT[3:0]				CPR[3:0]			
		7:0	CTR[3:0]							CEN

© 2026 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x017C	PWM_CMPMUPD4	31:24								
		23:16					CUPRUPD[3:0]			
		15:8					CPRUPD[3:0]			
		7:0	CTRUPD[3:0]							CENUPD
0x0180	PWM_CMPV5	31:24								CVM
		23:16	CV[23:16]							
		15:8	CV[15:8]							
		7:0	CV[7:0]							
0x0184	PWM_CMPVUPD5	31:24								CVMUPD
		23:16	CVUPD[23:16]							
		15:8	CVUPD[15:8]							
		7:0	CVUPD[7:0]							
0x0188	PWM_CMPM5	31:24								
		23:16	CUPRCNT[3:0]				CUPR[3:0]			
		15:8	CPRCNT[3:0]				CPR[3:0]			
		7:0	CTR[3:0]							CEN
0x018C	PWM_CMPMUPD5	31:24								
		23:16					CUPRUPD[3:0]			
		15:8					CPRUPD[3:0]			
		7:0	CTRUPD[3:0]							CENUPD
0x0190	PWM_CMPV6	31:24								CVM
		23:16	CV[23:16]							
		15:8	CV[15:8]							
		7:0	CV[7:0]							
0x0194	PWM_CMPVUPD6	31:24								CVMUPD
		23:16	CVUPD[23:16]							
		15:8	CVUPD[15:8]							
		7:0	CVUPD[7:0]							
0x0198	PWM_CMPM6	31:24								
		23:16	CUPRCNT[3:0]				CUPR[3:0]			
		15:8	CPRCNT[3:0]				CPR[3:0]			
		7:0	CTR[3:0]							CEN
0x019C	PWM_CMPMUPD6	31:24								
		23:16					CUPRUPD[3:0]			
		15:8					CPRUPD[3:0]			
		7:0	CTRUPD[3:0]							CENUPD
0x01A0	PWM_CMPV7	31:24								CVM
		23:16	CV[23:16]							
		15:8	CV[15:8]							
		7:0	CV[7:0]							
0x01A4	PWM_CMPVUPD7	31:24								CVMUPD
		23:16	CVUPD[23:16]							
		15:8	CVUPD[15:8]							
		7:0	CVUPD[7:0]							
0x01A8	PWM_CMPM7	31:24								
		23:16	CUPRCNT[3:0]				CUPR[3:0]			
		15:8	CPRCNT[3:0]				CPR[3:0]			
		7:0	CTR[3:0]							CEN
0x01AC	PWM_CMPMUPD7	31:24								
		23:16					CUPRUPD[3:0]			
		15:8					CPRUPD[3:0]			
		7:0	CTRUPD[3:0]							CENUPD
0x01B0 ... 0x01FF	Reserved									
0x0200	PWM_CMR0	31:24								
		23:16					PPM	DTLI	DTHI	DTE
		15:8			TCTS	DPOLI	UPDS	CES	CPOL	CALG
		7:0					CPRE[3:0]			

© 2026 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0204	PWM_CDTY0	31:24								
		23:16	CDTY[23:16]							
		15:8	CDTY[15:8]							
		7:0	CDTY[7:0]							
0x0208	PWM_CDTYUPD0	31:24								
		23:16	CDTYUPD[23:16]							
		15:8	CDTYUPD[15:8]							
		7:0	CDTYUPD[7:0]							
0x020C	PWM_CPRD0	31:24								
		23:16	CPRD[23:16]							
		15:8	CPRD[15:8]							
		7:0	CPRD[7:0]							
0x0210	PWM_CPRDUPD0	31:24								
		23:16	CPRDUPD[23:16]							
		15:8	CPRDUPD[15:8]							
		7:0	CPRDUPD[7:0]							
0x0214	PWM_CCNT0	31:24								
		23:16	CNT[23:16]							
		15:8	CNT[15:8]							
		7:0	CNT[7:0]							
0x0218	PWM_DT0	31:24	DTL[15:8]							
		23:16	DTL[7:0]							
		15:8	DTH[15:8]							
		7:0	DTH[7:0]							
0x021C	PWM_DTUPD0	31:24	DTLUPD[15:8]							
		23:16	DTLUPD[7:0]							
		15:8	DTHUPD[15:8]							
		7:0	DTHUPD[7:0]							
0x0220	PWM_CMR1	31:24								
		23:16					PPM	DTLI	DTHI	DTE
		15:8			TCTS	DPOLI	UPDS	CES	CPOL	CALG
		7:0					CPRE[3:0]			
0x0224	PWM_CDTY1	31:24								
		23:16	CDTY[23:16]							
		15:8	CDTY[15:8]							
		7:0	CDTY[7:0]							
0x0228	PWM_CDTYUPD1	31:24								
		23:16	CDTYUPD[23:16]							
		15:8	CDTYUPD[15:8]							
		7:0	CDTYUPD[7:0]							
0x022C	PWM_CPRD1	31:24								
		23:16	CPRD[23:16]							
		15:8	CPRD[15:8]							
		7:0	CPRD[7:0]							
0x0230	PWM_CPRDUPD1	31:24								
		23:16	CPRDUPD[23:16]							
		15:8	CPRDUPD[15:8]							
		7:0	CPRDUPD[7:0]							
0x0234	PWM_CCNT1	31:24								
		23:16	CNT[23:16]							
		15:8	CNT[15:8]							
		7:0	CNT[7:0]							
0x0238	PWM_DT1	31:24	DTL[15:8]							
		23:16	DTL[7:0]							
		15:8	DTH[15:8]							
		7:0	DTH[7:0]							
0x023C	PWM_DTUPD1	31:24	DTLUPD[15:8]							
		23:16	DTLUPD[7:0]							
		15:8	DTHUPD[15:8]							
		7:0	DTHUPD[7:0]							

© 2026 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0240	PWM_CMR2	31:24								
		23:16					PPM	DTLI	DTHI	DTE
		15:8			TCTS	DPOLI	UPDS	CES	CPOL	CALG
		7:0					CPRE[3:0]			
0x0244	PWM_CDTY2	31:24								
		23:16	CDTY[23:16]							
		15:8	CDTY[15:8]							
		7:0	CDTY[7:0]							
0x0248	PWM_CDTYUPD2	31:24								
		23:16	CDTYUPD[23:16]							
		15:8	CDTYUPD[15:8]							
		7:0	CDTYUPD[7:0]							
0x024C	PWM_CPRD2	31:24								
		23:16	CPRD[23:16]							
		15:8	CPRD[15:8]							
		7:0	CPRD[7:0]							
0x0250	PWM_CPRDUPD2	31:24								
		23:16	CPRDUPD[23:16]							
		15:8	CPRDUPD[15:8]							
		7:0	CPRDUPD[7:0]							
0x0254	PWM_CCNT2	31:24								
		23:16	CNT[23:16]							
		15:8	CNT[15:8]							
		7:0	CNT[7:0]							
0x0258	PWM_DT2	31:24	DTL[15:8]							
		23:16	DTL[7:0]							
		15:8	DTH[15:8]							
		7:0	DTH[7:0]							
0x025C	PWM_DTUPD2	31:24	DTLUPD[15:8]							
		23:16	DTLUPD[7:0]							
		15:8	DTHUPD[15:8]							
		7:0	DTHUPD[7:0]							
0x0260 ... 0x03FF	Reserved									
0x0400	PWM_CMUPD0	31:24								
		23:16								
		15:8			CPOLINVUP				CPOLUP	
		7:0								
0x0404 ... 0x041F	Reserved									
0x0420	PWM_CMUPD1	31:24								
		23:16								
		15:8			CPOLINVUP				CPOLUP	
		7:0								
0x0424 ... 0x042B	Reserved									
0x042C	PWM_ETRG1	31:24	RFEN	TRGSRC	TRGFILT	TRGEDGE			TRGMODE[1:0]	
		23:16	MAXCNT[23:16]							
		15:8	MAXCNT[15:8]							
		7:0	MAXCNT[7:0]							
0x0430	PWM_LEBR1	31:24								
		23:16					PWMHREN	PWMHFEN	PWMLREN	PWMLFEN
		15:8								
		7:0		LEBDELAY[6:0]						
0x0434 ... 0x043F	Reserved									

© 2026 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0440	PWM_CMUPD2	31:24								
		23:16								
		15:8			CPOLINVUP				CPOLUP	
		7:0								
0x0444 ... 0x044B	Reserved									
0x044C	PWM_ETRG2	31:24	RFEN	TRGSRC	TRGFILT	TRGEDGE			TRGMODE[1:0]	
		23:16	MAXCNT[23:16]							
		15:8	MAXCNT[15:8]							
		7:0	MAXCNT[7:0]							
0x0450	PWM_LEBR2	31:24								
		23:16					PWMHREN	PWMHFEN	PWMLREN	PWMLFEN
		15:8								
		7:0		LEBDELAY[6:0]						

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.1. PWM Clock Register

Name: PWM_CLK
Offset: 0x00
Reset: 0x00000000
Property: Read/Write

This register can only be written if bits WPSWS0 and WPHWS0 are cleared in the PWM Write Protection Status Register.

Bit	31	30	29	28	27	26	25	24
					PREB[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DIVB[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
					PREA[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DIVA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 27:24 – PREB[3:0] CLKB Source Clock Selection

Value	Name	Description
0	CLK	Peripheral clock
1	CLK_DIV2	Peripheral clock/2
2	CLK_DIV4	Peripheral clock/4
3	CLK_DIV8	Peripheral clock/8
4	CLK_DIV16	Peripheral clock/16
5	CLK_DIV32	Peripheral clock/32
6	CLK_DIV64	Peripheral clock/64
7	CLK_DIV128	Peripheral clock/128
8	CLK_DIV256	Peripheral clock/256
9	CLK_DIV512	Peripheral clock/512
10	CLK_DIV1024	Peripheral clock/1024
Other	–	Reserved

Bits 23:16 – DIVB[7:0] CLKB Divide Factor

Value	Name	Description
0	CLKB_POFF	CLKB clock is turned off
1	PREB	CLKB clock is clock selected by PREB
2-255	PREB_DIV	CLKB clock is clock selected by PREB divided by DIVB factor

Bits 11:8 – PREA[3:0] CLKA Source Clock Selection

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Name	Description
0	CLK	Peripheral clock
1	CLK_DIV2	Peripheral clock/2
2	CLK_DIV4	Peripheral clock/4
3	CLK_DIV8	Peripheral clock/8
4	CLK_DIV16	Peripheral clock/16
5	CLK_DIV32	Peripheral clock/32
6	CLK_DIV64	Peripheral clock/64
7	CLK_DIV128	Peripheral clock/128
8	CLK_DIV256	Peripheral clock/256
9	CLK_DIV512	Peripheral clock/512
10	CLK_DIV1024	Peripheral clock/1024
Other	–	Reserved

Bits 7:0 – DIVA[7:0] CLKA Divide Factor

Value	Name	Description
0	CLKA_POFF	CLKA clock is turned off
1	PREA	CLKA clock is clock selected by PREA
2-255	PREA_DIV	CLKA clock is clock selected by PREA divided by DIVA factor

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.2. PWM Enable Register

Name: PWM_ENA
Offset: 0x04
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						CHID2	CHID1	CHID0
Access						W	W	W
Reset						–	–	–

Bits 0, 1, 2 – CHIDx Channel ID

Value	Description
0	No effect.
1	Enable PWM output for channel x.

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.3. PWM Disable Register

Name: PWM_DIS
Offset: 0x08
Reset: –
Property: Write-only

This register can only be written if bits WPSWS1 and WPHWS1 are cleared in the PWM Write Protection Status Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						CHID2	CHID1	CHID0
Access						W	W	W
Reset						–	–	–

Bits 0, 1, 2 – CHIDx Channel ID

Value	Description
0	No effect.
1	Disable PWM output for channel x.

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.4. PWM Status Register

Name: PWM_SR
Offset: 0x0C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						CHID2	CHID1	CHID0
Access						R	R	R
Reset						0	0	0

Bits 0, 1, 2 – CHIDx Channel ID

Value	Description
0	PWM output for channel x is disabled.
1	PWM output for channel x is enabled.

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.5. PWM Interrupt Enable Register 1

Name: PWM_IER1
Offset: 0x10
Reset: –
Property: Write-only

This register can only be written if bits WPSWS1 and WPHWS1 are cleared in the PWM Write Protection Status Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						FCHID2	FCHID1	FCHID0
Access						W	W	W
Reset						–	–	–

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						CHID2	CHID1	CHID0
Access						W	W	W
Reset						–	–	–

Bits 16, 17, 18 – FCHIDx Fault Protection Trigger on Channel x Interrupt Enable

Bits 0, 1, 2 – CHIDx Counter Event on Channel x Interrupt Enable

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.6. PWM Interrupt Disable Register 1

Name: PWM_IDR1
Offset: 0x14
Reset: –
Property: Write-only

This register can only be written if bits WPSWS1 and WPHWS1 are cleared in the PWM Write Protection Status Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						FCHID2	FCHID1	FCHID0
Access						W	W	W
Reset						–	–	–

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						CHID2	CHID1	CHID0
Access						W	W	W
Reset						–	–	–

Bits 16, 17, 18 – FCHIDx Fault Protection Trigger on Channel x Interrupt Disable

Bits 0, 1, 2 – CHIDx Counter Event on Channel x Interrupt Disable

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.7. PWM Interrupt Mask Register 1

Name: PWM_IMR1
Offset: 0x18
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						FCHID2	FCHID1	FCHID0
Access						R	R	R
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						CHID2	CHID1	CHID0
Access						R	R	R
Reset						0	0	0

Bits 16, 17, 18 – FCHIDx Fault Protection Trigger on Channel x Interrupt Mask

Bits 0, 1, 2 – CHIDx Counter Event on Channel x Interrupt Mask

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.8. PWM Interrupt Status Register 1

Name: PWM_ISR1
Offset: 0x1C
Reset: 0x00000000
Property: Read-only

Note: Reading PWM_ISR1 automatically clears CHIDx and FCHIDx flags.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						FCHID2	FCHID1	FCHID0
Access						R	R	R
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						CHID2	CHID1	CHID0
Access						R	R	R
Reset						0	0	0

Bits 16, 17, 18 – FCHIDx Fault Protection Trigger on Channel x

Value	Description
0	No new trigger of the fault protection since the last read of PWM_ISR1.
1	At least one trigger of the fault protection since the last read of PWM_ISR1.

Bits 0, 1, 2 – CHIDx Counter Event on Channel x

Value	Description
0	No new counter event has occurred since the last read of PWM_ISR1.
1	At least one counter event has occurred since the last read of PWM_ISR1.

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.9. PWM Sync Channels Mode Register

Name: PWM_SCM
Offset: 0x20
Reset: 0x00000000
Property: Read/Write

This register can only be written if bits WPSWS2 and WPHWS2 are cleared in the PWM Write Protection Status Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	PTRCS[2:0]			PTRM			UPDM[1:0]	
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	0	0	0			0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						SYNC2	SYNC1	SYNC0
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 23:21 – PTRCS[2:0] Peripheral DMA Controller Transfer Request Comparison Selection
Selection of the comparison used to set the flag WRDY and the corresponding Peripheral DMA Controller transfer request.

Bit 20 – PTRM Peripheral DMA Controller Transfer Request Mode

UPDM	PTRM	WRDY Flag and Peripheral DMA Controller Transfer Request
0	x	The WRDY flag in PWM Interrupt Status Register 2 and the PDC transfer request are never set to '1'.
1	x	The WRDY flag in PWM Interrupt Status Register 2 is set to '1' as soon as the update period is elapsed, the Peripheral DMA Controller transfer request is never set to '1'.
2	0	The WRDY flag in PWM Interrupt Status Register 2 and the PDC transfer request are set to '1' as soon as the update period is elapsed.
	1	The WRDY flag in PWM Interrupt Status Register 2 and the PDC transfer request are set to '1' as soon as the selected comparison matches.

Bits 17:16 – UPDM[1:0] Synchronous Channels Update Mode

Value	Name	Description
0	MODE0	Manual write of double buffer registers and manual update of synchronous channels 1[(1)].
1	MODE1	Manual write of double buffer registers and automatic update of synchronous channels 2[(2)].
2	MODE2	The WRDY flag in PWM Interrupt Status Register 2 and the PDC transfer request are set to '1' as soon as the update period is elapsed.

© 2026 Microchip Technology Inc. and its subsidiaries

Notes:

1. The update occurs at the beginning of the next PWM period, when the UPDULOCK bit in PWM Sync Channels Update Control Register is set.
2. The update occurs when the Update Period is elapsed.

Bits 0, 1, 2 – SYNCx Synchronous Channel x

Value	Description
0	Channel x is not a synchronous channel.
1	Channel x is a synchronous channel.

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.10. PWM Sync Channels Update Control Register

Name: PWM_SCUC
Offset: 0x28
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								UPDULOCK
Access								R/W
Reset								0

Bit 0 – UPDULOCK Synchronous Channels Update Unlock
This bit is automatically reset when the update is done.

Value	Description
0	No effect
1	If the UPDM field is set to '0' in PWM Sync Channels Mode Register, writing the UPDULOCK bit to '1' triggers the update of the period value, the duty-cycle and the dead-time values of synchronous channels at the beginning of the next PWM period. If the field UPDM is set to '1' or '2', writing the UPDULOCK bit to '1' triggers only the update of the period value and of the dead-time values of synchronous channels.

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.11. PWM Sync Channels Update Period Register

Name: PWM_SCUP
Offset: 0x2C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	UPRCNT[3:0]				UPR[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:4 – UPRCNT[3:0] Update Period Counter
Reports the value of the update period counter.

Bits 3:0 – UPR[3:0] Update Period
Defines the time between each update of the synchronous channels if automatic trigger of the update is activated (UPDM = 1 or UPDM = 2 in PWM Sync Channels Mode Register). This time is equal to UPR+1 periods of the synchronous channels.

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.12. PWM Sync Channels Update Period Update Register

Name: PWM_SCUPUPD
Offset: 0x30
Reset: –
Property: Write-only

This register acts as a double buffer for the UPR value. This prevents an unexpected automatic trigger of the update of synchronous channels.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					UPRUPD[3:0]			
Access					W	W	W	W
Reset					–	–	–	–

Bits 3:0 – UPRUPD[3:0] Update Period Update
Defines the time between each update of the synchronous channels if automatic trigger of the update is activated (UPDM = 1 or UPDM = 2 in PWM Sync Channels Mode Register). This time is equal to UPR+1 periods of the synchronous channels.

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.13. PWM Interrupt Enable Register 2

Name: PWM_IER2
Offset: 0x34
Reset: –
Property: Write-only

This register can only be written if bits WPSWS1 and WPHWS1 are cleared in the PWM Write Protection Status Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	CMPU7	CMPU6	CMPU5	CMPU4	CMPU3	CMPU2	CMPU1	CMPU0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	CMPM7	CMPM6	CMPM5	CMPM4	CMPM3	CMPM2	CMPM1	CMPM0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
					UNRE	TXBUFE	ENDTX	WRDY
Access					W	W	W	W
Reset					–	–	–	–

Bits 16, 17, 18, 19, 20, 21, 22, 23 – CMPUx Comparison x Update Interrupt Enable

Bits 8, 9, 10, 11, 12, 13, 14, 15 – CMPMx Comparison x Match Interrupt Enable

Bit 3 – UNRE Synchronous Channels Update Underrun Error Interrupt Enable

Bit 2 – TXBUFE PDC TX Buffer Empty Interrupt Enable

Bit 1 – ENDTX PDC End of TX Buffer Interrupt Enable

Bit 0 – WRDY Write Ready for Synchronous Channels Update Interrupt Enable

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.14. PWM Interrupt Disable Register 2

Name: PWM_IDR2
Offset: 0x38
Reset: –
Property: Write-only

This register can only be written if bits WPSWS1 and WPHWS1 are cleared in the PWM Write Protection Status Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	CMPU7	CMPU6	CMPU5	CMPU4	CMPU3	CMPU2	CMPU1	CMPU0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	CMPM7	CMPM6	CMPM5	CMPM4	CMPM3	CMPM2	CMPM1	CMPM0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
					UNRE	TXBUFE	ENDTX	WRDY
Access					W	W	W	W
Reset					–	–	–	–

Bits 16, 17, 18, 19, 20, 21, 22, 23 – CMPUx Comparison x Update Interrupt Disable

Bits 8, 9, 10, 11, 12, 13, 14, 15 – CMPMx Comparison x Match Interrupt Disable

Bit 3 – UNRE Synchronous Channels Update Underrun Error Interrupt Disable

Bit 2 – TXBUFE PDC TX Buffer Empty Interrupt Disable

Bit 1 – ENDTX PDC End of TX Buffer Interrupt Disable

Bit 0 – WRDY Write Ready for Synchronous Channels Update Interrupt Disable

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.15. PWM Interrupt Mask Register 2

Name: PWM_IMR2
Offset: 0x3C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	CMPU7	CMPU6	CMPU5	CMPU4	CMPU3	CMPU2	CMPU1	CMPU0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CMPM7	CMPM6	CMPM5	CMPM4	CMPM3	CMPM2	CMPM1	CMPM0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
					UNRE	TXBUFE	ENDTX	WRDY
Access					R	R	R	R
Reset					0	0	0	0

Bits 16, 17, 18, 19, 20, 21, 22, 23 – CMPUx Comparison x Update Interrupt Mask

Bits 8, 9, 10, 11, 12, 13, 14, 15 – CMPMx Comparison x Match Interrupt Mask

Bit 3 – UNRE Synchronous Channels Update Underrun Error Interrupt Mask

Bit 2 – TXBUFE PDC TX Buffer Empty Interrupt Mask

Bit 1 – ENDTX PDC End of TX Buffer Interrupt Mask

Bit 0 – WRDY Write Ready for Synchronous Channels Update Interrupt Mask

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.16. PWM Interrupt Status Register 2

Name: PWM_ISR2
Offset: 0x40
Reset: 0x00000006
Property: Read-only

Reading PWM_ISR2 automatically clears flags WRDY, UNRE and CMPSx.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	CMPU7	CMPU6	CMPU5	CMPU4	CMPU3	CMPU2	CMPU1	CMPU0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CMPM7	CMPM6	CMPM5	CMPM4	CMPM3	CMPM2	CMPM1	CMPM0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
					UNRE	TXBUFE	ENDTX	WRDY
Access					R	R	R	R
Reset					0	1	1	0

Bits 16, 17, 18, 19, 20, 21, 22, 23 – CMPUx Comparison x Update

Value	Description
0	The comparison x has not been updated since the last read of the PWM_ISR2 register.
1	The comparison x has been updated at least one time since the last read of the PWM_ISR2 register.

Bits 8, 9, 10, 11, 12, 13, 14, 15 – CMPMx Comparison x Match

Value	Description
0	The comparison x has not matched since the last read of the PWM_ISR2 register.
1	The comparison x has matched at least one time since the last read of the PWM_ISR2 register.

Bit 3 – UNRE Synchronous Channels Update Underrun Error

Value	Description
0	No Synchronous Channels Update Underrun has occurred since the last read of the PWM_ISR2 register.
1	At least one Synchronous Channels Update Underrun has occurred since the last read of the PWM_ISR2 register.

Bit 2 – TXBUFE PDC TX Buffer Empty

Value	Description
0	PWM_TCR or PWM_TCNR has a value other than 0.
1	Both PWM_TCR and PWM_TCNR have a value other than 0.

Bit 1 – ENDTX PDC End of TX Buffer

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	The Transmit Counter register has not reached 0 since the last write of the PDC.
1	The Transmit Counter register has reached 0 since the last write of the PDC.

Bit 0 – WRDY Write Ready for Synchronous Channels Update

Value	Description
0	New duty-cycle and dead-time values for the synchronous channels cannot be written.
1	New duty-cycle and dead-time values for the synchronous channels can be written.

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.17. PWM Output Override Value Register

Name: PWM_OOV
Offset: 0x44
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						OOVL2	OOVL1	OOVL0
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						OOVH2	OOVH1	OOVH0
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 16, 17, 18 – OOVLx Output Override Value for PWML output of the channel x

Value	Description
0	Override value is 0 for PWML output of channel x.
1	Override value is 1 for PWML output of channel x.

Bits 0, 1, 2 – OOVHx Output Override Value for PWMH output of the channel x

Value	Description
0	Override value is 0 for PWMH output of channel x.
1	Override value is 1 for PWMH output of channel x.

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.18. PWM Output Selection Register

Name: PWM_OS
Offset: 0x48
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						OSL2	OSL1	OSL0
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						OSH2	OSH1	OSH0
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 16, 17, 18 – OSLx Output Selection for PWML output of the channel x

Value	Description
0	Dead-time generator output DTOLx selected as PWML output of channel x.
1	Output override value OOVLx selected as PWML output of channel x.

Bits 0, 1, 2 – OSHx Output Selection for PWMH output of the channel x

Value	Description
0	Dead-time generator output DTOHx selected as PWMH output of channel x.
1	Output override value OOVHx selected as PWMH output of channel x.

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.19. PWM Output Selection Set Register

Name: PWM_OSS
Offset: 0x4C
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						OSSL2	OSSL1	OSSL0
Access						W	W	W
Reset						–	–	–

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						OSSH2	OSSH1	OSSH0
Access						W	W	W
Reset						–	–	–

Bits 16, 17, 18 – OSSLx Output Selection Set for PWML output of the channel x

Value	Description
0	No effect.
1	Output override value OOVLx selected as PWML output of channel x.

Bits 0, 1, 2 – OSSHx Output Selection Set for PWMH output of the channel x

Value	Description
0	No effect.
1	Output override value OOVHx selected as PWMH output of channel x.

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.20. PWM Output Selection Clear Register

Name: PWM_OSC
Offset: 0x50
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						OSCL2	OSCL1	OSCL0
Access						W	W	W
Reset						–	–	–

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						OSCH2	OSCH1	OSCH0
Access						W	W	W
Reset						–	–	–

Bits 16, 17, 18 – OSCLx Output Selection Clear for PWML output of the channel x

Value	Description
0	No effect.
1	Dead-time generator output DTOLx selected as PWML output of channel x.

Bits 0, 1, 2 – OSCHx Output Selection Clear for PWMH output of the channel x

Value	Description
0	No effect.
1	Dead-time generator output DTOHx selected as PWMH output of channel x.

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.21. PWM Output Selection Set Update Register

Name: PWM_OSSUPD
Offset: 0x54
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						OSSUPL2	OSSUPL1	OSSUPL0
Access						W	W	W
Reset						–	–	–

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						OSSUPH2	OSSUPH1	OSSUPH0
Access						W	W	W
Reset						–	–	–

Bits 16, 17, 18 – OSSUPLx Output Selection Set for PWML output of the channel x

Value	Description
0	No effect.
1	Output override value OOVLx selected as PWML output of channel x at the beginning of the next channel x PWM period.

Bits 0, 1, 2 – OSSUPHx Output Selection Set for PWMH output of the channel x

Value	Description
0	No effect.
1	Output override value OOVHx selected as PWMH output of channel x at the beginning of the next channel x PWM period.

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.22. PWM Output Selection Clear Update Register

Name: PWM_OSCUPD
Offset: 0x58
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						OSCUPL2	OSCUPL1	OSCUPL0
Access						W	W	W
Reset						–	–	–

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						OSCUPH2	OSCUPH1	OSCUPH0
Access						W	W	W
Reset						–	–	–

Bits 16, 17, 18 – OSCUPLx Output Selection Clear for PWML output of the channel x

Value	Description
0	No effect.
1	Dead-time generator output DTOLx selected as PWML output of channel x at the beginning of the next channel x PWM period.

Bits 0, 1, 2 – OSCUPHx Output Selection Clear for PWMH output of the channel x

Value	Description
0	No effect.
1	Dead-time generator output DTOHx selected as PWMH output of channel x at the beginning of the next channel x PWM period.

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.23. PWM Fault Mode Register

Name: PWM_FMR
Offset: 0x5C
Reset: 0x00000000
Property: Read/Write

This register can only be written if bits WPSWS5 and WPHWS5 are cleared in the PWM Write Protection Status Register.

See Fault Inputs for details on fault generation.

CAUTION To prevent an unexpected activation of the status flag FSy in the PWM Fault Status Register, the bit FMODy can be set to '1' only if the FPOLy bit has been previously configured to its final value.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	FFIL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	FMOD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FPOL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – FFIL[7:0] Fault Filtering
For each bit y of FFIL, where y is the fault input number:
0: The fault input y is not filtered.
1: The fault input y is filtered.

Bits 15:8 – FMOD[7:0] Fault Activation Mode
For each bit y of FMOD, where y is the fault input number:
0: The fault y is active until the fault condition is removed at the peripheral(1) level.
1: The fault y stays active until the fault condition is removed at the peripheral level(1)AND until it is cleared in the PWM Fault Clear Register.
Note:
1. The peripheral generating the fault.

Bits 7:0 – FPOL[7:0] Fault Polarity
For each bit y of FPOL, where y is the fault input number:
0: The fault y becomes active when the fault input y is at 0.
1: The fault y becomes active when the fault input y is at 1.

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.24. PWM Fault Status Register

Name: PWM_FSR
Offset: 0x60
Reset: 0x00000000
Property: Read-only

Refer to Fault Inputs for details on fault generation.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	FS[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FIV[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:8 – FS[7:0] Fault Status
For each bit y of FS, where y is the fault input number:
0: The fault y is not currently active.
1: The fault y is currently active.

Bits 7:0 – FIV[7:0] Fault Input Value
For each bit y of FIV, where y is the fault input number:
0: The current sampled value of the fault input y is 0 (after filtering if enabled).
1: The current sampled value of the fault input y is 1 (after filtering if enabled).

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.25. PWM Fault Clear Register

Name: PWM_FCR
Offset: 0x64
Reset: –
Property: Write-only

See Fault Inputs for details on fault generation.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	FCLR[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 7:0 – FCLR[7:0] Fault Clear
For each bit y of FCLR, where y is the fault input number:
0: No effect.
1: If bit y of FMOD field is set to '1' and if the fault input y is not at the level defined by the bit y of FPOL field, the fault y is cleared and becomes inactive (FMOD and FPOL fields belong to PWM Fault Mode Register), else writing this bit to '1' has no effect.

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.26. PWM Fault Protection Value Register 1

Name: PWM_FPV1
Offset: 0x68
Reset: 0x00000000
Property: Read/Write

This register can only be written if bits WPSWS5 and WPHWS5 are cleared in the PWM Write Protection Status Register.

See Fault Inputs for details on fault generation.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						FPVL2	FPVL1	FPVL0
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						FPVH2	FPVH1	FPVH0
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 16, 17, 18 – FPVLx Fault Protection Value for PWML output on channel x
This bit is taken into account only if the bit FPZLx is set to '0' in PWM Fault Protection Value Register 2.

Value	Description
0	PWML output of channel x is forced to '0' when fault occurs.
1	PWML output of channel x is forced to '1' when fault occurs.

Bits 0, 1, 2 – FPVHx Fault Protection Value for PWMH output on channel x
This bit is taken into account only if the bit FPZHx is set to '0' in PWM Fault Protection Value Register 2.

Value	Description
0	PWMH output of channel x is forced to '0' when fault occurs.
1	PWMH output of channel x is forced to '1' when fault occurs.

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.27. PWM Fault Protection Enable Register

Name: PWM_FPE
Offset: 0x6C
Reset: 0x00000000
Property: Read/Write

This register can only be written if bits WPSWS5 and WPHWS5 are cleared in the PWM Write Protection Status Register.

Only the first 8 bits (number of fault input pins) of fields FPEx are significant.

Refer to Fault Inputs for details on fault generation.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	FPE2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	FPE1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FPE0[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0:7, 8:15, 16:23 – FPEx Fault Protection Enable for channel x
For each bit y of FPEx, where y is the fault input number:
0: Fault y is not used for the fault protection of channel x.
1: Fault y is used for the fault protection of channel x.

CAUTION To prevent an unexpected activation of the fault protection, the bit y of FPEx field can be set to '1' only if the corresponding FPOL field has been previously configured to its final value in PWM Fault Mode Register.

