
Hardware Design Checklist

1.0 INTRODUCTION

This document provides a hardware design checklist for the Microchip LAN9691/2/3 family of switches. The checklist should be followed when utilizing the LAN parts in a new design. A summary of these items is provided in [Section 16.0, "Hardware Checklist Summary"](#). Detailed information on the following subjects can be found in the corresponding sections:

- [Section 2.0, "General Considerations"](#)
- [Section 3.0, "Power/Grounding"](#)
- [Section 4.0, "Reference Clocks"](#)
- [Section 5.0, "CPU System"](#)
- [Section 6.0, "Port Configuration"](#)
- [Section 7.0, "Parallel MAC Interface"](#)
- [Section 8.0, "Flexcom Interface"](#)
- [Section 9.0, "USB Device and Host ULPI Interface"](#)
- [Section 10.0, "Serial GPIO Controller \(SGPIO\)"](#)
- [Section 11.0, "MII Management Interface"](#)
- [Section 12.0, "GPIO Usage"](#)
- [Section 13.0, "JTAG Interface"](#)
- [Section 14.0, "Layout Considerations"](#)
- [Section 15.0, "Miscellaneous"](#)

2.0 GENERAL CONSIDERATIONS

2.1 Required References

The implementer should have the latest version of the following documents on hand:

- *LAN9691-LAN9692-LAN9693 Data Sheet*
- *LAN969x Errata*
- *EV09P11A, EVB-LAN9692-LM Reference Schematic Design* (Schematic)
- *EV09P11A, EVB-LAN9692-LM Reference Design Layout* (Layout file)
- *EV09P11A, EVB-LAN9692-LM Hardware Manual*

2.2 Pin Check

- Check the pinout of the part against the data sheet. Ensure that all pins match the data sheet and are configured as intended input, output or bidirectional for error checking.
- It is highly recommended to follow the reference design schematics as a basis and retain the reference design's use of GPIOs (parallel as well as serial) whenever possible to minimize software changes. Keep a log of major changes (such as port numbering, PHY addresses, and GPIO and SGPIO usage) and provide this log when there is a design review or when starting software customization.

2.3 Strapping Pins

- Configuration strapping pins allow various features of the device to be automatically configured to user-defined values. The strapping pins are latched on the rising edge of the nRESET signal. See [Figure 2-1](#) and [Table 2-1](#) for the description of the strapping pins and the corresponding configurations and modes. 1 kΩ or 4.7 kΩ resistors are recommended for pulling the strapping high or low, accordingly. Note that all GPIO pins have internal pull-up.

FIGURE 2-1: STRAPPING PINS

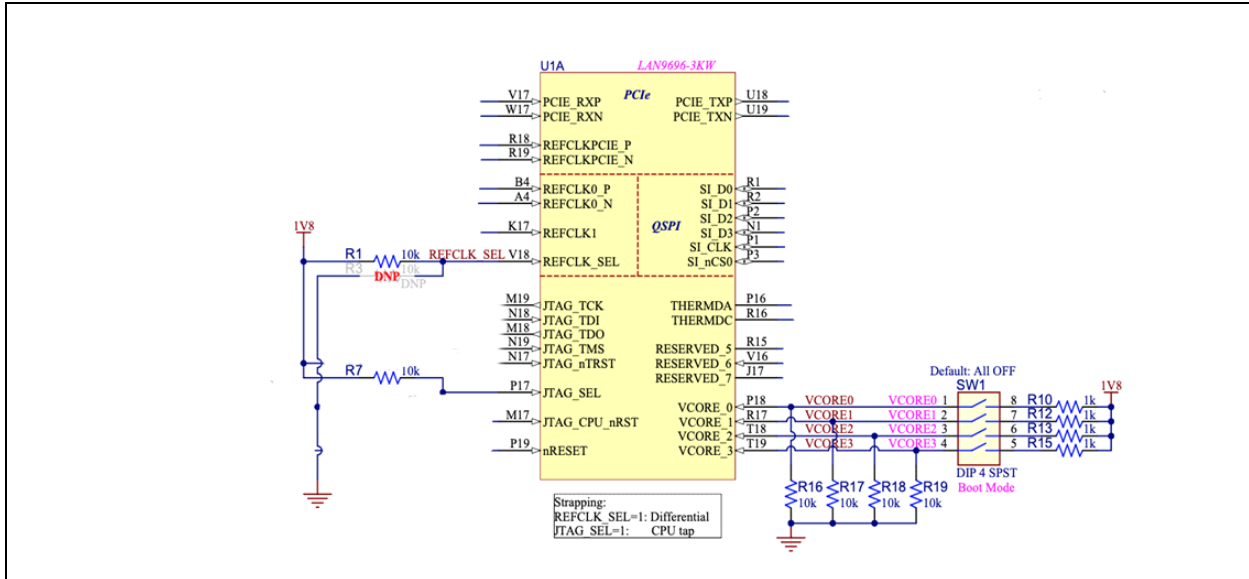


TABLE 2-1: PIN STRAPPING DESCRIPTIONS

Pin	Description
VCORE[3:0]	Bootloader configuration straps 3-0 0000: Boot from eMMC. Boot trace on FLEXCOM0 0001: Boot from QSPI0. Boot trace on FLEXCOM0 0010: Boot from SD Card. Boot trace on FLEXCOM0 0011: Boot from eMMC 0100: Boot from QSPI0 0101: Boot from SDCard 1000: TF-A Monitor on FLEXCOM0. OTP-emulation loaded from QSPI-HS 1010: TF-A Monitor on FLEXCOM0 1011: TF-A Monitor on FLEXCOM0-HS, 921600 Kbps 1101: Boot from QSPI-HS 1111: Internal CPU is disable (default). QSPI0/SI is configured as SPI client. (See Note 1.)
REFCLK_SEL	Core reference clock select. Low: REFCLK1 High: REFCLK0_N/REFCLK0_P (See Note 2.)
JTAG_SEL	Selects between CPU and Test TAP controller. High: Selects CPU TAP Controller Low: Selects Test TAP Controller

Note 1: TF-A (Trusted Firmware for ARM) and boot trace UART modes are default configured to 115200 baud/8/N.

2: REFCLK0_N/REFCLK0_P is always used as the SerDes reference clock. It is therefore likewise regarded as being for SyncE. REFCLK1 can be used if PTP/1588 and SyncE are in separate time domains. In this case, REFCLK1 can be regarded as an input clock reference for PTP/1588.

3.0 POWER/GROUNDING

3.1 Power Supplies

The chip has a number of power domains—mainly related for complying to different IO standards. Refer to [Figure 3-1](#) and [Table 3-1](#).

FIGURE 3-1: POWER SUPPLY PINS

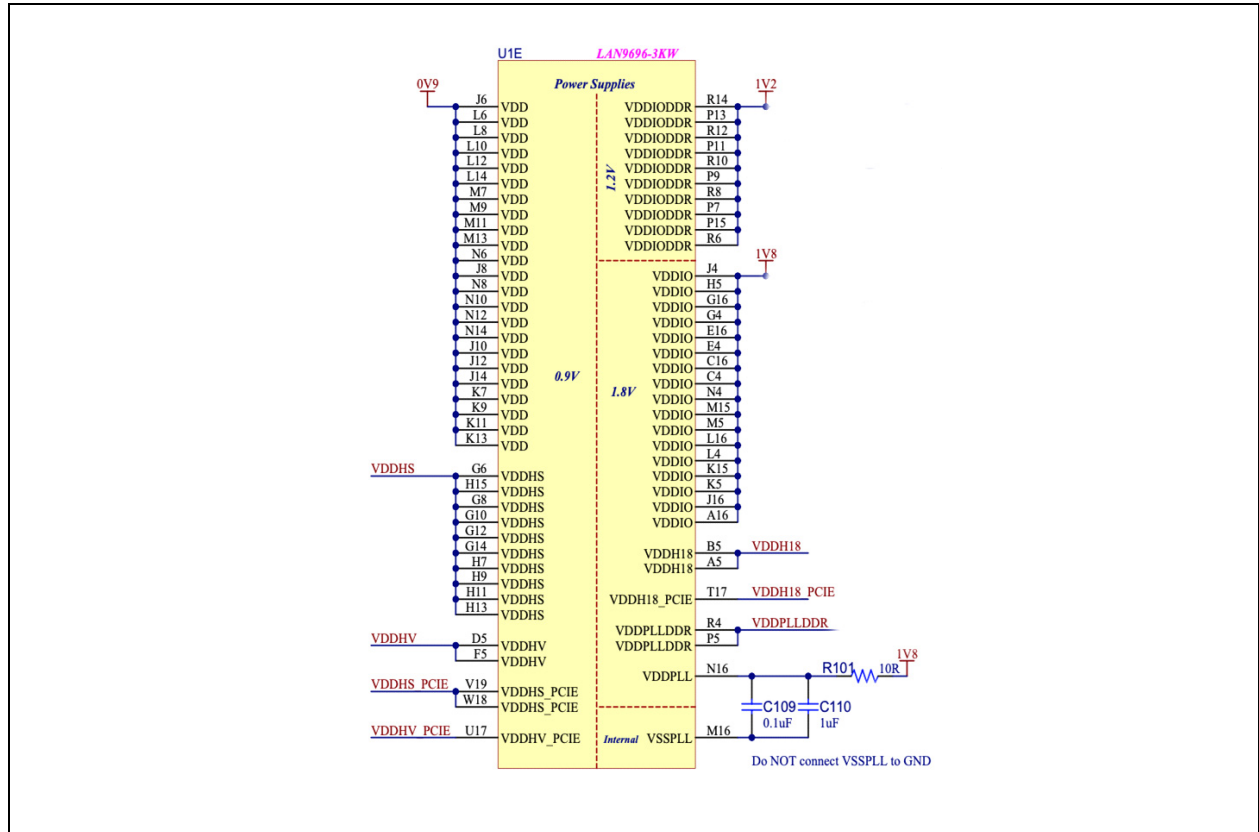


TABLE 3-1: POWER SUPPLY DESCRIPTION

Pin	Voltage	Description
VDD	0.9V	+0.9V digital core logic power supply input
VDDHS	0.9V	+0.9V SerDes driver power supply
VDDHV	0.9V	+0.9V SerDes PLL(VCO) power supply
VDDH18_PCIE	1.8V	+1.8V PCIe® SerDes power supply input
VDDHV_PCIE	0.9V	+0.9V PCIe® SerDes PLL(VCO) power supply
VDDHS_PCIE	0.9V	+0.9V PCIe® SerDes driver power supply
VDDIODDR	1.2V, 1.35V or 1.5V	+1.2V, 1.35V, or 1.5V DDR I/O power supply
VDDPLLDDR	1.8V	+1.8V DDR PLL power supply
VDDH18	1.8V	+1.8V SerDes power supply input
VDDIO	1.8V	+1.8V LVCMOS I/O power supply
VDDPLL	1.8V(Analog)	+1.8V system PLL power supply
VSSPLL	Analog	System PLL return, connected internally to VSS
VSS	GND	Common ground

LAN9691/2/3

- Each power supply voltage should have both bulk and high-frequency decoupling capacitors. The recommended bulk capacitors are 10 μ F (minimum), and high-frequency decoupling capacitors are 0.1 μ F (minimum) and maximum 0402 package. Surface-mount decoupling capacitors should be placed as close to the switch power supply pins as possible.
- Analog supplies must be isolated from the remaining board supplies using ferrite beads.
- If the DDR is not used, **VDDIODDDR** can be connected to **VSS**. **VDDPLLDDR** must still be connected to 1.8V.
- Even if the PCIe is not used, the **VDDHS_PCIE** and **VDDHV_PCIE** rails must be powered.

3.2 Power Supply Sequencing

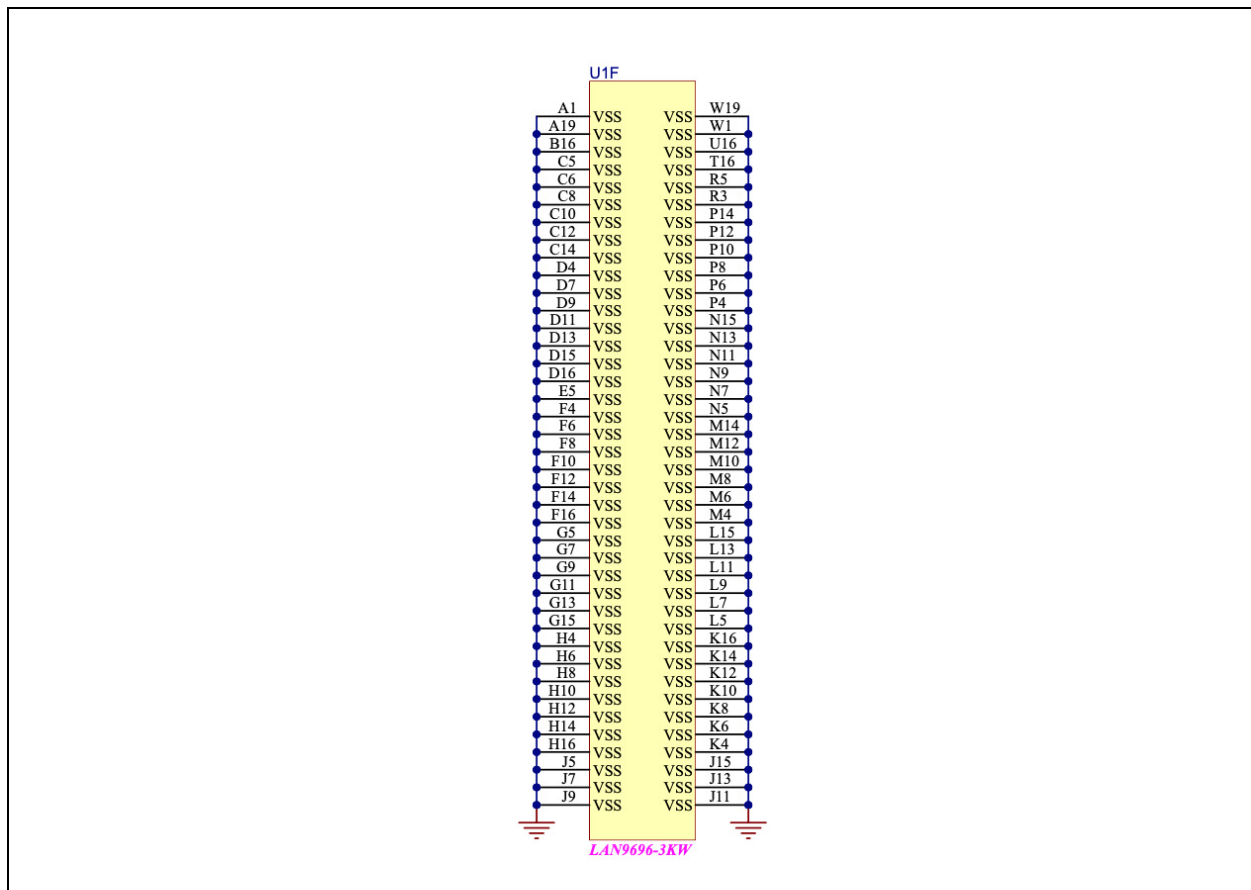
- The **nRESET** and **JTAG_nTRST** input pins must be held low until all power supply voltages have reached their recommended operating condition values.
- The overall strategy is not to let a higher-powered supply feeding lower-level powered grids through the internal protection diode network. Therefore, the general power sequence is: 0.9V -> 1.2V/1.35V/1.5V -> 1.8V.
- Maximum of 500 ms from 0.9V to 1.8V.