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.28. PWM Event Line 0 Mode Register

Name: PWM_ELMR0
Offset: 0x7C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	CSEL7	CSEL6	CSEL5	CSEL4	CSEL3	CSEL2	CSEL1	CSEL0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7 – CSELy Comparison y Selection

Value	Description
0	A pulse is not generated on the event line x when the comparison y matches.
1	A pulse is generated on the event line x when the comparison y match.

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.29. PWM Spread Spectrum Register

Name: PWM_SSPR
Offset: 0xA0
Reset: 0x00000000
Property: Read/Write

This register can only be written if bits WPSWS3 and WPHWS3 are cleared in the PWM Write Protection Status Register.

Only the first 24 bits (channel counter size) are significant.

Bit	31	30	29	28	27	26	25	24
								SPRDM
Access								R/W
Reset								0

Bit	23	22	21	20	19	18	17	16
	SPRD[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SPRD[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SPRD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 24 – SPRDM Spread Spectrum Counter Mode

Value	Description
0	Triangular mode. The spread spectrum counter starts to count from -SPRD when the channel 0 is enabled and counts upwards at each PWM period. When it reaches +SPRD, it restarts to count from -SPRD again.
1	Random mode. The spread spectrum counter is loaded with a new random value at each PWM period. This random value is uniformly distributed and is between -SPRD and +SPRD.

Bits 23:0 – SPRD[23:0] Spread Spectrum Limit Value
The spread spectrum limit value defines the range for the spread spectrum counter. It is introduced in order to achieve constant varying PWM period for the output waveform.

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.30. PWM Spread Spectrum Update Register

Name: PWM_SSPUP
Offset: 0xA4
Reset: –
Property: Write-only

This register can only be written if bits WPSWS3 and WPHWS3 are cleared in the PWM Write Protection Status Register.

This register acts as a double buffer for the SPRD value. This prevents an unexpected waveform when modifying the spread spectrum limit value.

Only the first 24 bits (channel counter size) are significant.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	SPRDUP[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	SPRDUP[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	SPRDUP[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 23:0 – SPRDUP[23:0] Spread Spectrum Limit Value Update
The spread spectrum limit value defines the range for the spread spectrum counter. It is introduced in order to achieve constant varying period for the output waveform.

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.31. PWM Debug Register

Name: PWM_DEBUG
Offset: 0xAC
Reset: 0x00000000
Property: Read/Write

This register can only be written if bits WPSWS2 and WPHWS2 are cleared in the PWM Write Protection Status Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								OUTMODE
Access								R/W
Reset								0

Bit 0 – OUTMODE PWM Output Mode when System is in Debug Mode
0 (NO_EFFECT): Keeps the PWM outputs running when the processor reports a debug operating mode.
1 (STUCK_AT): Forces the PWM outputs with the values configured in PWM_FPVx as soon as the processor reports a debug operating mode. See PWM_FPV1 and PWM_FPV2.

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.32. PWM Stepper Motor Mode Register

Name: PWM_SMMR
Offset: 0xB0
Reset: 0x00000000
Property: Read/Write

This register can only be written if bits WPSWS2 and WPHWS2 are cleared in the PWM Write Protection Status Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								R/W
Reset								0

Bit 16 – DOWNx Down Count

Value	Description
0	Up counter.
1	Down counter.

Bit 0 – GCENx Gray Count Enable

Value	Description
0	Disable Gray count generation on PWML[2*x], PWMH[2*x], PWML[2*x +1], PWMH[2*x +1]
1	Enable Gray count generation on PWML[2*x], PWMH[2*x], PWML[2*x +1], PWMH[2*x +1].

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.33. PWM Fault Protection Value Register 2

Name: PWM_FPV2
Offset: 0xC0
Reset: 0x00070007
Property: Read/Write

This register can only be written if bits WPSWS5 and WPHWS5 are cleared in the PWM Write Protection Status Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						FPZL2	FPZL1	FPZL0
Access						R/W	R/W	R/W
Reset						1	1	1

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						FPZH2	FPZH1	FPZH0
Access						R/W	R/W	R/W
Reset						1	1	1

Bits 16, 17, 18 – FPZLx Fault Protection to Hi-Z for PWML output on channel x

Value	Description
0	When fault occurs, PWML output of channel x is forced to value defined by the bit FPVLx in PWM Fault Protection Value Register 1.
1	When fault occurs, PWML output of channel x is forced to high-impedance state.

Bits 0, 1, 2 – FPZHx Fault Protection to Hi-Z for PWMH output on channel x

Value	Description
0	When fault occurs, PWMH output of channel x is forced to value defined by the bit FPVHx in PWM Fault Protection Value Register 1.
1	When fault occurs, PWMH output of channel x is forced to high-impedance state.

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.34. PWM Write Protection Control Register

Name: PWM_WPCR
Offset: 0xE4
Reset: –
Property: Write-only

See Register Write Protection for the list of registers that can be write-protected.

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	WPRG5	WPRG4	WPRG3	WPRG2	WPRG1	WPRG0	WPCMD[1:0]	
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x50574D	PASSWD	Writing any other value in this field aborts the write operation of the WPCMD field. Always reads as 0

Bits 2, 3, 4, 5, 6, 7 – WPRGx Write Protection Register Group x

Value	Description
0	The WPCMD command has no effect on the register group x.
1	The WPCMD command is applied to the register group x.

Bits 1:0 – WPCMD[1:0] Write Protection Command
This command is performed only if the WPKEY corresponds to 0x50574D (“PWM” in ASCII).

Value	Name	Description
0	DISABLE_SW_PROT	Disables the software write protection of the register groups of which the bit WPRGx is at ‘1’.
1	ENABLE_SW_PROT	Enables the software write protection of the register groups of which the bit WPRGx is at ‘1’.
2	ENABLE_HW_PROT	Enables the hardware write protection of the register groups of which the bit WPRGx is at ‘1’. Only a hardware reset of the PWM controller can disable the hardware write protection. Moreover, to meet security requirements, the PIO lines associated with the PWM can not be configured through the PIO interface.

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.35. PWM Write Protection Status Register

Name: PWM_WPSR
Offset: 0xE8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	WPVSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
			WPHWS5	WPHWS4	WPHWS3	WPHWS2	WPHWS1	WPHWS0
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	WPVS		WPSWS5	WPSWS4	WPSWS3	WPSWS2	WPSWS1	WPSWS0
Access	R		R	R	R	R	R	R
Reset	0		0	0	0	0	0	0

Bits 31:16 – WPVSRC[15:0] Write Protect Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bits 8, 9, 10, 11, 12, 13 – WPHWSx Write Protect HW Status

Value	Description
0	The HW write protection x of the register group x is disabled.
1	The HW write protection x of the register group x is enabled.

Bit 7 – WPVS Write Protect Violation Status

Value	Description
0	No write protection violation has occurred since the last read of PWM_WPSR.
1	At least one write protection violation has occurred since the last read of PWM_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

Bits 0, 1, 2, 3, 4, 5 – WPSWSx Write Protect SW Status

Value	Description
0	The SW write protection x of the register group x is disabled.
1	The SW write protection x of the register group x is enabled.

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.36. PWM Comparison x Value Register

Name: PWM_CMPVx
Offset: 0x0130 + x*0x04 [x=0..7]
Reset: 0x00000000
Property: Read/Write

Only the first 24 bits (channel counter size) of field CV are significant.

Bit	31	30	29	28	27	26	25	24
								CVM
Access								R/W
Reset								0

Bit	23	22	21	20	19	18	17	16
	CV[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CV[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CV[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 24 – CVM Comparison x Value Mode

Value	Description
0	The comparison x between the counter of the channel 0 and the comparison x value is performed when this counter is incrementing.
1	The comparison x between the counter of the channel 0 and the comparison x value is performed when this counter is decrementing. Note: This bit is not relevant if the counter of the channel 0 is left-aligned (CALG = 0 in PWM Channel Mode Register)

Bits 23:0 – CV[23:0] Comparison x Value
Define the comparison x value to be compared with the counter of the channel 0.

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.37. PWM Comparison x Value Update Register

Name: PWM_CMPVUPDx
Offset: 0x0134 + x*0x04 [x=0..7]
Reset: –
Property: Write-only

This register acts as a double buffer for the CV and CVM values. This prevents an unexpected comparison x match.

Only the first 24 bits (channel counter size) of field CVUPD are significant.

CAUTION The write of the register PWM_CMPVUPDx must be followed by a write of the register PWM_CMPMUPDx.

Bit	31	30	29	28	27	26	25	24
								CVMUPD
Access								W
Reset								–

Bit	23	22	21	20	19	18	17	16
	CVUPD[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	CVUPD[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	CVUPD[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 24 – CVMUPD Comparison x Value Mode Update
Note: This bit is not relevant if the counter of the channel 0 is left-aligned (CALG = 0 in PWM Channel Mode Register)

Value	Description
0	The comparison x between the counter of the channel 0 and the comparison x value is performed when this counter is incrementing.
1	The comparison x between the counter of the channel 0 and the comparison x value is performed when this counter is decrementing.

Bits 23:0 – CVUPD[23:0] Comparison x Value Update
Defines the comparison x value to be compared with the counter of the channel 0.

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.38. PWM Comparison x Mode Register

Name: PWM_CMPMx
Offset: 0x0138 + x*0x04 [x=0..7]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	CUPRCNT[3:0]				CUPR[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CPRCNT[3:0]				CPR[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CTR[3:0]							CEN
Access	R/W	R/W	R/W	R/W				R/W
Reset	0	0	0	0				0

Bits 23:20 – CUPRCNT[3:0] Comparison x Update Period Counter
Reports the value of the comparison x update period counter.
Note: The field CUPRCNT is read-only

Bits 19:16 – CUPR[3:0] Comparison x Update Period
Defines the time between each update of the comparison x mode and the comparison x value. This time is equal to CUPR+1 periods of the channel 0 counter.

Bits 15:12 – CPRCNT[3:0] Comparison x Period Counter
Reports the value of the comparison x period counter.
Note: The field CPRCNT is read-only

Bits 11:8 – CPR[3:0] Comparison x Period
CPR defines the maximum value of the comparison x period counter (CPRCNT). The comparison x value is performed periodically once every CPR+1 periods of the channel 0 counter.

Bits 7:4 – CTR[3:0] Comparison x Trigger
The comparison x is performed when the value of the comparison x period counter (CPRCNT) reaches the value defined by CTR.

Bit 0 – CEN Comparison x Enable

Value	Description
0	The comparison x is disabled and can not match.
1	The comparison x is enabled and can match.

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.39. PWM Comparison x Mode Update Register

Name: PWM_CMPMUPDx
Offset: 0x013C + x*0x04 [x=0..7]
Reset: –
Property: Write-only

This register acts as a double buffer for the CEN, CTR, CPR and CUPR values. This prevents an unexpected comparison x match.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					CUPRUPD[3:0]			
Access					W	W	W	W
Reset					–	–	–	–

Bit	15	14	13	12	11	10	9	8
					CPRUPD[3:0]			
Access					W	W	W	W
Reset					–	–	–	–

Bit	7	6	5	4	3	2	1	0
	CTRUPD[3:0]							CENUPD
Access	W	W	W	W				W
Reset	–	–	–	–				–

Bits 19:16 – CUPRUPD[3:0] Comparison x Update Period Update
Defines the time between each update of the comparison x mode and the comparison x value. This time is equal to CUPR+1 periods of the channel 0 counter.

Bits 11:8 – CPRUPD[3:0] Comparison x Period Update
CPR defines the maximum value of the comparison x period counter (CPRCNT). The comparison x value is performed periodically once every CPR+1 periods of the channel 0 counter.

Bits 7:4 – CTRUPD[3:0] Comparison x Trigger Update
The comparison x is performed when the value of the comparison x period counter (CPRCNT) reaches the value defined by CTR.

Bit 0 – CENUPD Comparison x Enable Update

Value	Description
0	The comparison x is disabled and can not match.
1	The comparison x is enabled and can match.

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.40. PWM Channel Mode Register

Name: PWM_CMRx
Offset: 0x0200 + x*0x04 [x=0..2]
Reset: 0x00000000
Property: Read/Write

This register can only be written if bits WPSWS2 and WPHWS2 are cleared in the PWM Write Protection Status Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					PPM	DTLI	DTHI	DTE
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
			TCTS	DPOLI	UPDS	CES	CPOL	CALG
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
					CPRE[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 19 – PPM Push-Pull Mode
The Push-Pull mode is enabled for channel x.

Value	Description
0	The Push-Pull mode is disabled for channel x.
1	The Push-Pull mode is enabled for channel x.

Bit 18 – DTLI Dead-Time PWMLx Output Inverted

Value	Description
0	The dead-time PWMLx output is not inverted.
1	The dead-time PWMLx output is inverted.

Bit 17 – DTHI Dead-Time PWMHx Output Inverted

Value	Description
0	The dead-time PWMHx output is not inverted.
1	The dead-time PWMHx output is inverted.

Bit 16 – DTE Dead-Time Generator Enable

Value	Description
0	The dead-time generator is disabled.
1	The dead-time generator is enabled.

Bit 13 – TCTS Timer Counter Trigger Selection

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	The comparator of the channel x (OCx) is used as the trigger source for the Timer Counter (TC).
1	The counter events of the channel x is used as the trigger source for the Timer Counter (TC).

Bit 12 – DPOLI Disabled Polarity Inverted

Value	Description
0	When the PWM channel x is disabled (CHIDx(PWM_SR) = 0), the OCx output waveform is the same as the one defined by the CPOL bit.
1	When the PWM channel x is disabled (CHIDx(PWM_SR) = 0), the OCx output waveform is inverted compared to the one defined by the CPOL bit.

Bit 11 – UPDS Update Selection

If the PWM period is center-aligned (CALG=1):
0: The update occurs at the next end of the PWM period after writing the update register(s).
1: The update occurs at the next end of the PWM half period after writing the update register(s).
If the PWM period is left-aligned (CALG=0), the update always occurs at the end of the PWM period after writing the update register(s).

Bit 10 – CES Counter Event Selection

If the PWM period is center-aligned (CALG=1):
0: The channel counter event occurs at the end of the PWM period.
1: The channel counter event occurs at the end of the PWM period and at half the PWM period.
If the PWM period is left-aligned (CALG=0), the channel counter event occurs at the end of the period and the CES bit has no effect.

Bit 9 – CPOL Channel Polarity

Value	Description
0	The OCx output waveform (output from the comparator) starts at a low level.
1	The OCx output waveform (output from the comparator) starts at a high level.

Bit 8 – CALG Channel Alignment

Value	Description
0	The period is left-aligned.
1	The period is center-aligned.

Bits 3:0 – CPRE[3:0] Channel Prescaler

Value	Name	Description
	MCK	Peripheral clock
1	MCK_DIV_2	Peripheral clock/2
2	MCK_DIV_4	Peripheral clock/4
3	MCK_DIV_8	Peripheral clock/8
4	MCK_DIV_16	Peripheral clock/16
5	MCK_DIV_32	Peripheral clock/32
6	MCK_DIV_64	Peripheral clock/64
7	MCK_DIV_128	Peripheral clock/128
8	MCK_DIV_256	Peripheral clock/256
9	MCK_DIV_512	Peripheral clock/512
10	MCK_DIV_1024	Peripheral clock/1024
11	CLKA	Clock A
12	CLKB	Clock B

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.41. PWM Channel Duty Cycle Register

Name: PWM_CDTYx
Offset: 0x0204 + x*0x04 [x=0..2]
Reset: 0x00000000
Property: Read/Write

Only the first 24 bits (channel counter size) are significant.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	CDTY[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CDTY[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CDTY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – CDTY[23:0] Channel Duty-Cycle
Defines the waveform duty-cycle. This value must be defined between 0 and CPRD (PWM_CPRDx).

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.42. PWM Channel Duty Cycle Update Register

Name: PWM_CDTYUPDx
Offset: 0x0208 + x*0x04 [x=0..2]
Reset: –
Property: Write-only

This register acts as a double buffer for the CDTY value. This prevents an unexpected waveform when modifying the waveform duty-cycle.

Only the first 24 bits (channel counter size) are significant.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	CDTYUPD[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	CDTYUPD[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	CDTYUPD[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 23:0 – CDTYUPD[23:0] Channel Duty-Cycle Update
Defines the waveform duty-cycle. This value must be defined between 0 and CPRD (PWM_CPRDx).

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.43. PWM Channel Period Register

Name: PWM_CPRDx
Offset: 0x020C + x*0x04 [x=0..2]
Reset: 0x00000000
Property: Read/Write

This register can only be written if bits WPSWS3 and WPHWS3 are cleared in the PWM Write Protection Status Register.

Only the first 24 bits (channel counter size) are significant.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	CPRD[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CPRD[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CPRD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – CPRD[23:0] Channel Period

If the waveform is left-aligned, then the output waveform period depends on the channel counter source clock and can be calculated:
– By using the PWM peripheral clock divided by a given prescaler value “X” (where X = 2^{PREA} is 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, or 1024). The resulting period formula is:

$$\frac{(X \times \mathrm{CPRD})}{f_{\mathrm{peripheral\ clock}}}$$

– By using the PWM peripheral clock divided by a given prescaler value “X” (see above) and by either the DIVA or the DIVB divider. The formula becomes, respectively:

$$\frac{(X \times \mathrm{CPRD} \times \mathrm{DIVA})}{f_{\mathrm{peripheral\ clock}}} \text{ or } \frac{(X \times \mathrm{CPRD} \times \mathrm{DIVB})}{f_{\mathrm{peripheral\ clock}}}$$

If the waveform is center-aligned, then the output waveform period depends on the channel counter source clock and can be calculated:
– By using the PWM peripheral clock divided by a given prescaler value “X” (where X = 2^{PREA} is 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, or 1024). The resulting period formula is:

$$\frac{(2 \times X \times \mathrm{CPRD})}{f_{\mathrm{peripheral\ clock}}}$$

– By using the PWM peripheral clock divided by a given prescaler value “X” (see above) and by either the DIVA or the DIVB divider. The formula becomes, respectively:

© 2026 Microchip Technology Inc. and its subsidiaries

$$\frac{(2 \times X \times \text{CPRD} \times \text{DIVA})}{f_{\text{peripheral clock}}} \text{ or } \frac{(2 \times X \times \text{CPRD} \times \text{DIVB})}{f_{\text{peripheral clock}}}$$

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.44. PWM Channel Period Update Register

Name: PWM_CPRDUPDx
Offset: 0x0210 + x*0x04 [x=0..2]
Reset: –
Property: Write-only

This register can only be written if bits WPSWS3 and WPHWS3 are cleared in the PWM Write Protection Status Register.

This register acts as a double buffer for the CPRD value. This prevents an unexpected waveform when modifying the waveform period.

Only the first 24 bits (channel counter size) are significant.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	CPRDUPD[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	CPRDUPD[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	CPRDUPD[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 23:0 – CPRDUPD[23:0] Channel Period Update

If the waveform is left-aligned, then the output waveform period depends on the channel counter source clock and can be calculated:
– By using the PWM peripheral clock divided by a given prescaler value “X” (where X = 2^{PREA} is 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, or 1024). The resulting period formula is:

$$\frac{(X \times \text{CPRDUPD})}{f_{\text{peripheral clock}}}$$

– By using the PWM peripheral clock divided by a given prescaler value “X” (see above) and by either the DIVA or the DIVB divider. The formula becomes, respectively:

$$\frac{(X \times \text{CPRDUPD} \times \text{DIVA})}{f_{\text{peripheral clock}}} \text{ or } \frac{(X \times \text{CPRDUPD} \times \text{DIVB})}{f_{\text{peripheral clock}}}$$

If the waveform is center-aligned, then the output waveform period depends on the channel counter source clock and can be calculated:
– By using the PWM peripheral clock divided by a given prescaler value “X” (where X = 2^{PREA} is 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, or 1024). The resulting period formula is:

© 2026 Microchip Technology Inc. and its subsidiaries

$$\frac{(2 \times X \times \text{CPRDUPD})}{f_{\text{peripheral clock}}}$$

– By using the PWM peripheral clock divided by a given prescaler value “X” (see above) and by either the DIVA or the DIVB divider. The formula becomes, respectively:

$$\frac{(2 \times X \times \text{CPRDUPD} \times \text{DIVA})}{f_{\text{peripheral clock}}} \text{ or } \frac{(2 \times X \times \text{CPRDUPD} \times \text{DIVB})}{f_{\text{peripheral clock}}}$$

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.45. PWM Channel Counter Register

Name: PWM_CCNTx
Offset: 0x0214 + x*0x04 [x=0..2]
Reset: 0x00000000
Property: Read-only

Only the first 24 bits (channel counter size) are significant.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	CNT[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CNT[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CNT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – CNT[23:0] Channel Counter Register
Channel counter value. This register is reset when:

- the channel is enabled (writing CHIDx in the PWM_ENA register).
- the channel counter reaches CPRD value defined in the PWM_CPRDx register if the waveform is left-aligned.

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.46. PWM Channel Dead Time Register

Name: PWM_DTx
Offset: 0x0218 + x*0x04 [x=0..2]
Reset: 0x00000000
Property: Read/Write

This register can only be written if bits WPSWS4 and WPHWS4 are cleared in the PWM Write Protection Status Register.

Only the first 12 bits (dead-time counter size) of fields DTH and DTL are significant.

Bit	31	30	29	28	27	26	25	24
	DTL[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DTL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DTH[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DTH[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – DTL[15:0] Dead-Time Value for PWMLx Output
Defines the dead-time value for PWMLx output. This value must be defined between 0 and CDTY (PWM_CDTYx).

Bits 15:0 – DTH[15:0] Dead-Time Value for PWMHx Output
Defines the dead-time value for PWMHx output. This value must be defined between 0 and the value (CPRD – CDTY) (PWM_CPRDx and PWM_CDTYx).

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.47. PWM Channel Dead Time Update Register

Name: PWM_DTUPDx
Offset: 0x021C + x*0x04 [x=0..2]
Reset: –
Property: Write-only

This register can only be written if bits WPSWS4 and WPHWS4 are cleared in the PWM Write Protection Status Register.

This register acts as a double buffer for the DTH and DTL values. This prevents an unexpected waveform when modifying the dead-time values.

Only the first 12 bits (dead-time counter size) of fields DTHUPD and DTLUPD are significant.

Bit	31	30	29	28	27	26	25	24
	DTLUPD[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	DTLUPD[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	DTHUPD[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	DTHUPD[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:16 – DTLUPD[15:0] Dead-Time Value Update for PWMLx Output
Defines the dead-time value for PWMLx output. This value must be defined between 0 and CDTY (PWM_CDTYx). This value is applied only at the beginning of the next channel x PWM period.

Bits 15:0 – DTHUPD[15:0] Dead-Time Value Update for PWMHx Output
Defines the dead-time value for PWMHx output. This value must be defined between 0 and the value (CPRD – CDTY) (PWM_CPRDx and PWM_CDTYx). This value is applied only at the beginning of the next channel x PWM period.

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.48. PWM Channel Mode Update Register

Name: PWM_CMUPDx
Offset: 0x0400 + x*0x04 [x=0..2]
Reset: –
Property: Write-only

This register can only be written if bits WPSWS2 and WPHWS2 are cleared in the PWM Write Protection Status Register.

This register acts as a double buffer for the CPOL value. This prevents an unexpected waveform when modifying the polarity value.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			CPOLINVUP				CPOLUP	
Access			W				W	
Reset			–				–	

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 13 – CPOLINVUP Channel Polarity Inversion Update
If this bit is written at ‘1’, the write of the bit CPOLUP is not taken into account.

Value	Description
0	No effect.
1	The OCx output waveform (output from the comparator) is inverted.

Bit 9 – CPOLUP Channel Polarity Update
The write of this bit is taken into account only if the bit CPOLINVUP is written at ‘0’ at the same time.

Value	Description
0	The OCx output waveform (output from the comparator) starts at a low level.
1	The OCx output waveform (output from the comparator) starts at a high level.

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.49. PWM External Trigger Register

Name: PWM_ETRGx
Offset: 0x042C + (x-1)*0x04 [x=1..2]
Reset: 0x00000000
Property: Read/Write

This register can only be written if bits WPSWS2 and WPHWS2 are cleared in the PWM Write Protection Status Register.

Bit	31	30	29	28	27	26	25	24
	RFEN	TRGSRC	TRGFILT	TRGEDGE			TRGMODE[1:0]	
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	0	0	0			0	0

Bit	23	22	21	20	19	18	17	16
	MAXCNT[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	MAXCNT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MAXCNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 31 – RFEN Recoverable Fault Enable

Value	Description
0	The TRGINx signal does not generate a recoverable fault.
1	The TRGINx signal generate a recoverable fault in place of the fault x input.

Bit 30 – TRGSRC Trigger Source

Value	Description
0	The TRGINx signal is driven by the PWMEXTRGx input.
1	The TRGINx signal is driven by the output of the analog comparator configured by the UART.

Bit 29 – TRGFILT Filtered input

Value	Description
0	The external trigger input x is not filtered.
1	The external trigger input x is filtered.

Bit 28 – TRGEDGE Edge Selection

Value	Name	Description
0	FALLING_ZERO	TRGMODE = 1: TRGINx event detection on falling edge. TRGMODE = 2, 3: TRGINx active level is 0
1	RISING_ONE	TRGMODE = 1: TRGINx event detection on rising edge. TRGMODE = 2, 3: TRGINx active level is 1

© 2026 Microchip Technology Inc. and its subsidiaries

Bits 25:24 – TRGMODE[1:0] External Trigger Mode

Value	Name	Description
0	OFF	External trigger is not enabled.
1	MODE1	External PWM Reset Mode
2	MODE2	External PWM Start Mode
3	MODE3	Cycle-by-cycle Duty Mode

Bits 23:0 – MAXCNT[23:0] Maximum Counter value

Maximum channel x counter value measured at the TRGINx event since the last read of the register. At the TRGINx event, if the channel x counter value is greater than the stored MAXCNT value, then MAXCNT is updated by the channel x counter value.

© 2026 Microchip Technology Inc. and its subsidiaries

46.7.50. PWM Leading-Edge Blanking Register

Name: PWM_LEBRx
Offset: 0x0430 + (x-1)*0x04 [x=1..2]
Reset: 0x00000000
Property: Read/Write

This register can only be written if bits WPSWS2 and WPHWS2 are cleared in the PWM Write Protection Status Register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					PWMHREN	PWMHFEN	PWMLREN	PWMLFEN
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		LEBDELAY[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit 19 – PWMHREN PWMH Rising Edge Enable

Value	Description
0	Leading-edge blanking is disabled on PWMHx output rising edge.
1	Leading-edge blanking is enabled on PWMHx output rising edge.

Bit 18 – PWMHFEN PWMH Falling Edge Enable

Value	Description
0	Leading-edge blanking is disabled on PWMHx output falling edge.
1	Leading-edge blanking is enabled on PWMHx output falling edge.

Bit 17 – PWMLREN PWML Rising Edge Enable

Value	Description
0	Leading-edge blanking is disabled on PWMLx output rising edge.
1	Leading-edge blanking is enabled on PWMLx output rising edge.

Bit 16 – PWMLFEN PWML Falling Edge Enable

Value	Description
0	Leading-edge blanking is disabled on PWMLx output falling edge.
1	Leading-edge blanking is enabled on PWMLx output falling edge.

© 2026 Microchip Technology Inc. and its subsidiaries

Bits 6:0 – LEBDELAY[6:0] Leading-Edge Blanking Delay for TRGINx

Leading-edge blanking duration for external trigger x input. The delay is calculated according to the following formula:

LEBDELAY = ($f_{peripheral\ clock}$ × Delay) + 1

© 2026 Microchip Technology Inc. and its subsidiaries

47. Universal Asynchronous Receiver Transmitter (UART)

47.1. Description

The Universal Asynchronous Receiver Transmitter (UART) features a two-pin UART that can be used for communication and trace purposes and offers an ideal medium for in-situ programming solutions.

Moreover, the association with a peripheral DMA controller (PDC) permits packet handling for these tasks with processor time reduced to a minimum.

The UART receiver can be bypassed to get direct access to the analog comparator.

The optical link establishes electrically isolated serial communication with hand-held equipment, such as calibrators compliant with the ANSI-C12.18 or IEC62056-21 standard.

47.2. Embedded Characteristics

- Two-pin UART
  - Independent Receiver and Transmitter with a common programmable Baud Rate Generator
  - Baud Rate can be driven by processor-independent generic source clock
  - Even, odd, mark or space parity generation
  - Parity, framing and overrun error detection
  - Automatic echo, Local Loopback and Remote Loopback Channel modes
  - Digital filter on receive line
  - Interrupt generation
  - Support for two PDC channels with connection to receiver and transmitter
  - Comparison function on received character
  - Optical link modulator and demodulator for communication compliant with ANSI-C12.18 or IEC62056-21 Standards
  - Receiver time-out
- Register Write Protection
- Analog Comparator Controller (ACC)
  - Bypasses UART receiver when ACC is enabled
  - Gives direct access to analog comparator (AC) output
  - Detects edges on AC output
  - Generates interrupt on AC output changes

© 2026 Microchip Technology Inc. and its subsidiaries

47.3. Block Diagram

Figure 47-1. UART Block Diagram

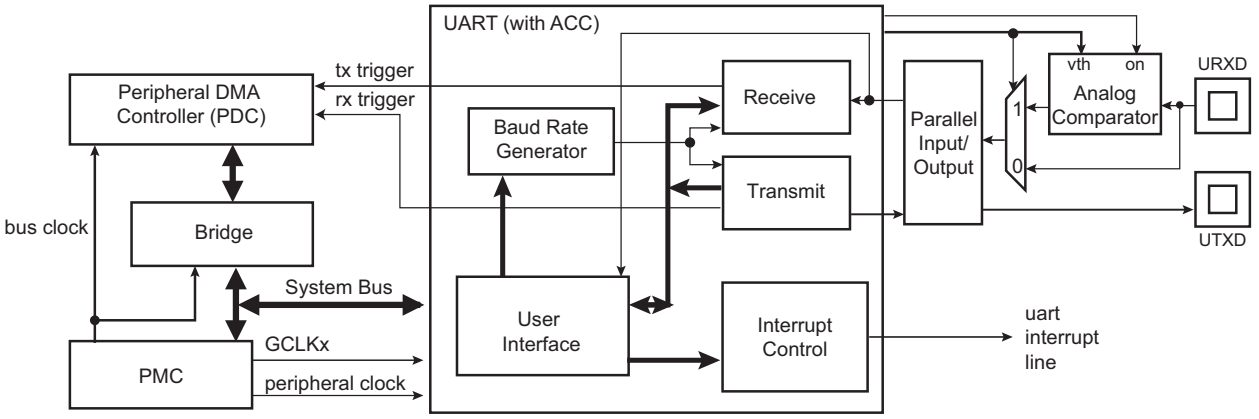


Table 47-1. UART Pin Description

Pin Name	Description	Type
URXD	UART Receive Data	Input
UTXD	UART Trasnmit Data	Output

47.4. Product Dependencies

47.4.1. I/O Lines

The UART pins are multiplexed with PIO lines. The user must first configure the corresponding PIO Controller to enable I/O line operations of the UART.

47.4.2. Power Management

The UART clock can be controlled through the Power Management Controller (PMC). In this case, the user must first configure the PMC to enable the UART clock. Usually, the peripheral identifier used for this purpose is 1.

47.4.3. Interrupt Sources

The UART interrupt line is connected to one of the interrupt sources of the Interrupt Controller. Interrupt handling requires programming of the Interrupt Controller before configuring the UART.

47.4.4. Optical Interface

The UART optical interface requires configuration of the PMC to generate 4096 kHz or 8192 kHz on the PLLA prior to any transfer.

47.5. Functional Description

The UART operates in Asynchronous mode only and supports only 8-bit character handling (with parity). It has no clock pin.

The UART is made up of a receiver and a transmitter that operate independently, and a common baud rate generator.

The UART receiver can be bypassed to access the analog comparator directly. When the UART receiver is bypassed, the UART transmitter can be used to transmit characters.

47.5.1. Baud Rate Generator

The baud rate generator provides the bit period clock named baud rate clock to both the receiver and the transmitter.

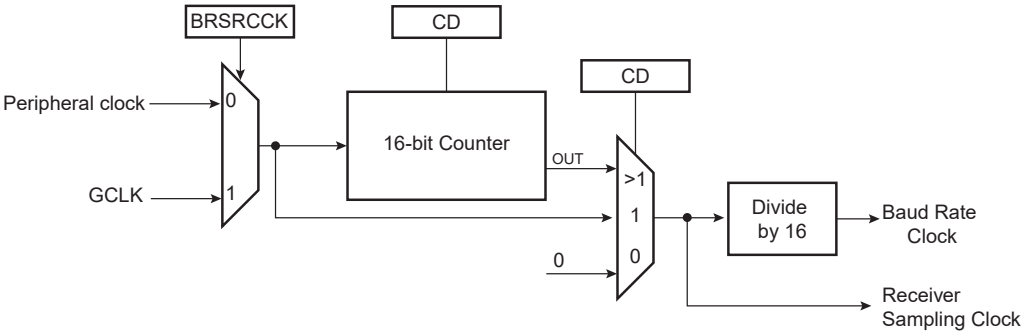
© 2026 Microchip Technology Inc. and its subsidiaries

The baud rate clock is the peripheral clock divided by 16 times the clock divisor (CD) value written in the Baud Rate Generator register (UART_BRGR). If UART_BRGR is set to 0, the baud rate clock is disabled and the UART remains inactive. The maximum allowable baud rate is peripheral clock or GCLK divided by 16. The minimum allowable baud rate is peripheral clock divided by (16 x 65536). The clock source driving the baud rate generator (peripheral clock or GCLK) can be selected by writing UART_MR_BRSRCCK.

If GCLK is selected, the baud rate is independent of the processor/bus clock. Thus the processor clock can be changed while UART is enabled. The processor clock frequency changes must be performed only by programming PMC_MCKR.PRES (refer to "Power Management Controller (PMC)"). Other methods to modify the processor/bus clock frequency (PLL multiplier, etc.) are forbidden when UART is enabled.

The peripheral clock frequency must be at least three times higher than GCLK.

Figure 47-2. Baud Rate Generator



47.5.2. Receiver

47.5.2.1. Receiver Reset, Enable and Disable

After device reset, the UART receiver is disabled and must be enabled before being used. The receiver can be enabled by writing the Control Register (UART_CR) with the bit RXEN to '1'. At this command, the receiver starts looking for a start bit.

The programmer can disable the receiver by writing UART_CR.RXDIS to '1'. If the receiver is waiting for a start bit, it is immediately stopped. However, if the receiver has already detected a start bit and is receiving the data, it waits for the stop bit before actually stopping its operation.

The receiver can be put in reset state by writing UART_CR.RSTRX to '1'. In this case, the receiver immediately stops its current operations and is disabled, whatever its current state. If RSTRX is applied when data is being processed, this data is lost.

47.5.2.2. Start Detection and Data Sampling

The UART only supports asynchronous operations, and this affects only its receiver. The UART receiver detects the start of a received character by sampling the URXD signal until it detects a valid start bit. A low level (space) on URXD is interpreted as a valid start bit if it is detected for more than seven cycles of the sampling clock, which is 16 times the baud rate. Hence, a space that is longer than 7/16 of the bit period is detected as a valid start bit. A space which is 7/16 of a bit period or shorter is ignored and the receiver continues to wait for a valid start bit.

When a valid start bit has been detected, the receiver samples the URXD at the theoretical midpoint of each bit. It is assumed that each bit lasts 16 cycles of the sampling clock (1-bit period) so the bit sampling point is eight cycles (0.5-bit period) after the start of the bit. The first sampling point is therefore 24 cycles (1.5-bit periods) after detecting the falling edge of the start bit.

Each subsequent bit is sampled 16 cycles (1-bit period) after the previous one.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 47-3. Start Bit Detection

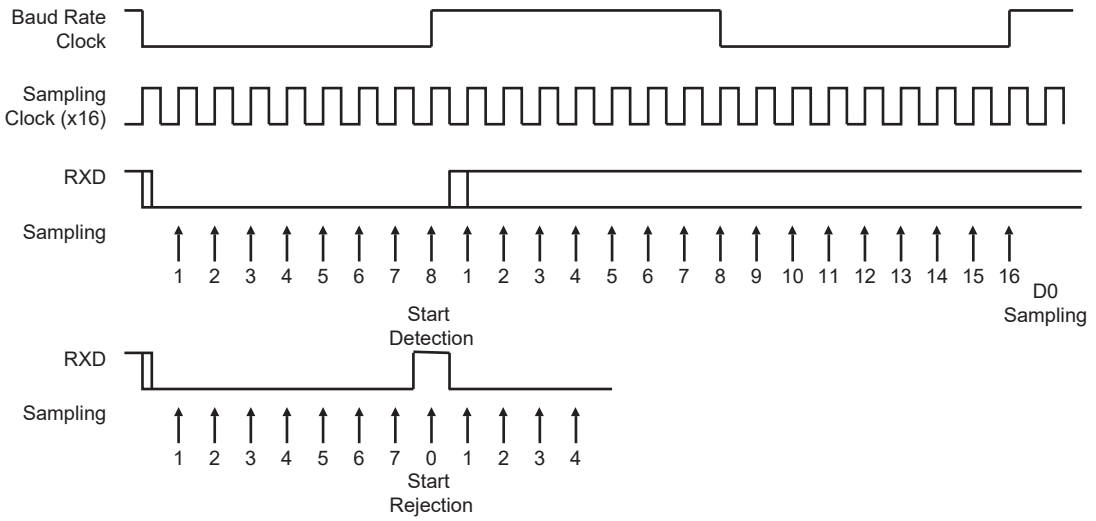
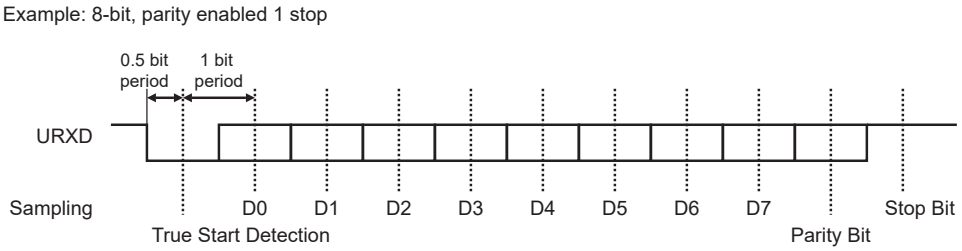


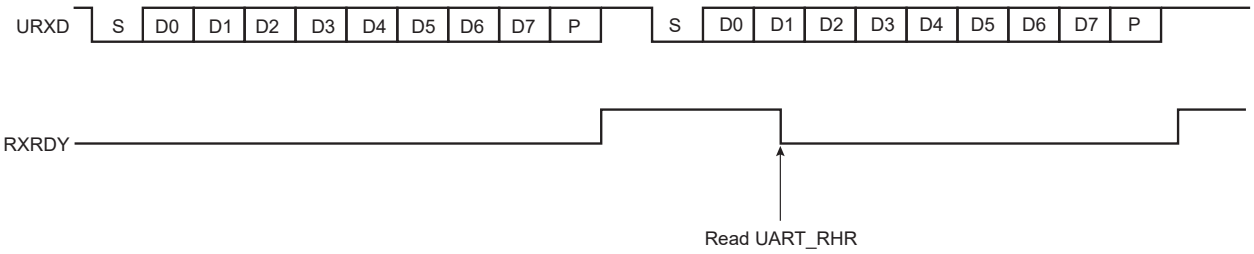
Figure 47-4. Character Reception



47.5.2.3. Receiver Ready

When a complete character is received, it is transferred to the Receive Holding rregister (UART_RHR) and the RXRDY status bit in the Status register (UART_SR) is set. The bit RXRDY is automatically cleared when UART_RHR is read.

Figure 47-5. Receiver Ready

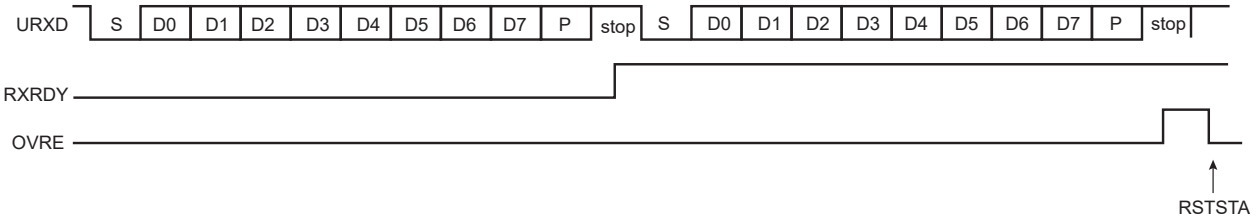


47.5.2.4. Receiver Overrun

The OVRE status bit in UART_SR is set if UART_RHR has not been read by the software (or the PDC) since the last transfer, the RXRDY bit is still set and a new character is received. OVRE is cleared when the software writes a 1 to the bit RSTSTA (Reset Status) in UART_CR.

© 2026 Microchip Technology Inc. and its subsidiaries

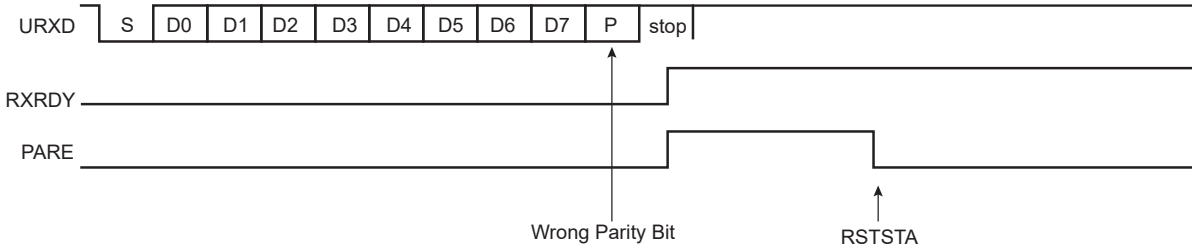
Figure 47-6. Receiver Overrun



47.5.2.5. Parity Error

Each time a character is received, the receiver calculates the parity of the received data bits, in accordance with UART_MR.PAR. It then compares the result with the received parity bit. If different, UART_SR.PARE is set at the same time RXRDY is set. The parity bit is cleared when UART_CR.RSTSTA is written at 1. If a new character is received before the reset status command is written, PARE remains at 1.

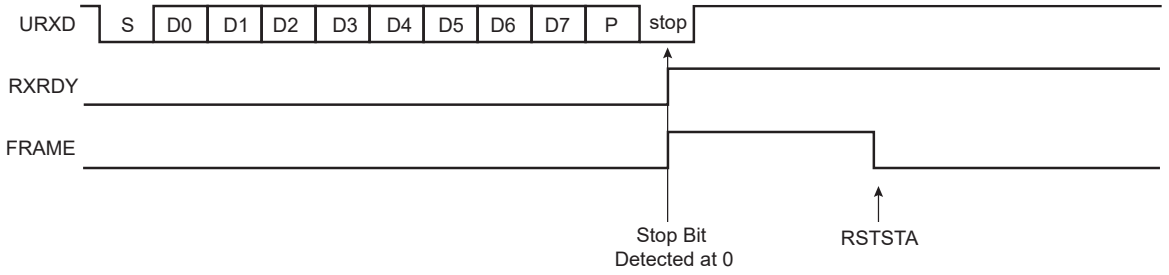
Figure 47-7. Parity Error



47.5.2.6. Receiver Framing Error

When a start bit is detected, it generates a character reception when all the data bits have been sampled. The stop bit is also sampled and when it is detected at 0, UART_SR.FRAME is set at the same time the RXRDY bit is set. The FRAME bit remains high until UART_CR.RSTSTA is written at 1.

Figure 47-8. Receiver Framing Error



47.5.2.7. Receiver Digital Filter

The UART embeds a digital filter on the receive line. It is disabled by default and can be enabled by writing UART_MR.FILTER to ‘1’. When enabled, the receive line is sampled using the 16x bit clock and a three-sample filter (majority 2 over 3) determines the value of the line.

47.5.2.8. Receiver Time-out

The Receiver Time-out provides support in handling variable-length frames. This feature detects an idle condition on the URXD line. When a time-out is detected, UART_SR.TIMEOUT rises and can generate an interrupt, thus indicating to the driver an end of frame.

The time-out delay period (during which the receiver waits for a new character) is programmed in the field TO of the Receiver Time-out register (UART_RTOR). If TO is written to 0, the Receiver Time-out is disabled and no time-out is detected. UART_SR.TIMEOUT remains at ‘0’. Otherwise, the

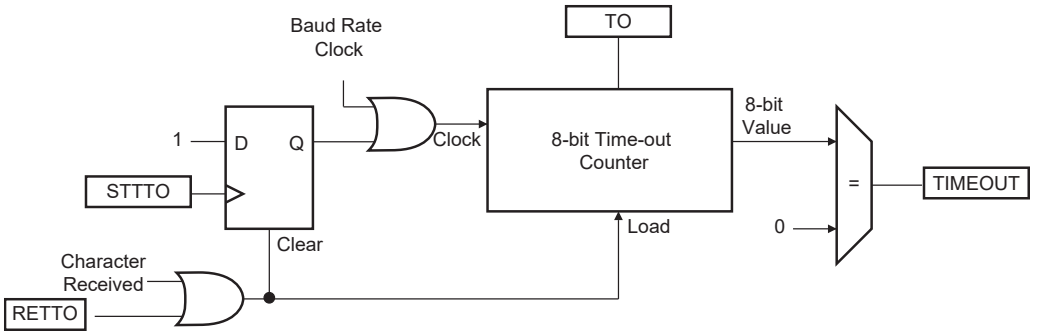
© 2026 Microchip Technology Inc. and its subsidiaries

receiver loads an 8-bit counter with the value programmed in TO. This counter is decremented at each bit period and reloaded each time a new character is received. If the counter reaches 0, UART_SR.TIMEOUT rises. Then, the user can either:

- stop the counter clock until a new character is received. This is performed by writing a '1' to UART_CR.STTTO. In this case, the idle state on URXD before a new character is received does not provide a time-out. This prevents having to handle an interrupt before a character is received and allows waiting for the next idle state on URXD after a frame is received, or
- obtain an interrupt while no character is received. This is performed by writing a '1' to UART_CR.RETTO. If RETTO is performed, the counter starts counting down immediately from the TO value. This enables generation of a periodic interrupt so that a user time-out can be handled, for example when no key is pressed on a keyboard.

The figure below shows the block diagram of the Receiver Time-out feature.

Figure 47-9. Receiver Time-out Block Diagram



The table below gives the maximum time-out period for some standard baud rates.

Table 47-2. Maximum Time-out Period

Baud Rate (bit/s)	Bit Time (µs)	Time-out (µs)
600	1,667	425,085
1,200	833	212,415
2,400	417	106,335
4,800	208	53,040
9,600	104	26,520
14,400	69	17,595
19,200	52	13,260
28,800	35	8,925
38,400	26	6,630
56,000	18	4,590
57,600	17	4,335
200,000	5	1,275

47.5.3. Transmitter

47.5.3.1. Transmitter Reset, Enable and Disable

After device reset, the UART transmitter is disabled and must be enabled before being used. The transmitter is enabled by writing UART_CR.TXEN to '1'. From this command, the transmitter waits for a character to be written in the Transmit Holding register (UART_THR) before actually starting the transmission.

The programmer can disable the transmitter by writing UART_CR.TXDIS to '1'. If the transmitter is not operating, it is immediately stopped. However, if a character is being processed into the internal

© 2026 Microchip Technology Inc. and its subsidiaries

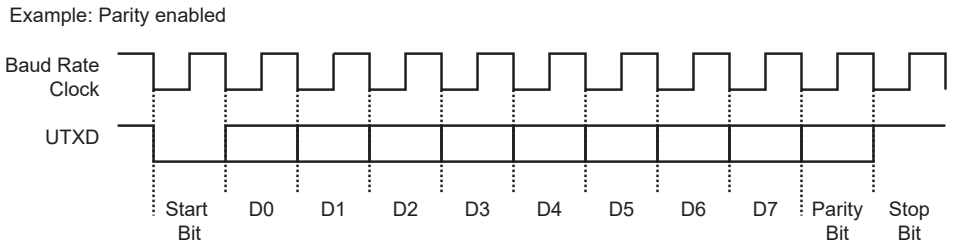
shift register and/or a character has been written in the UART_THR, the characters are completed before the transmitter is actually stopped.

The programmer can also put the transmitter in its reset state by writing the UART_CR.RSTTX to '1'. This immediately stops the transmitter, whether or not it is processing characters.

47.5.3.2. Transmit Format

The UART transmitter drives the pin UTXD at the baud rate clock speed. The line is driven depending on the format defined in UART_MR and the data stored in the internal shift register. One start bit at level 0, then the 8 data bits, from the lowest to the highest bit, one optional parity bit and one stop bit at 1 are consecutively shifted out as shown in the following figure. UART_MR.PARE defines whether or not a parity bit is shifted out. When a parity bit is enabled, it can be selected between an odd parity, an even parity, or a fixed space or mark bit.

Figure 47-10. Character Transmission

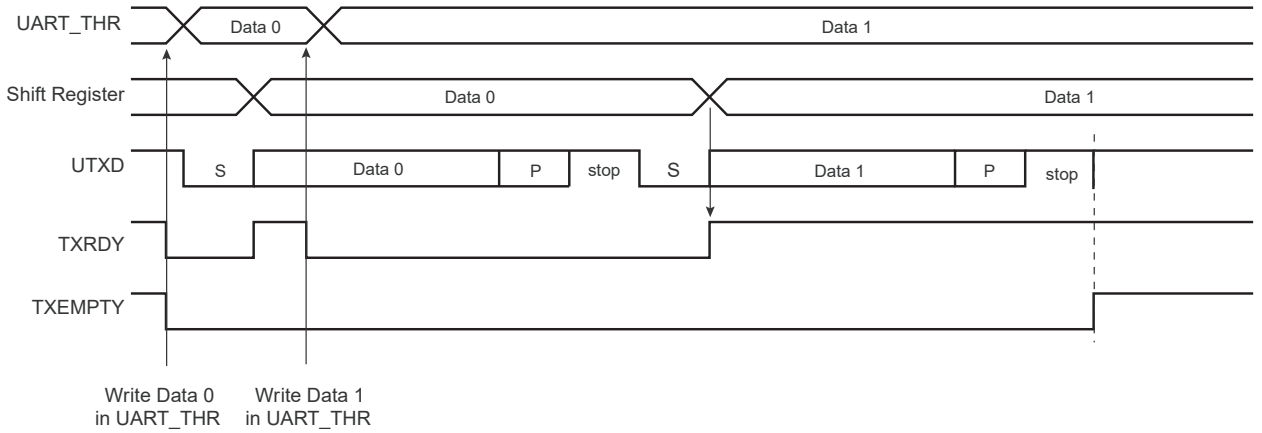


47.5.3.3. Transmitter Control

When the transmitter is enabled, UART_SR.TXRDY is set. The transmission starts when the programmer writes in the UART_THR, and after the written character is transferred from UART_THR to the internal shift register. TXRDY remains high until a second character is written in UART_THR. As soon as the first character is completed, the last character written in UART_THR is transferred into the internal shift register and TXRDY rises again, showing that the holding register is empty.

When both the internal shift register and UART_THR are empty, i.e., all the characters written in UART_THR have been processed, TXEMPTY rises after the last stop bit has been completed.

Figure 47-11. Transmitter Control



47.5.4. Analog Comparator Control

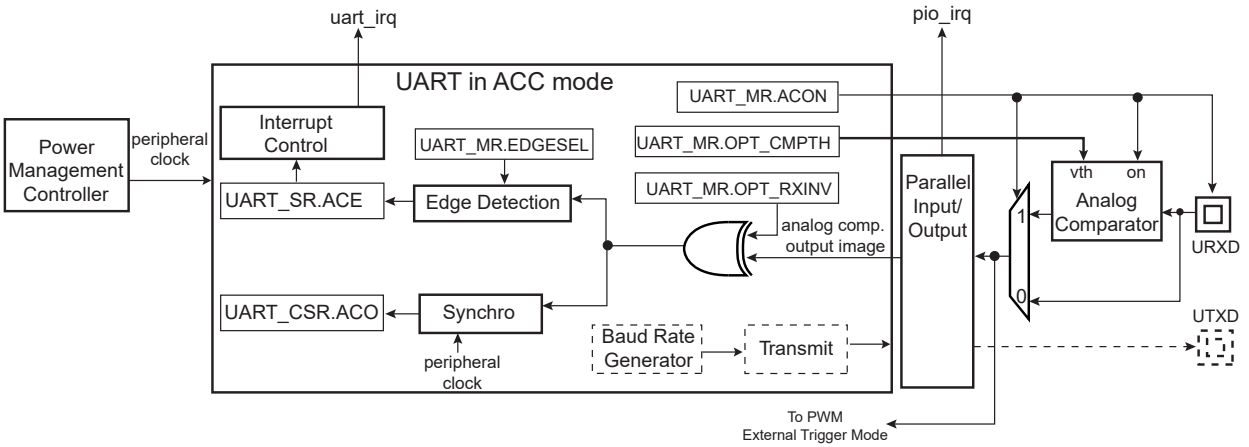
The UART receiver can be disabled and bypassed to obtain control of the analog comparator by writing a '1' to UART_MR.ACON. The comparator output is monitored and any event on the analog comparator output is reported in the flag UART_SR.ACE. The type of event (rising edge, falling edge, any edge) is configurable in UART_MR.EDGESEL.

© 2026 Microchip Technology Inc. and its subsidiaries

The pin URXD is the input of the analog comparator and compared with a predefined threshold configured in UART_MR.OPT_CMPTH.

The analog comparator output is resynchronized on the peripheral clock and reported in UART_CSR.ACO.

Figure 47-12. Analog Comparator Controller



47.5.5. Optical Interface

To use the optical interface circuitry, the PLLA clock must be ready and programmed to generate a frequency within the range of 4096 up to 8192 kHz. This range allows a modulation by a clock with an adjustable frequency from 30 up to 60 kHz.

The optical interface is enabled by writing a '1' to UART_MR.OPT_EN (see UART Mode Register).

When UART_MR.OPT_EN=1 or if UART_MR.ACON=1, the URXD pad is automatically configured in Analog mode and the analog comparator is enabled (see Optical Interface Block Diagram).

To match the characteristics of the off-chip optical receiver circuitry, the voltage reference threshold of the embedded comparator can be adjusted from VDD3V3/10 up to VDD/2 by programming UART_MR.OPT_CMPTH.

The NRZ output of the UART transmitter sub-module is modulated with the 30 up to 60 kHz modulation clock prior to driving the PIO controller.

A logical 0 on the UART transmitter sub-module output generates the said modulated signal (see Optical Interface Waveforms) having a frequency programmable from 30 kHz up to 60 kHz. 38 kHz is the default value assuming the PLLA clock frequency is 8192 kHz. A logical 1 on the UART transmitter sub-module output generates a stuck-at 1 output signal (no modulation). The idle polarity of the modulated signal is 1 (UART_MR.OPT_MDINV = 0).

The idle polarity of the modulated signal can be inverted by writing a '1' to UART_MR.OPT_MDINV.

The duty cycle of the modulated signal can be adjusted from 6.25% up to 50% (default value) by steps of 6.25% by programming UART_MR.OPT_DUTY.

If UART_MR.OPT_DMOD=1, the receive signal is demodulated. The duty cycle of the signal at the output of the analog comparator must be greater than 12%.

To reduce the power consumption of the optical link circuitry, the clock can be turned off and automatically turned on by the UART when activity occurs on receive line. This function is enabled only if UART_MR.OPT_WKUP=1.

When there is no further activity on the optical link (detected by example by a software timeout, etc.), the software stops the clock by writing a 1 in UART_CR.OPT_SLEEP. The clock of the UART is

© 2026 Microchip Technology Inc. and its subsidiaries

stopped and only the analog comparator must be enabled (UART_MR.ACON=1). As soon as a 1 is detected at the output of the analog comparator, the clock is immediately provided to the UART and the optical link is ready to receive and transmit.

Figure 47-13. Optical Interface Block Diagram

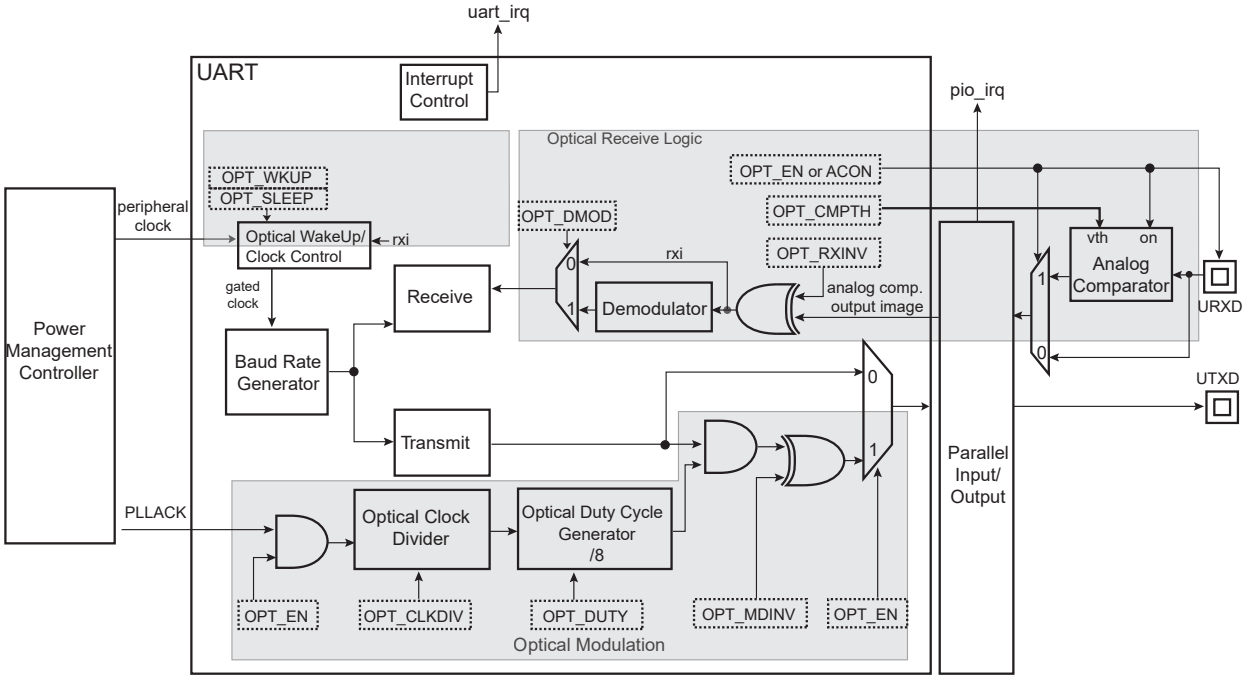
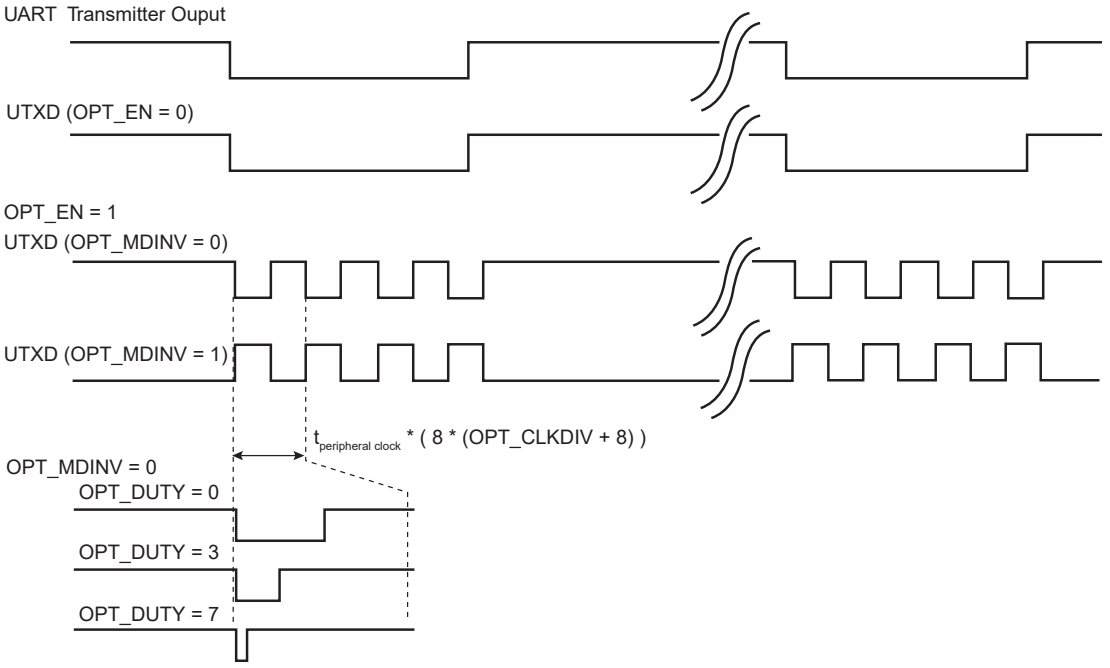


Figure 47-14. Optical Interface Waveforms



The default configuration values of the optical link circuitries allow the 38 kHz modulation, a 50% duty cycle and an idle polarity, allowing a direct drive of an IR LED through a resistor.

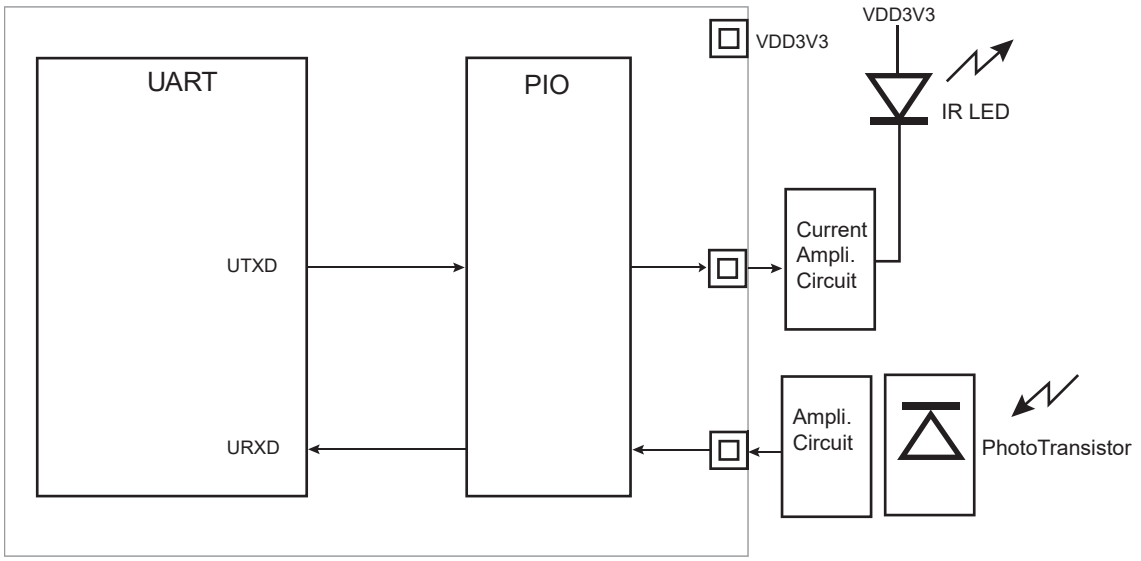
© 2026 Microchip Technology Inc. and its subsidiaries

Refer to the section “Electrical Characteristics” for drive capability of the buffer associated with the UTXD output.

In case of direct drive of the IR LED, the PIO must be programmed in Open-Drain mode.

If an off-chip current amplifier is used to drive the transmitting of the IR LED, the PIO must be programmed in Non Open-Drain drive mode for the line index driving the UTXD output, or in Open-Drain mode depending on the type of external circuitry.

Figure 47-15. Optical Interface Connected to IR Components



47.5.6. Peripheral DMA Controller (PDC)

Both the receiver and the transmitter of the UART are connected to a PDC.

The PDC channels are programmed via registers that are mapped within the UART user interface from the offset 0x100. The status bits are reported in UART_SR and generate an interrupt.

The RXRDY bit triggers the PDC channel data transfer of the receiver. This results in a read of the data in UART_RHR. TXRDY triggers the PDC channel data transfer of the transmitter. This results in a write of data in UART_THR.