3.3 Ground

- Create at least one unbroken ground plane (**GND**) layer in the stack-up.
- Typically, there should be a solid ground plane beneath each of the outer layers to ensure that the signals on these layers have a good reference plane.
- Shield ground on signal layers should serve as “quiet” ground for PHY copper media signals and SFP cages. It couples capacitively to the ground planes, providing a low-impedance return path for high-frequency noise.
- **VSSPLL** must not be connected to system ground. See [Figure 3-1](#).

See [Figure 3-2](#) for the reference circuit.

FIGURE 3-2: GROUND PINS



4.0 REFERENCE CLOCKS

The switch has a number of reference clocks to serve its different parts. See [Figure 4-1](#) and [Table 4-1](#).

FIGURE 4-1: REFERENCE CLOCKS

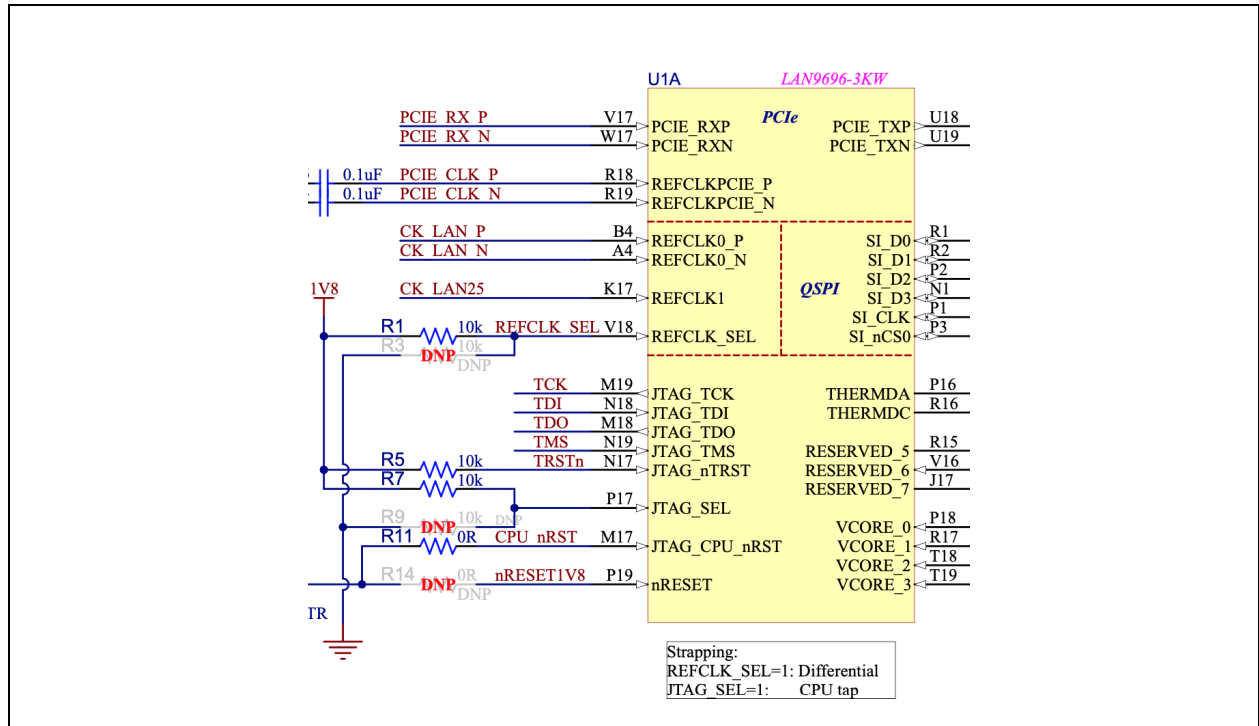


TABLE 4-1: REFERENCE CLOCK DESCRIPTIONS

Pin	Buffer Type	Description
REFCLK0_P/N Pin B4/A4	CML	This is the reference clock used for the SerDes interfaces and, optionally, the core clock reference. The frequency of REFCLK0_P/N is fixed at 156.25 MHz. A quality oscillator is therefore recommended. Check P/N must be AC-coupled. LVPECL or CML signals are recommended.
REFCLK1 Pin K17	LVC MOS, 1.8V	This is a 25 MHz single-ended core clock reference when REFCLK_SEL is low. It is not used when REFCLK_SEL is high.
REFCLKPCIE_P/N Pin R18/R19	CML	This is a PCIe® SerDes reference clock. If unused, the clock input can be floating or pulled to VDDIO18. The reference clock is fixed to 100 MHz +/- 300 ppm. The PCIe reference clock is typically supplied from the host. Check P/N must be AC-coupled. LVPECL or CML signals are recommended.

Note 1: The data sheet requirements for maximum clock jitter must be accounted for in the board design when selecting clock source (oscillator) and clock distribution (buffer) components.

4.1 Differential Clock

- REFCLK0_P/N and REFCLKPCIE_P/N are CML type. Each P/N pin of the clock input has an internal 50R termination to the common-mode voltage input, which results in a 100Ω differential termination from P to N.
- REFCLK0_P/N and REFCLKPCIE_P/N have a common-mode input voltage of 0.9V. The differential peak-to-peak in/out swing is 600 mV ~ 1600 mV.

5.0 CPU SYSTEM

The switch can be managed by either the internal VCORE CPU or an external CPU. The selection between internal or external CPU is based on the **VCORE[3:0]** strapping pins described in [Section 2.3, "Strapping Pins"](#).

5.1 Internal CPU Auto Boot Mode

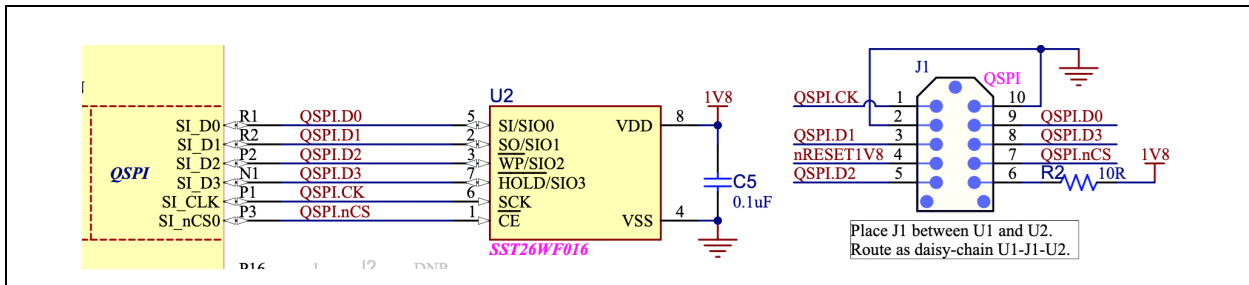
- The internal VCORE CPU system is based on an embedded ARM Cortex A7 microprocessor running at 1 GHz with support for multiple memories, such as 16-bit DDR4, DDR3, DDR3L, Quad SPI, eMMC Flash, and SD card.
- When the VCORE CPU is enabled, it automatically boots up from an internal boot ROM. The firmware uses **VCORE[3:0]** to determine the second stage boot loader options. The secondary bootloader can be stored in eMMC Flash, QSPI0 NOR Flash, or SD Card.
- The internal VCORE CPU optionally provides secure boot and secure execution of firmware based on Microchip customers' Root-of-Trust manufacturing process. The VCORE CPU can also boot into TF-A Monitor and therefore initializes the UART interface located on Flexcom0, so that the secondary bootloader can be loaded from an external device into the LAN9691/2/3's CPU system.

5.2 QSPI Interface

- The LAN9691/2/3 supports two QSPI interfaces. The Quad Serial Peripheral Interface, QSPI, is a synchronous serial data link that provides communication with external devices. 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.
- QSPI0 is selected to be the secondary bootloader, when **VCORE[3:0]** is 0001, 0100, 1000, and 1101. In that case, QSPI0 is configured in Serial Memory mode. The QSPI0 then acts as a serial Flash memory controller. QSPI0 is therefore used for reading data from the serial Flash memory, allowing the CPU to execute code from it (XIP execute in place). The QSPI0 can also be used to control the serial Flash memory (i.e., Program, Erase, Lock, and so on) by sending specific commands. In this mode, the QSPI is compatible with single-bit SPI, Dual SPI, Quad, and Octal SPI protocols.

[Figure 5-1](#) shows the connection between the QSPI0 and a 16 Mb NOR Flash supporting x1/x2/x4 Serial Peripheral Interface (SPI) Protocol and SQI protocol. New boards use the 8 MB version of the NOR Flash device.

FIGURE 5-1: QSPI0 CONNECTION TO NOR FLASH



- A Flash programming header in the Microchip reference design is used to support on-PCB Flash programming.
- It is recommended to force nRESET1V8 low during Flash programming, so that the switch is held in Reset and does not drive the QSPI signals. Otherwise, both the switch and the Flash Programmer drive the QSPI signals, and that will cause Flash programming failures. On the Microchip reference board, this can be done by either shorting J5 or tying pin 4 of J1 to GND. Pulling **VCORE[3:0]** to a different mode than booting from the QSPI0 can also ensure the switch does not drive the QSPI signals attached to the Flash.
- The client SPI interface shares the same pins of the QSPI0/SPI interface. The client SPI is enabled when **VCORE[3:0]** are set to 1111 (default value). In that mode, the internal CPU is disabled. External CPU (with an SI host interface) is required to configure the switch through the SPI client interface.

[Table 5-1](#) shows the pin definition of the QSPI0/SPI and SPI client interface.

TABLE 5-1: QSPI0 DEFINITION IN DIFFERENT MODES

Pin Name	Quad SPI Operation	Dual SPI Operation	Single-bit SPI Operation	Client SPI Operation
SI_CLK	QSPI clock output	Dual SPI clock output	SPI clock output	Client SPI clock input
SI_nCS0	QSPI chip select output	Dual SPI chip select output	SPI chip select output	Client SPI chip select input
SI_D0	Data input/output 0	Data input/output 0	MOSI output	MISO output
SI_D1	Data input/output 1	Data input/output 1	MISO input	MOSI input
SI_D2	Data input/output 2	Not used	Not used	Not used
SI_D3	Data input/output 3	Not used	Not used	Not used

- The QSPI0 has dedicated HW pins, while the QSPI1 uses pins that are overlaid on the GPIO pins in Alternate mode. Furthermore, QSPI0 also supports Octal SPI mode with either data input/output pins. QSPI0_DQS and QSPI0_D4/D5/D6/D7 are overlaid on GPIO 20~GPIO24 in Octal mode. Note that QSPI0_D4-D7 follow after QSPI1_D0-D3, which could seem that D4-D7 belong to QSPI1, but this is not the case. See [Table 12-1](#).

The location of QSPI1 on the GPIO pins with regards to Alternate mode selection is shown in [Table 5-2](#).

TABLE 5-2: QSPI1 INTERFACE LOCATION TO GPIO PINS

Interface	Overlaid	SCK	nCS	IO[3]	IO[2]	IO[1]	IO[0]
QSPI_1	ALT2	GPIO14	GPIO15	GPIO19	GPIO18	GPIO17	GPIO16
QSPI_1	ALT4	GPIO25	GPIO26	GPIO30	GPIO29	GPIO28	GPIO27

5.3 DDR3/3L/4 Memory Controller–SDRAM Interface

- The memory controller operates at a maximum of 600 MHz to support the speed of 2400 MT/s on the DDR interface. It supports two-byte lanes (16-bit data bus). The memory controller supports SDRAM with data width = x8 or x16. The maximum amount of physical memory that can be attached to the controller (per byte lane) is 1 GB and therefore a total of 2 GB. See [Table 5-3](#).

TABLE 5-3: DRR CONTROLLER INTERFACES

DDR Interface	In/Out	Remarks
DDR_A[17:0]	Out	Address bus
DDR_BA[1:0], DDR_BG[1:0]	Out	Bank address, bank group
DDR_Ck_t/c, DDR_CKE[1:0]	Out	DDR_CK_t/c must be routed as 100Ω differential pair.
DDR_3WE_n/ DDR_3CAS_n/ DDR_3RAS_n	Out	Active-low command signals. Write Enable, Column Address Strobe, and Row Address Strobe.
DDR_DQ[7:0], DDR_DM0	In/Out	Data bus. Maximum skew is 1 mm within each byte lane. On-die termination of DQ signals should be used as applicable. Byte lanes may be swapped freely. Bit swizzling is allowed for all bits within a byte lane. However, the first bit in each byte lane must be connected directly for write leveling training to succeed. For a two-lane design, these first bits are DQ0 and DQ8.
DDR_DQ[15:8], DDR_DM1	In/Out	
DDR_DQS[1:0]_P/N	Out	DDR_DQSx_P/N must be routed as 100Ω differential pairs.
DDR_CS_n[1:0]	Out	Active-low chip select.
DDR_ODT[1:0]	Out	On-die termination.
DDR_PAR	Out	Parity signal for DDR4
DDR_ALERT_n	Out	DDR4 active-low alert. Pull to VDDIODDR.

LAN9691/2/3

TABLE 5-3: DDR CONTROLLER INTERFACES (CONTINUED)

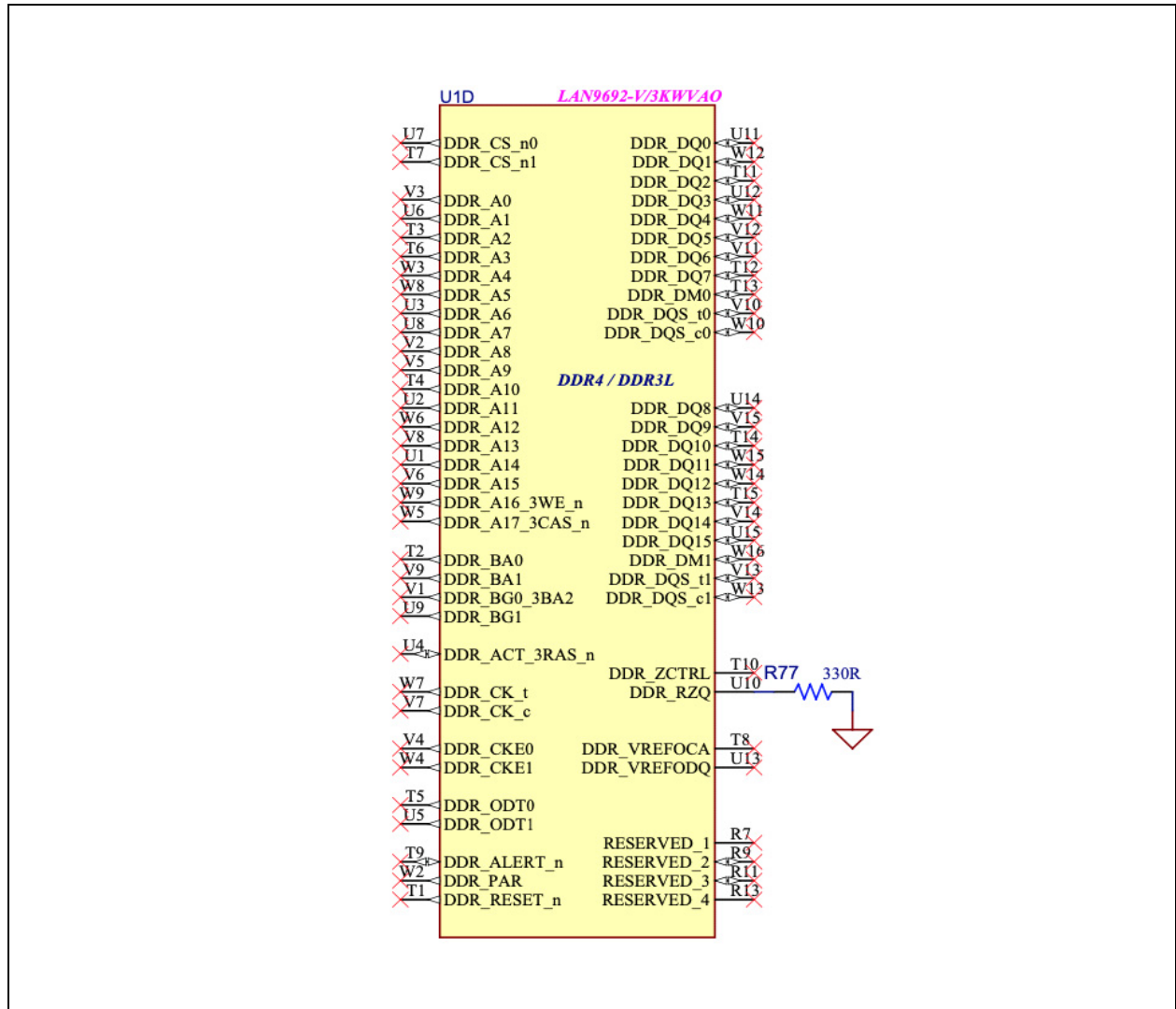
DDR Interface	In/Out	Remarks
DDR_RESET_n	Out	DDR active-low Reset output.
DDR_RZQ	Out	Bias: 240Ω to VSS. Must always be connected to VSS through a resistor even when the DDR interface is unused.
DDR_ZCTRL	Out	The pin should be left floating.
DDR_VREFOCA, DDR_VREFODQ	Out/Analog	The pin should be left floating.

- Command lane signals must use midpoint termination at the end of the trace. Use a qualified Sink/Source DDR Termination Regulator (like MIC5166) on the Command lane signals.
- The 1 mm length matching recommendation only applies to the byte lanes. The length matching requirement for the command/address bus can be relaxed to 2 mm to 3 mm, as it is SDR, not DDR.
- Trace impedance for the DDR traces should be within the range of 40Ω to 80Ω.
- Use a high-frequency decoupling capacitor for every **VDD** and **VDDQ** pin of the DDR device. One 22 μF bulk capacitor on **VDD** of the DDR device is also recommended.
- It is recommended to use the same DDR device as the reference designs use, or designers must ensure that the substitute is fully compliant. Otherwise, fine tuning of DDR parameters might be needed through FWU. Microchip reference boards typically use a single chip, DDR4–AS4C512M16D4, which is a 512M*16 bit = 1 GB memory.
- It is strongly recommended to perform simulations of the DDR interface for the specific application and topology. There will be no support from the Microchip SW team to make it run in case of issues. The FWU can be used for DDR configuration and/or debugging. (FWU is a Firmware Utility provided by Microchip using the TFA port and boot mode.)
- Check that the switch DDR supply, Vdd_IODDR, matches the DDR RAM; 1.5V on DDR3, 1.35 on DDR3L, and 1.2V on DDR4.
- The LAN9691/2/3 uses in-line ECC where part of the DDR memory is used for the ECC code. This means approximately 12% of the memory is reserved for ECC. Therefore, the LAN9691/2/3 only needs to use one x16 DDR RAM.

Note: It is recommended to use the same DDR4 DRAM utilized in the reference designs. Otherwise, further DDR tuning may be required. Alliance AS4C512M16D4 and Micron MT40A512M16 are suitable options. Therefore, any device that matches the same internal memory organization as these devices and runs a minimum of 1.2 GHz is acceptable.

DDR memory is not necessary when the LAN9691/2/3 is managed by an external CPU or when running VelocityDrive-SP (previously called LMStAX by SW development) on the internal CPU. In that case, the **VDDIODDDR** can be connected to **VSS**. **VDDPLDDR** must still be connected to 1.8V and can be left unfiltered. See [Figure 5-2](#) and [Figure 5-3](#) for the recommended settings on the DDR interface and related power supplies when DDR is not used. R77 does not need to be 240Ω. It may be any value from 200Ω to 1 kΩ.

FIGURE 5-2: UNUSED DDR INTERFACE



The LAN9601/2/3 supports two Secure Digital Multimedia Card Controllers, SDMMC0 and SDMMC1. The Secure

SD Interface	In/Out	Overlaid	GPIO	Remarks
--------------	--------	----------	------	---------

TABLE 5-4: SDMMC SIGNALS IN SD CARD MODE MAPPING TO GPIO PINS (CONTINUED)

SD Interface	In/Out	Overlaid	GPIO	Remarks
SD1_CMD	Out	ALT4	GPIO33	Command signal must be pulled high with 4.7 kΩ to 10 kΩ resistor.
SD1_CLK	Out	ALT1	GPIO34	Clock
SD1_DAT[3:0]	In/Out	ALT1	GPIO[38:35]	Data bit 3:0
CARD1_nDETECT	In	ALT1	GPIO41	Active-low detect signal
CARD1_WP	In	ALT1	GPIO40	Write protect signal
SD1_1V8SEL	Out	ALT1	GPIO39	SD card signal voltage selection

Table 5-5 shows the GPIO pin mapping in eMMC mode.

TABLE 5-5: SDMMC SIGNALS IN EMMC MODE MAPPING TO GPIO PINS

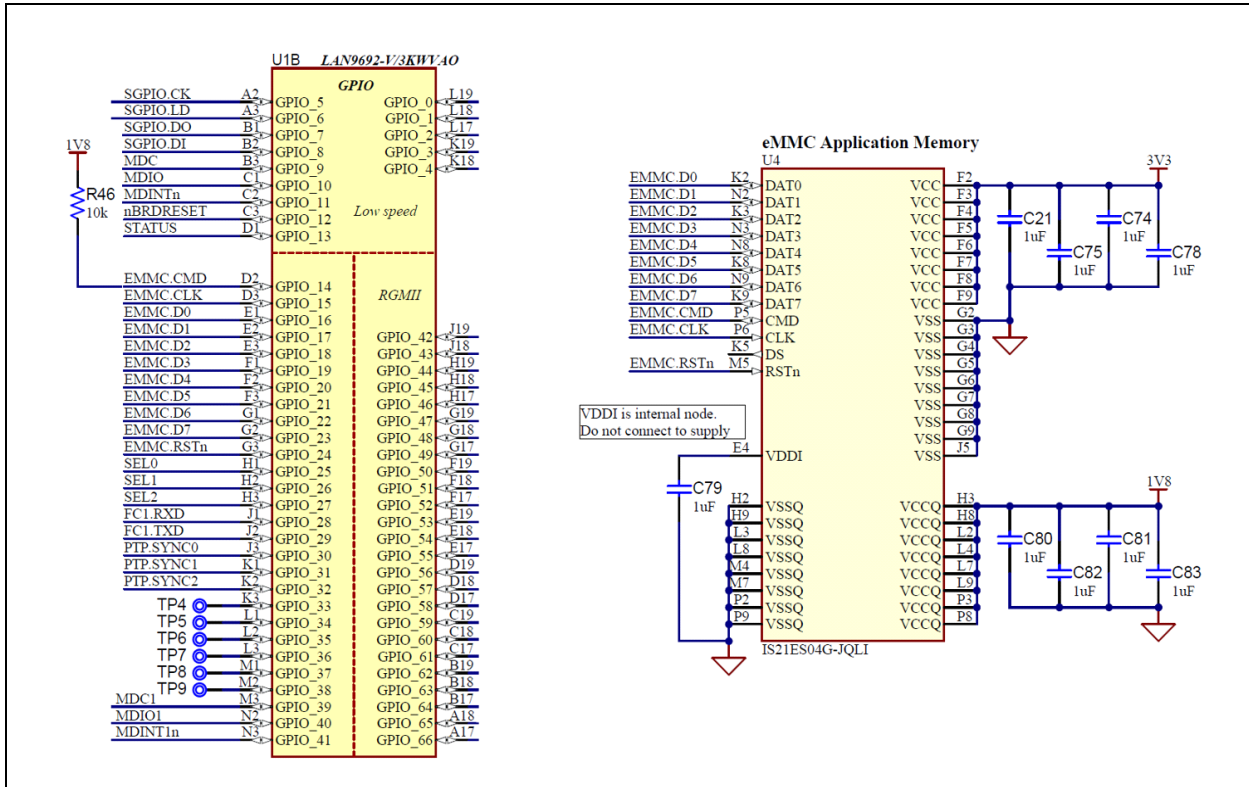
eMMC Interface	In/Out	Overlaid	GPIO	Remarks
MMC0_CMD	Out	ALT1	GPIO14	Command signal must be pulled high with 4.7 kΩ to 10 kΩ resistor.
MMC0_CLK	Out	ALT1	GPIO15	Clock
MMC0_DAT[7:0]	In/Out	ALT1	GPIO[23:16]	Data bit 7:0
MMC0_RSTN	Out	ALT1	GPIO24	Active-low Reset signal
MMC1_CMD	Out	ALT1	GPIO33	Command signal must be pulled high with 4.7 kΩ to 10 kΩ resistor.
MMC1_CLK	Out	ALT1	GPIO34	Clock
MMC1_DAT[3:0]	In/Out	ALT1	GPIO[38:35]	Data bit 3:0
MMC1_RSTN	Out	ALT4	GPIO39	Active-low Reset signal

- SDMMC0 is selected to be the secondary bootloader, when **VCORE[3:0]** is 0000, or 0011. In that case, SDMMC0 is configured in eMMC Boot Operation mode. In Boot Operation mode, the processor can read boot data from the eMMC device by keeping the CMD line low after power-on before issuing the CMD1. The data can be read from either one of the boot partitions or the user area according to **BOOT_PARTITION_ENABLE** in the Extended CSD register.
- SDMMC0 is also selected to be the secondary bootloader, when **VCORE[3:0]** is 0010, or 0101. In that case, SDMMC0 is configured in 4-bit SD card Boot Operation mode.

LAN9691/2/3

When the LAN9691/2/3 is configured to eMMC Boot mode, the SDMMC Controller 0 connects to a 4 GB or 8 GB eMMC NAND Flash, which stores the firmware and the application configurations. See reference connection in [Figure 5-4](#).

FIGURE 5-4: EMMC APPLICATION MEMORY AND LAN969X



5.5 External CPU Mode

An external CPU can be connected to the LAN9691/2/3 device through the PCIe® client interface or the SPI client interface. (See [Table 5-1](#).) Through these interfaces, the external CPU has access to the complete set of registers in LAN9691/2/3.

5.5.1 SPI CLIENT INTERFACE

- The SPI client interface is enabled when **VCORE[3:0]** is 1111. The SPI client interface shares the same pins as the QSPI0 boot controller interface. Among the chip selection signals, only **SI_nCS0** supports this mode.
- When the external CPU uses the SPI client interface to read the Switch registers, the switch must prepare read data after receiving the last address bit. The access time of the register, that is read, must be satisfied before shifting out the first bit of read data, so there must be a delay between the last address bit and the first data bit. The following are some ways to satisfy the needed delay:
 - Use a **SI_CLK** with a period of minimum twice the access time for the register target. For example, for normal switch core targets: $1/(2 \times 400 \text{ ns}) = 1.25 \text{ MHz}$ (maximum).
 - Pause the **SI_CLK** between shifting of serial address bit 0 and the first data bit with enough time to satisfy the access time for the register target.
 - Configure the device to send out padding bytes before transmitting the read data to satisfy the access time for the register target. For example, one dummy byte allows enough read time for the SI clock to run up to 16 MHz in a single initiator system.