47.5.7. Comparison Function on Received Character

When a comparison is performed on a received character, the result of the comparison is reported on UART_SR.CMP when UART_RHR is loaded with the new received character. The CMP flag is cleared by writing a ‘1’ to UART_CR.RSTSTA.

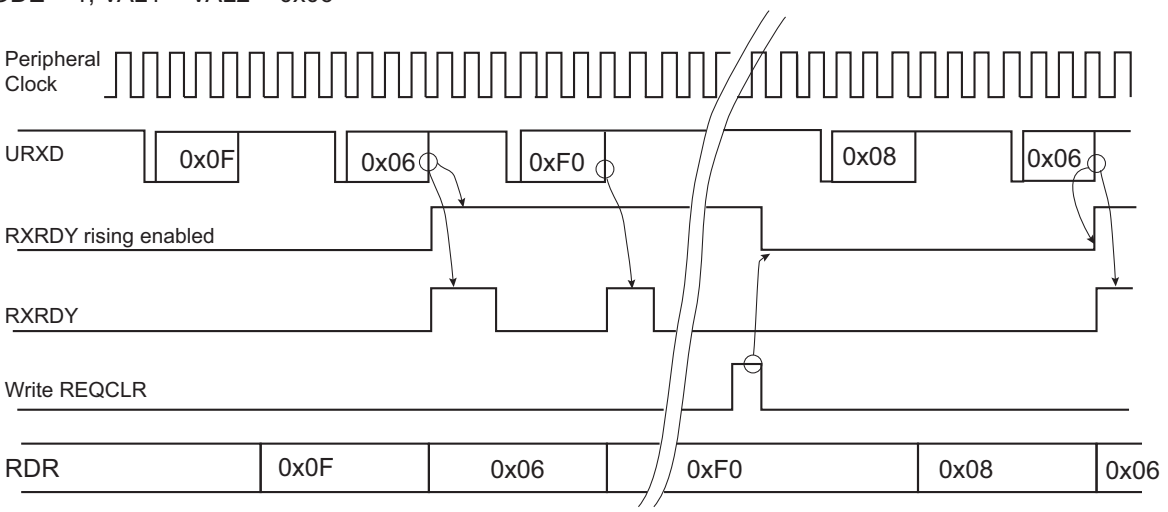
UART_CMPR (see UART Comparison Register) can be programmed to provide different comparison methods. These are listed below:

- If VAL1 equals VAL2, then the comparison is performed on a single value and the flag is set to 1 if the received character equals VAL1.
- If VAL1 is strictly lower than VAL2, then any value between VAL1 and VAL2 sets the CMP flag.
- If VAL1 is strictly higher than VAL2, then the flag CMP is set to 1 if either received character equals VAL1 or VAL2.

By programming CMPMODE to 1, the comparison function result triggers the start of the loading of UART_RHR (see the figure below). The trigger condition occurs as soon as the received character value matches the condition defined by the programming of VAL1, VAL2 and CMPPAR in UART_CMPR. The comparison trigger event can be restarted by writing a ‘1’ to UART_CR.REQCLR.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 47-16. Receive Holding Register Management



47.5.8. Register Write Protection

To prevent any single software error from corrupting UART behavior, certain registers in the address space can be write-protected by setting the WPEN bit in the UART Write Protection Mode Register (UART_WPMR).

The following registers can be write-protected:

- UART Mode Register
- UART Baud Rate Generator Register
- UART Comparison Register
- UART Receiver Time-out Register

47.5.9. Test Modes

The UART supports three test modes. These modes of operation are programmed by using UART_MR.CHMODE.

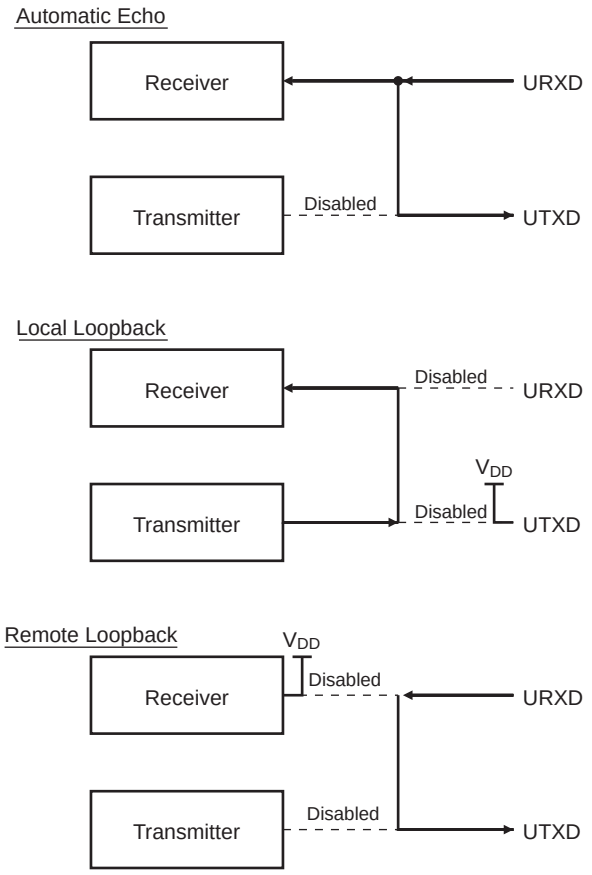
The Automatic Echo mode allows a bit-by-bit retransmission. When a bit is received on the URXD line, it is sent to the UTXD line. The transmitter operates normally, but has no effect on the UTXD line.

The Local Loopback mode allows the transmitted characters to be received. UTXD and URXD pins are not used and the output of the transmitter is internally connected to the input of the receiver. The URXD pin level has no effect and the UTXD line is held high, as in idle state.

The Remote Loopback mode directly connects the URXD pin to the UTXD line. The transmitter and the receiver are disabled and have no effect. This mode allows a bit-by-bit retransmission.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 47-17. Test Modes



© 2026 Microchip Technology Inc. and its subsidiaries

47.6. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	UART_CR	31:24								
		23:16								
		15:8		OPT_SLEEP		REQCLR	STTTO	RETTO		RSTSTA
		7:0	TXDIS	TXEN	RXDIS	RXEN	RSTTX	RSTRX		
0x04	UART_MR	31:24		OPT_CMPTH[2:0]				OPT_DUTY[2:0]		
		23:16				OPT_CLKDIV[4:0]				
		15:8	CHMODE[1:0]			BRSRCCK	PAR[2:0]			OPTI_WKUP
		7:0	EDGESEL[1:0]		ACON	FILTER	OPT_DMOD	OPT_MDINV	OPT_RXINV	OPT_EN
0x08	UART_IER	31:24								
		23:16								ACE
		15:8	CMP			RXBUFF	TXBUFE		TXEMPTY	TIMEOUT
		7:0	PARE	FRAME	OVRE	ENDTX	ENDRX		TXRDY	RXRDY
0x0C	UART_IDR	31:24								
		23:16								ACE
		15:8	CMP			RXBUFF	TXBUFE		TXEMPTY	TIMEOUT
		7:0	PARE	FRAME	OVRE	ENDTX	ENDRX		TXRDY	RXRDY
0x10	UART_IMR	31:24								
		23:16								ACE
		15:8	CMP			RXBUFF	TXBUFE		TXEMPTY	TIMEOUT
		7:0	PARE	FRAME	OVRE	ENDTX	ENDRX		TXRDY	RXRDY
0x14	UART_SR	31:24								
		23:16								ACE
		15:8	CMP			RXBUFF	TXBUFE		TXEMPTY	TIMEOUT
		7:0	PARE	FRAME	OVRE	ENDTX	ENDRX		TXRDY	RXRDY
0x18	UART_RHR	31:24								
		23:16								
		15:8								
		7:0	RXCHR[7:0]							
0x1C	UART_THR	31:24								
		23:16								
		15:8								
		7:0	TXCHR[7:0]							
0x20	UART_BRGR	31:24								
		23:16								
		15:8	CD[15:8]							
		7:0	CD[7:0]							
0x24	UART_CMPR	31:24								
		23:16	VAL2[7:0]							
		15:8		CMPPAR		CMPMODE				
		7:0	VAL1[7:0]							
0x28	UART_RTOR	31:24								
		23:16								
		15:8								
		7:0	TO[7:0]							
0x2C	UART_CSR	31:24								
		23:16								
		15:8								
		7:0								ACO
0x30 ... 0xE3	Reserved									
0xE4	UART_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0								WPEN
0xE8 ... 0xFF	Reserved									

© 2026 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0100 ... 0x128	Reserved for PDC registers	31:24								
		23:16								
		15:8								
		7:0								

© 2026 Microchip Technology Inc. and its subsidiaries

47.6.1. UART Control Register

Name: UART_CR
Offset: 0x00
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		OPT_SLEEP		REQCLR	STTTO	RETTO		RSTSTA
Access		W		W	W	W		W
Reset		–		–	–	–		–

Bit	7	6	5	4	3	2	1	0
	TXDIS	TXEN	RXDIS	RXEN	RSTTX	RSTRX		
Access	W	W	W	W	W	W		
Reset	–	–	–	–	–	–		

Bit 14 – OPT_SLEEP Optical Logic Sleep Mode Command

Value	Description
0	No effect.
1	The peripheral clock is stopped and enabled on the next activity on analog comparator output.

Bit 12 – REQCLR Request Clear

Value	Description
0	No effect.
1	Restarts the comparison trigger to enable loading of the Receiver Holding register.

Bit 11 – STTTO Start Time-out

Value	Description
0	No effect.
1	Starts waiting for a character before clocking the time-out counter. Resets status bit TIMEOUT in UART_SR.

Bit 10 – RETTO Rearm Time-out

Value	Description
0	No effect.
1	Restarts time-out.

Bit 8 – RSTSTA Reset Status

Value	Description
0	No effect.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
1	Resets the status bits PARE, FRAME, CMP and OVRE in the UART_SR.

Bit 7 – TXDIS Transmitter Disable

Value	Description
0	No effect.
1	The transmitter is disabled. If a character is being processed and a character has been written in the UART_THR and RSTTX is not set, both characters are completed before the transmitter is stopped.

Bit 6 – TXEN Transmitter Enable

Value	Description
0	No effect.
1	The transmitter is enabled if TXDIS is 0.

Bit 5 – RXDIS Receiver Disable

Value	Description
0	No effect.
1	The receiver is disabled. If a character is being processed and RSTRX is not set, the character is completed before the receiver is stopped.

Bit 4 – RXEN Receiver Enable

Value	Description
0	No effect.
1	The receiver is enabled if RXDIS is 0.

Bit 3 – RSTTX Reset Transmitter

Value	Description
0	No effect.
1	The transmitter logic is reset and disabled. If a character is being transmitted, the transmission is aborted.

Bit 2 – RSTRX Reset Receiver

Value	Description
0	No effect.
1	The receiver logic is reset and disabled. If a character is being received, the reception is aborted.

© 2026 Microchip Technology Inc. and its subsidiaries

47.6.2. UART Mode Register

Name: UART_MR
Offset: 0x04
Reset: 0x00130000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
		OPT_CMPTH[2:0]				OPT_DUTY[2:0]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bit	23	22	21	20	19	18	17	16
				OPT_CLKDIV[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				1	0	0	1	1

Bit	15	14	13	12	11	10	9	8
	CHMODE[1:0]			BRSRCCK	PAR[2:0]			OPTI_WKUP
Access	R/W	R/W		R/W	R/W	R/W	R/W	R/W
Reset	0	0		0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	EDGESEL[1:0]		ACON	FILTER	OPT_DMOD	OPT_MDINV	OPT_RXINV	OPT_EN
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 30:28 – OPT_CMPTH[2:0] Receive Path Comparator Threshold

Value	Name	Description
0	VDD3V3_DIV2	Comparator threshold is VDD3V3/2 volts.
1	VDD3V3_DIV2P5	Comparator threshold is VDD3V3/2.5 volts.
2	VDD3V3_DIV3P3	Comparator threshold is VDD3V3/3.3 volts.
3	VDD3V3_DIV5	Comparator threshold is VDD3V3/5 volts.
4	VDD3V3_DIV10	Comparator threshold is VDD3V3/10 volts.

Bits 26:24 – OPT_DUTY[2:0] Optical Link Modulation Clock Duty Cycle

Value	Name	Description
0	DUTY_50	Modulation clock duty cycle Is 50%.
1	DUTY_43P75	Modulation clock duty cycle Is 43.75%.
2	DUTY_37P5	Modulation clock duty cycle Is 37.5%.
3	DUTY_31P25	Modulation clock duty cycle Is 31.75%.
4	DUTY_25	Modulation clock duty cycle Is 25%.
5	DUTY_18P75	Modulation clock duty cycle Is 18.75%.
6	DUTY_12P5	Modulation clock duty cycle Is 12.5%.
7	DUTY_6P25	Modulation clock duty cycle Is 6.25%.

Bits 20:16 – OPT_CLKDIV[4:0] Optical Link Clock Divider

Value	Description
0–31	The optical modulation clock frequency is defined by PLLACK / (8 * (OPT_CLKDIV + 8)).

Bits 15:14 – CHMODE[1:0] Channel Mode

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Name	Description
0	NORMAL	Normal mode
1	AUTOMATIC	Automatic echo
2	LOCAL_LOOPBACK	Local loopback
3	REMOTE_LOOPBACK	Remote loopback

Bit 12 – BRSRCCK Baud Rate Source Clock
0 (PERIPH_CLK): The baud rate is driven by the peripheral clock
1 (GCLK): The baud rate is driven by a PMC-programmable clock GCLK (refer to section "Power Management Controller (PMC)").

Bits 11:9 – PAR[2:0] Parity Type

Value	Name	Description
0	EVEN	Even parity
1	ODD	Odd parity
2	SPACE	Space: parity forced to 0
3	MARK	Mark: parity forced to 1
4	NO	No parity

Bit 8 – OPTI_WKUP Optical Link Activity Wake-up Enable

Value	Name	Description
0	DISABLED	To detect any activity on the output of analog comparator, the clock is always active and OPT_EN must be written to 1.
1	ENABLED	If OPT_EN=0 and a logical 1 is detected after inversion (if RXINV=1) on analog comparator output, the clock if automatically enabled for all UART sub-modules. After a period of inactivity on URXD (time-out) the software can instruct the UART to disabled the clock of all sub-modules to reduce power consumption by applying the UART_CR.OPT_SLEEP command.

Bits 7:6 – EDGESEL[1:0] Analog Comparator Output Edge Selection

Value	Name	Description
0	RISING	UART_SR.ACE is set if a rising edge is detected on analog comparator output.
1	FALLING	UART_SR.ACE is set if a falling edge is detected on analog comparator output.
2	ANY_EDGE	UART_SR.ACE is set as soon as an edge is detected on analog comparator output.

Bit 5 – ACON Analog Comparator Enable

Value	Name	Description
0	DISABLED	The analog comparator is disabled. If OPT_EN=1, the analog comparator is enabled.
1	ENABLED	The analog comparator is enabled.

Bit 4 – FILTER Receiver Digital Filter
0 (DISABLED): UART does not filter the receive line.
1 (ENABLED): UART filters the receive line using a three-sample filter (16x-bit clock) (2 over 3 majority).

Bit 3 – OPT_DMOD Optical Demodulation Enable

Value	Name	Description
0	DISABLED	The optical demodulator is disabled. External demodulation must be enabled.
1	ENABLED	The optical demodulator is enabled.

Bit 2 – OPT_MDINV UART Modulated Data Inverted

Value	Name	Description
0	DISABLED	The output of the modulator is not inverted.

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Name	Description
1	ENABLED	The output of the modulator is inverted.

Bit 1 – OPT_RXINV UART Receive Data Inverted

Value	Name	Description
0	DISABLED	The comparator data output is not inverted before entering UART.
1	ENABLED	The comparator data output is inverted before entering UART.

Bit 0 – OPT_EN UART Optical Interface Enable

Value	Name	Description
0	DISABLED	Disables the UART optical link.
1	ENABLED	Enables the UART optical link.

© 2026 Microchip Technology Inc. and its subsidiaries

47.6.3. UART Interrupt Enable Register

Name: UART_IER
Offset: 0x08
Reset: –
Property: Write-only

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								ACE
Access								W
Reset								–

Bit	15	14	13	12	11	10	9	8
	CMP			RXBUFF	TXBUFE		TXEMPTY	TIMEOUT
Access	W			W	W		W	W
Reset	–			–	–		–	–

Bit	7	6	5	4	3	2	1	0
	PARE	FRAME	OVRE	ENDTX	ENDRX		TXRDY	RXRDY
Access	W	W	W	W	W		W	W
Reset	–	–	–	–	–		–	–

Bit 16 – ACE Enable Analog Comparator Event Interrupt

Bit 15 – CMP Enable Comparison Interrupt

Bit 12 – RXBUFF Enable Buffer Full Interrupt

Bit 11 – TXBUFE Enable Buffer Empty Interrupt

Bit 9 – TXEMPTY Enable TXEMPTY Interrupt

Bit 8 – TIMEOUT Enable Time-out Interrupt

Bit 7 – PARE Enable Parity Error Interrupt

Bit 6 – FRAME Enable Framing Error Interrupt

Bit 5 – OVRE Enable Overrun Error Interrupt

Bit 4 – ENDTX Enable End of Transmit Interrupt

Bit 3 – ENDRX Enable End of Receive Transfer Interrupt

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 1 – TXRDY Enable TXRDY Interrupt

Bit 0 – RXRDY Enable RXRDY Interrupt

© 2026 Microchip Technology Inc. and its subsidiaries

47.6.4. UART Interrupt Disable Register

Name: UART_IDR
Offset: 0x0C
Reset: –
Property: Write-only

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								ACE
Access								W
Reset								–

Bit	15	14	13	12	11	10	9	8
	CMP			RXBUFF	TXBUFE		TXEMPTY	TIMEOUT
Access	W			W	W		W	W
Reset	–			–	–		–	–

Bit	7	6	5	4	3	2	1	0
	PARE	FRAME	OVRE	ENDTX	ENDRX		TXRDY	RXRDY
Access	W	W	W	W	W		W	W
Reset	–	–	–	–	–		–	–

Bit 16 – ACE Disable Analog Comparator Event Interrupt

Bit 15 – CMP Disable Comparison Interrupt

Bit 12 – RXBUFF Disable Buffer Full Interrupt

Bit 11 – TXBUFE Disable Buffer Empty Interrupt

Bit 9 – TXEMPTY Disable TXEMPTY Interrupt

Bit 8 – TIMEOUT Disable Time-out Interrupt

Bit 7 – PARE Disable Parity Error Interrupt

Bit 6 – FRAME Disable Framing Error Interrupt

Bit 5 – OVRE Disable Overrun Error Interrupt

Bit 4 – ENDTX Disable End of Transmit Interrupt

Bit 3 – ENDRX Disable End of Receive Transfer Interrupt

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 1 – TXRDY Disable TXRDY Interrupt

Bit 0 – RXRDY Disable RXRDY Interrupt

© 2026 Microchip Technology Inc. and its subsidiaries

47.6.5. UART Interrupt Mask Register

Name: UART_IMR
Offset: 0x10
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is disabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								ACE
Access								W
Reset								–

Bit	15	14	13	12	11	10	9	8
	CMP			RXBUFF	TXBUFE		TXEMPTY	TIMEOUT
Access	R			R	R		R	R
Reset	0			0	0		0	0

Bit	7	6	5	4	3	2	1	0
	PARE	FRAME	OVRE	ENDTX	ENDRX		TXRDY	RXRDY
Access	R	R	R	R	R		R	R
Reset	0	0	0	0	0		0	0

Bit 16 – ACE Mask Analog Comparator Event Interrupt

Bit 15 – CMP Mask Comparison Interrupt

Bit 12 – RXBUFF Mask RXBUFF Interrupt

Bit 11 – TXBUFE Mask TXBUFE Interrupt

Bit 9 – TXEMPTY Mask TXEMPTY Interrupt

Bit 8 – TIMEOUT Mask Time-out Interrupt

Bit 7 – PARE Mask Parity Error Interrupt

Bit 6 – FRAME Mask Framing Error Interrupt

Bit 5 – OVRE Mask Overrun Error Interrupt

Bit 4 – ENDTX Mask End of Transmit Interrupt

Bit 3 – ENDRX Mask End of Receive Transfer Interrupt

© 2026 Microchip Technology Inc. and its subsidiaries

Bit 1 – TXRDY Mask TXRDY Interrupt

Bit 0 – RXRDY Mask RXRDY Interrupt

© 2026 Microchip Technology Inc. and its subsidiaries

47.6.6. UART Interrupt Status Register

Name: UART_SR
Offset: 0x14
Reset: 0x00001818
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								ACE
Access								R
Reset								0

Bit	15	14	13	12	11	10	9	8
	CMP			RXBUFF	TXBUFE		TXEMPTY	TIMEOUT
Access	R			R	R		R	R
Reset	0			1	1		0	0

Bit	7	6	5	4	3	2	1	0
	PARE	FRAME	OVRE	ENDTX	ENDRX		TXRDY	RXRDY
Access	R	R	R	R	R		R	R
Reset	0	0	0	1	1		0	0

Bit 16 – ACE Analog Comparator Event Interrupt (Cleared by writing UART_CR.RSTSTA)

Value	Description
0	No event occurred since the last read of UART_SR.
1	The event (configured in UART_MR.EDGESEL) occurred since the last read of UART_SR.

Bit 15 – CMP Comparison Match (Cleared by writing UART_CR.RSTSTA)

Value	Description
0	No received character matches the comparison criteria programmed in VAL1, VAL2 fields and in CMPPAR bit since the last RSTSTA.
1	The received character matches the comparison criteria.

Bit 12 – RXBUFF Receive Buffer Full

Value	Description
0	The buffer full signal from the receiver PDC channel is inactive.
1	The buffer full signal from the receiver PDC channel is active.

Bit 11 – TXBUFE Transmission Buffer Empty

Value	Description
0	The buffer empty signal from the transmitter PDC channel is inactive.
1	The buffer empty signal from the transmitter PDC channel is active.

Bit 9 – TXEMPTY Transmitter Empty (Cleared by writing UART_THR)

© 2026 Microchip Technology Inc. and its subsidiaries

Value	Description
0	There are characters in UART_THR, or characters being processed by the transmitter, or the transmitter is disabled.
1	There are no characters in UART_THR and there are no characters being processed by the transmitter.

Bit 8 – TIMEOUT Receiver Time-out (Cleared by writing UART_CR.STTTO)

Value	Description
0	There has not been a time-out since the last Start Time-out command (STTTO in UART_CR) or the Time-out register is 0.
1	There has been a time-out since the last Start Time-out command (STTTO in UART_CR).

Bit 7 – PARE Parity Error (Cleared by writing UART_CR.RSTSTA)

Value	Description
0	No parity error has occurred since the last RSTSTA.
1	At least one parity error has occurred since the last RSTSTA.

Bit 6 – FRAME Framing Error (Cleared by writing UART_CR.RSTSTA)

Value	Description
0	No framing error has occurred since the last RSTSTA.
1	At least one framing error has occurred since the last RSTSTA.

Bit 5 – OVRE Overrun Error (Cleared by writing UART_CR.RSTSTA)

Value	Description
0	No overrun error has occurred since the last RSTSTA.
1	At least one overrun error has occurred since the last RSTSTA.

Bit 4 – ENDTX End of Transmitter Transfer

Value	Description
0	The end of transfer signal from the transmitter PDC channel is inactive.
1	The end of transfer signal from the transmitter PDC channel is active.

Bit 3 – ENDRX End of Receiver Transfer

Value	Description
0	The end of transfer signal from the receiver PDC channel is inactive.
1	The end of transfer signal from the receiver PDC channel is active.

Bit 1 – TXRDY Transmitter Ready (Cleared by writing UART_THR)

Value	Description
0	A character has been written to UART_THR and not yet transferred to the internal shift register, or the transmitter is disabled.
1	There is no character written to UART_THR not yet transferred to the internal shift register.

Bit 0 – RXRDY Receiver Ready (Cleared by reading UART_RHR)

Value	Description
0	No character has been received since the last read of the UART_RHR, or the receiver is disabled.
1	At least one complete character has been received, transferred to UART_RHR and not yet read.

© 2026 Microchip Technology Inc. and its subsidiaries

47.6.7. UART Receiver Holding Register

Name: UART_RHR
Offset: 0x18
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	RXCHR[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – RXCHR[7:0] Received Character
Last received character if RXRDY is set.

© 2026 Microchip Technology Inc. and its subsidiaries

47.6.8. UART Transmit Holding Register

Name: UART_THR
Offset: 0x1C
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	TXCHR[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 7:0 – TXCHR[7:0] Character to be Transmitted
Next character to be transmitted after the current character if TXRDY is not set.

© 2026 Microchip Technology Inc. and its subsidiaries

47.6.9. UART Baud Rate Generator Register

Name: UART_BRGR
Offset: 0x20
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	CD[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – CD[15:0] Clock Divisor

Value	Description
0	Baud rate clock is disabled
1 to 65,535	If BRSRCCK = 0: $CD = \frac{f_{peripheral\ clock}}{16 \times Baud\ Rate}$ If BRSRCCK = 1: $CD = \frac{f_{GCLKx}}{16 \times Baud\ Rate}$

© 2026 Microchip Technology Inc. and its subsidiaries

47.6.10. UART Comparison Register

Name: UART_CMPR
Offset: 0x24
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	VAL2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
		CMPPAR		CMPMODE				
Access		R/W		R/W				
Reset		0		0				

Bit	7	6	5	4	3	2	1	0
	VAL1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – VAL2[7:0] Second Comparison Value for Received Character

Value	Description
0-255	The received character must be lower or equal to the value of VAL2 and higher or equal to VAL1 to set CMP flag in UART_SR.

Bit 14 – CMPPAR Compare Parity

Value	Description
0	The parity is not checked and a bad parity cannot prevent from waking up the system.
1	The parity is checked and a matching condition on data can be cancelled by an error on parity bit, so no wake-up is performed.

Bit 12 – CMPMODE Comparison Mode

Value	Name	Description
0	FLAG_ONLY	Any character is received and comparison function drives CMP flag.
1	START_CONDITION	Comparison condition must be met to start reception.

Bits 7:0 – VAL1[7:0] First Comparison Value for Received Character

Value	Description
0-255	The received character must be higher or equal to the value of VAL1 and lower or equal to VAL2 to set CMP flag in UART_SR.

© 2026 Microchip Technology Inc. and its subsidiaries

47.6.11. UART Receiver Time-out Register

Name: UART_RTOR
Offset: 0x28
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	TO[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – TO[7:0] Time-out Value

Value	Description
0	The receiver time-out is disabled.
1–255	The receiver time-out is enabled and the time-out delay is TO x bit period.

© 2026 Microchip Technology Inc. and its subsidiaries

47.6.12. UART Current Status Register

Name: UART_CSR
Offset: 0x2C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								ACO
Access								R
Reset								0

Bit 0 – ACO Analog Comparator Output
Current value of the analog comparator output (resynchronized on peripheral clock).

© 2026 Microchip Technology Inc. and its subsidiaries

47.6.13. UART Write Protection Mode Register

Name: UART_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
								WPEN
Access								R/W
Reset								0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x554152	PASSWD	Writing any other value in this field aborts the write operation. Always reads as 0.

Bit 0 – WPEN Write Protection Enable

See Register Write Protection for the list of registers that can be protected.

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x554152 (UART in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x554152 (UART in ASCII).

© 2026 Microchip Technology Inc. and its subsidiaries

48. Electrical Characteristics

48.1. Electrical Parameters Usage

The sections that follow provide the limiting values for device electrical parameters.

- Unless otherwise noted, these values are valid over the junction temperature range T_J= [-40°C + 100°C].
- Parameters annotated as “Simulation data” are not production-tested. Their limiting values come from simulations run in corner case conditions and were verified by electrical characterization over a limited number of samples.
- These limits may be affected by the board on which the device is mounted. In particular, noisy supply and ground conditions must be avoided and care must be taken to provide:
  - a PCB with a low impedance ground plane (a single unbroken ground plane is strongly recommended)
  - low impedance decoupling of the device power supply inputs. A 10 nF to 220 nF Ceramic X7R (or X5R) capacitor placed very close to each power supply input is a minimum requirement. See specific recommendations regarding analog pins or functions in the corresponding sections. To reduce any potential electromagnetic compatibility (EMC) related issues, it is good practice to double this decoupling capacitor whenever possible with a high frequency one, for example, one 100 pF (C0G or NP0) per power supply input.
  - low impedance power supply decoupling of external components. This recommendation aims at avoiding large current spikes flowing into the PCB ground and power planes.
- In addition, although the device is specified with wide operating supply ranges on most of its supply inputs, large and fast supply variations may lead to unpredictable device behavior including, but not limited to, out-of-specification operation. It is therefore mandatory to keep the power supply variations within the limits given in Table 48-1 during device operation.

Table 48-1. Maximum Power Supply Variations

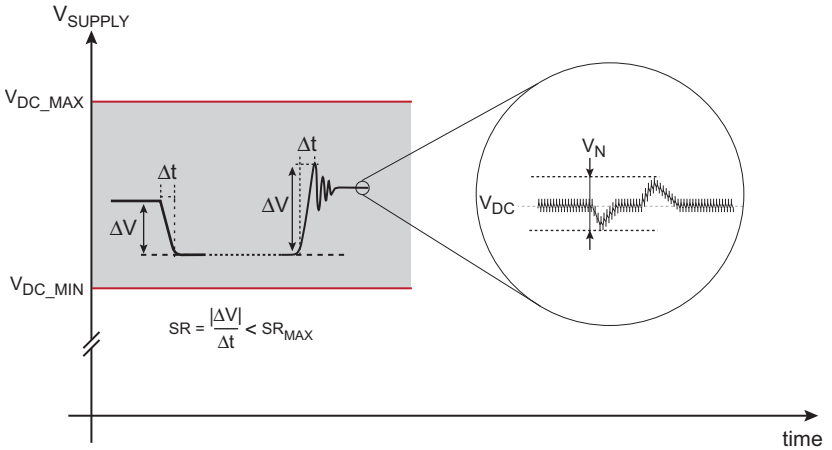
Symbol	Parameter	Conditions	Min	Max	Unit
V_N	Peak-to-peak ripple and noise voltage	Applies to: (VDDCORE, VDDPLL)(4) VBAT, VDD3V3	–	2	% V_{DC}(1)
SR	Slew rate of power supply variations	$\Delta V \le 5\%\ V_{DC\_MIN}$(2, 3)	–	±5	mV/µs

Notes:

1. V_{DC} is the power supply DC value.
2. V_{DC_MIN} is the minimum operating voltage of the supply input as described in table Recommended Operating Conditions on Power Supply Inputs.
3. ΔV is the variation amplitude. The slew rate specification applies when $\Delta V \ge V_N$.
4. If powered externally.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 48-1. Maximum Power Supply Variation



48.2. Absolute Maximum Ratings

Table 48-2. Absolute Maximum Ratings

Storage Temperature	-60°C to +150°C	**Note:** Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied. **Exposure to absolute maximum rating conditions for extended periods may affect device reliability.**
Power Supply Inputs with respect to ground pins:		
VDDCORE, VDDPLL	-0.3V to 1.4V	
VBAT, VDD3V3, VDDLCD	-0.3V to 4.0V	
Voltage on VDD3V3 Digital Input Pins with Respect to Ground	-0.3V to 4.0V	
Voltage on VBAT Digital Input Pins with Respect to Ground	-0.3V to 4.0V	
Injected Current into any input pin	± 1 mA	
Total Injected Current in all input pins of a common power supply pair	± 10 mA	
Total DC Output Current on all I/O lines in one segment:		
128-pin EP-TQFP	100 mA	
VQFN64	100 mA	
Maximum DC Output Current per output pin	± 20 mA	

Note: All I/O pins are internally clamped to their respective VDD and GND rails as defined in the table Pinout and Multiplexing.