Refer to [Section 5.2, "QSPI Interface"](#) for the description of the SPI Client interface signals.

5.5.2 PCIE® GEN2 END POINT INTERFACE

- The LAN9691/2/3 implements a single-lane PCIe® Gen2 End Point controller that can be hooked up to any PCIe-capable system. The PCIe client interface can be used by an external CPU to read or write to the Switch registers. The PCIe end point interface needs to be enabled by software in any of the booting modes. Through PCIe, an external CPU can access (and control) the device.
- Ethernet frames can be injected and extracted using registers, or the device can be configured to DMA Ethernet frames autonomously to/from PCIe memory.
- The PCIe end point interface consists of the following dedicated pins:
 - PCIe_Rx P/N
 - PCIe_Tx P/N
 - PCIeCLKP P/N
- Optional PERST# signal is available from the configurable GPIO pins and must be enabled by software. See [Table 5-6](#).

TABLE 5-6: PCIE ADDITIONAL SIGNALS MAPPING TO GPIO PINS

Interface	In/Out	Overlaid	PERST#
PCIe	In	ALT3	GPIO0
	In	ALT3	GPIO36

- AC coupling is recommended on the PCIe link between the switch and the external CPU and also on the differential PCIe reference clock.

6.0 PORT CONFIGURATION

6.1 SerDes Interface

- The device supports ten identical physical SerDes interfaces (denoted S0 to S9). Each of the SerDes has a maximum speed of 10 Gbps. All SerDes interfaces are available to all SKUs in the LAN9691/2/3 family. Different SKUs only have different total bandwidth limitation as shown in [Table 6-1](#).
- The SerDes can be configured in different speeds—100M, 1G, 2.5G, 5G, and 10G. It also supports different protocols, such as BX, KX, FX, KR, SFI, SGMII, QSGMII, QXGMII, DXGMII, and USXGMII. Refer to the data sheet for the complete list of speed and protocols supported by each SerDes interface.
- The SerDes Macros have 100Ω internal termination and biasing. Check the use of AC-coupling for DC level adjustment and EMI suppression.
- When the SerDes interfaces are used to connect to an external PHY, AC coupling is recommended between the switch and the external PHY.
- Unused SerDes interfaces must be left floating.
- Take caution in the naming of the **R_x** and **T_x** pairs on the switch and PHY. Typically, **R_x** is the output pair, and **T_x** is the input pair on a PHY. This is different from how they are named on a switch. Make sure that the output pair from the switch is connected to the input pair of the PHY and vice versa.
- It is highly recommended to simulate all 10G traces and route them on the top or bottom layer to avoid reflections from stubs when using vias and connectors. Trace bends of 90 degrees should be avoided.

6.2 Port Mapping

- The device has 30 logical port modules (denoted D0 to D29) that connect to ten physical SerDes and two RGMII interfaces. Each SKU in the LAN9691/2/3 family has a unique overall maximum bandwidth limitation as shown in [Table 6-1](#).

TABLE 6-1: BANDWIDTH LIMITATIONS

Part Numbers	LAN9691	LAN9692	LAN9693
Total I/O bandwidth	46G	66G	102G

- Port 28 (D28) has a fixed allocation to the RGMII 0, and port 29 has a fixed allocation to RGMII 1. The remaining 28 ports can be connected to the remaining ten SerDes using QSGMII or USXGMII extenders in various multiplexing scenarios. Alternatively, ten of the ports can have a 1:1 mapping to the ten SerDes.
- [Table 6-2](#) specifies the maximum speed of each internal port module and the possible mapping to the ten SerDes. In addition, [Table 6-2](#) indicates that only port D0, D4, D8, D12, D16, D20, D24, D25, D26, and D27 can connect directly to the SerDes as a single-port interface. Those single-port interfaces can be configured to different line speeds, and different port modules (MAC+PCS) are needed depending on the speeds supported. A port supporting 5 Gbps or 10 Gbps as the maximum line speed has a DEV5G or DEV10G to support the highest line speed. In addition, both ports have a shadow DEV2G5 port module to support the lower speeds (10/100/1000/2500 Mbps).
- Some of the ports can only be enabled in one of the multiplexing modes—QSGMII (4*1G), 10G-QXGMII (4*2.5), or 10G-DXGMII(2*5G) like D9. On the other side, D8 can work at a 10G maximum speed. It can be enabled in QSGMII2 where D8 will be one of the 1G ports. It can also be enabled in 10G-QXGMII0 where D8 will work as one of 2.5G ports in the four-port group. It can also be enabled in DXGMII0 where it is one of the two 5G ports in the group.

Note: DEV10G supports 10GBase-K/R, but also supports 5GBase-K/R, and a DEV2G5 supports SGMII+/2500Base-X, as well as SGMII/1000Base-X and 100Base-FX.

TABLE 6-2: LOGICAL PORT ASSIGNMENTS TO INTERNAL DEVICES AND SERDES MACROS

Logical Port	DEV2G5	DEV5G	DEV10G	DEV RGMII	SerDes Interface	QSGMII	10G QXGMII	10G DXGMII
D28	—	—	—	RGMII0	—	—	—	—
D29	—	—	—	RGMII1	—	—	—	—
D0	0	—	0	—	S0	QSGMII0	—	—
D1	1	—	—	—	—		—	—
D2	2	—	—	—	—		—	—
D3	3	—	—	—	—		—	—
D4	4	—	1	—	S1	QSGMII1	—	—
D5	5	—	—	—	—		—	—
D6	6	—	—	—	—		—	—
D7	7	—	—	—	—		—	—
D8	8	—	2	—	S2	QSGMII2	10G-QXGMII0	10G-DXGMII0
D9	9	0	—	—	—			—
D10	10	—	—	—	—			—
D11	11	—	—	—	—			—
D12	12	—	3	—	S3	QSGMII3	10G-QXGMII1	10G-DXGMII1
D13	13	1	—	—	—			—
D14	14	—	—	—	—			—
D15	15	—	—	—	—			—
D16	16	—	4	—	S4	QSGMII4	10G-QXGMII2	10G-DXGMII2
D17	17	2	—	—	—			—
D18	18	—	—	—	—			—
D19	19	—	—	—	—			—
D20	20	—	5	—	S5	QSGMII5	10G-QXGMII3	10G-DXGMII3
D21	21	3	—	—	—			—
D22	22	—	—	—	—			—
D23	23	—	—	—	—			—
D24	24	—	6	—	S6	—	—	—
D25	25	—	7	—	S7	—	—	—
D26	26	—	8	—	S8	—	—	—
D27	27	—	9	—	S9	—	—	—

- Check if the customer application is not exceeding the maximum number of port speeds for the used switch derivative. The maximum numbers are given in [Table 6-3](#).

TABLE 6-3: MAXIMUM PORT SPEEDS SUPPORTED

Port Mapping	LAN9691	LAN9692	LAN9693
Maximum bandwidth [Gbps]	46	66	102
Maximum number of ports	30	30	30
Maximum number of 1G ports	30	30	30
Maximum number of 1G SGMII ports	10	10	10

TABLE 6-3: MAXIMUM PORT SPEEDS SUPPORTED (CONTINUED)

Port Mapping	LAN9691	LAN9692	LAN9693
Maximum number of 100 FX ports	10	10	10
Maximum number of 2.5G ports	10 direct or 16+2 through 10g-QXGMII	10 direct or 16 + 6 through 10g-QXGMII	10 direct or 16 + 6 through 10g-QXGMII
Maximum number of 5G ports	9 direct or 8+1 through 10G-DXGMII	10 direct or 8+5 through 10g-DXGMII	10 direct or 8+6 through 10g-DXGMII
Maximum number of 10G ports	4	6	10

6.3 MAC-to-MAC Connections

- The SerDes macros have 100Ω internal termination and biasing. Check the use of AC-coupling for DC level adjustment and EMI suppression. Make sure that the signal direction and polarity are correct. The SerDes macros can invert P and N to ease the layout.
- It is highly recommended to simulate all 10G traces and route them on the top or bottom layer to avoid reflections from stubs when using vias and connectors. Trace bends of 90 degrees should not be made.
- Signal detection can be omitted since MAC-to-MAC or backplane connection is always connected.

6.4 SFP Ports

- It is important to track the individual SFP signals required by the SFP MSA. The switch does not need to control all signals, but the signals must all be pulled up to 3V3. SFP modules have built-in AC caps mandatory by the SFP MSA. See [Table 6-4](#).

TABLE 6-4: SFP SIGNALS SEEN FROM CASING

SFP Pin	SFP In/Out	Remark
RX	Out	No AC-coupling as it is built into the module
TX	In	No 100Ω differential termination across P and N
SCL	In	Should have a pull-up resistor
SDA	In/Out	Should have a pull-up resistor
TX_FAULT	Out	4.7 kΩ to 10 kΩ pull-up resistor to 3V3
TX_DISABLE	In	SFP modules have an internal pull-up resistor, so no pull is needed on the board. If controlled, additional circuitry may be required to avoid powering the Tx LED during Reset state.
MODULE_DETECT	Out	4.7 kΩ to 10 kΩ pull-up resistor to 3V3
RATE_SEL1	In	RATE_SEL signals do not need a pull, so they may be left unconnected.
RATE_SEL2	In	RATE_SEL signals do not need a pull, so they may be left unconnected.
RX_LOS	Out	4.7 kΩ to 10 kΩ pull-up resistor to 3V3

- The switch application depends on the use of the MODULE_DETECT to initiate the I²C reading of the SFP MSA ROM to automatically determine the SFP module type and speed.
- Most optical modules do not squelch the Rx signal when losing signal from link partner. Therefore, the Rx signal will still be active and prevent the switch internal signal detect circuitry to denote loss of signal to the PCS layer. It is therefore crucial to route the SFP RX_LOS signal to the switch. The switch has two ways to route the RX_LOS to the PCS layer, either as a GPIO input (parallel) or as an SGPIO input (serial).
- The SGPIO controller has 32*4 programmable inputs. Any SGPIO input can be mapped to any port or SerDes loss-of-signal input through the HW_SGPIO_TO_SD_MAP_CFG and HW_SGPIO_TO_SERDES_SD_MAP_CFG, respectively, by writing the global SGPIO index of the selected SGPIO to these registers.
- Alternatively, loss-of-signal can also be taken directly from overlaid functions on the regular GPIOs (named

SFP_SD0~SFP_SD9). (See [Table 6-5](#).) In that case, the input from the SIO controller is ignored. Those SFP_SDx are programmable signal detection signals. To enable parallel signal detect “n,” first configure which indices from the serial GPIO controller must be overwritten by setting GPIO_SD_DEV_MAP[n].G_SD_DEV_MAP and SGPIO_SD_SERDES_MAP[n].G_SD_SERDES_MAP. Then, enable the SFPn_SD function on the GPIOs. Refer to [Table 6-5](#).

TABLE 6-5: PARALLEL SIGNAL DETECT INPUT MAPPING

Overlaid	SFP_SD 0	SFP_SD 1	SFP_SD 2	SFP_SD 3	SFP_SD 4	SFP_SD 5	SFP_SD 6	SFP_SD 7	SFP_SD 8	SFP_S D9
ALT2	GPIO44	GPIO45	GPIO46	GPIO47	GPIO48	GPIO49	GPIO50	GPIO51	GPIO52	GPIO53
ALT1	GPIO57	GPIO58	GPIO59	GPIO60	—	—	—	—	—	—

7.0 PARALLEL MAC INTERFACE

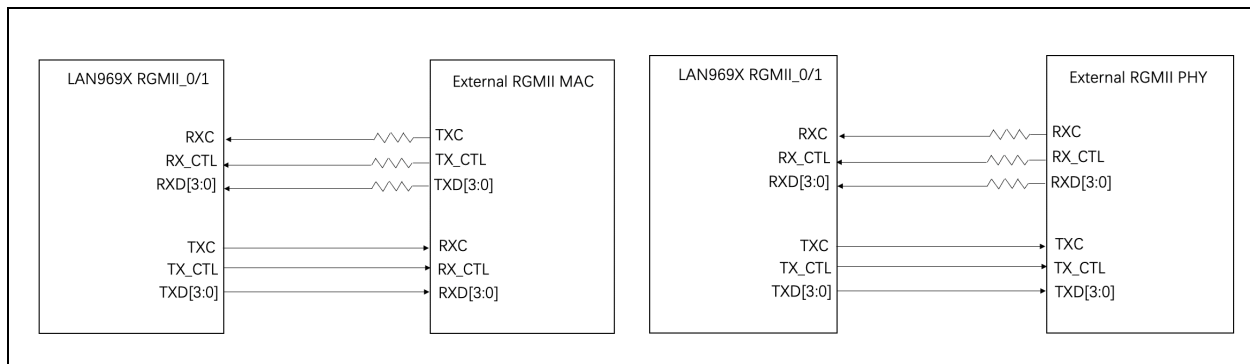
- The LAN9691/2/3 supports two parallel MAC interfaces. The two interfaces are overlaid on the GPIO pins. Each of them can be individually set to one of the following configurations:
 - RGMII: Supports 1000, 100, and 10 Mbps data rates.
 - RMII: Supports 100 and 10 Mbps data rates, inputs or outputs 50 MHz RMII reference clock.
- RGMII0/RMII0 maps to switch port 28, and RGMII1/RMII1 maps to switch port 29.

7.1 RGMII Interface

Follow the design guidelines listed below when connecting the RGMII interface to an external device:

- Route RGMII signals over a continuous ground plane layer for 50Ω impedance control. If the RGMII signals must go through a connector, use a high-speed connector, keep traces short, and ensure that signal integrity is not degraded.
- Match the signal traces within 7 mm (300 mils) for best performance. All transmit signals should be matched and all receive signals should be matched. The transmit and receive signal groups do not need to be matched with each other. Internal delay mode for selected clocks can be enabled to skew clock and data.
- The outputs from the LAN9691/2/3 have internal default ~32Ω series termination. ~16Ω and ~11Ω source termination can also be supported through register configurations. Microchip LAN9691/2/3 EVB does not use external source series termination resistors on the output side of the RGMII, and this has been simulated and proved. Depending on the layout, some designs may need external source termination resistors. [Figure 7-1](#) shows typical RGMII connections.
- Both devices should use the same I/O supply voltage. The LAN9691/2/3 has fixed 1.8V on the RGMII interfaces. The two LAN9691/2/3 RGMII interface connections with external RGMII MAC or PHY devices are shown in [Figure 7-1](#).

FIGURE 7-1: RGMII CONNECTION TO EXTERNAL DEVICE



- The two RGMII interfaces can optionally be configured to support RGMII-ID mode with internal delay for the egress output clock RX_CLK. If internal delay is not enabled by either the LAN9691/2/3 or the link partner sides, a PCB trace delay of approximately 2.0 ns is required.

7.2 RMII Interface

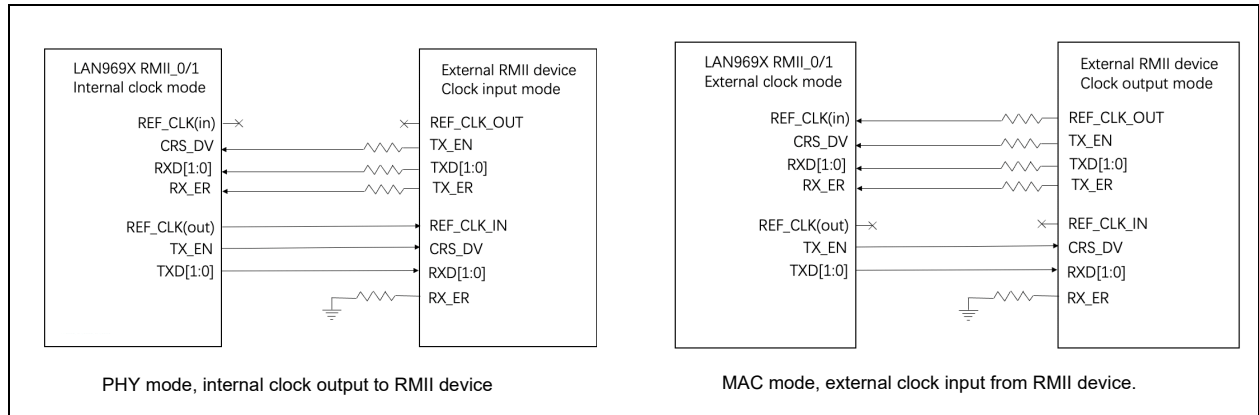
Follow the succeeding design guidelines when enabling the RMII interface to an external device:

- Route RMII signals over a continuous ground plane layer for 50Ω impedance control.
- The outputs from LAN9691/2/3 have internal default ~32Ω series termination. ~16Ω and ~11Ω source terminations can also be supported through register configurations. External series resistors on the output side of the RMII may or may not be needed depending on the layout. It is recommended that customers simulate the RMII signals. [Figure 7-2](#) illustrates typical RMII connections.

- Both devices should use the same I/O supply voltage. The LAN9691/2/3 has fixed 1.8V on the RGMII interfaces.

The two RMII signal connection options (PHY mode and MAC mode) to an external RMII device are shown in [Figure 7-2](#).

FIGURE 7-2: RMII PHY MODE AND MAC MODE



7.3 MAC Interface Location on GPIO Pins in Alternate Mode

Table 7-1 shows which GPIO pins are allocated for the two parallel MAC interfaces.

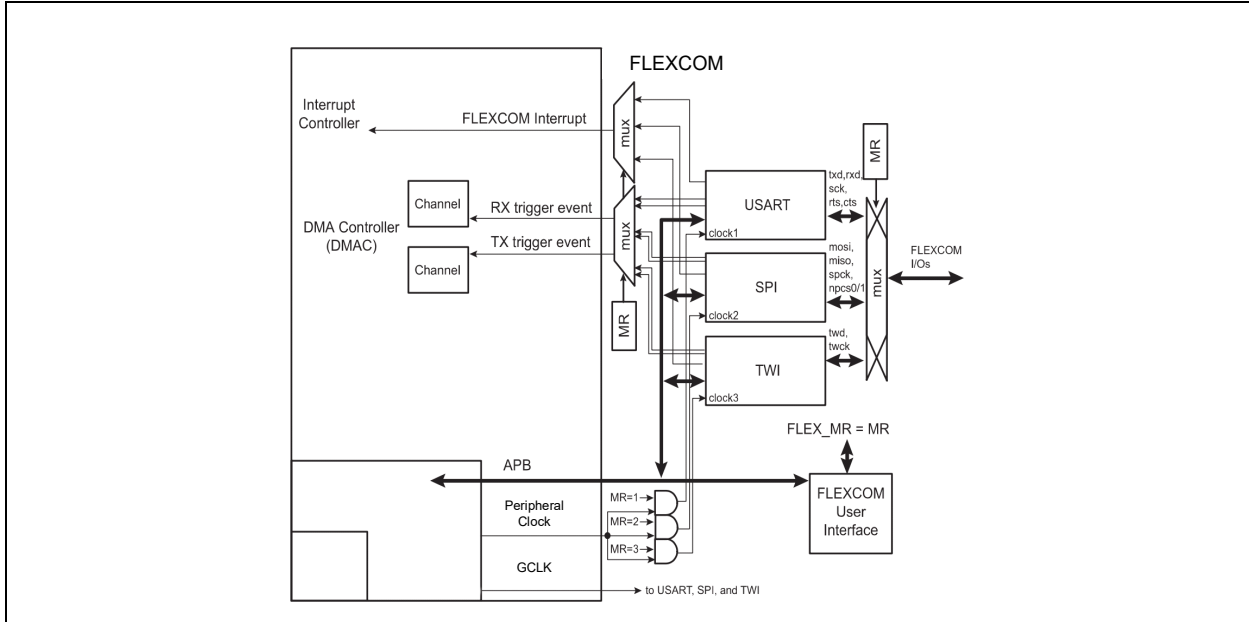
TABLE 7-1: MAC INTERFACE LOCATION TO GPIO PINS

RGMII Interface RMII	RXC REFCLK	RXCTL CRS_DV	RXD[3]	RXD[2] RXER	RXD[1] RXD[1]	RGMII Interface RMII
MAC_RX_0 (input)	GPIO58	GPIO59	GPIO57	GPIO56	GPIO55	GPIO54
MAC_RX_1 (input)	GPIO46	GPIO47	GPIO45	GPIO44	GPIO43	GPIO42
RGMII Interface RMII	TXC REFCLK	TXCTL TX_EN	TXD[3]	TXD[2]	TXD[1] TXD[1]	TXD[0] TXD[0]
MAC_TX_0 (output)	GPIO60	GPIO61	GPIO65	GPIO64	GPIO63	GPIO62
MAC_TX_1 (output)	GPIO48	GPIO49	GPIO53	GPIO52	GPIO51	GPIO50

8.0 FLEXCOM INTERFACE

The Flexible Serial Communication Controller (FLEXCOM) offers several serial communication protocols that are managed by the three submodules USART, SPI, and TWI (I²C). The block diagram of a Flexcom is shown in [Figure 8-1](#).

FIGURE 8-1: FLEXCOM BLOCK DIAGRAM



- LAN9691/2/3 supports up to 4x Flexcom Controllers. Each of them can be configured in USART/UART, SPI, or TWI mode. The pins used for interfacing each Flexcom are multiplexed with the GPIO lines. If some of the I/O lines of a Flexcom are not used by the application, they can still be used for other purposes by the GPIO controller.
- Each Flexcom interface has its own TXD, RXD, and SCK signals, and there are additional 21 Shared Flexcom pins, which can be used for RTS and CTS in USART/UART mode or Chip Select in SPI mode. See [Table 8-1](#) and [Table 8-2](#).

TABLE 8-1: FLEXCOM MAPPING OPTIONS GPIO PINS

Interface	Index	Overlaid	High-speed	SCK	RDX	TXD
Flexcom0	A	ALT1	—	GPIO2	GPIO3	GPIO4
Flexcom1	A	ALT2	—	GPIO27	GPIO28	GPIO29
Flexcom2	A	ALT3	—	GPIO14	GPIO15	GPIO16
	B	ALT1	—	GPIO64	GPIO65	GPIO66
Flexcom3	A	ALT2	—	GPIO54	GPIO55	GPIO56

TABLE 8-2: FLEXCOM SHARED TO MAPPING GPIO PINS

Shared	0	1	2	3	4	5	6	7	8	9	10
GPIO	0	1	12	13	20/30	21/31	22/32	23/33	34	35	36
Overlaid	ALT2	ALT2	ALT2	ALT2	ALT3	ALT3	ALT3	ALT3	ALT4	ALT4	ALT4
Shared		11	12	13	14	15	16	17	18	19	20
GPIO	—	57	58	59	60	61	62	63	64	65	66
Overlaid	—	ALT2	ALT2	ALT2	ALT2	ALT2	ALT2	ALT2	ALT2	ALT2	ALT2

Note 1: The FLEX_Shared signals can be configured and assigned to each of the Flexcom interface.

TABLE 8-3: FLEXCOM IO MAPPING TO INTERFACE SIGNALS

Interface Mode	Signal	Flexcom
USART	SCK	FLEX_SCK (not used in UART mode)
	RX	FLEX_RXD
	TX	FLEX_TXD
	RTS	FLEX_Shared
	CTS	FLEX_Shared, if used must always be driven.
TWI/HS	SCL	FLEX_RXD
	SDA	FLEX_TXD
	SMB Alert	FLEX_SCK
	SCL_GATE	10 TWI_SCL_GATE or FLEX_Shared. SCL_GATE only supported by FLEXCOM1
SPI Host	CLK	FLEX_SCK
	MOSI	FLEX_TXD
	MISO	FLEX_RXD
	CS	FLEX_Shared, always driven.
SPI Client	CLK	FLEX_SCK
	MOSI	FLEX_TXD
	MISO	FLEX_RXD
	CS	FLEX_Shared The external host must always drive this signal.

8.1 USART/UART Interface

- The Universal Synchronous Asynchronous Receiver Transceiver (USART) provides one full-duplex universal synchronous or asynchronous serial link. Data frame format is widely programmable (data length, parity, and 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 FLEX_TXD and FLEX_RXD are used for sending and receiving data in both USART and UART mode.
- FLEX_SCK is used in USART mode for data sampling.
- FLEX_SCK is not used in UART mode. The receiver samples the RXD input with an oversampling of either 16 or eight (8) times the baud rate clock in UART mode.
- 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 RTS and CTS pins. RTS and CTS are using the FLEX_Shared signals.

8.2 TWI/I²C Interface

- 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 TWI 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 host or as client with sequential or single-byte access. Multiple initiator capability is supported.
- FLEX_TXD is used as the data signal (or SDA for I²C) of the two-wire interface, and FLEX_RXD is used as the clock signal (or SCL) of the two-wire interface. Standard mode speed (100 kHz), Fast mode speed (400 kHz), Fast mode Plus speed (1 MHz), High-speed mode (3.4 MHz), and clock stretching are all supported. Both 7-bit and 10-bit (only in Initiator mode) target addressing are supported. The host can also read/write with one-, two-, or three-byte internal address and one-byte data.

8.3 TWI/I²C Clock Muxing

- The LAN9691/2/3 has a built-in support for connecting to multiple I²C devices that use the same I²C address (for example, SFP modules). This is done by having multiplexed clock outputs, **TWI_SCL_GATE[0:9]**, together with one shared SDA. Clock muxing is only available when using the Flexcom1.
- Depending on which device it needs to communicate, the software can enable or disable the various clocks. **TWI_SCL_GATE[0:9]** are overlaid GPIO pins. Refer to [Table 8-4](#).

TABLE 8-4: GATED CLOCK MUX MAPPING TO GPIO PINS

TWI_SCL	GATE 0	GATE 1	GATE 2	GATE 3	GATE 4	GATE 5	GATE 6	GATE 7	GATE 8	GATE 9
Overlaid	ALT3	ALT3	ALT3	ALT3	ALT3	ALT3	ALT3	ALT3	ALT3	ALT3
	GPIO57	GPIO58	GPIO59	GPIO60	GPIO61	GPIO62	GPIO63	GPIO64	GPIO65	GPIO66

8.4 SPI Host or Client Interface

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.

- FLEX_TXD is used as MOSI, and FLEX_RXD is used as MISO. FLEX_SCK is used as the SPI clock signal.
- Out of the 21 **FLEX_Shared** IO pins, one or more chip select signals can be enabled for the Flexcom in SPI mode.
- Check for pull-up, if used as SPI_nCS.

9.0 USB DEVICE AND HOST ULPI INTERFACE

- The LAN9691/2/3 supports a USB device through its host ULPI (UTMI+ low-pin interface) interface. An external USB PHY (e.g., the USB3343) is needed to provide a USB host/device interface. The interface supports USB 2.0 high speed transmission speed. The ULPI is overlaid on GPIO Pins. [Table 9-1](#) shows the location on GPIO.

TABLE 9-1: GPIO PIN LOCATION

Interface	Overlaid	USB_CLK	USB_D[0:7]	USB_STP	USB_DIR	USB_NXT
ULPI	ALT2	GPIO30	GPIO31~GPIO38	GPIO39	GPIO40	GPIO41

- Other USB interface control signals can also be supported by hardware through overlaid GPIO pins like the USB power enable (**USB_POWER_E_O**) and overcurrent detection (**USB_OVER_DETECT**) signals during USB host operation as well as the USB Reset output (**USB2PHY_RESET_O**) to reset the external USB PHY. [Table 9-1](#) details the possible GPIO pins available for those control signals.