© 2026 Microchip Technology Inc. and its subsidiaries

48.3. Recommended Operating Conditions

48.3.1. Power Supply Inputs

Table 48-3. Recommended Operating Conditions on Power Supply Inputs

Supply Name	Parameter	Conditions	Min	Max	Unit
VDD3V3	• Core voltage regulator input • LCD voltage regulator input, • Analog block supply (ADC, Temp Sensor and voltage reference) • I/O Buffers lines Power Supply • Flash memory charge pumps supply for read, erase and program operations	12-bit ADC, LCD and Internal Voltage Reference not operating	2.25	3.6	V
		12-bit ADC operating with external reference voltage. LCD with external supply	2.5	3.6	V
		All analog blocks and LCD operating	2.8	3.6	V
VDDCORE	Core Logic Power Supply (including CPU, Memories and Peripherals)	• f_{CPU0} ≤ 200 MHz, • f_{MCK0} ≤ 200 MHz, • $f_{MCK0DIV}$ ≤ 100 MHz, • $f_{MCK0DIV2}$ ≤ 100 MHz • $f_{MCK1DIV}$ ≤ 120 MHz Applies only when VDDCORE and VDDPLL are powered from an external regulator. VDDCORE and VDDPLL are powered from one single power source such that V(VDDCORE,VDDPLL) ≤ 50mV	1.04	1.21	V
VDDPLL	PLLs and 12-48 MHz Crystal Oscillators supply				
VBAT	Power supply of the backup area if no VDD3V3	–	1.65	3.6	V
VDDLCD	LCD voltage regulator input or output for the LCD Driver, and its logic	At any time, VDDLCD ≤ VDD3V3	2.5	3.6	V
t_{R_VDD}	Power supply slope at power-up	Applies to any of the power supply inputs listed above	0.2	20	mV/µs
t_{F_VDD}	Power supply slope at power-down	Applies to any of the power supply inputs listed above	-20	-1	mV/µs

48.3.2. Input Pins

Table 48-4. Recommended Operating Conditions on Input Pins(1)

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{GPIO_IN}	Input voltage range on GPIOs referenced to VDD3V3	On any pin configured as a digital input	-0.3	–	VDD3V3 +0.3	V
V_{VDDBU_IN}	Input voltage range on inputs referenced to VDDBU (VBAT or VDD3V3)	–	-0.3	–	(VBAT,VDD3V3) +0.3	V
V_{IN}	Input line voltage range on inputs(2, 3)	–	-0.3	–	V_{DD3V3} + 0.3V	V
I_{IN}	DC current injection on inputs(4)	–	–	–	±0.2	mA
I_{TOT_INJ}	Total current injection per power rail or per ground rail(5)	–	–	–	±2	mA

© 2026 Microchip Technology Inc. and its subsidiaries

Notes:

1. All I/O pins are internally clamped to their respective VDD and GND rails as defined in the table "Pin Description".
2. Input voltages $V_{IN} \leq 0V$ or $V_{IN} \geq VDD3V3$ lead to negative or positive current injection on inputs.
3. For analog inputs, input voltages $V_{IN} \geq \min(VDD3V3, V_{VREFP})$ lead to saturated A/D conversion to 0xFFF.
4. Current injection on A/D converter analog inputs may degrade the analog performance of the corresponding channel or the analog performance of other analog channels.
5. Corresponds to the sum of the positive currents into one power rail and respectively to the sum of the negative currents into one ground rail as defined in the table Pinout and Multiplexing.

48.3.3. Thermal Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
T_A	Ambient temperature range	–	-40	–	+85	°C
T_J	Junction temperature range	–	-40	–	+100	

Power Dissipation (in W) = PD = P_{INT} + $P_{I/O}$, where

- Internal Chip Power Dissipation:
  - P_{INT} = VDD x (IDD – S(IOH))
- I/O Pin Power Dissipation:
  - $P_{I/O}$ = S ({VDD – VOH} x IOH) + S (VOL x IOL))

48.3.3.1. Thermal Resistance Data

Table 48-5. Thermal Resistance Data

Symbol	Parameter	Conditions	(1)	Unit
Θ_{JA}	Junction-to-ambient thermal resistance	JESD51-2, four-layer (2s2p), 0m/s air flow	17.3	°C/W
Θ_{JC}	Junction-to-case thermal resistance	–	9	
Θ_{JA}	Junction-to-ambient thermal resistance	JESD51-2, four-layer (2s2p), 0m/s air flow	21.5	°C/W
Θ_{JC}	Junction-to-case thermal resistance	–	12.2	

Note:

1. Junction to ambient thermal resistance, Theta-JA (Θ_{JA}) numbers are achieved by package simulations.

48.3.4. Clocks

Table 48-6. Recommended Operating Conditions on Internal Clocks

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{CPU0_CLK}	Processor clock (CPU_CLK0) frequency	$V_{DDCORE} \geq 1.04V$	–	–	200	MHz
f_{MCK0xx}	Main Bus clock (MCK) frequency - MCK0 - MCK0DIV - MCK0DIV2	$V_{DDCORE} \geq 1.04V$	– – –	– – –	200 100 100	MHz
$f_{MCK1DIV}$	Main Bus clock (MCK) frequency	$V_{DDCORE} \geq 1.04V$	–	–	100	MHz

© 2026 Microchip Technology Inc. and its subsidiaries

48.4. Recommended Power Supply Sequencing for Externally Supplied VDDCORE/ VDDPLL

48.4.1. Power-Up and Power-Down

At power-up, from a power supply sequencing perspective, the power supply inputs are categorized into independent groups. The figure below shows the recommended power-up sequence by group.

Note that:

- VBAT
  - When supplied from a pre-charged storage element (battery or super-capacitor), VBAT is an always-on supply input and is therefore not part of the power supply sequencing.
  - When no storage element is used on VBAT in the application, VBAT must be tied to VDD3V3.

Figure 48-2. Recommended Power-Up Sequence with VDDCORE/VDDPLL Externally Supplied

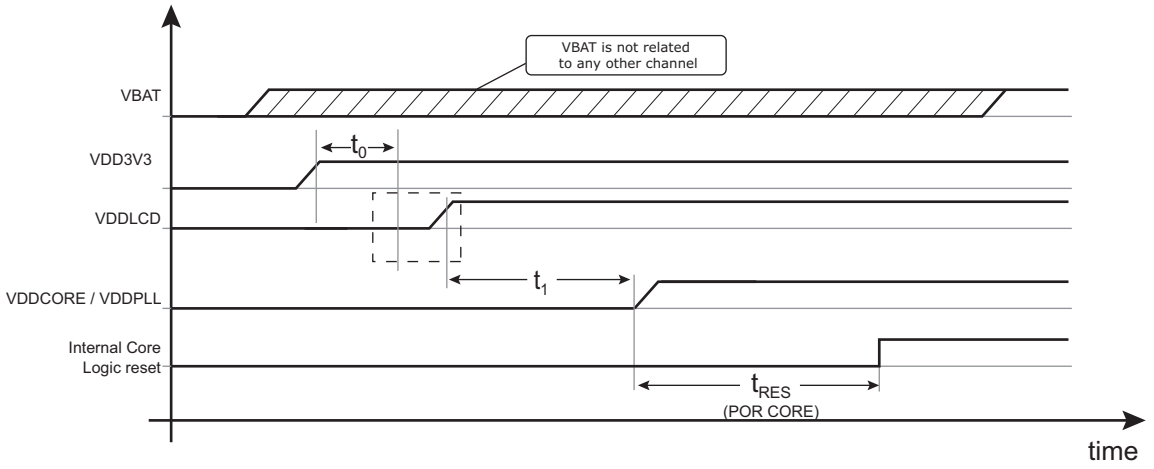
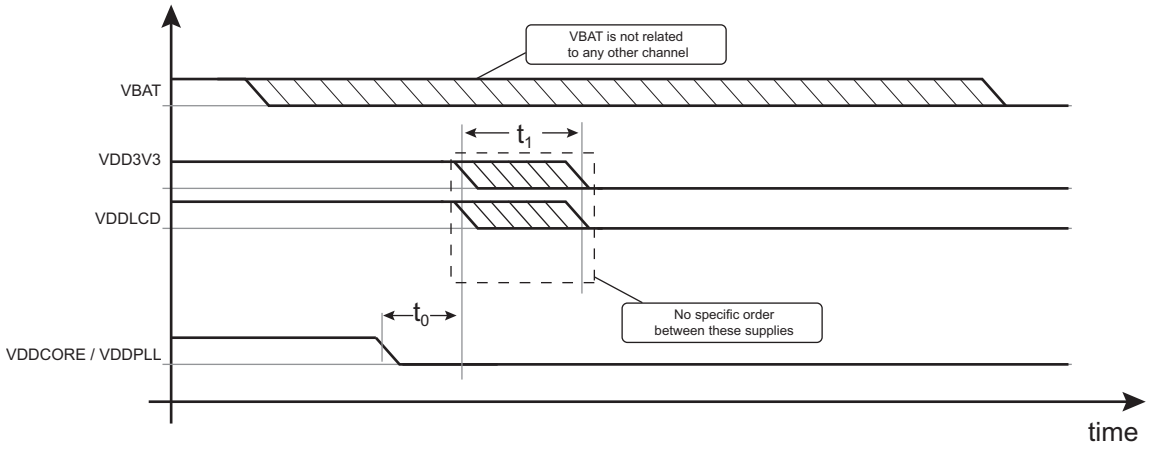


Table 48-7. Recommended Power Sequence at Power-Up

Symbol	Parameter	Conditions	Min	Max	Unit
t_0	VDDCORE delay	Delay from VDD3V3 established to VDDLCD turn-on	0	–	ms
t_1	Periphery supply timing	Delay from the last established periphery supply (VDDLCD) to VDDCORE turn-on	0	1	
t_{RES}	Reset time	See VDDCORE Power-On-Reset		–	

Figure 48-3. Recommended Power-Down Sequence with VDDCORE/VDDPLL Externally Supplied



© 2026 Microchip Technology Inc. and its subsidiaries

Table 48-8. Power-Down Timing Specifications

Symbol	Parameter	Conditions	Min	Max	Unit
t_0	VDDCORE delay	Delay from VDDCORE out of its operating range to the first of (VDDLCD, VDD3V3) out-of-range	0	–	ms
t_1	Periphery supply timing	Delay from the periphery supply out-of-range to the last one	–	10	

48.5. I/O Characteristics

Unless otherwise specified, V_{DD} refers to the voltage of the associated Power Rail of the I/O line, as defined in the table Pinout and Multiplexing, e.g., for PA2, V_{DD} refers to the voltage applied on VDD3V3.

48.5.1. I/O DC Characteristics

Table 48-9. Input DC Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
V_{IL}	Low-level input voltage(1)	FWUP, WKUP/TMP[2:0], JTAGSEL, TST	–	0.3 x VDDBU	V
		PIOA/B/C/D IOs	–	0.3 x VDD3V3	V
V_{IH}	High-level input voltage(1)	FWUP, WKUP/TMP[2:0], JTAGSEL, TST	0.7 x VDDBU	–	V
		PIOA/B/C/D IOs	0.7 x VDD3V3	–	
V_{hys}	Hysteresis voltage(1)	FWUP, WKUP/TMP[2:0], JTAGSEL, TST	350	–	mV
		PIOA/B/C/D IOs	260	–	
I_{IH}	Input-high leakage current(1)	No pull-up or pull-down; VIN=VDD; VDD3V3 Max, T_A = 85°C	–	200	nA
I_{IL}	Input-low leakage current(1)	No pull-up or pull-down; VIN=GND; VDD3V3 Max, T_A = 85°C	–	200	nA
R_{PULL}	Programmable pull-up or pull-down resistor	Digital input mode	80	120	kΩ
C_{IN}	Input capacitance(1)		–	4	pF

Note:

1. Simulation data.

Table 48-10. Output DC Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
V_{OL}	Low-level output voltage @ I_{OLMAX}	SHDN, RTCOUT0, PIOA/B/C/D IOs	–	0.2 x VDD	V
V_{OH}	High-level output voltage @ I_{OHMAX}	SHDN, RTCOUT0, PIOA/B/C/D IOs	0.8 x VDD	–	
I_{OL}	Low-level output current	SHDN, RTCOUT0	–	+0.03(1)	mA
		PIOA/B/C/D IOs	–	+11	
I_{OH}	High-level output current	SHDN, RTCOUT0	–	-0.03(1)	mA
		PIOA/B/C/D IOs	–	-11	

Note:

1. Total maximum current draw allowed.

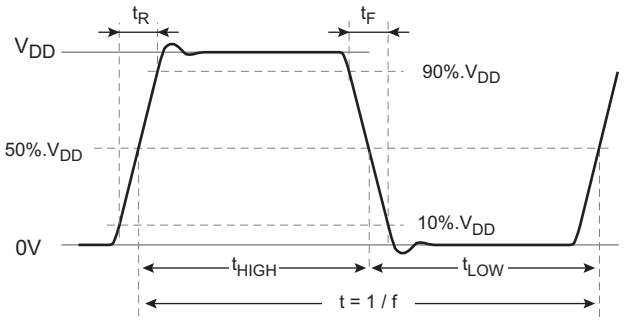
48.5.2. I/O AC Characteristics

48.5.2.1. Output Driver AC Characteristics

The timing definitions necessary to specify the maximum operating frequency of an output driver are given in the figure below.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 48-4. Timing Definitions of a Digital Output Signal



t: Period of the digital output signal

f = 1 / t: Frequency of the digital output signal

t_{HIGH}: Time during which the output waveform is greater than $V_{DD}/2$

$t_{LOW} = t - t_{HIGH}$: Time during which the output waveform is less than $V_{DD}/2$

$d = t_{HIGH} / t$: Output waveform duty cycle

In Output Driver AC Characteristics, Output Load = 10pF, 3V < VDD3V3 < 3.6V, the maximum operating frequency f_{MAX} guarantees that the driver's output waveform fulfills the following characteristics :

- $t_R < 0.75 / f_{MAX}$ and $t_F < 0.75 / f_{MAX}$
- d: the duty cycle of the output waveform is between 45% and 55%

The f_{MAX} parameter indicates the speed limit of an output driver across various operating conditions: supply voltage range, load capacitance, slew rate programming. The effective maximum output frequency of a specific output line may be limited by the peripheral that drives this line.

The table below gives the AC output characteristics of the output drivers in the following conditions:

- Output load: 10pF at the pin
- VDD3V3 range: 3V < VDD < 3.6V
- Four slew rate settings programmable in the PIO Controller user interface

Table 48-11. Output Driver AC Characteristics, Output Load = 10pF, 3V < VDD3V3 < 3.6V

Symbol	Parameter	I/O Type	Conditions			Unit
			Slew Rate	Min	Max	
t_R	Rise time (2, 3)	PIOA/B/C/D IOs	Fast	1.76	3.1	ns
			Medium Fast	4.1	9.5	
			Medium	6.3	14	
			Slow	8.6	19.6	
t_F	Fall time(2, 3)	PIOA/B/C/D IOs	Fast	1.88	3.5	ns
			Medium Fast	4.9	10	
			Medium	7.6	15.2	
			Slow	10.3	20.8	
f_{MAX}	Maximum frequency(1, 2, 3)	PIOA/B/C/D IOs	Fast	–	65	MHz
			Medium Fast	–	20	
			Medium	–	15	
			Slow	–	10	

© 2026 Microchip Technology Inc. and its subsidiaries

Notes:

1. f_{MAX} may be limited by the peripheral that drives the I/O line.
2. Measured between 10% and 90%.
3. Simulation data.

48.5.2.2. Input AC Characteristics

The table below provides the input characteristics of the I/O lines when configured as a digital input. In particular, these values apply when the XIN input is used as a clock input of the device (main crystal oscillator set in Bypass mode). They do not apply for the XIN32 input which is designed for slow signals with frequencies up to 100 kHz. Parameters V_{IL} and V_{IH} are defined in Output DC Characteristics.

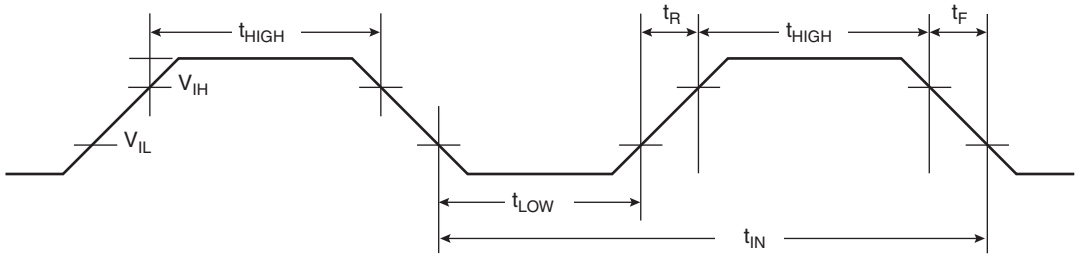
Table 48-12. Input AC Characteristics

Symbol	Parameter	Conditions	Min	Max	Units
f_{IN}	Input frequency(1)	–	10	50	MHz
t_{IN}	Input period	–	20	100	ns
Duty	Duty cycle	–	40	60	%
t_{HIGH}	Time at high level	–	8	60	ns
t_{LOW}	Time at low level	–	8	60	ns
t_R	Rise time	–	–	10	ns
t_F	Fall time	–	–	10	ns

Note:

1. The maximum input frequency may be limited by the peripheral receiving this signal.

Figure 48-5. Digital Input AC Characteristics



48.6. Digital Peripherals Characteristics

The timing specifications in this section are extracted from circuit simulations run in corner cases. Theses values are not tested in production.

© 2026 Microchip Technology Inc. and its subsidiaries

48.6.1. QSPI Characteristics

Figure 48-6. QSPI SPI Mode 0 or QSPI Serial Memory Mode, Single Data Rate, Mode 0

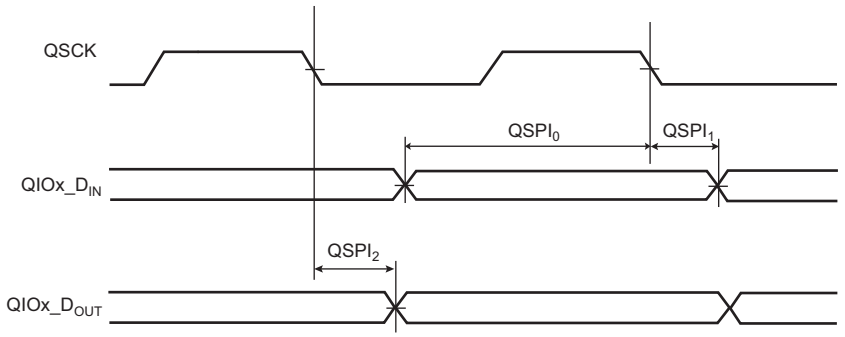


Figure 48-7. QSPI SPI Mode 3

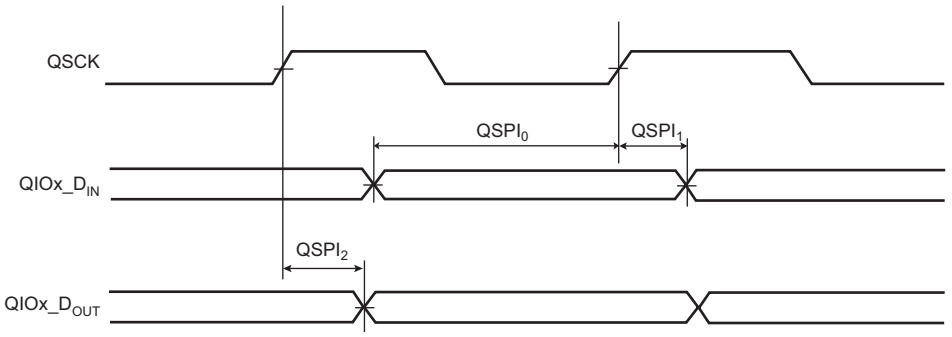
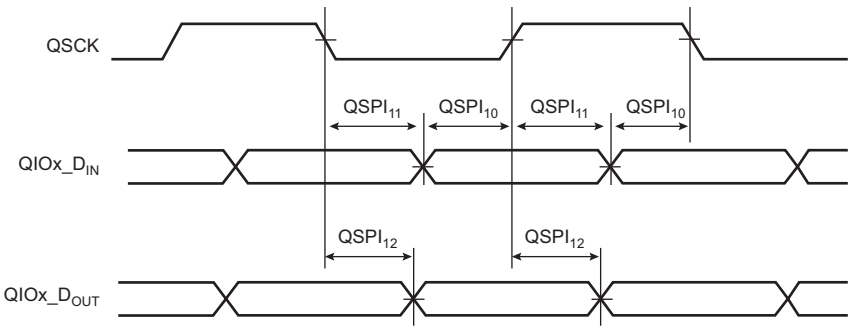


Figure 48-8. QSPI Serial Memory Mode, Double Data Rate, Mode 0



48.6.1.1. Maximum QSPI Frequency

The following formulas give maximum QSPI frequency in Read and Write modes.

48.6.1.1.1. Host Write in Single Data Rate Mode or QSPI SPI Mode

The maximum speed of the QSPI peripheral clock (MCK0DIV) is 100 MHz. However the transfer speed may be limited to a lower value if the output driver maximum frequency is less than 100 MHz (See I/O Characteristics).

48.6.1.1.2. Host Read in Single Data Rate Mode

$$f_{QSCK}\text{max} = \frac{1}{QSPI_0 + t_{VALID}}$$

t_{VALID} is the client time response to output data after detecting a QSCK edge.

For a QSPI SPI Mode client device with t_{VALID} (or t_V) = 12 ns, with $QSPI_0$ = 1.8ns, f_{QSCK}max = 72.4 MHz.

© 2026 Microchip Technology Inc. and its subsidiaries

For a QSPI Flash memory device with t_{VALID} (or t_V) = 6 ns, and $QSPI_0$ = 1.0ns, the formula returns a value of 128 MHz. In worst case conditions, this exceeds the maximum allowed frequency of the QSPI. In this case, the limitation is due to the controller and not to the client.

48.6.1.2. QSPI Timings

Timings are given in the following domains:

- 3.3V domain: VDD3V3 from 3.00V to 3.6V, external load = 10pF, Slew rate = Fast

Table 48-13. QSPI Timings in QSPI SPI Mode and Single Data Rate Mode

Symbol	Parameter	Conditions	Min	Max	Unit
$QSPI_0$	QIOx data in to QSCK rising edge (input setup time)	3.3V domain	1.8	–	ns
$QSPI_1$	QIOx data in to QSCK rising edge (input hold time)	3.3V domain	0.5	–	ns
$QSPI_2$	QSCK rising edge to QIOx data out valid	3.3V domain	0	2.5	ns

Table 48-14. QSPI Timings in Double Data Rate Mode

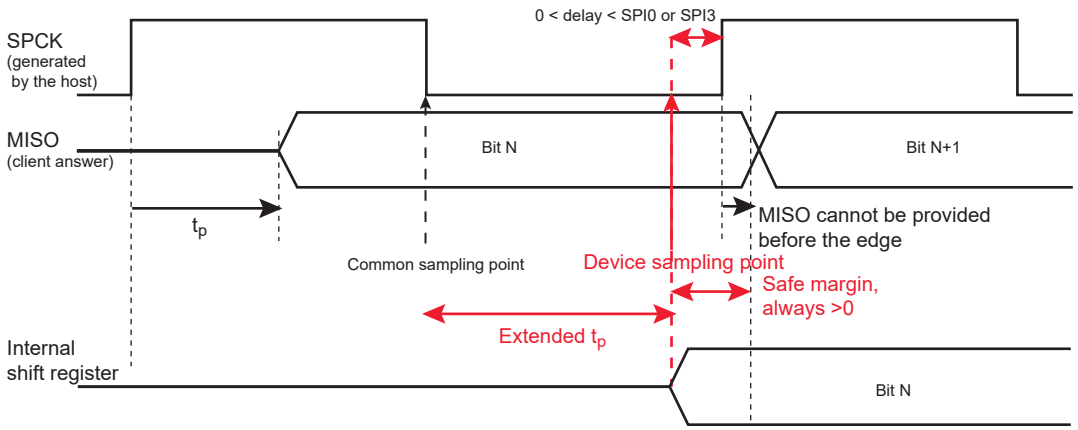
Symbol	Parameter	Conditions	Min	Max	Unit
$QSPI_{10}$	QIOx data in to QSCK edge (rising or falling, input setup time)	3.3V domain	1.7	–	ns
$QSPI_{11}$	QIOx data in to QSCK edge (rising or falling, input hold time)	3.3V domain	0.7	–	ns
$QSPI_{12}$	QSCK edge (rising or falling) to QIOx data out valid	3.3V domain	0	2.5	ns

48.6.2. FLEXCOM Characteristics

In Figure 48-10 and Figure 48-11 below, the MOSI line shifting edge is represented with a hold time equal to 0. However, it is important to note that for this device, the MISO line is sampled prior to the MOSI line shifting edge. As shown in Figure 48-9, the device sampling point extends the propagation delay (t_p) for client and routing delays to more than half the SPI clock period, whereas the common sampling point allows only less than half the SPI clock period.

As an example, an SPI client working in Mode 0 can be safely driven if the SPI host is configured in Mode 0.

Figure 48-9. FLEXCOM in SPI Mode: MISO Capture in Host Mode



© 2026 Microchip Technology Inc. and its subsidiaries

Figure 48-10. FLEXCOM in SPI Host Mode 1 and 2

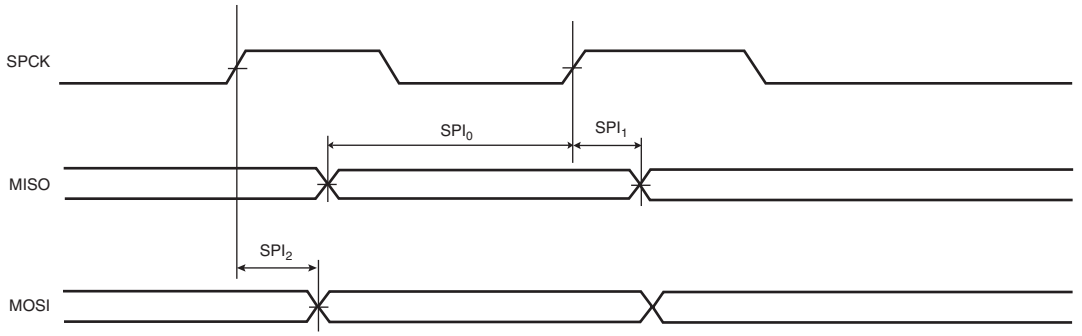


Figure 48-11. FLEXCOM in SPI Host Mode 0 and 3

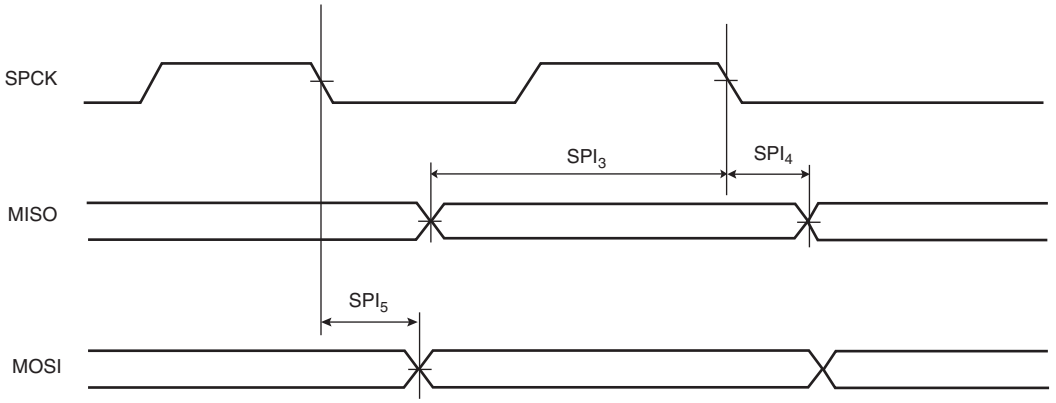
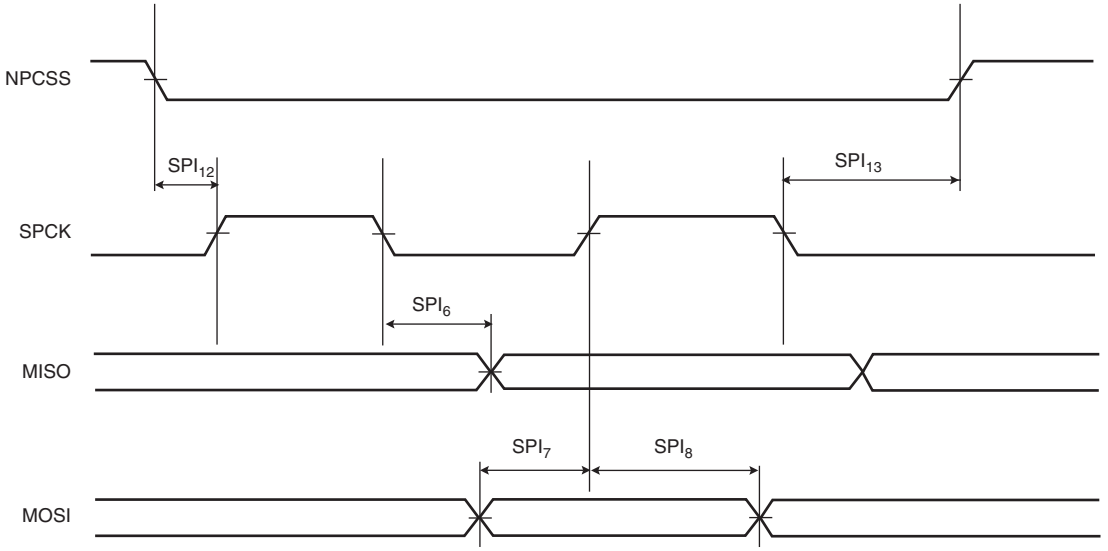


Figure 48-12. FLEXCOM in SPI Client Mode 0 and 3



© 2026 Microchip Technology Inc. and its subsidiaries

Figure 48-13. FLEXCOM in SPI Client Mode 1 and 2

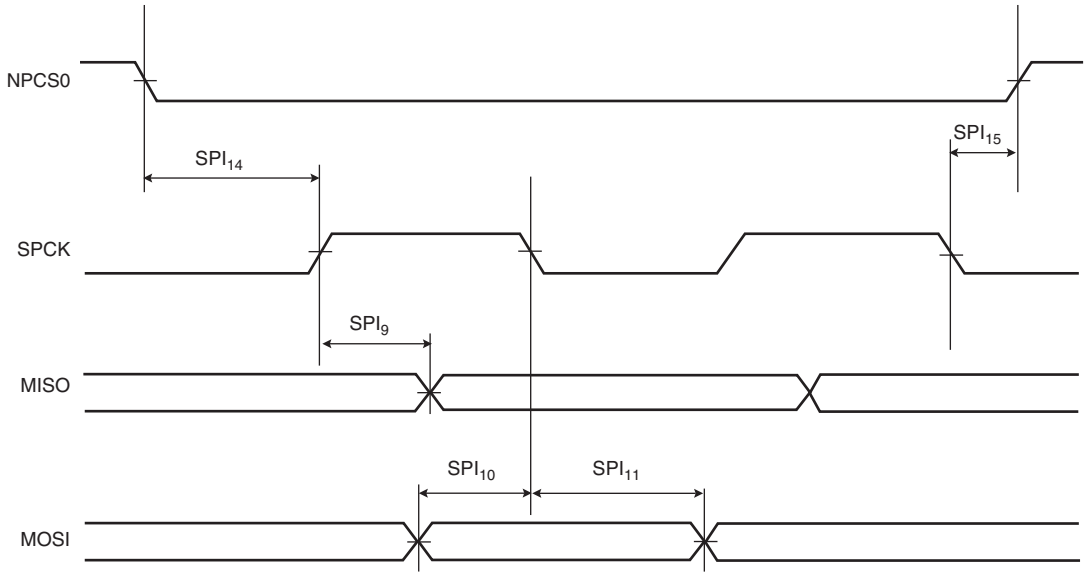
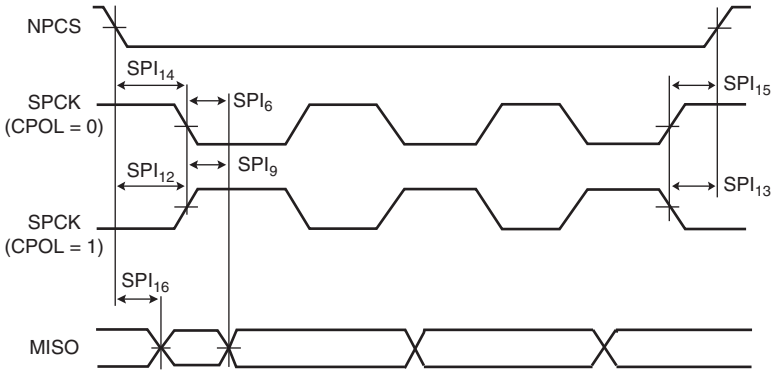


Figure 48-14. FLEXCOM in SPI Client - NPCS Timings



48.6.2.1. Maximum FLEXCOM SPI Frequency

The following formulas give the maximum SPI frequency in host Read and Write modes and in client Read and Write modes.