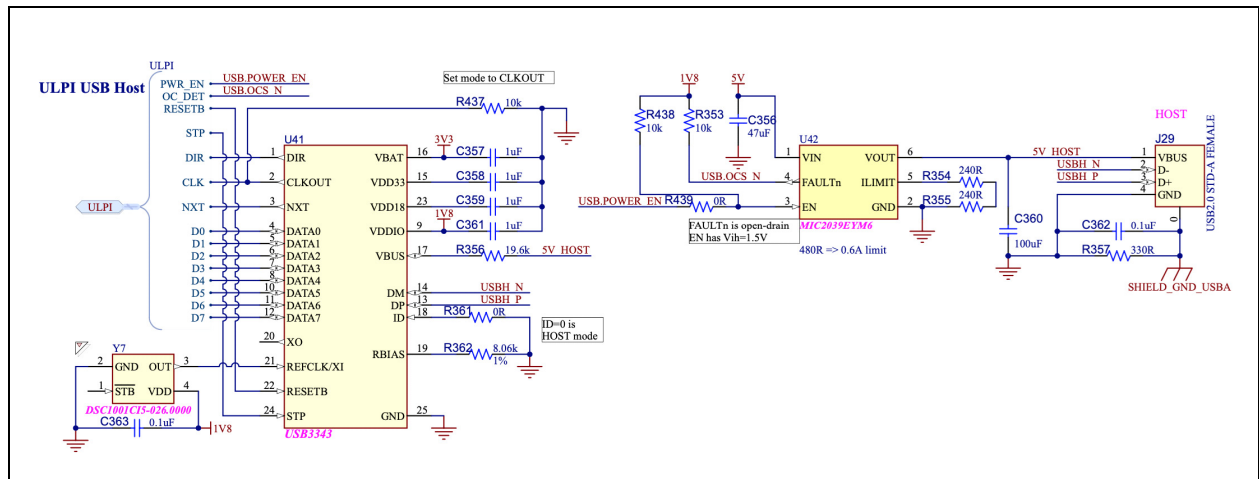
TABLE 9-2: USB CONTROL SIGNALS TO GPIO PINS

Interface	Overlaid	USB_POWER_E_O	USB2PHY_RESET_O	USB_OVER_DETECT
USB	ALT3	GPIO1	GPIO12	GPIO13
	ALT4	GPIO17	GPIO18	GPIO19

[Figure 9-1](#) shows an example of implementing USB host interface through an external USB PHY. In this example, **FAULTn** from MIC2039EYM is an open-drain output.

- Check that the input to **USB_OC_DET** (USB over current detection) matches the **VDDIO** supply of 1.8V or need a resistor divider.

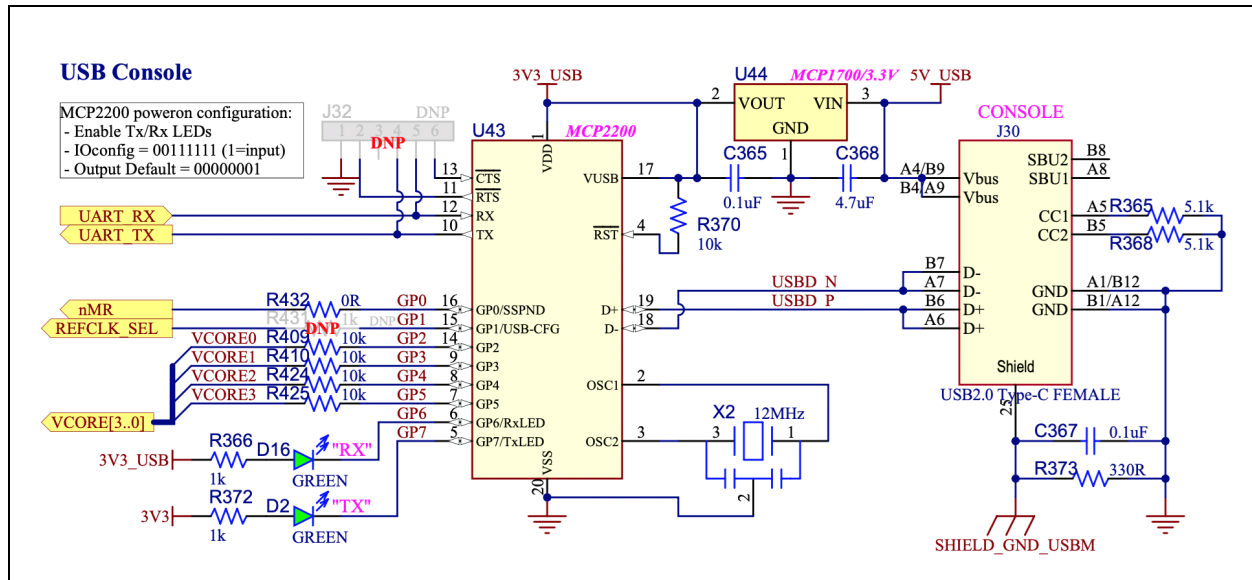
FIGURE 9-1: USB HOST INTERFACE WITH EXTERNAL USB PHY



LAN9691/2/3

If both a USB host and a USB device are needed, an external USB-to-UART/I²C serial converter device can be connected to one of the Flexcom UARTs. Figure 9-2 illustrates an example from the Microchip reference design. The figure shows that the serial GPIO's can be used to control the Reset and Boot mode of the LAN9691/2/3.

FIGURE 9-2: USB CONSOLE PORT WITH USB TO UART CONVERTER



10.0 SERIAL GPIO CONTROLLER (SGPIO)

10.1 SGPIO Signals

The serial GPIO controller uses a four-pin serial interface to extend the number of available general purpose I/O pins. The primary purpose of the SGPIO controller is to connect control signals from SFP modules and to act as an LED controller. The four-pin serial interface pins are overlaid on the GPIOs. Refer to [Table 10-1](#) and [Figure 10-1](#).

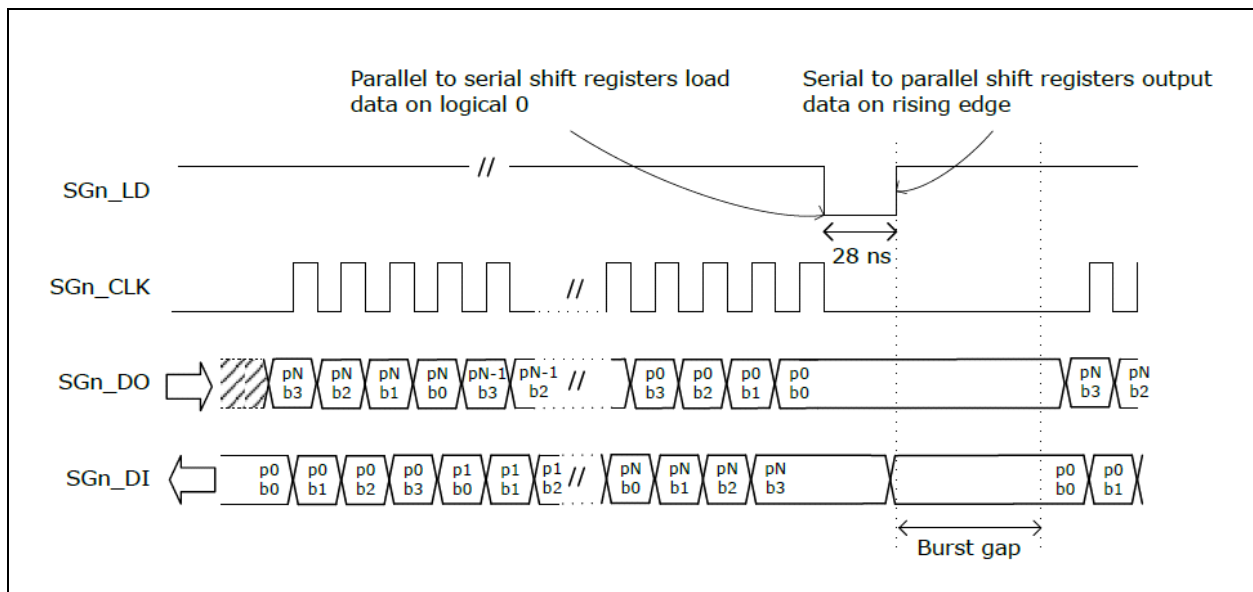
TABLE 10-1: SERIAL GPIO CONTROLLER SIGNALS

Signal	Overlaid	SGPIO_0
SG0_CLK 1	ALT1	GPIO5
SG0_DO	ALT1	GPIO6
SG0_DI	ALT1	GPIO7
SG0_LD	ALT1	GPIO8

Note 1: It is recommended to use a split end-termination on the CLK.

[Figure 10-1](#) illustrates the I/O timing of the serial GPIO controller. Serial data are output on the SG_DO pin clocked by SG_CLK in bursts. After each burst, there is an assertion of the SG_LD signal. At the same time, as shifting out serial outputs on SG_DO, the serial GPIO controller also samples the SG_DI input. The values sampled on SG_DI are made available to the software.

FIGURE 10-1: SGPIO TIMING



- The maximum length of a burst is 128 bits data cycles organized by 32 ports with 4-bit port width. However, each port can be enabled or disabled individually, and the port width (number of bits per port) is also centrally configurable. All enabled ports will have the same port width. The configuration of port enabling and port width applies to both serial input and output.
- SG_LD can be used to ensure that outputs are stable when serial data are being shifted through the shift registers (for example, the TX_DIS output signals for the SFP modules). This can be done by using the SG_LD signal to load the serial data onto a shift register's parallel output pins after the burst has completed. The 74HC595 is one of the shift registers supporting load input. If the serial GPIO controller is used for serial LED output, then SG_LD is optional because it is usually not detected when serial data are updated (shift through the chain). In that case, the 74HC164, which does not have the load input, can also be used.

10.2 SGPIO Constraints

- When a serial output bit is configured to support link/activity LED, it needs to know which switch port status to display on which SGPIO bit. The SGPIO controller uses a 1:1 mapping. This means SGPIO port 0 displays link/activity for switch port 0, while SGPIO port 1 displays link/activity for switch port 1, and so on.
- The serial input function of the SGPIO controller can automatically route Signal detection (RX_LOS) to the internal PCS block of a certain port. Per default, the signal detection function also uses 1:1 mapping between the switch port number and the SGPIO port number, so that bit 0 of each SGPIO port can be enabled for RX_LOS input.
- There might be more serial output bits than serial input bits in a typical design especially when serial LED is supported. Since the port enabling and port width configuration are shared by the SGPIO output and input, the same number of serial bits in the output and input streams is present. This means the same number of external shift registers is required for both output and input chains. Therefore, some shift registers are wasted on the serial input chain. To save the shift registers for the unused SGPIO input bits, the serial input bits can be looped—the SG_DI input signal is also connected to the serial input pins of the last shift register to create a loop—so that the bits in that loop can be duplicated and the length of the whole serial stream is expanded.

11.0 MII MANAGEMENT INTERFACE

- For the software to access the external PHY registers, the MIIM host interface should be connected from the switch to the external PHY. The MIIM host controller supports both Clause 22 and Clause 45. The maximum clock frequency is 25 MHz.
- The LAN9691/2/3 supports two MIIM host interfaces. These are overlaid on the GPIOs. Each MIIM host interface has a clock output from the switch, MDC, a bidirectional data input/output signal, MDIO, and an interrupt input, **MIIM_IRQ**, used for receiving interrupt from the external PHY. The **MIIM_IRQ** (inputs) are not a part of the MII-Management controllers but are part of the IO Interrupt controller. The naming and placement of the **MIIM_IRQ** pins close to each MIIM host interface are simply made to ease routing of all control signals going to the CuPHY.
- Besides the MIIM host interfaces, LAN9691/2/3 also supports an MIIM client interface which can be used for an external CPU to access the Switch registers. The MIIM client interface must first be enabled (i.e., by the switch application or from the SPI client interface.) The MIIM client interface can be configured (by a strapping pin—**MIIMC_ADDR**) to answer one of two direct PHY addresses. Setting the **MIIM_0_ADDR** pin to 0 configures the MIIM client to use PHY address 0, while setting it to 1 configures it to use PHY address 31. See [Table 11-1](#).

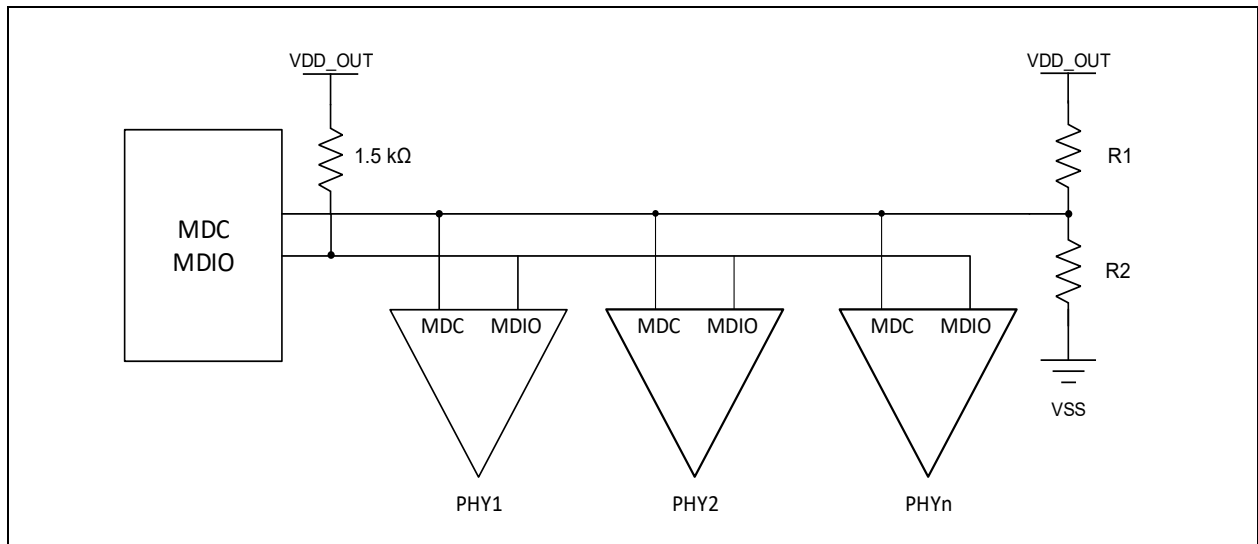
TABLE 11-1: MII MANAGEMENT CONTROLLER SIGNALS

Signal	MIIM_0 Host	MIIM_0 Client	MIIM_1 Host		
Overlaid	ALT1	ALT2	ALT3	ALT3	ALT1
MDC	GPIO9	GPIO9	GPIO27	GPIO39	GPIO61
MDIO	GPIO10	GPIO10	GPIO28	GPIO40	GPIO62
MIIM_IRQ	GPIO11	N/A	GPIO29	GPIO41	GPIO63
MIIM_0_ADDR	N/A	GPIO11	N/A	N/A	N/A

The MDIO signal is changed or sampled on the falling edge of the MDC clock by the controller. The **MDIO** pin is tri-stated in-between access and when expecting read data from the PHY.