48.6.2.1.1. Host Write Mode

The SPI sends data to a client device only, e.g. an LCD. The limit is given by SPI_2 (or SPI_5) timing. Since it gives a maximum frequency above the output driver maximum frequency (see I/O AC Characteristics), the max SPI frequency is the one from the output driver.

48.6.2.1.2. Host Read Mode

$$f_{SPCK}\text{max} = \frac{1}{\text{SPI}_0(\text{or SPI}_3) + t_{\text{valid}}}$$

t_{valid} is the client time response to output data after detecting an SPCK edge.

For a nonvolatile memory with t_{VALID} (or t_v) = 5 ns, using SPI_3, f_{SPCK}max = 41.6 MHz.

48.6.2.1.3. Client Read Mode

In client mode, SPCK is the input clock for the SPI. The max SPCK frequency is given by setup and hold timings SPI_7/SPI_8(or SPI_{10}/SPI_{11}). Since this gives a frequency well above the output driver maximum frequency, the limit in client Read mode is given by the SPCK output driver.

© 2026 Microchip Technology Inc. and its subsidiaries

48.6.2.1.4. Client Write Mode

$$f_{SPCK}\max = \frac{1}{2x(SPI_{6max}(\text{or } SPI_{9max}) + t_{setup})}$$

t_{setup} is the setup time from the host before sampling data.

48.6.2.2. FLEXCOM SPI Timings

The timings in the table below are given in the following domains:

- 3.3V domain: VDD3V3 from 3.00V to 3.6V, maximum external capacitor = 10 pF, Slew rate = FAST

Table 48-15. FLEXCOM SPI Timings

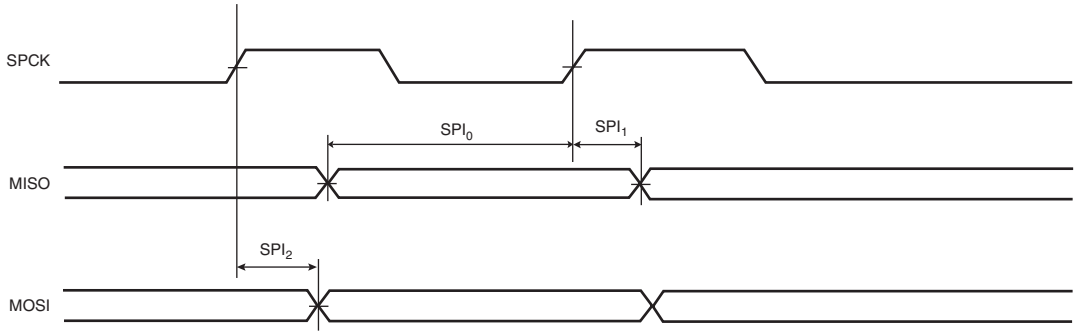
Symbol	Parameter	Conditions	Min	Max	Unit
Host Mode					
SPI_0	MISO setup time before SPCK rises	3.3V domain	19	–	ns
SPI_1	MISO hold time after SPCK rises	3.3V domain	0	–	ns
SPI_2	SPCK rising to MOSI delay	3.3V domain	0	2.8	ns
SPI_3	MISO setup time before SPCK falls	3.3V domain	19	–	ns
SPI_4	MISO hold time after SPCK falls	3.3V domain	0	–	ns
SPI_5	SPCK falling to MOSI delay	3.3V domain	0	2.8	ns
Client Mode					
SPI_6	SPCK falling to MISO delay	3.3V domain	6.2	18	ns
SPI_7	MOSI setup time before SPCK rises	3.3V domain	2.3	–	ns
SPI_8	MOSI hold time after SPCK rises	3.3V domain	1.4	–	ns
SPI_9	SPCK rising to MISO delay	3.3V domain	6.3	18	ns
SPI_{10}	MOSI setup time before SPCK falls (client)	3.3V domain	2.3	–	ns
SPI_{11}	MOSI hold time after SPCK falls (client)	3.3V domain	1.4	–	ns
SPI_{12}	NPCS setup to SPCK rising (client)	3.3V domain	6.3	–	ns
SPI_{13}	NPCS hold after SPCK falling (client)	3.3V domain	0	–	ns
SPI_{14}	NPCS setup to SPCK falling (client)	3.3V domain	6.5	–	ns
SPI_{15}	NPCS hold after SPCK falling (client)	3.3V domain	0.4	–	ns
SPI_{16}	NPCS falling to MISO valid (client)	3.3V domain	21	–	ns

48.6.3. FLEXCOM USART Timings in Asynchronous Mode

The maximum baud rate that can be achieved is MCK0DIV/8, if the FLEX_US_MR.OVER = 1.

48.6.4. MCSPI Characteristics

Figure 48-15. MCSPI in SPI Legacy Host Mode 1 and 2



© 2026 Microchip Technology Inc. and its subsidiaries

Figure 48-16. MCSPI in SPI Legacy Host Mode 0 and 3

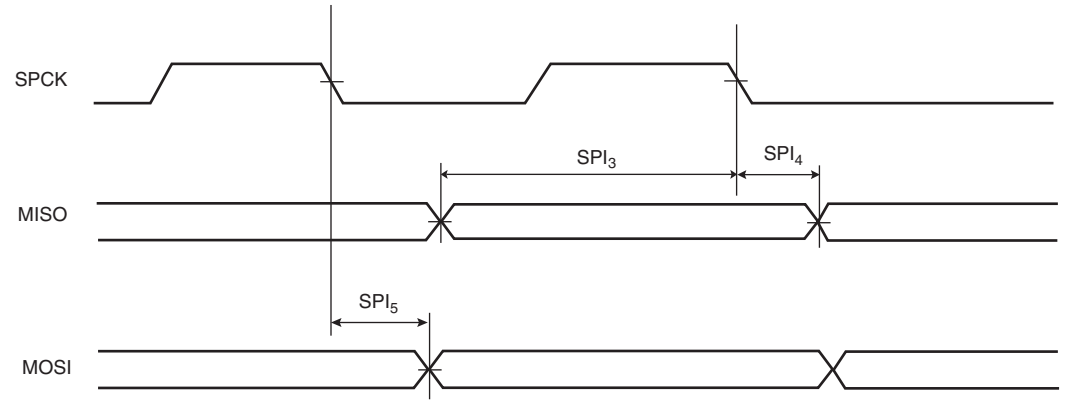


Figure 48-17. MCSPI in SPI Legacy Client Mode 0 and 3

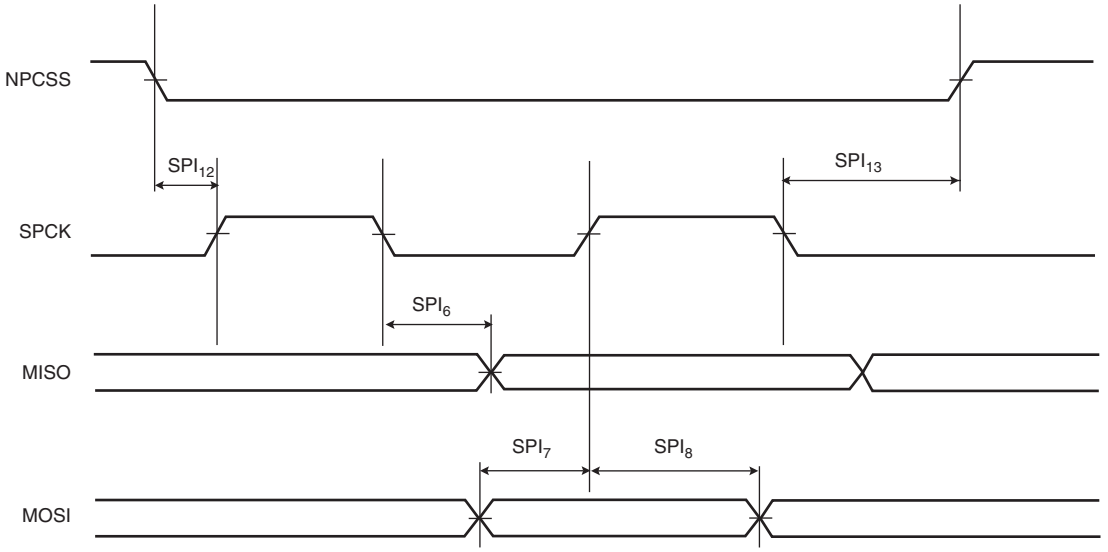
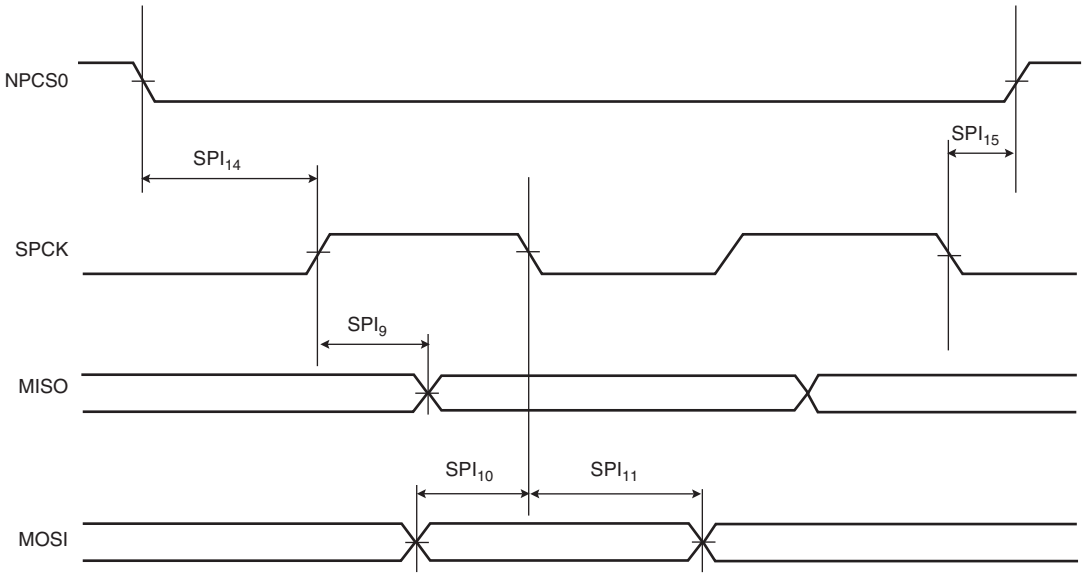
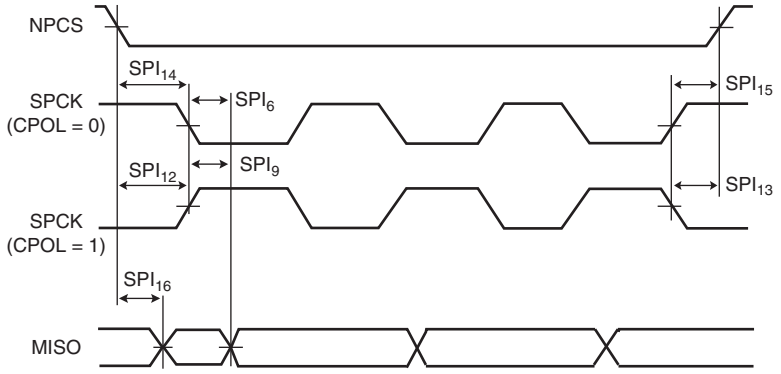


Figure 48-18. MCSPI in SPI Legacy Client Mode 1 and 2



© 2026 Microchip Technology Inc. and its subsidiaries

Figure 48-19. MCSPI in SPI Legacy Mode Client - NPCS Timings



48.6.4.1. Maximum MCSPI Frequency in SPI Legacy Mode

The following formulas give the maximum SPI frequency in host Read and Write modes and in client Read and Write modes.

48.6.4.1.1. Host Write Mode

The MCSPI sends data to a client device only, e.g. an LCD. The limit is given by SPI_2 (or SPI_5) timing. Since it gives a maximum frequency above the output driver maximum frequency (see I/O AC Characteristics), the max SPI frequency is the one from the output driver.

48.6.4.1.2. Host Read Mode

$$f_{SPCK}\text{max} = \frac{1}{\text{SPI}_0(\text{or SPI}_3) + t_{\text{valid}}}$$

t_{valid} is the client time response to output data after detecting an SPCK edge.

For a nonvolatile memory with t_{VALID} (or t_v) = 5 ns, using SPI_3, f_{SPCK}max = 50 MHz.

48.6.4.1.3. Client Read Mode

In client mode, SPCK is the input clock for the SPI. The max SPCK frequency is given by setup and hold timings SPI_7/SPI_8 (or SPI_{10}/SPI_{11}). Since this gives a frequency well above the output driver maximum frequency, the limit in client Read mode is given by the SPCK output driver.

48.6.4.1.4. Client Write Mode

$$f_{SPCK}\text{max} = \frac{1}{2x(SPI_{6\text{max}}(\text{or SPI}_{9\text{max}}) + t_{\text{setup}})}$$

t_{setup} is the setup time from the host before sampling data.

48.6.4.2. Maximum MCSPI Frequency in SPI Two-Wire Mode

f_{SPCK}max = 27 MHz

48.6.4.3. MCSPI Timings

The timings in the table below are given in the following domains:

- 3.3V domain: VDD3V3 from 3.00V to 3.6V, maximum external capacitor = 10 pF, Slew Rate = FAST

Table 48-16. MCSPI SPI Timings in SPI Legacy Mode

Symbol	Parameter	Conditions	Min	Max	Unit
Host Mode					
SPI_0	MISO setup time before SPCK rises	3.3V domain	19	–	ns
SPI_1	MISO hold time after SPCK rises	3.3V domain	0	–	ns
SPI_2	SPCK rising to MOSI delay	3.3V domain	0	0	ns
SPI_3	MISO setup time before SPCK falls	3.3V domain	15	–	ns

© 2026 Microchip Technology Inc. and its subsidiaries

Table 48-16. MCSPI SPI Timings in SPI Legacy Mode (continued)

Symbol	Parameter	Conditions	Min	Max	Unit
SPI_4	MISO hold time after SPCK falls	3.3V domain	0	–	ns
SPI_5	SPCK falling to MOSI delay	3.3V domain	0	3.7	ns
Client Mode					
SPI_6	SPCK falling to MISO delay	3.3V domain	0	12.8	ns
SPI_7	MOSI setup time before SPCK rises	3.3V domain	–	1.5	ns
SPI_8	MOSI hold time after SPCK rises	3.3V domain	0.7	–	ns
SPI_9	SPCK rising to MISO delay	3.3V domain	5.6	13.3	ns
SPI_{10}	MOSI setup time before SPCK falls (client)	3.3V domain	–	1.5	ns
SPI_{11}	MOSI hold time after SPCK falls (client)	3.3V domain	0.6	–	ns
SPI_{12}	NPCS setup to SPCK rising (client)	3.3V domain	36.8	–	ns
SPI_{13}	NPCS hold after SPCK falling (client)	3.3V domain	35.8	–	ns
SPI_{14}	NPCS setup to SPCK falling (client)	3.3V domain	37.2	–	ns
SPI_{15}	NPCS hold after SPCK falling (client)	3.3V domain	36.2	–	ns
SPI_{16}	NPCS falling to MISO valid (client)	3.3V domain	15.5	–	ns

48.7. Embedded Analog Peripherals Characteristics

48.7.1. Core Voltage Regulator

Table 48-17. Core Voltage Regulator Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
V_{DD3V3}	Supply voltage range (VDD3V3)	–	2.25	3.6	V
V_{DDOUT}	DC output voltage[4]	–	1.08	1.23	V
I_{INRUSH}	Inrush current [3, 5]	–	–	200	mA
C_{IN}	Input decoupling capacitor[1]	–	2.2	–	µF
C_{OUT}	Output capacitor[2]	Capacitance ESR	1.8 5	2.8 50	µF mΩ
t_{ON}	Turn-on time[5]	C_{OUT}= 2.2µF	–	200	µs

Notes:

1. An X5R or X7R ceramic capacitor must be connected between VDD3V3 and the closest GND pin of the device. This decoupling capacitor is mandatory to reduce inrush current and to improve transient response and noise rejection.
2. An external X5R or X7R capacitor must be connected between VDDOUT and the device's closest GND pin.
3. Current needed to charge the external bypass/decoupling capacitor network.
4. This regulator is designed to supply internal circuitry only. No external load authorized.
5. Simulation data.

48.7.2. LCD Voltage Regulator and LCD Output Buffers

The LCD Voltage Regulator is a complete solution to drive an LCD display. It integrates a low-power LDO regulator with programmable output voltage and buffers to drive the LCD lines. A capacitor is required at the LDO regulator output (VDDLCD). This regulator can be set in Active (Normal) mode, in Bypass mode (HiZ mode), or in OFF mode.

- In Normal mode, the VDDLCD LDO regulator output can be selected from 2.4V to 3.5V using LCDVROUT bits in the Supply Controller Mode Register (SUPC_MR), with the conditions:

© 2026 Microchip Technology Inc. and its subsidiaries

  - VDDLCD ≤ VDD3V3 - 100 mV.
- In Bypass mode (HiZ mode), the VDDLCD is set in high impedance (through the LCDMODE bits in SUPC_MR register), and can be forced externally. This mode can be used to save the LDO operating current.
- In OFF mode, the VDDLCD output is pulled down.

Important: When using an external or the internal voltage regulator, VDD3V3 must be still supplied with the conditions: 2.4V≤ VDDLCD ≤ VDD3V3 and VDD3V3 ≥ 2.5V.

Table 48-18. LCD Voltage Regulator Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
V_{VDD3V3}	Input supply voltage range	–	2.5	3.6	V
V_{DDLCD}	Programmable output range	Refer to Table 48-19			V
V_{ACC}	V_{DDLCD} accuracy	With respect to programmed output range	-50	+50	mV
I_{VDD3V3}	Current consumption	–	–	7	µA
C_{OUT}	Output capacitor on VDDLCD[1]		1	10	µF
		ESR for 1 µF capacitor	5	20	mΩ
		ESR for 10 µF capacitor	1	10	mΩ
t_{ON}	Start-up time[1]	C_{OUT}= 1 µF, VDDLCD max	–	2.5	ms

Note:

1. Simulation data.

Table 48-19. VDDLCD Voltage Selection at VDD3V3 = 3.6V

VROUT	0	1	2	3	4	5	6	7	8	9	10	11	12	13	12	13
V_{DDLCD}	2.375	2.45	2.525	2.60	2.675	2.75	2.825	2.90	2.975	3.05	3.125	3.20	3.275	3.35	3.425	3.50

Table 48-20. LCD Analog Buffers Bias Generation Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
I_{DDIN}	Current consumption (VDD3V3)	Buffered	–	52	µA
		Unbuffered	–	5	
Z_{OUT}	Buffer output impedance[1]	Buffered	500	1300	Ω
		Unbuffered	300k	1Meg	
C_{LOAD}	Capacitive output load[1]	–	10p	50n	F

Note:

1. Simulation data.

48.7.3. VDDCORE Supply Monitor

Table 48-21. DC Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
V_{IT-}	Negative-going input threshold voltage (VDDCORE) [1]	–	1.02	1.07	V
V_{hys}	Hysteresis voltage[2]	V_{IT+} - V_{IT-}	25	35	mV
t_{DET}	Detection time[2]	–	–	300	ns
t_{START}	Start-up time[2]	From disabled state to enabled state	–	300	µs
I_{VDD3V3}	Current consumption (VDD3V3)[2]	Supply Monitor enabled	–	20	µA

© 2026 Microchip Technology Inc. and its subsidiaries

Notes:

1. Operation of the device is granted down to the V_{IT-} threshold.
2. Simulation data.

48.7.4. VDDCORE Power-On-Reset

Table 48-22. DC Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
V_{IT-}	Negative-going input threshold voltage (VDDCORE)	–	0.75	0.94	V
V_{IT+}	Positive-going input threshold voltage (VDDCORE)	–	0.88	0.985	V
V_{hys}	Hysteresis voltage(1)	–	10	160	mV
t_{RES}	Reset time(1)	–	–	5	ms

Note:

1. Simulation data.

48.7.5. VDD3V3 Supply Monitor

Table 48-23. VDD3V3 Supply Monitor

Symbol	Parameter	Conditions	Min	Max	Unit
V_{IT-}	Programmable range of negative-going input threshold voltage (VDD3V3)	–	2.4	3.45	V
V_{ACC}	V_{IT-} Accuracy	With respect to programmed threshold value(4)	-2	+2	%
V_{hys}	Hysteresis(2, 3)	–	20	50	mV
I_{DDON}	Current consumption(1, 3)	ON with a 100% duty cycle.	–	20	µA
t_{DET}	Detection time(3)		–	300	ns
t_{ON}	Start-up time(3)	From OFF to ON	–	300	µs

Notes:

1. The average current consumption can be reduced by using the supply monitor in Sampling mode. Refer to Supply Controller (SUPC).
2. $V_{hys}= V_{IT+} - V_{IT-}$. V_{IT+} is the positive-going input threshold voltage (VDD3V3).
3. Simulation data.
4. See the following table: VDD3V3 Supply Monitor V_{IT-} Threshold Selection.

Table 48-24. VDD3V3 Supply Monitor V_{IT-} Threshold Selection

Digital Code	Threshold (V)
0	2.40
1	2.47
2	2.54
3	2.61
4	2.68
5	2.75
6	2.82
7	2.89
8	2.96
9	3.03
10	3.10

© 2026 Microchip Technology Inc. and its subsidiaries

Table 48-24. VDD3V3 Supply Monitor V_{IT-} Threshold Selection (continued)	
Digital Code	Threshold (V)
11	3.17
12	3.24
13	3.31
14	3.38
15	3.45

48.7.6. VBAT Supply Monitor

Table 48-25. DC Characteristics[1]

Symbol	Parameter	Conditions	Min	Max	Unit
V_{IT+}	Positive-going input threshold voltage (VBAT)	–	1.45	1.595	V
V_{IT-}	Negative-going input threshold voltage (VBAT)	–	1.41	1.56	V
V_{hys}	Hysteresis[2]	–	25	55	mV
t_{res}	Reset time-out period[2]	–	60	280	µs

Notes:

1. The product is granted to be functional at V_{IT-}. In this table, V_{IT-} corresponds to the negative-going input threshold voltage. Operation of the device backup section, and in particular of the Slow RC oscillator, is granted down to the VBAT Supply Monitor V_{IT-} threshold.
2. Simulation data.

48.7.7. VDD3V3 Power-On-Reset

Table 48-26. DC Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
V_{IT+}	Positive-going input threshold voltage (VDD3V3)	–	2	2.24	V
V_{IT-}	Negative-going input threshold voltage (VDD3V3)	–	1.9	2.15	V
V_{hys}	Hysteresis voltage (V_{IT+} - V_{IT-})	–	70	160	mV
t_{res}	Reset time-out period[1]	–	10	80	µs

Note:

1. Simulation data.

48.7.8. Power Switch

The power switch has its two inputs monitored by the VBAT Supply Monitor on the VBAT pin and by the VDD3V3 Power-On-Reset on VDD3V3 pin. Refer to Power Switch for more details.

48.7.9. Slow RC Oscillator

Table 48-27. Slow RC Oscillator Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
f_{SLOWRC}	Frequency accuracy	VDDBU[1, 2] = 3V T_A= 0 to +50°C	30.4	33.6	kHz
		VDDBU[1, 2] = 1.65V to 3.6V T_A= -40 to +85°C	29.1	35.5	

© 2026 Microchip Technology Inc. and its subsidiaries

Notes:

1. VDDBU is the output of the power switch (VBAT or VDD3V3).
2. Functional down to V_{IT-} of the VBAT Supply Monitor.

48.7.10. Main RC Oscillator

Table 48-28. Main RC Oscillator Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
I_{DD}	Current consumption on VDDCORE(1)	–	–	320	µA
t_{START}	Start-up time(1)	–	–	15	µs
f_O	Nominal output frequency	–	12		MHz
f_{ACC}	Output frequency accuracy	0°C < T_A < +70°C(1)	-5	+2	%
		-40°C < T_A < +85°C, over VDDCORE range	-8	+2	
f_{STEP}	Frequency trimming step size(1)	–	40	150	kHz
t_{f_SET}	Output frequency settling time	On frequency change (MOSCRCF)	–	2	µs

Note:

1. Simulation data.

48.7.11. Crystal Oscillators Design Considerations

When choosing a crystal for the 32.768 kHz Crystal Oscillator or for the Main Crystal Oscillator, several parameters must be taken into account. Important parameters are as follows:

Crystal Load CapacitanceC_L or $C_{CRYSTAL}$
The total capacitance loading the crystal, including the oscillator's internal parasitics and the PCB parasitics, must match the load capacitance for which the crystal's frequency is specified. Any mismatch in the load capacitance with respect to the crystal specification leads to inaccurate oscillation frequency.

Crystal Drive Level
Use only crystals with specified drive levels greater than the specified MCU oscillator drive level. Applications that do not respect this criterion may damage the crystal.

Crystal Equivalent Series Resistance (ESR)
Use only crystals with a specified ESR lower than the specified MCU oscillator ESR. In applications where this criterion is not respected, the crystal oscillator may not start.

Crystal Shunt Capacitance (C_S or C_0)
Use only crystals with a specified shunt capacitance lower than the specified MCU oscillator shunt capacitance. In applications where this criterion is not respected, the crystal oscillator may not start.

PCB Layout Considerations
To minimize inductive and capacitive parasitics associated with XIN, XOUT, XIN32, XOUT32 nets, it is recommended to route them as short as possible. It is also of prime importance to keep those nets away from noisy switching signals (clock, data, PWM, etc.). A good practice is to shield them with a quiet ground net to avoid coupling to neighboring signals.

48.7.12. Slow Clock Crystal Oscillator

Table 48-29. 32.768 kHz Crystal Oscillator Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
V_{VDDBU}	Supply voltage (VDDBU)(1, 3)	–	V_{IT-}	3.6	V
I_{VDDBU}	Current consumption (VDDBU)(2)	–	–	750	nA

© 2026 Microchip Technology Inc. and its subsidiaries

Table 48-29. 32.768 kHz Crystal Oscillator Characteristics (continued)

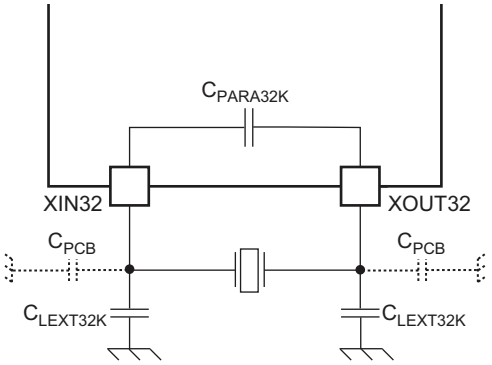
Symbol	Parameter	Conditions	Min	Max	Unit
$t_{SWITCH\_TD\_SLCK}$	Slow clock source switching time after crystal oscillator enable to crystal oscillator ready[4]			[4]	s
f_{OSC}	Operating frequency	–		32.768	kHz
Duty	Duty cycle	–	40	60	%
$C_{PARA32K}$	Internal parasitic capacitance[2]	Between XIN32 and XOUT32	–	1	pF

Notes:

1. VDDBU is the output of the power switch connected to VBAT or VDD3V3.
2. Simulation data.
3. Functional down to V_{IT-} of the VBAT Supply Monitor.
4. This time is defined as 98400 Slow RC Periods. Oscillator Enable and Status bits in Supply Controller registers.

The oscillator supports a bypass mode. See Input AC Characteristics.

Figure 48-20. 32.768 kHz Crystal Oscillator



$C_{LEXT32K} = 2 \times (C_{CRYSTAL} - C_{PARA32K} - C_{PCB} / 2)$

where C_{PCB} is the ground referenced parasitic capacitance of the printed circuit board (PCB) on XIN32 and XOUT32 tracks. As an example, if the crystal is specified for a 12.5 pF load, with C_{PCB} = 1 pF (on XIN32 and on XOUT32), $C_{LEXT32K}$ = 2 x (12.5 - 1 - 0.5) = 22 pF.

The table below summarizes recommendations for crystal selection.

Table 48-30. Recommended 32.768kHz Crystal Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
$C_{CRYSTAL}$	Crystal load capacitance	As specified by the crystal manufacturer	[1]	[1]	pF
ESR	Equivalent Series Resistor (R_S)	–	–	[1]	kΩ
C_{SHUNT}	Shunt capacitance	–	–	[1]	pF
P_{ON}	Drive level	–	15	–	µW

Note:

1. See Table 48-31.

© 2026 Microchip Technology Inc. and its subsidiaries

Table 48-31. ESR, C_{SHUNT} and C_{LOAD} Selection

ESR (kΩ)	C_{SHUNT} (pF)	$C_{CRYSTAL}$ (pF)	
		Min	Max
50	1	5	14
	2	5.5	13
	3	6.5	12
	4	8	10.5
70	1	5	11.5
	2	7	10
80	1	5.5	10
	2	8	8
90	1	6	9.5
100	1	6.5	8.5

48.7.13. Main Crystal Oscillator

Table 48-32. Main Crystal Oscillator Characteristics

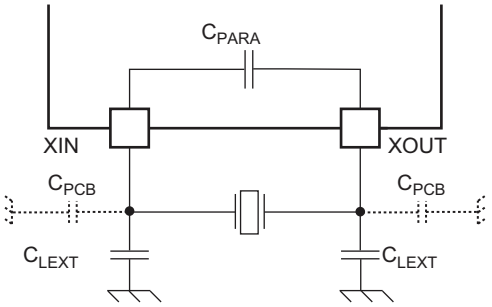
Symbol	Parameter	Conditions	Min	Max	Unit
$V_{VDDCORE}$	Supply voltage (VDDCORE)(2)	–	V_{IT-}	(3)	V
$I_{VDDCORE}$	Current consumption (VDDCORE)(1)	–	–	500	µA
t_{START}	Start-up time(1)	(4)	–	3	ms
f_{OSC}	Operating frequency	–	12	48	MHz
Duty	Duty cycle	–	40	60	%
C_{PARA}	Internal parasitic capacitance(1)	Between XIN and XOUT	–	1	pF

Notes:

1. Simulation data.
2. Functional down to V_{IT-} of the VDDCORE Supply Monitor.
3. VDDCORE max.
4. For all 12 MHz to 48 MHZ crystals complying with the Recommended Crystal Characteristics given in tables Table 48-33, Table 48-34 and Table 48-35.

The oscillator supports a Bypass mode. See Input AC Characteristics.

Figure 48-21. Main Crystal Oscillator Schematic



$C_{LEXT} = 2 \times (C_{CRYSTAL} - C_{PARA} - C_{PCB} / 2)$, where C_{PCB} is the ground referenced parasitic capacitance of the printed circuit board (PCB) on XIN and XOUT tracks. As an example, if the crystal is specified for a 9 pF load, with C_{PCB} = 1 pF (on XIN and on XOUT), C_{LEXT} = 2 x (9 - 1 - 0.5) = 15 pF.

© 2026 Microchip Technology Inc. and its subsidiaries

The table below summarizes recommendations for crystal selection. Three sets of crystal characteristics are supported with this oscillator corresponding to the three operating frequency ranges.