- Because the **MDIO** is an open-drain output, the **MDIO** should be pulled high with a resistor of around 1.5 k Ω . When connecting **MDC/MDIO** to multiple PHYs, the layout scheme in [Figure 11-1](#) with split end termination of 330 Ω on **MDC** is recommended.

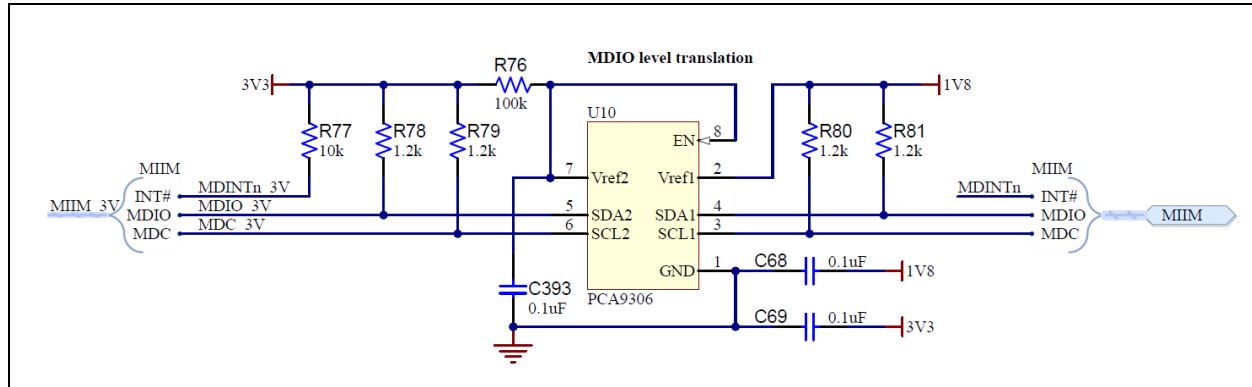
FIGURE 11-1: DAISY-CHAINING MDC/MDIO TO MULTIPLE PHYs



LAN9691/2/3

- The MIIM host controller uses PHY addresses to select each of the external PHYs, so the PHY addresses must be configured differently for each PHY on the same MIIM interface. Up to 32 (0-31) addresses are supported per IEEE.
- The LAN9691/2/3 IO supply is 1.8V. External PHY's often use 2.5V or 3.3V as their IO supply. [Figure 11-2](#) displays an example of implementing the level translation.

FIGURE 11-2: EXAMPLE LEVEL TRANSLATION OF THE MII-MANAGEMENT BUS



Note: Because the LAN9691/2/3 offers two external MII-management interfaces, splitting Clause 22 and Clause 45 PHYs among different buses is recommended. Some CuPHYs, which normally use Clause 22, cannot cooperate with multi-Gig PHYs using Clause 45. An alternative solution could be running the multi-Gig PHY in Clause 22 as well.

12.0 GPIO USAGE

- Each GPIO pin can be assigned to one of up to five functions. Ensure that there are no functional overlaps in the design.

TABLE 12-1: GPIO ALTERNATIVE FUNCTION ASSIGNMENT

GPIO	ALT-RGMII	ALT1	ALT2	ALT3	ALT4
GPIO0	—	IRQ0_IO	FLEX_SHARED0	PCIE_PERST	—
GPIO1	—	IRQ0_IO	FLEX_SHARED1	USB_POWER_E_O	—
GPIO2	—	FLEX0_SCK	—	—	—
GPIO3	—	FLEX0_RXD	—	—	—
GPIO4	—	FLEX0_TXD	—	—	—
GPIO5	—	SG0_CLK	—	—	—
GPIO6	—	SG0_DO	—	—	—
GPIO7	—	SG0_DI	—	—	—
GPIO8	—	SG0_LD	—	—	—
GPIO9	—	MIIM0_MDC	MIIMC_MDC	CPU_CLK	—
GPIO10	—	MIIM0_MDIO	MIIMC_MDIO	DDR_CLK	—
GPIO11	—	IRQ2_IO / MIIM0_IRQ	MIIMC_ADDR	SWITCH_CORE_CLK	—
GPIO12	—	IRQ3_IO	FLEX_SHARED2	USB2PHY_RESET_O	—
GPIO13	—	IRQ4_IO	FLEX_SHARED3	USB_OVER_DETECT	—
GPIO14	—	SDM-MC0_CMD	QSPI1_SCK	FLEX2_SCK	—
GPIO15	—	SDMMC0_CK	QSPI1_nCS	FLEX2_RXD	—
GPIO16	—	SDMMC0_D0	QSPI1_IO0	FLEX2_TXD	—
GPIO17	—	SDMMC0_D1	QSPI1_IO1	PTPSYNC0	USB_POWER_E_O
GPIO18	—	SDMMC0_D2	QSPI1_IO2	PTPSYNC1	USB2PHY_RESET_O
GPIO19	—	SDMMC0_D3	QSPI1_IO3	PTPSYNC2	USB_OVER_DETECT
GPIO20	—	SDMMC0_D4	QSPI0_IO4	FLEX_SHARED4	SDMMC0_S-D_LED
GPIO21	—	SDMMC0_D5	QSPI0_IO5	FLEX_SHARED5	SDMMC0_S-D_WP
GPIO22	—	SDMMC0_D6	QSPI0_IO6	FLEX_SHARED6	SDMMC0_S-D_CD
GPIO23	—	SDMMC0_D7	QSPI0_IO7	FLEX_SHARED7	SDMMC0_S-D_1V8SEL
GPIO24	—	SDM-MC0_RSTN	QSPI0_DQS (read data strobe)	—	—
GPIO25	—	PWM	FUSA1	CANTX0	QSPI1_SCK
GPIO26	—	TACHO	FUSA2	CANRX0	QSPI_nCS
GPIO27	—	SYNCE_RC0	FLEX1_SCK	MIIM1_MDC	QSPI_IO0
GPIO28	—	SYNCE_RC1	FLEX1_RXD	MIIM1_MDIO	QSPI_IO1
GPIO29	—	SYNCE_RC2	FLEX1_TXD	IRQ5_IO / MIIM1_IRQ	QSPI_IO2

TABLE 12-1: GPIO ALTERNATIVE FUNCTION ASSIGNMENT (CONTINUED)

GPIO	ALT-RGMII	ALT1	ALT2	ALT3	ALT4
GPIO30	—	PTPSYNC0	USB_CLK	FLEX_SHARED4	QSPI_IO3
GPIO31	—	PTPSYNC1	USB_D0	FLEX_SHARED5	—
GPIO32	—	PTPSYNC2	USB_D1	FLEX_SHARED6	—
GPIO33	—	SDM-MC1_CMD	USB_D2	FLEX_SHARED7	—
GPIO34	—	SDMMC1_CK	USB_D3	CANTX1	FLEX_SHARED8
GPIO35	—	SDMMC1_D0	USB_D4	CANRX1	FLEX_SHARED9
GPIO36	—	SDMMC1_D1	USB_D5	PCIE_PERST	FLEX_SHARED10
GPIO37	—	SDMMC1_D2	USB_D6	CANTX0	—
GPIO38	—	SDMMC1_D3	USB_D7	CANRX0	—
GPIO39	—	SDMMC1_S-D_1v8SEL	USB_STP	MIIM1_MDC	SDM-MC1_RSTN
GPIO40	—	SDMMC1_S-D_WP	USB_DIR	MIIM1_MDIO	—
GPIO41	—	SDMMC1_S-D_CD	USB_NXT	IRQ5_IO / MIIM1_IRQ	—
GPIO42	RGMII1.RX_D0	PTP.SYNC3	CANTX1	—	—
GPIO43	RGMII1.RX_D1	PTP.SYNC4	CANRX1	—	—
GPIO44	RGMII1.RX_D2	PTP.SYNC5	SFP_SD0	—	—
GPIO45	RGMII1.RX_D3	PTP.SYNC6	SFP_SD1	—	—
GPIO46	RGMII1.RX_CLK	PTP.SYNC7	SFP_SD2	—	—
GPIO47	RGMII1.RX_CTL	—	SFP_SD3	—	—
GPIO48	RGMII1.TX_D0	—	SFP_SD4	—	—
GPIO49	RGMII1.TX_D1	—	SFP_SD5	—	—
GPIO50	RGMII1.TX_D2	—	SFP_SD6	—	—
GPIO51	RGMII1.TX_D3	—	SFP_SD7	—	—
GPIO52	RGMII1.TX_CLK	PWM	SFP_SD8	—	—
GPIO53	RGMII1.TX_CTL	TACHO	SFP_SD9	—	—
GPIO54	RGMII0.RX_D0	SYNCE_RC1	FLEX3_SCK	—	—
GPIO55	RGMII0.RX_D1	SYNCE_RC0	FLEX3_RXD	—	—
GPIO56	RGMII0.RX_D2	SYNCE_RC2	FLEX3_TXD	—	—
GPIO57	RGMII0.RX_D3	SFP_SD0	FLEX_SHARED11	TWI_SCL_GATE0	PTP.SYNC3
GPIO58	RGMII0.RX_CLK	SFP_SD1	FLEX_SHARED12	TWI_SCL_GATE1	PTP.SYNC4
GPIO59	RGMII0.RX_CTL	SFP_SD2	FLEX_SHARED13	TWI_SCL_GATE2	PTP.SYNC5
GPIO60	RGMII0.TX_D0	SFP_SD3	FLEX_SHARED14	TWI_SCL_GATE3	PTP.SYNC6
GPIO61	RGMII0.TX_D1	MIIM1_MDC	FLEX_SHARED15	TWI_SCL_GATE4	—
GPIO62	RGMII0.TX_D2	MIIM1_MDIO	FLEX_SHARED16	TWI_SCL_GATE5	—
GPIO63	RGMII0.TX_D3	MIIM1_IRQ/IRQ5	FLEX_SHARED17	TWI_SCL_GATE6	—

TABLE 12-1: GPIO ALTERNATIVE FUNCTION ASSIGNMENT (CONTINUED)

GPIO	ALT-RGMII	ALT1	ALT2	ALT3	ALT4
GPIO64	RGMII0.TX_CLK	FLEX2_SCK	FLEX- _SHARED18	TWI_SCL_GATE7	—
GPIO65	RGMII0.TX_CTL	FLEX2_RXD	FLEX- _SHARED19	TWI_SCL_GATE8	—
GPIO66	—	FLEX2_TXD	FLEX- _SHARED20	TWI_SCL_GATE9	—

HAO Signal	HAO	HAO	HAO
------------	-----	-----	-----

14.0 LAYOUT CONSIDERATIONS

- The following standard high-speed design layout techniques must be followed:
 - Minimize the number of signal vias.
 - No 90-degree bends for high-speed differential signals. Forty-five-degree bends or better curved traces should be used.
 - Traces must refer to a single power or ground plane. Do not let high-speed traces pass plane splits on an adjacent layer.
 - To minimize crosstalk, the characteristic impedances of signals should be determined predominantly by the distance to the reference plane and not the distance to the neighboring traces.
 - Allow as much clearance between traces as possible. Do not let two traces neighbor at a fixed low distance for extended lengths of trace.
 - Reduce trace length as much as possible, especially for high-speed single-ended signals (e.g., DDR) and very high-speed differential signals (e.g., QSGMII, XFI/SFI, USXGMII).
- Special care must be taken for the high-speed differential pairs (XFI/SFI/QSGMII/SGMII/PCIe):
 - Must be routed as 100Ω differential traces. PCIe and XFI signals may be routed as 85Ω differential traces for lower loss.
 - Allow as much clearance as possible between differential pairs. The well-known rule-of-thumb of “3*W” (W = width of trace) is the absolute minimum. Allow for 5*W or even 10*W clearance wherever possible to minimize crosstalk.
 - If guard traces or planes are used between individual pairs, use the same clearance constraints trace-to-plane as pair-to-pair (e.g., 3*W, 5*W or 10*W) to reduce the impact of the guard on the impedance of the trace, especially the (P or N) trace closest to the guard. Tie the guard to the inner layer ground plane frequently along the way. Untied guards are good antennas.
 - Intra-pair skew (delta between lengths of P and N traces) must be as low as possible. As an example, for a 10 Gbit/sec trace, an intra-pair skew of 3 mm translates into 20 ps or 20% of a bit, causing the differential receiver to read the signal with excessive jitter.
- DDR SDRAM Interface. Source-synchronous interfaces such as DDR have same-length constraints:
 - Byte lane 0 (DQ[7:0], DM0, DQS0_c/t): 1 mm.
 - Byte lane 1 (DQ[15:8], DM1, DQS1_c/t): 1 mm.
 - Command lane (address, clock, and control): 2-3 mm.
 - The command lane trace lengths between each RAM must also match in length. The maximum skew is 1 mm.
 - Command lane/Address bus signals must use midpoint termination at the end of the trace (A[17:0], BA[2:0], BG[1:0], Ck_t/c, CKE[1:0], ODT[1:0], ACT_n, CS_n[1:0], PAR). The midpoint must be sourced with a quality Sink/Source DDR termination regulator.
 - Trace impedance for the DDR traces should be within the 40Ω-to-60Ω range.
 - Differential signals CK_t/c and DQS[4:0]_t/c should be routed with a differential trace impedance of 100Ω.
 - Remember bias on **DDR_RZQ**: 240Ω to VSS.
 - **DDR_ALERT_n** pulled to VDD or reference supply, **VDDIODDR**.
- **DDR_ZCTRL** must be left floating. The reference voltage is generated internally.
- Low-bandwidth serial interfaces have fast edges, too. Even though the effective bandwidths of these interfaces are low, it is important to avoid reflections on the multi-drop clocks (MDC, SGPIO_CLK, and SPI_CLK). Route the clock nets daisy-chained from the source to the first consumer, then to the next consumer, and finally to the parallel termination network.
- The pull-ups on MDIO and I2C_SDA must be present and can be placed wherever convenient.
- All power supplies must be routed as planes. The following are additional constraints:
 - Limit via sharing (where several power pins are connected to a single via). Ideally, each power pin should be connected to its associated power plane through one or more dedicated via.
 - Surface-mount local decoupling capacitors must be used and should be placed as close to the power supply pins as possible.
 - Analog supplies must be isolated from the remaining board supplies using ferrite beads.
- Simulate the completed design, and possibly selected signals during layout:
 - Simulate power integrity (e.g., power drops/current densities). Even though Microchip devices are low power. For example, small voltage drops on a 1.0V supply quickly add up to several percent of nominal voltage. If

power integrity simulation shows issues, users may solve them using wider planes or by using thicker planes (e.g., ½ oz -> 2 oz copper) or, in some cases, a remote sense point (located at the power consumer) for the DC/DC converter circuit.