Table 48-33. Recommended Crystal Characteristics (Set 1)

Symbol	Parameter	Conditions	Min	Max	Unit
f_0	Fundamental frequency	–	12	24	MHz
$C_{CRYSTAL}$	Crystal load capacitance	As specified by the crystal manufacturer	7.5	14.5	pF
ESR	Equivalent Series Resistor (R_S)	–	–	120	Ω
C_{SHUNT}	Shunt capacitance	–	–	5	pF
P_{ON}	Drive level	–	40	–	µW

Table 48-34. Recommended Crystal Characteristics (Set 2)

Symbol	Parameter	Conditions	Min	Max	Unit
f_0	Fundamental frequency	–	24	32	MHz
$C_{CRYSTAL}$	Crystal load capacitance	As specified by the crystal manufacturer	7.5	9.4	pF
ESR	Equivalent Series Resistor (R_S)	–	–	100	Ω
C_{SHUNT}	Shunt capacitance	–	–	5	pF
P_{ON}	Drive level	–	50	–	µW

Table 48-35. Recommended Crystal Characteristics (Set 3)

Symbol	Parameter	Conditions	Min	Max	Unit
f_0	Fundamental frequency	–	32	48	MHz
$C_{CRYSTAL}$	Crystal load capacitance	As specified by the crystal manufacturer	4	5.3	pF
ESR	Equivalent Series Resistor (R_S)	–	–	100	Ω
C_{SHUNT}	Shunt capacitance	–	–	3	pF
P_{ON}	Drive level	–	50	–	µW

48.7.14. PLLA Characteristics

Table 48-36. PLLA Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
V_{DDPLL}	Supply voltage range	–	1.04	1.21	V
I_{VDDPLL}	Current consumption(3)		–	3	µA/MHz
f_{IN}	Input frequency	–	32.768		kHz
f_{PLLACK}	VCO frequency range(1)	–	380	600	MHz
t_{LOCK}	Lock time(3)	–	–	10	ms
t_{START}	Start-up time(3)	–	–	5	ms

Notes:

1. Internal frequency range of the PLLA. Two output clocks are provided. PLLACK0 and PLLACK1 for core and peripherals frequency selection. Refer to section Power Management Controller (PMC).
2. Slow clock crystal oscillator output allowed only.
3. Simulation data.

48.7.15. PLLB and PLLC Characteristics

Table 48-37. PLLB and PLLC Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
V_{DDPLL}	Supply voltage range	–	1.04	1.21	V

© 2026 Microchip Technology Inc. and its subsidiaries

Table 48-37. PLLB and PLLC Characteristics (continued)

Symbol	Parameter	Conditions	Min	Max	Unit
I_{VDDPLL}	Current consumption(3)		–	5	µA/MHz
f_{IN}	Input frequency range	–	5	50	MHz
f_{PLLACK}	VCO frequency range PLLBCK/PLLCCK	–	300	600	MHz
t_{LOCK}	Lock time(3)	–	–	150	µs
t_{START}	Start-up time(3)	–	–	50	µs

Notes:

1. Internal frequency range of the PLLB. Two output clocks are provided. PLLBCK and PLLCCK for core and peripherals frequency selection. Refer to section Power Management Controller (PMC).
2. Slow clock crystal oscillator output allowed only.
3. Simulation data.

48.7.16. Temperature Sensor Characteristics

The temperature sensor provides an output voltage (V_{TEMP}) that is proportional to absolute temperature (PTAT). This voltage can be measured through the channel number 7 of the 12-bit ADC. Improvement of the raw performance of the temperature sensor acquisition can be achieved by performing a single temperature point calibration to remove the initial inaccuracies (V_{TEMP} and ADC offsets).

Table 48-38. Temperature Sensor Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
V_{DD3V3}	Supply voltage range (VDD3V3)	–	2.5	3.6	V
$V_{VREFTEMP}$	Bandgap output voltage	T_A = 25° C	1.19	1.21	V
$dV_{VREFTEMP}$	Bandgap output voltage drift with temperature(3)	$dV_{VREFTEMP} = V_{REFTEMP}(T_A) - V_{VREFTEMP}$ Over T_A range [-40°C to +85°C]	-25	+25	mV
V_{TEMP}	Output voltage	T_A = 25° C	0.585	0.665	V
t_S	V_{TEMP} settling time	When V_{TEMP} is sampled by the 12-bit ADC, the required track time to ensure 1°C accurate settling	–	10	µs
T_{ACC}	Temperature reading accuracy(1, 2, 3)	After offset calibration Over T_A range [-40°C to +85°C]	-5	+5	°C
		After offset calibration Over T_A range [-20°C to +70°C]	-4	+4	°C
t_{ON}	Start-up time(3)	–	–	12	µs

Notes:

1. Does not include errors due to A/D conversion process.
2. Using the output voltage sensitivity to temperature dV_{TEMP}/dT = 2.07 mV / °C.
3. Simulation data.

48.7.17. Optical UART RX Transceiver Characteristics

The following table gives the description of the optical link transceiver for electrically isolated serial communication with hand-held equipment, such as calibrators compliant with standards ANSI-C12.18 or IEC62056-21 (only available on UART). The Analog Comparator embedded in the transceiver can also used in standalone. Refer to section Universal Asynchronous Receiver Transmitter (UART).

© 2026 Microchip Technology Inc. and its subsidiaries

Table 48-39. Transceiver Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
VDD3V3	Supply voltage range (VDD3V3)	–	2.7	3.6	V
V_{TH}	Comparator threshold	V_{TH} relative to the programmed threshold. Refer to the field OPT_CMPTH in UART_MR (UART)	35	35	mV
V_{HYST}	Hysteresis	–	10	40	mV
t_P	Propagation delay(1)	With 100 mVpp square wave input around threshold	–	2.5	µs
t_{ON}	Start-up time(1)	–	–	200	µs

Note:

1. Simulation data.

48.7.18. 12-bit ADC Characteristics

Table 48-40. ADC Power Supply Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
V_{DD3V3}	Supply voltage range	Internal voltage reference OFF(1) Internal voltage reference ON(2)	2.5 2.8	VDD3V3	V

Notes:

1. VREFP must be supplied with an external voltage.
2. A capacitor must be connected between VREFP and GND. See the following table.

Table 48-41. ADC Voltage Reference Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
V_{VREFP}	Positive input voltage range	Internal voltage reference OFF only	2.5	V_{DD3V3}	V
V_{ACC}	Voltage accuracy(1, 2)	–	-2.5	+2.5	%
t_{ON}	Start-up time(2)	–	–	100	µS
V_{NOISE}	Output voltage noise(2)	Integrated rms value, bandwidth 1 Hz to 500 kHz	60	100	µVrms
R_{VREFP}	VREFP input resistance to ground(2)	ADC ON	7.2	12	kΩ
		ADC OFF	1	–	MΩ
$R_{LOADVREFP}$	Resistive load drive capability(2)	External load on VREFP	50	–	kΩ

Notes:

1. A ceramic X5R or X7R capacitor of 220nF maximum on VREFP is mandatory.
2. Simulation data.

Table 48-42. ADC Timing Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
f_{CKADC}	ADC clock frequency	–	0.2	20	MHz
t_{CONV}	ADC conversion time(1)	–	20	–	t_{CKADC}
f_S	Sampling rate(2)	–	–	1	MS/s
t_{START}	Start-up time(5)	OFF to ON(4)	–	4	µs
t_{TRACK}	Track and hold time(3)	–	300	–	ns

© 2026 Microchip Technology Inc. and its subsidiaries

Notes:

1. $t_{CONV} = t_{CH} + t_{TRACK} + 14 \times t_{CKADC}$ with $t_{CKADC} = 1 / f_{CKADC}$. $t_{CH} = 0$ when the ADC operates in the same input mode (single-ended or differential) for the current conversion than for the previous one. $t_{CH} = 2$ when the ADC input mode is changed to perform the current conversion.
2. $f_S = 1 / t_{CONV}$.
3. Refer to Track and Hold Time versus Source Impedance – Sampling Rate.
4. ADC_MR.SLEEP = 0 and ADC_MR.FWUP = 1.
5. Simulation data.

Table 48-43. ADC Supply Monitor

Symbol	Parameter	Conditions	Min	Max	Unit
V_{IT-}	Negative-going threshold voltage (VDD3V3) detection(1)	Voltage threshold in low range Voltage threshold in high range	2.30 2.60	2.60 2.90	V
V_{hys}	Hysteresis	–	5	35	mV
t_{DET}	Detection time	–	–	20	µs
t_{ON}	Start-up time(2)	From OFF to ON	–	100	µs

Notes:

1. Voltage threshold range selection from ADC_ACR.SMVT.
   Low or high range detection threshold is in correlation with the VREFP voltage supply and ADC voltage supply range. See Table 48-40.
2. Simulation data.

Table 48-44. ADC Analog Input Characteristics

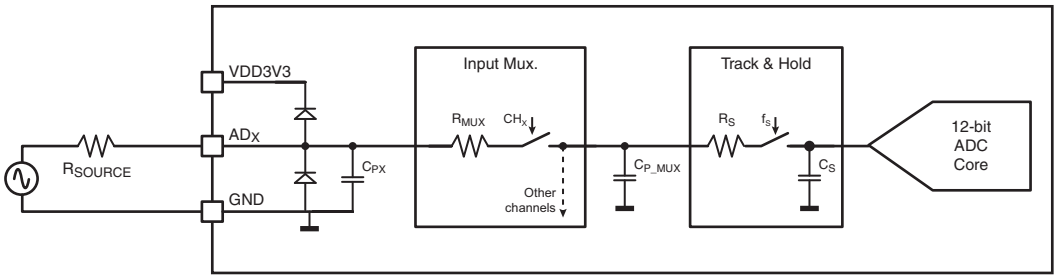
Symbol	Parameter	Conditions	Min	Max	Unit
V_{FS}	Analog input full scale range(1)	ADC_COR.DIFFx = 0	0	V_{VREFP}	V
		ADC_COR.DIFFx = 1	$-V_{VREFP}$	V_{VREFP}	
V_{INCM}	Common mode input range in Differential Input mode(2)	ADC_COR.DIFFx = 1	0.4 x V_{DD3V3}	0.6 x V_{DD3V3}	V
C_S	ADC sampling capacitance(5)	–	–	3	pF
C_{P_ADx}	ADx input parasitic capacitance(4, 5)	ADx pin configured as analog input	–	7	
R_{ON}	Internal series resistor(4, 5)	VDD3V3 ≥ 1.7V	–	8	kΩ
		VDD3V3 ≥ 2.4V	–	2	
Z_{IN}	Common mode input impedance(3, 5)	On ADx pin	1 / (f_S.C_S)	–	Ω

Notes:

1. $V_{FS} = V_{ADx}$ in Single-ended mode, and $V_{FS} = (V_{ADx} - V_{ADx+1})$ in Differential mode.
2. $V_{INCM} = (V_{ADx} + V_{ADx+1}) / 2$.
3. Assuming conversion on one single channel.
4. With respect to the equivalent model of Figure 48-23.
5. Simulation data.

© 2026 Microchip Technology Inc. and its subsidiaries

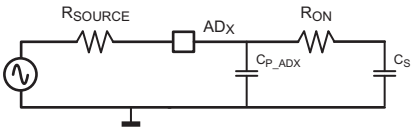
Figure 48-22. Acquisition Path Block Diagram



For tracking time calculation, during the sampling phase of the converter, this acquisition path can be reduced to the equivalent model of Figure 48-23 where:

- $R_{ON} = R_{MUX} + R_S$
- $C_{P\_ADX} = C_{PX} + C_{P\_MUX}$

Figure 48-23. Equivalent Model of the Acquisition Path



See Track and Hold Time versus Source Impedance – Sampling Rate for further details on usage of this model.

Table 48-45. Static Performance Characteristics[(1)]

Symbol	Parameter	Conditions	Min	Max	Unit
RES_{ADC}	Native ADC resolution	–	12		Bits
INL	Integral non-linearity	f_{CKADC} = 1 MHz ADC_EMR.OSR = $(000)_2$	-4	+4	LSB
DNL	Differential non-linearity		-2	+2	LSB
OE	Offset error		-5	+5	LSB
GE	Gain error		-20	+20	LSB

Notes:

1. In this table, errors are expressed in LSB where:
   - LSB = $V_{VREFP} / 2^{12}$ in Single-ended mode (ADC_CCR.DIFFx = 0)
   - LSB = $V_{VREFP} / 2^{11}$ in Differential mode (ADC_CCR.DIFFx = 1)
2. Errors with respect to the best-fit line method.

In the following table, unless otherwise specified, the specifications are given with the following assumptions.

- Source resistance = 50Ω
- ADC_EMR.OSR<2:0> = $(000)_2$
- Noise bandwidth = [0;f_S/2], V_{INPP} = 0.95 x V_{VREFP}
- High-speed
  - f_{CKADC} = 20 MHz, f_S = 1MS/s,
  - $f_{IN} \leq$ 125 kHz

© 2026 Microchip Technology Inc. and its subsidiaries

Table 48-46. Dynamic Performance Characteristics[1]

Symbol	Parameter	Conditions	Min	Max	Unit
Single-ended Input mode (ADC_CCR.DIFFx = 0)					
SNR	Signal to noise ratio	–	60	–	dB
THD	Total harmonic distortion	–	–	-64	dB
SINAD	Signal to noise and distortion	–	58	–	dB
ENOB	Effective number of bits	–	9.5	–	bits
Differential Input mode (ADC_CCR.DIFFx = 1)					
SNR	Signal to noise ratio	–	64	–	dB
THD	Total harmonic distortion	–	–	-68	dB
SINAD	Signal to noise and distortion	–	62	–	dB
ENOB	Effective number of bits	–	10.5	–	bits

Note:

1. Simulation data.

48.7.18.1. Track and Hold Time versus Source Impedance – Sampling Rate

Referring to Figure 48-23, during the tracking phase, the 12-bit ADC charges its sampling capacitor C_S through various serial resistors modeled as R_{SOURCE} (source output resistor) and R_{ON} (multiplexer series resistor and the sampling switch series resistor). In case of high output source resistance (e.g. a low power resistive divider), the tracking time must be increased to ensure full settling of the sampling capacitor voltage. Of course, programming a long tracking time may impact the sampling frequency (f_S). The following formula gives the minimum tracking time that ensures a 12-bit accurate settling.

$t_{TRACK} \geq 8 \times (R_{SOURCE} + R_{ON}) \times C_S$

The ADC Controller (ADCC) counts the tracking time in ADC clock cycles (t_{CKADC}). This time can be adjusted between 6 and 54 cycles by means of the fields ADC_MR.TRACKTIM and ADC_EMR.TRACKX = TRACKTIMx4. At maximum ADC clock frequency (20 MHz), the maximum tracking time that can be programmed is 2.7 µs. This limits 12-bit accurate sampling to sources having R_{SOURCE} in the 100 kΩ range. To overcome this limitation, the ADC clock frequency can be decreased.

The following two examples show typical use cases of tracking time and sampling frequency calculation.

Example: Calculated tracking time is lower than the default (minimum) tracking time.

- Assuming: f_{CKADC} = 8 MHz (t_{CKADC} = 125 ns), R_{SOURCE} = 10 kΩ
- The minimum required track time is: $t_{TRACK} = 8 \times (10\ k\Omega + 2\ k\Omega) \times 3pF = 288$ ns
- t_{TRACK} is less than the minimum tracking time (6 x t_{CKADC} = 750 ns): set TRACKTIM = 0 and TRACKX = 0
- The real tracking time is 6 x t_{CKADC} (750 ns) and the conversion time is $t_{CONV} = t_{CH} + t_{TRACK} + 14 \times t_{CKADC} = 20 \times t_{CKADC}$ with t_{CH} = 0
- The sampling rate is: f_S = 8 MHz / 20 = 400 kS/s
- The maximum allowable source resistance is: $R_{SOURCE\_MAX} = (6 \times t_{CKADC}) / (3pF \times 8) - 2\ k\Omega = 29.25\ k\Omega$

Example: Calculated tracking time is greater than the default (minimum) tracking time.

- Assuming: f_{CKADC} = 20 MHz (t_{CKADC} = 50 ns), R_{SOURCE} = 16 kΩ
- The minimum required track time is: $t_{TRACK} = 8 \times (16\ k\Omega + 2\ k\Omega) \times 3\ pF = 432$ ns
- t_{TRACK} is greater than the minimum tracking time (6 x t_{CKADC} = 300ns): set TRACKTIM=4 and TRACKX=TRACKTIMx4

© 2026 Microchip Technology Inc. and its subsidiaries

- The real track time is: (4 x (4 + 1) - 10) = 10 x t_{CK_ADC} = 500 ns
- The conversion time is t_{CONV} = t_{CH} + t_{TRACK} + 14 x t_{CKADC}= 24 x t_{CKADC} with t_{CH} = 0
- The sampling rate is: f_S = 20 MHz / 24 = 833.3 kS/s
- The maximum allowable source resistance is: R_{SOURCE_MAX} = (10 x t_{CKADC}) / (3pF x 8) - 2 kΩ = 18.8 kΩ

48.7.19. VBAT Resistive Load

Table 48-47. VBAT Resistive Load

Symbol	Parameter	Conditions	Max	Unit
R_{low}	Low value of ZBAT resistive load(1)	Impedance on VBAT during measurement	24	kΩ
R_{high}	High value of ZBAT resistive load(1)		120	

Note:

1. Simulation data.

48.7.20. Analog Comparator Characteristics

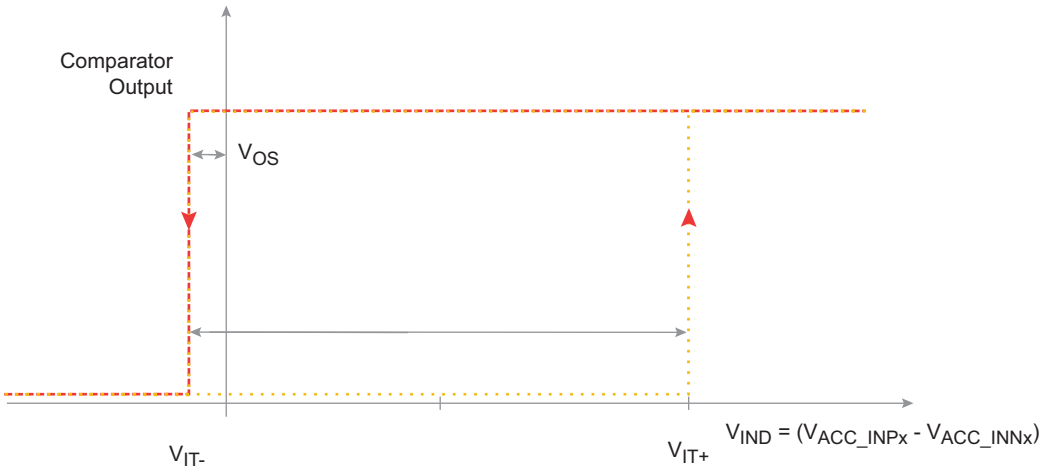
Table 48-48. Analog Comparator Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
V_{VDD3V3}	Supply voltage range	–	2.25	3.6	V
t_{START}	Start-up time(1)	–	–	100	µs
t_P	Propagation delay(1)	–	–	0.1	µs
V_{CM}	Common mode input range	On the positive and negative inputs	0	$V_{VDD3V3\ -1V}$	V
V_{OS}	Input offset voltage	–	-15	+15	mV
V_{hys}	Comparator hysteresis(1)	–	–	10	mV

Notes:

1. Simulation data.
2. Measured with 100 mV threshold overdrive.

Figure 48-24. Analog Comparator Input/Output Transfer Function



© 2026 Microchip Technology Inc. and its subsidiaries

48.8. Embedded Flash Characteristics

48.8.1. Embedded Flash DC Characteristics

Table 48-49. DC Flash Characteristics

Symbol	Parameter	Conditions	Typ	Max	Unit
I_{CC}	Active Current[1]	Random Read: - Total (VDDCORE + VDD3V3)	–	7	mA
		Program: - Total (VDDCORE + VDD3V3)	–	5	mA
		Erase: - Total (VDDCORE + VDD3V3)	–	7	mA

Note:

1. Simulation data.

48.8.2. Embedded Flash AC Characteristics

Table 48-50. AC Flash Characteristics

Parameter	Conditions	Min	Max	Unit
Program/ Erase operation cycle time[1, 3]	Write page (512 bytes)	–	3	ms
	Erase 4-Kbyte, 8-Kbyte, 16-Kbyte or 32-Kbyte block[2]	–	155	ms
	Erase sector (128-Kbyte)	–	160	ms
	Erase one plane, any size	–	250	ms
	Full chip erase (Hardware erase with erase signal), any size	–	800	ms
Erase Suspend/Resume[3]	Suspend Erase command to Suspended Erase	–	50	µs
Write Suspend/Resume[3]	Suspend Write command to Suspended Write	–	20	µs
Clear GPNVM or Lock bits[3]	–	–	150	ms
Set GPNVM or Lock bits[3]	–	–	200	µs
Set Security bit[3]	–	–	150	ms
Hardware Erase signal assertion time	Time needed to start a full Flash erase operation	1	20	ms
Data retention	Not powered or powered @ 85°C	10	–	years
	Not powered or powered @ 25°C	20	–	
Endurance	Write/Erase cycles per page, block or sector @ 85°C	–	100K	cycles

Notes:

1. Page from main memory or user signature array.
2. Same Erase time due to parallel erase of blocks.
3. Simulation data.

48.8.2.1. Flash Wait States and Operating Frequency

The maximum operating frequency given in the following table is limited by the Embedded Flash access time when the processor is fetching code out of it. The table gives the device maximum operating frequency depending on the FWS field of the EFC_FMR register. This field defines the number of wait states required to access the Embedded Flash memory.

© 2026 Microchip Technology Inc. and its subsidiaries

Table 48-51. Flash Wait State Versus Core Operating Frequency

FWS (Flash Wait State)	Core Frequency(1, 2)	Unit
0	25	MHz
1	51	
2	76	
3	102	
4	127	
5	153	
6	179	
7	204	

Notes:

1. Over the full temperature and VDDCORE/VDD3V3 voltage range of the device.
2. Simulation data.

48.9. Power Supply Current Consumption

This section provides information about the current consumption on different power supply rails of the device. It gives current consumption in low-power modes (Backup mode, Wait mode, Sleep mode) and in Active mode (the application running from memory, by peripheral).

48.9.1. Backup Mode Current Consumption

Backup mode configurations and measurements are defined as follows:

- Configuration A and B are typical use cases with crystal oscillator, anti-tamper pins enabled
- Configuration C is the same as Configuration B, but with SLCD enabled

Remember: In Backup mode, the core voltage regulator is off and thus all the digital functions powered by VDDCORE are off. The 32 kHz Slow RC oscillator is an always-on clock source.

48.9.1.1. Backup Mode Configuration A

- 32.768 kHz Crystal and RC Oscillators are enabled
- RTC running
- RTT enabled on 1 Hz mode
- Force wake-up (FWUP) enabled
- Anti-tamper input TMP0 enabled, RTCOUT0 enabled at 1Hz
- All power rails OFF except VBAT, Current measurement as per below

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 48-25. Measurement Setup for Configuration A

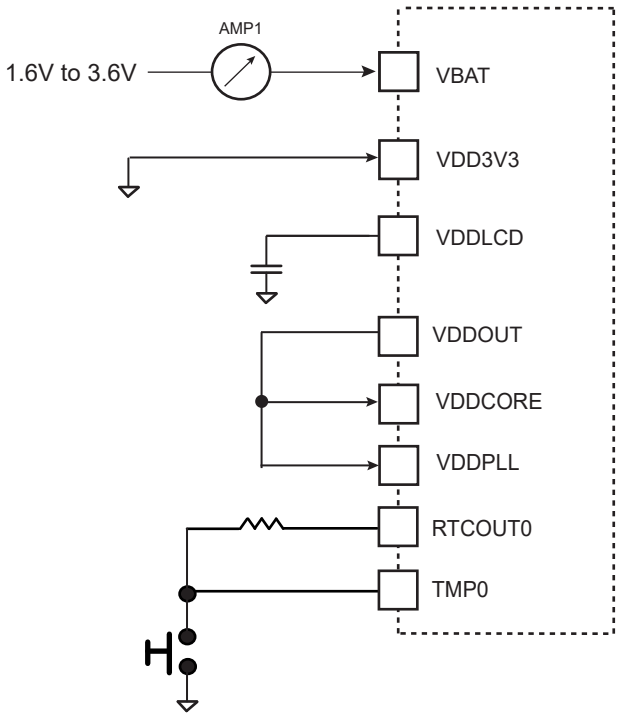


Table 48-52. Current Consumption Values for Backup Mode Configuration A

Conditions	Configuration A	Unit
VBAT = 3.6V		µA
25°C	3.4	
85°C	6.5	
VBAT = 3.0V		
25°C	2.3	
85°C	4.9	
VBAT = 1.6V		
25°C	1.2	
85°C	3.2	

48.9.1.2. Backup Mode Configuration B

- Power Switch on VDD3V3
- RTC running
- RTT enabled on 1 Hz mode
- Force wake-up (FWUP) enabled
- Anti-tamper input TMP0 enabled, RTCOUT0 enabled at 1Hz
- All power rails OFF except VBAT and VDD3V3, Current measurement as per below

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 48-26. Measurement Setup for Configurations B

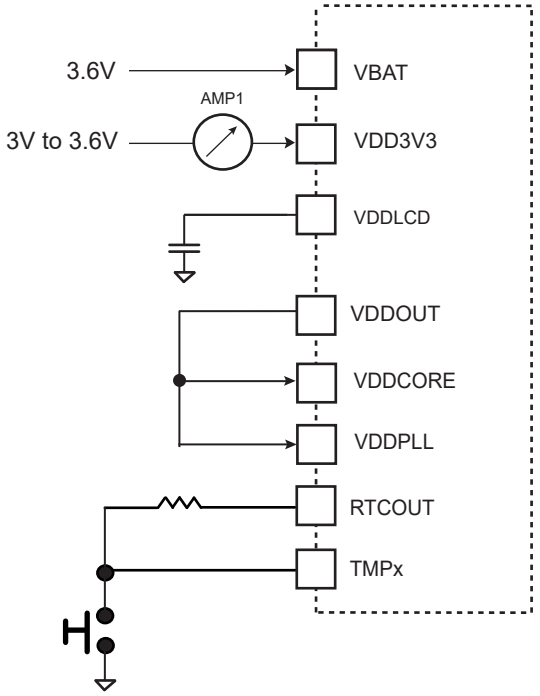


Table 48-53. Current Consumption Values for Backup Mode Configuration B

Conditions	Configuration B	Unit
VDD3V3 = 3.6V		µA
25°C	6.3	
85°C	15	
VDD3V3 = 3.0V		
25°C	4.8	
85°C	11	

48.9.1.3. Backup Mode Configuration C

- Power Switch on VDD3V3
- RTC running
- RTT enabled on 1 Hz mode
- Force wake-up (FWUP) enabled
- SLCD controller in Low-power mode, static bias and x64 slow clock buffer on-time drive time
- SLCD voltage regulator used
- Anti-tamper input TMP0 enabled, RTCOUT0 enabled at 1Hz
- All power rails OFF except VBAT and VDD3V3, Current measurement as per below

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 48-27. Measurement Setup for Configuration C

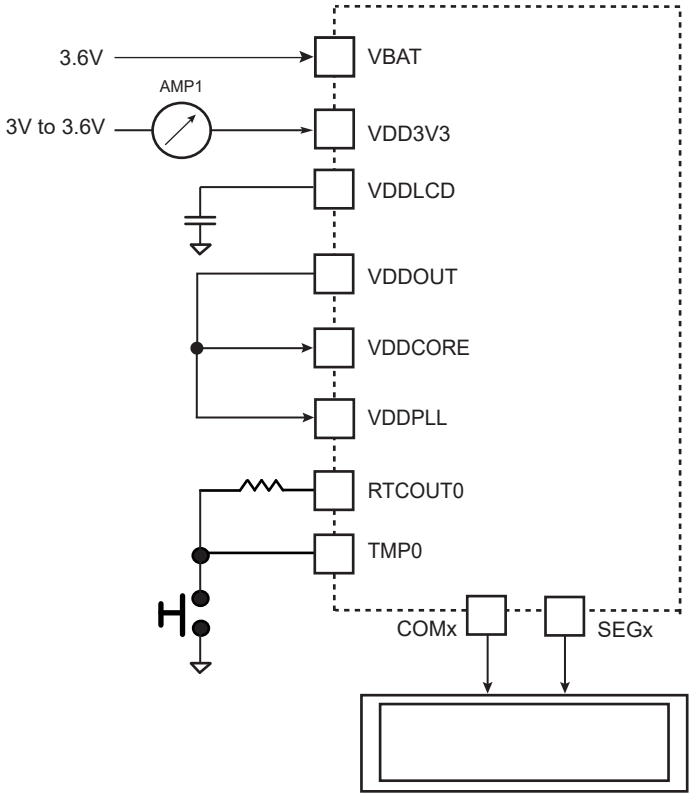


Table 48-54. Current Consumption Values for Backup Mode Configuration C

Conditions	Configuration C	Unit
VDD3V3 = 3.6V		
25°C	10.5	µA
85°C	20	

48.9.2. Wait Mode Current Consumption

Wait mode configuration and measurements are defined in Wait Mode Configuration.

Remember: In Wait mode, the core voltage regulator is on, but the device is not clocked. Flash Power mode can be either in Read-Idle mode or Deep Power-down mode. Wait mode provides a much faster wake-up compared to Backup mode.

48.9.2.1. Wait Mode Configuration

- 32.768 kHz crystal oscillator enabled
- 12 MHz RC oscillator running
- Main crystal and PLLs stopped
- RTC running
- RTT enabled on 1 Hz mode.
- One wake-up pin (WKUPx) used in Fast Wake-up mode
- Anti-tamper inputs TMP0 and RTCOUT0 enabled at 1Hz
- GPIO lines in default state

© 2026 Microchip Technology Inc. and its subsidiaries

- Current measurement as illustrated in the figure below

Figure 48-28. Measurement Setup for Wait Mode Configuration

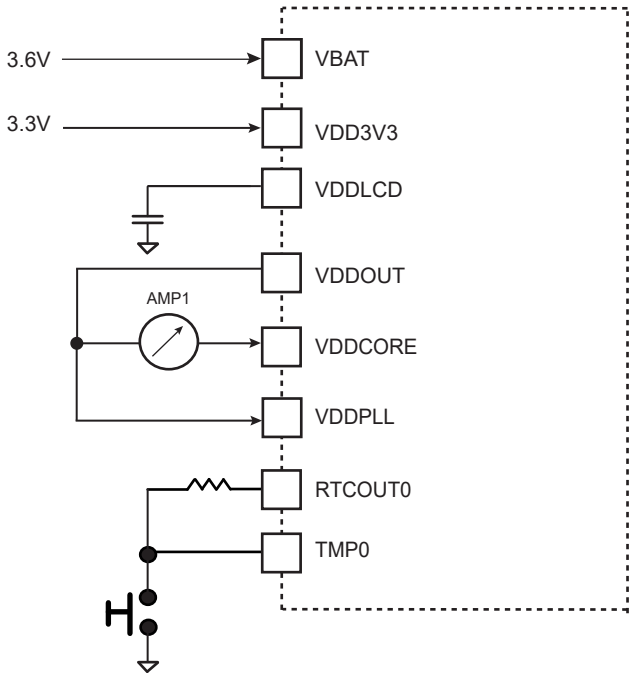


Table 48-55. Current Consumption in Wait Mode

Conditions	IDD_IN/IO - AMP1		Unit
	@25°C	@85°C	
Device powered, all clocks stopped, core not running	4	29	mA

48.9.3. Sleep Mode Current Consumption

Sleep mode configuration and measurements are defined in this section.

Remember: In Sleep mode, power consumption of the device is optimized with respect to response time.

- VDD3V3 = 3.3V
- VDDCORE/VDDPLL = Internal Voltage regulator used
- T_A = 25°C
- Core clock (CPU_CLK0) stopped
- Sub-system main clocks (MCK0, MCK0DIV,MCK0DIV2), running at various frequencies from PLLs.
- All peripheral clocks deactivated
- No activity on I/O lines
- Current measurement as illustrated in the figure below

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 48-29. Measurement Setup for Sleep Mode

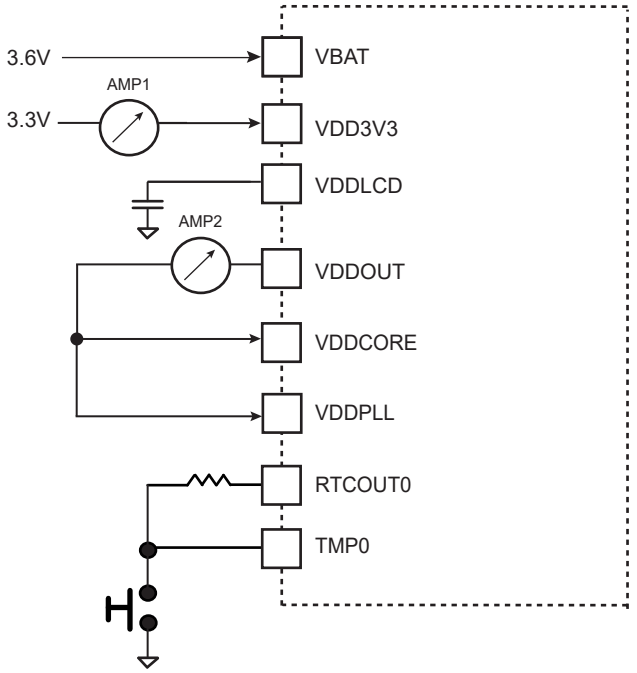


Table 48-56. Typical Sleep Mode Current Consumption Versus Frequency

Core Clock / Main Clock (MHz)	AMP1	AMP2	Unit
Core Clock/Main Clock MCK0, MCK0DIV, MCK0DIV2			
200/200, 100, 200	21	19	mA
100/100, 100, 100	13	12	mA
50/50, 50, 50	9	8	mA

48.9.4. Active Mode Power Consumption

The current consumption configuration for Active mode is as follows:

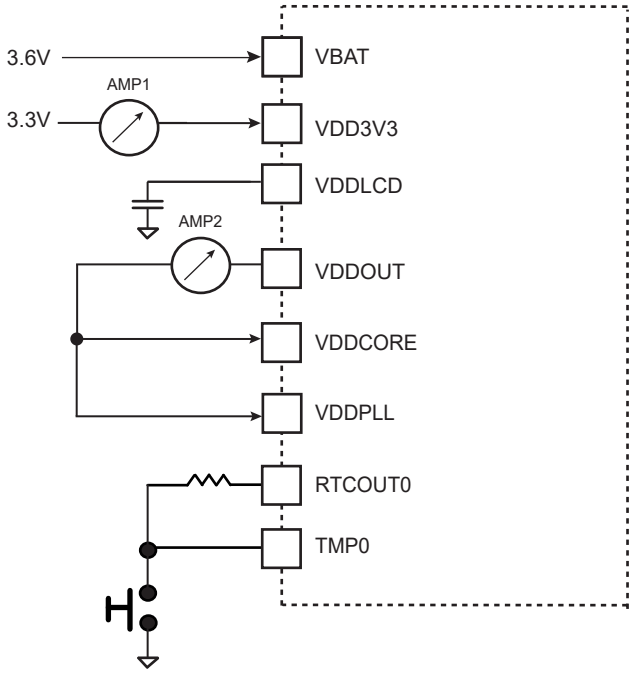
- VDD3V3 = 3.3V
- VDDCORE = internal voltage regulator used
- TA = 25°C
- Core clock (CPU_CLK0) running at various frequencies
- Sub-system main Clocks (MCK0, MCK0DIV, MCK0DIV2), running at various frequencies from PLLs.
- All peripheral clocks deactivated
- No activity on IO lines
- Flash Wait State (FWS) in EEFC_FMR adjusted versus core frequency
- Current measurement as illustrated in the figure below

Active mode power consumption values are obtained using a PIC32CXMTG-EK, with a limited number of samples using typical process devices, at typical voltage and room temperature.

These values are not tested in production.

© 2026 Microchip Technology Inc. and its subsidiaries

Figure 48-30. Measurement Setup for Active Mode



48.9.4.1. CoreMark™

- Core running CoreMark from different memory modes:
  - Flash with cache disabled
  - Flash with cache enabled

PLLA and PLLB are used to generate the required frequencies.

Table 48-57. Current Consumption

Core Clock/Main Clock (MHz)	Cache Disabled		Cache Enabled		Unit
Core Clock/Main Clock MCK0, MCK0DIV, MCK0DIV2	AMP1	AMP2	AMP1	AMP2	
200/200, 100, 200	37	29	40	37	mA
100/100, 100, 100	25	19.5	25	22	mA
50/50, 50, 50	18	13	14	13	mA

© 2026 Microchip Technology Inc. and its subsidiaries

49. Mechanical Characteristics

Packages listed here are rated as per reference J-STD-020D = Time and Conditions

MSL LEVEL: MSL-3

TIME: 1 68 hours

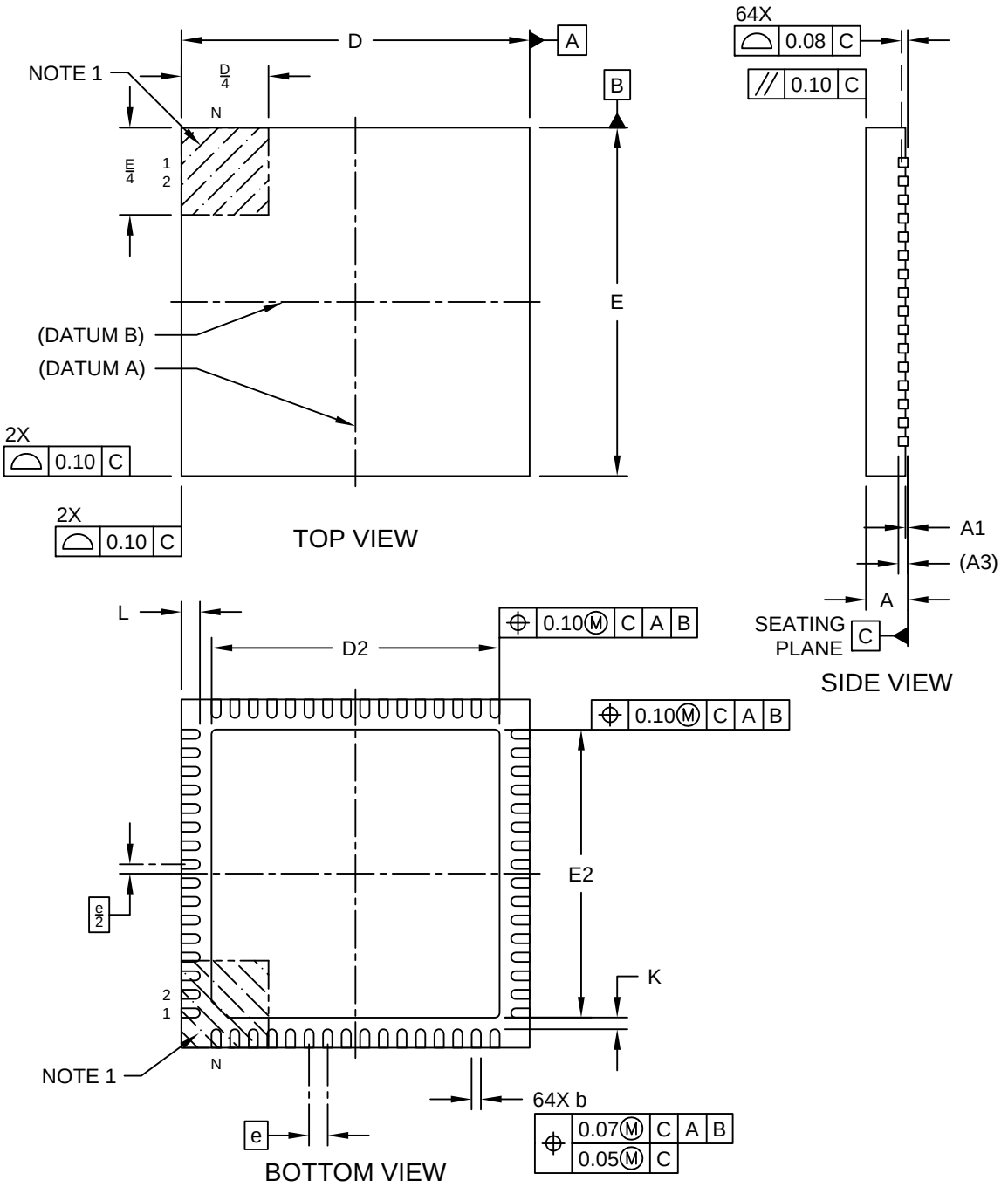
CONDITIONS: ≤ 30°C/60% RH

© 2026 Microchip Technology Inc. and its subsidiaries

49.1. 64-lead VQFN Mechanical Characteristics

64-Lead Very Thin Plastic Quad Flat, No Lead Package (SHB) - 7.5x7.5 mm Body [SHB] With 6.2 mm Exposed Pad; Atmel Legacy Global Package Code ZFR

Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging

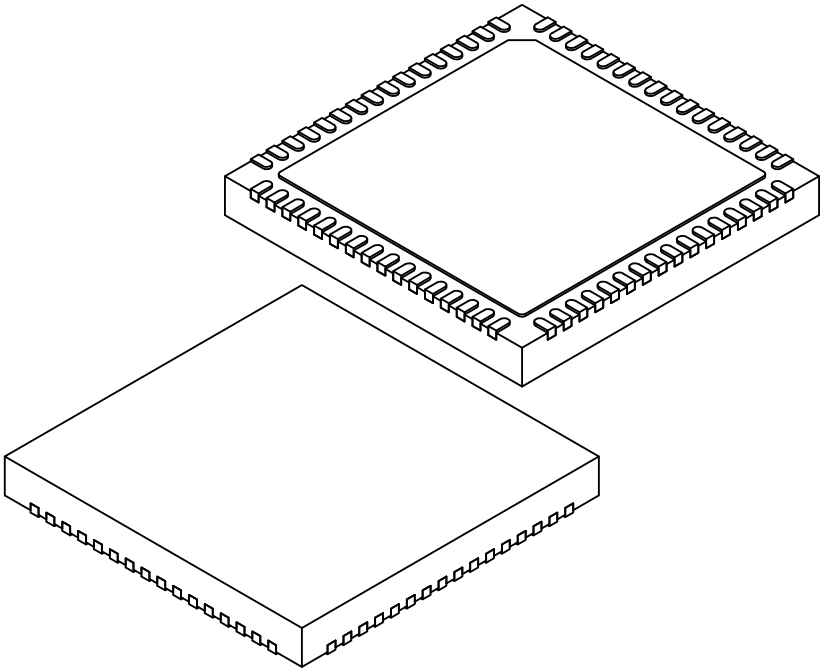


Microchip Technology Drawing C04-21410 Rev A Sheet 1 of 2

© 2026 Microchip Technology Inc. and its subsidiaries

64-Lead Very Thin Plastic Quad Flat, No Lead Package (SHB) - 7.5x7.5 mm Body [SHB] With 6.2 mm Exposed Pad; Atmel Legacy Global Package Code ZFR

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Terminals	N		64	
Pitch	e		0.40 BSC	
Overall Height	A	0.80	0.85	0.90
Standoff	A1	0.00	0.02	0.05
Terminal Thickness	A3		0.20 REF	
Overall Length	D		7.50 BSC	
Exposed Pad Length	D2	6.05	6.20	6.35
Overall Width	E		7.50 BSC	
Exposed Pad Width	E2	6.05	6.20	6.35
Terminal Width	b	0.15	0.20	0.25
Terminal Length	L	0.30	0.40	0.50
Terminal-to-Exposed-Pad	K	0.20	-	-

Notes:

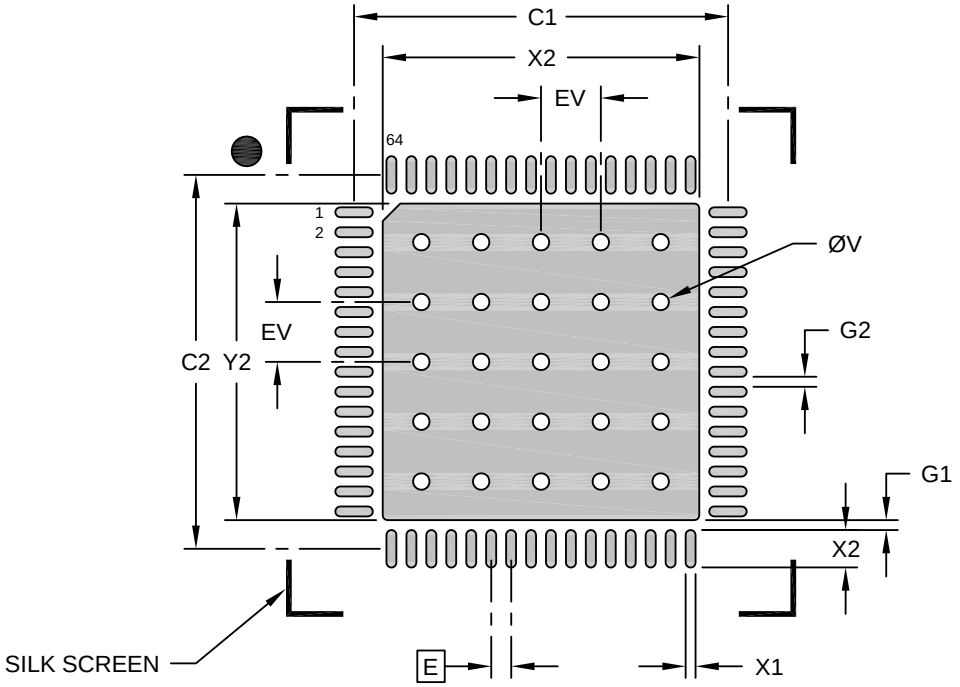
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated
3. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-21410 Rev A Sheet 1 of 2

© 2026 Microchip Technology Inc. and its subsidiaries

64-Lead Very Thin Plastic Quad Flat, No Lead Package (SHB) - 7.5x7.5 mm Body [SHB] With 6.2 mm Exposed Pad; Atmel Legacy Global Package Code ZFR

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.40 BSC	
Optional Center Pad Width	X2			6.35
Optional Center Pad Length	Y2			6.35
Contact Pad Spacing	C1		7.50	
Contact Pad Spacing	C2		7.50	
Contact Pad Width (X64)	X1			0.20
Contact Pad Length (X64)	Y1			0.75
Contact Pad to Center Pad (X64)	G1	0.20		
Contact Pad to Contact Pad (X60)	G2	0.20		
Thermal Via Diameter	V		0.33	
Thermal Via Pitch	EV		0.20	

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
2. For best soldering results, thermal vias, if used, should be filled or tented to avoid solder loss during reflow process

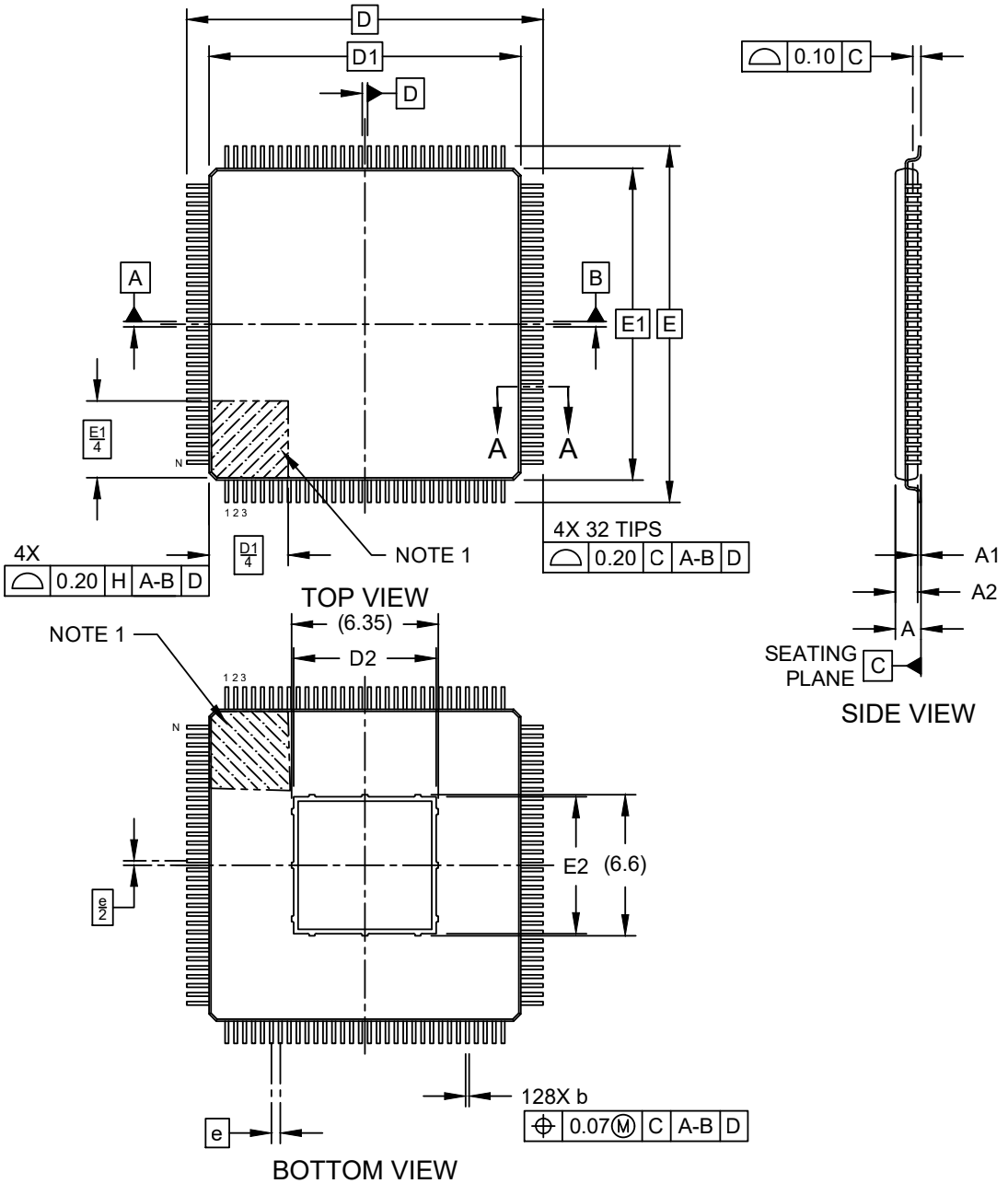
Microchip Technology Drawing C04-23410 Rev A

© 2026 Microchip Technology Inc. and its subsidiaries

49.2. 128-pin EP-TQFP Mechanical Characteristics

128-Lead Thin Quad Flat, Plastic (X9B) - 14x14x1 mm Body [TQFP]
With 6.15x6.40 mm Exposed Pad

Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging

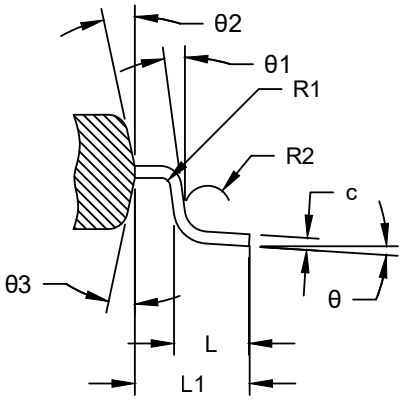


Microchip Technology Drawing C04-21321 Rev C Sheet 1 of 2

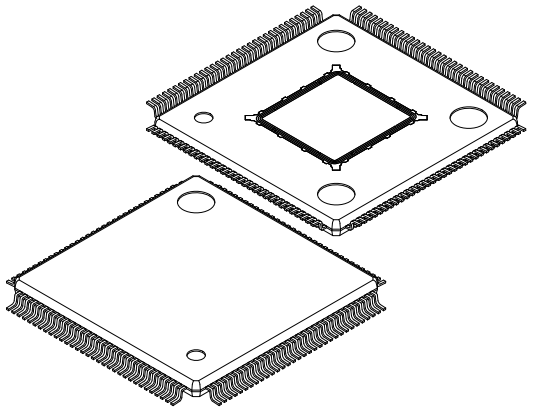
© 2026 Microchip Technology Inc. and its subsidiaries

128-Lead Thin Quad Flat, Plastic (X9B) - 14x14x1 mm Body [TQFP] With 6.15x6.40 mm Exposed Pad

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



SECTION A-A



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Leads	N		128	
Lead Pitch	e		0.40 BSC	
Overall Height	A	-	-	1.20
Standoff	A1	0.05	0.10	0.15
Molded Package Thickness	A2	0.95	1.00	1.05
Overall Length	D		16.00 BSC	
Molded Package Length	D1		14.00 BSC	
Exposed Pad Length	D2	6.05	6.15	6.25
Overall Width	E		16.00 BSC	
Molded Package Width	E1		14.00 BSC	
Exposed Pad Width	E2	6.30	6.40	6.50
Foot Length	L	0.45	0.60	0.75
Footprint	L1		1.00 REF	
Lead Width	b	0.13	0.16	0.23
Lead Thickness	c	0.09	-	0.20
Lead Bend Radius	R1	0.08	-	-
Lead Bend Radius	R2	0.08	-	0.20
Lead Foot Angle	θ	0°	3.5°	7°
Lead Angle	θ1	0°	-	-
Mold Draft Angle Top	θ2	11°	12°	13°
Mold Draft Angle Bottom	θ3	11°	12°	13°

Notes:

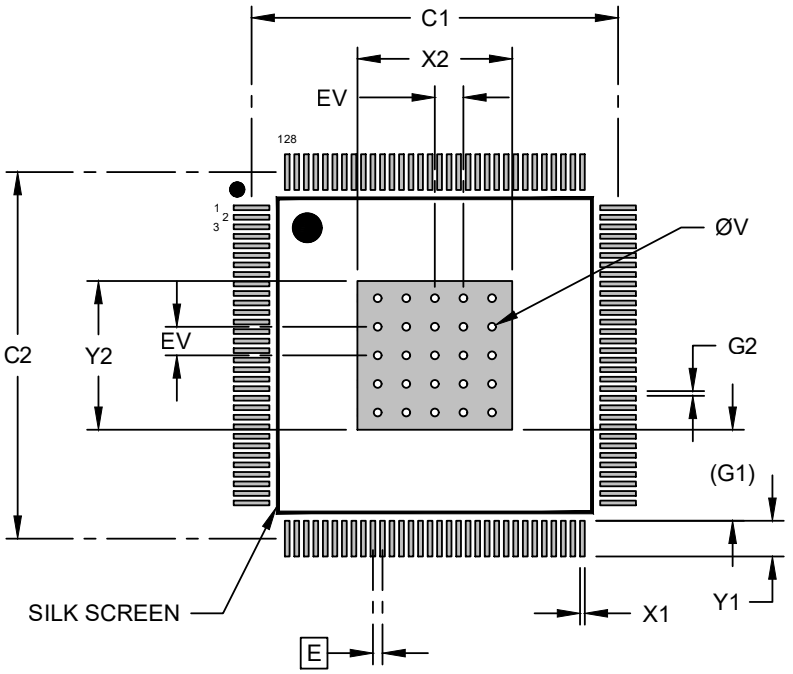
1. The Pin 1 visual index feature may vary, but it must be located within the hatched area.
2. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. The theoretically exact value is shown without tolerances.
   REF: Reference Dimension, usually without tolerance, is for information purposes only.

Microchip Technology Drawing C04-21321 Rev C Sheet 2 of 2

© 2026 Microchip Technology Inc. and its subsidiaries

128-Lead Thin Quad Flat, Plastic (X9B) - 14x14x1 mm Body [TQFP] With 6.15x6.40 mm Exposed Pad

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.40 BSC	
Optional Center Pad Width	X2			6.60
Optional Center Pad Length	Y2			6.35
Contact Pad Spacing	C1		15.40	
Contact Pad Spacing	C2		15.40	
Contact Pad Width (X128)	X1			0.20
Contact Pad Length (X128)	Y1			1.50
Contact Pad to Center Pad (X128)	G1		3.73 REF	
Contact Pad to Contact Pad (X124)	G2	0.20		
Thermal Via Diameter	V		0.33	
Thermal Via Pitch	EV		1.20	

Notes:

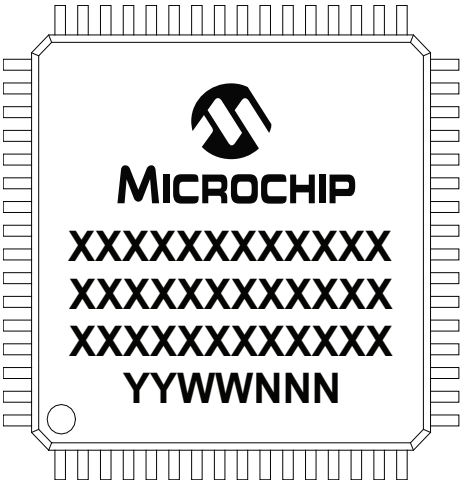
1. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. The theoretically exact value is shown without tolerances.
2. For best soldering results, thermal vias, if used, should be filled or tented to avoid solder loss during the reflow process.

Microchip Technology Drawing C04-23321 Rev C

© 2026 Microchip Technology Inc. and its subsidiaries

50. Marking

Top marking follows the scheme below:



with possible values:

Line	Description	Values
1	Company logo	Microchip logo
2	Company name	Microchip
3	Device name	PIC32CXMTG
4	Packaging code	SHB for 64-lead VQFN-EP 7.5 x 7.5 x 0.9 mm, pitch 0.4 mm X9B for 128-lead TQFP-EP 14 x 14 x 1.0 mm, pitch 0.4 mm
5	Not used	–
6	Lot traceability	YYWWNNN

© 2026 Microchip Technology Inc. and its subsidiaries

51. Ordering Information

Ordering Code	Flash (Kbytes)	Package	Carrier Type	Package Type	Temperature Operating Range
PIC32CX2051MTG128-I/X9B	2x1024	EP-TQFP128	Tray	Green	Industrial (-40°C to +85°C)
PIC32CX2051MTG128T-I/X9B			Tape & Reel		
PIC32CX2051MTG64-I/SHB		VQFN64	Tray		
PIC32CX2051MTG64T-I/SHB			Tape & Reel		
PIC32CX1025MTG128-I/X9B	2x512	EP-TQFP128	Tray		
PIC32CX1025MTG128T-I/X9B			Tape & Reel		
PIC32CX1025MTG64-I/SHB		VQFN64	Tray		
PIC32CX1025MTG64T-I/SHB			Tape & Reel		
PIC32CX5112MTG128-I/X9B	2x256	EP-TQFP128	Tray		
PIC32CX5112MTG128T-I/X9B			Tape & Reel		

© 2026 Microchip Technology Inc. and its subsidiaries

52. Revision History

Revision C - 03/2026

The following changes were implemented for this revision:

Section	Changes
Mechanical Characteristics	Updated figures.

Rev. B - 11/2022

Note: Microchip is in the process of implementing inclusive language in our documentation. During this update, you may see the old terms and the new terms in our documentation text. We apologize for any confusion.

Section	Changes
Minor editorial changes throughout.	
Pinout and Multiplexing	Corrected pin 62 namein table 128-pin EP-TQFP Multiplexing.
Input/Output Lines	Shutdown (SHDN) Pin: corrected pin name.
	Flash Memory Hardware ERASE Signal: updated.
ROM Code and Boot Strategies	GPNVM Bits Overview: updated table.
Supply Controller (SUPC)	SUPC_SR: index 12 now 'reserved'.
Clock Generator	PLL Controls: updated.
Power Management Controller (PMC)	Block Diagram: updated.
Special Function Registers (SFR)	Register Write Protection: list of registers updated. Register Summary: offset 0xA0 now reserved (was Core Debug Configuration Register). SFR_PWM_DEBUG: index 1 now 'reserved' (was CORE1)
Chip Identifier (CHIPID)	Chip ID Registers: updated table.
Flexible Serial Communication Controller (FLEXCOM)	Throughout: Deleted references to Host High-Speed mode. Bus Clear Command, FIFO Pointer Error, FLEX_TWI_CR: modified Sniffer Mode: updated Sniffer description. FLEX_TWI_SMR: updated BSEL description.
Quad Serial Peripheral Interface (QSPI)	Corrected clock name from GCK to GCLK throughout.
Analog-to-Digital Converter (ADC) Controller	ADC_MR: updated reset value.
Analog Comparator Controller (ACC)	ACC_ISR: updated reset value.
True Random Number Generator (TRNG)	TRNG_WPSR: modified SWETYP description (value 5).
Pulse Width Modulation Controller (PWM)	Fault Protection: figure updated.
Electrical Characteristics	VDD3V3 Power-On-Reset: corrected unit in table. Analog Comparator Characterisitics: corrected unit in table.

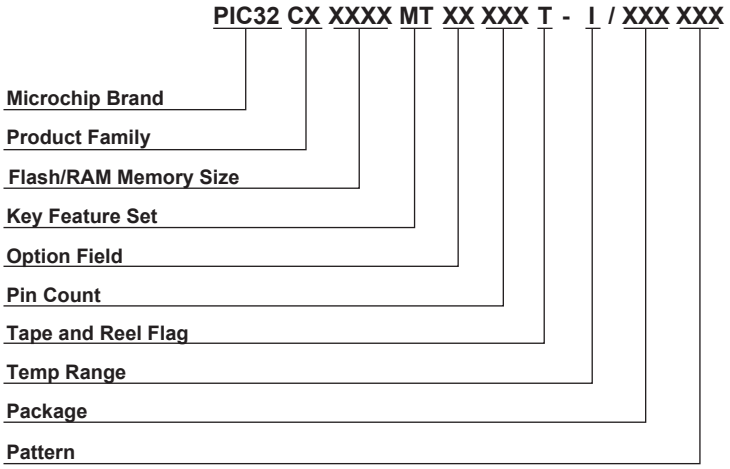
Rev. A - 03/2022

First Issue.

© 2026 Microchip Technology Inc. and its subsidiaries

53. Product Identification System

To order or obtain information, for example, on pricing or delivery, contact Microchip: https://www.microchip.com/en-us/about/contact-us.



Microchip Brand:	PIC32	= PIC32
Product Family:	CX	= Mid-performance CM4
Flash/RAM Memory Size (in Kb):	2051	= 2x1024/512 + 32 + 16
	1025	= 2x512/256 + 32 + 16
	5112	= 2x256/128 + 32 + 16
Key Feature Set:	Metering	= MT
Option Field:	G	= Single Core
Pin Count:	128	= 128-pin
	64	= 64-lead
Tape and Reel Flag:	Blank	= Standard packaging (tray)
	T	= Tape and Reel
Temp Range:	I	= -40°C to +85°C (Industrial)
Package:	X9B	= 128-pin Thin Quad Flat Pack 14x14 mm Body with 4.7x4.7 mm Exposed Pad
	SHB	= 64-lead Very Thin Plastic Quad Flat, No Lead Package 7.5x7.5 mm Body with 6.2 mm Exposed Pad
Pattern:	Three-digit QTP, SQTP code or special requirement (blank otherwise)	

Examples:

- PIC32CX2051MTG128-I/X9B:
  - PIC32: Microchip PIC32 microcontroller
  - CX: ARM Cortex-M4 processor-based
  - 2051: 2x1024 KBytes Flash Memory / 512KBytes SRAM
  - MT-G: Single-Core
  - 128: 128-pin
  - I: Industrial Temp
  - Tray
  - X9B: EP-TQFP128

© 2026 Microchip Technology Inc. and its subsidiaries

Note: Tape and Reel identifier only appears in the catalog part number description. This identifier is used for ordering purposes and is not printed on the device package

© 2026 Microchip Technology Inc. and its subsidiaries

Microchip Information

Trademarks

The “Microchip” name and logo, the “M” logo, and other names, logos, and brands are registered and unregistered trademarks of Microchip Technology Incorporated or its affiliates and/or subsidiaries in the United States and/or other countries (“Microchip Trademarks”). Information regarding Microchip Trademarks can be found at https://www.microchip.com/en-us/about/legal-information/microchip-trademarks.

ISBN: 979-8-3371-2964-8

Legal Notice

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. Contact your local Microchip sales office for additional support or, obtain additional support at www.microchip.com/en-us/support/design-help/client-support-services.

THIS INFORMATION IS PROVIDED BY MICROCHIP “AS IS”. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP’S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION.

Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip products:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip products are strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is “unbreakable”. Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.

© 2026 Microchip Technology Inc. and its subsidiaries

Product Page Links

PIC32CX1025MTG, PIC32CX2051MTG, PIC32CX5112MTG

© 2026 Microchip Technology Inc. and its subsidiaries