- Simulate signal integrity. In some cases, DDR/SGMII/SERDES/QSGMII/SFI/XFI signals are extremely high speed that it is better to simulate on the layout file than to measure on the actual board. If a full simulation is not possible, simulating at least a single lane of a multi-lane interface can still provide valuable insight. Microchip supplies IBIS or IBIS-AMI models for its devices' high-speed inputs/outputs. Use the more accurate IBIS-AMI models both for the target signal and neighboring crosstalk aggressors whenever applicable.

15.0 MISCELLANEOUS

This section covers the miscellaneous signals mainly supported as overlaid functions on the GPIO pins in alternate mode. Many signals have alternative routes. Avoiding using one functionality does not avoid using another. This is indicated in the index and overlaid columns.

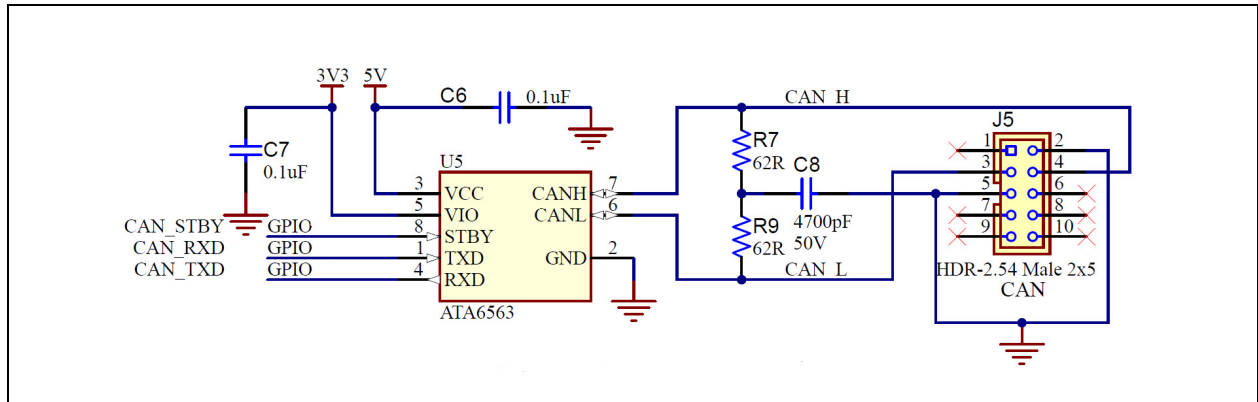
15.1 CAN FD

TABLE 15-1: CAN CONTROLLER MAPPING TO GPIO PINS

Interface	Index	Overlaid	Rx	Tx
CAN FD 0	A	ALT3	GPIO26	GPIO25
	B	ALT3	GPIO38	GPIO37
CAN FD 1	A	ALT3	GPIO35	GPIO34
	B	ALT2	GPIO43	GPIO42

The CAN interface is intended to be connected to a CAN transceiver, like the ATA6563. [Figure 15-1](#) shows an example for how to connect the interface to a CAN transceiver.

FIGURE 15-1: CAN BUS TRANSCEIVER



15.2 FAN Controller

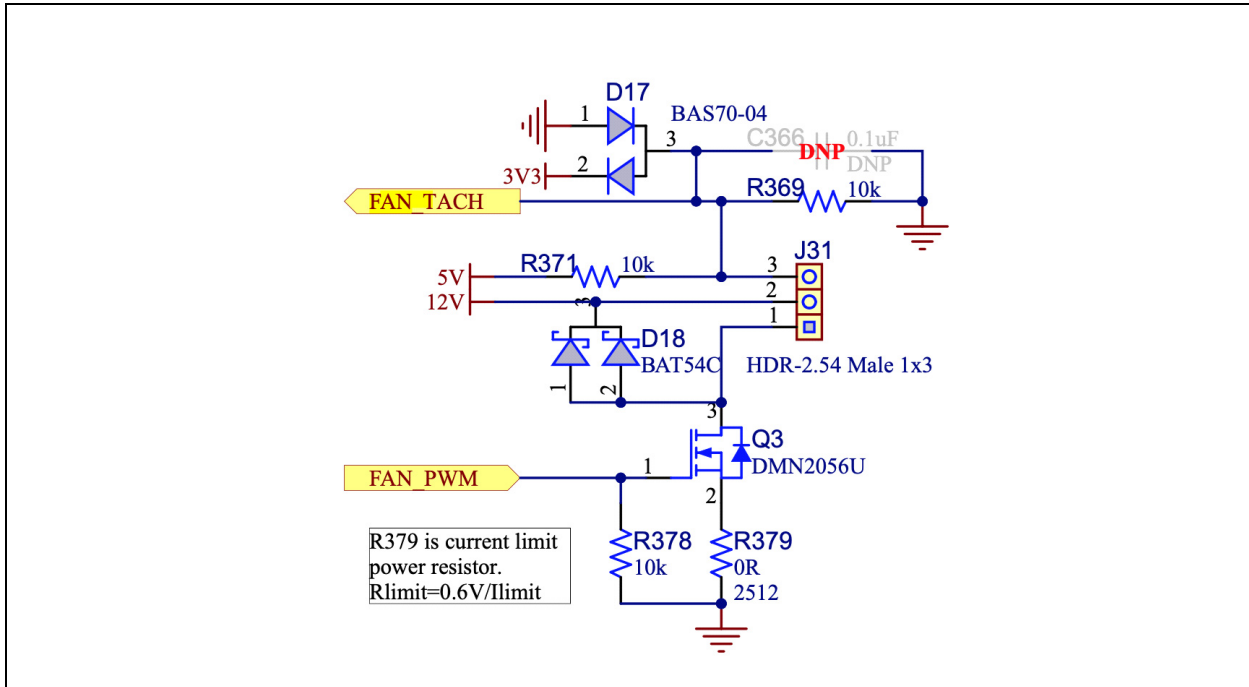
TABLE 15-2: FAN CONTROLLER MAPPING TO GPIO PINS

Signal	Index	Overlaid	GPIO	Remarks
TACHO	A	ALT1	GPIO26	Hardware-based pulse counting input
	B	ALT1	GPIO53	
PWM	A	ALT1	GPIO25	Pulse width modulation output
	B	ALT1	GPIO52	

LAN9691/2/3

Figure 15-2 shows how the PWM and TACHO signals are used with a 12V FAN. The FAN connector is powered from a 12V supply through a PWM-controlled FETs for adjustable fan power by doing ground chopping.

FIGURE 15-2: 12V FAN CIRCUITRY



15.3 SyncE Recovered Clocks

TABLE 15-3: RECOVERED CLOCK OUTPUT MAPPING TO GPIO PINS

Signal	Index	Overlaid	RC0	RC1	RC2
SYNCE_RC	A	ALT1	GPIO27	GPIO28	GPIO29
	B	ALT1	GPIO55	GPIO54	GPIO56

15.4 Functional Safety

The switch's two PLL internal status can be monitored through FuSa[1] and FuSa[2]. FuSa[1] outputs the CPU clock PLL divided by 16. FuSa[2] is the Dying Gasp Interrupt output signaled from the device when the CPU clock PLL has lost lock. See Table 15-4.

TABLE 15-4: FUNCTIONAL SAFETY MAPPING TO GPIO PINS

Signal	Index	Overlaid	FUSA1	FUSA2
Functional safety	—	ALT2	GPIO25	GPIO26

15.5 Clock Monitor Output

As part of Functional Safety (FuSa), selected PLL output signals can be monitored. The PLL output signal is divided by 16.

TABLE 15-5: CLOCK MONITOR OUTPUT MAPPING TO GPIO PINS

Signal	Overlaid	CPU_CLK	DDR_CLK	SWITCH_CORE_CLK
CLK_MON	ALT3	GPIO9	GPIO10	GPIO11

15.6 PTP Engines

There are eight PTP engines that can be set up individually as input or output and for varied frequencies. They are all overlaid on GPIO pins as shown in [Table 15-6](#).

TABLE 15-6: PTP ENGINE MAPPING TO GPIO PINS

Engine	Index	Overlaid	PTP0	PTP1	PTP2	PTP3	PTP4	PTP5	PTP6	PTP7
PTP	A	ALT1	GPIO30	GPIO31	GPIO32	GPIO42	GPIO43	GPIO44	GPIO45	GPIO46
	B	ALT3	GPIO17	GPIO18	GPIO19	—	—	—	—	—
	C	ALT4	—	—	—	GPIO57	GPIO58	GPIO59	GPIO60	—

15.7 Interrupt Controller

- There are six interrupt signals which can be configured individually as either interrupt source or destination to the internal or external CPU system. They are all overlaid on GPIO pins as shown in [Table 15-7](#).
- If interrupt signal is being used, remember to have pull-up to the appropriate supply voltage, VDDIO.

TABLE 15-7: IRQ MAPPING TO GPIO PINS

Interrupt	Index	Overlaid	IRQ0_IO	IRQ1_IO	IRQ2_IO	IRQ3_IO	IRQ4_IO	IRQ5_IO
IRQ_IO	A	ALT1	GPIO0	GPIO1	GPIO11	GPIO12	GPIO13	—
	B	ALT3	—	—	—	—	—	GPIO29
	C	ALT3	—	—	—	—	—	GPIO41
	D	ALT1	—	—	—	—	—	GPIO63

LAN9691/2/3

NOTES:

16.0 HARDWARE CHECKLIST SUMMARY

TABLE 16-1: HARDWARE DESIGN CHECKLIST

Section	Check	Explanation	√	Notes
Section 2.0, "General Considerations"	Section 2.1, "Required References"	All necessary documents are on hand.		
	Section 2.2, "Pin Check"	The pins match the data sheet and the reference design's use of GPIO is retained to minimize software efforts.		
	Section 2.3, "Strapping Pins"	Check VCORE[3:0] to match booting mode and REFCLK_SEL to select the core reference clock source.		
Section 3.0, "Power/Grounding"		Each power rail has bulk and high-frequency decoupling capacitors.		
		Analog supplies are isolated from digital supplies through ferrite beads.		
		Check power sequencing.		
Section 4.0, "Reference Clocks"	Section 4.1, "Differential Clock"	Check if the differential clock complies with specifications. Refer to the device data sheet for the specification on reference clock input voltage range. Either CML or LCPECL is recommended.		
Section 5.0, "CPU System"	Section 5.1, "Internal CPU Auto Boot Mode"	Check the VCORE[3:0] strapping.		
	Section 5.2, "QSPI Interface"	Check NOR Flash data width, size, and connection. It is recommended to use the same family as the reference board.		
	Section 5.3, "DDR3/3L/4 Memory Controller-SDRAM Interface"	The trace lengths between each DDR LANE must match in length. The maximum skew is 1 mm. The differential signal impedance is 100Ω, single ended 40Ω ~60Ω. It is recommended to use the same DDR device as in the reference design. Simulate signal integrity for the interface.		
	Section 5.4, "SDMMC Controller"	Make sure the SDMMC usage matches the VCORE[3:0] booting mode configuration.		
	Section 5.5, "External CPU Mode"	Check the VCORE[3:0] strapping.		
		Check the SPI client or MIIM client if used as management interface from the external CPU to VLAN9691/2/3.		
Section 6.0, "Port Configuration"	Section 6.1, "SerDes Interface"	Check connection and possible AC-coupling.		
	Section 6.2, "Port Mapping"	Make sure the total bandwidth of the enabled ports does not exceed the limitation of the select SKU. Make sure the port mapping is supported as shown in Table 6-2 .		
	Section 6.3, "MAC-to-MAC Connections"	AC coupling is recommended. Check the signal directions.		
	Section 6.4, "SFP Ports"	Verify each SFP signal. RX_LOS from SFP modules are recommended to be connected to the switch through either SGPIO or GPIO.		

TABLE 16-1: HARDWARE DESIGN CHECKLIST (CONTINUED)

Section	Check	Explanation	√	Notes
Section 7.0, "Parallel MAC Interface"	Section 7.0, "Parallel MAC Interface"	Check if the link partner also supports 1.8V IO or if a voltage converter is needed.		
	Section 7.1, "RGMII Interface"	Match trace length within 7 mm for signals within TX or RX groups. Enable 2.0 ns internal delay if there is no trace delay. Simulate the RGMII interface signals.		
	Section 7.2, "RMII Interface"	Check clock directions in Input or Output mode.		
Section 8.0, "Flexcom Interface"	Section 8.0, "Flexcom Interface"	Check if the connected device supports 1.8V IO or if a voltage converter is needed.		
	Section 8.1, "USART/UART Interface"	Make UART available for CLI debugging, preferably the Flexcom 0 as this is the boot monitor UART.		
	Section 8.2, "TWI/I2C Interface"	Pulling high is required. Only Flexcom 1 supports the multiplexed clockout for I ² C slave devices with same addresses (like SFP modules).		
	Section 8.3, "TWI/I2C Clock Muxing"	Make sure the SPI usage matches the V _{CORE} [3:0] booting mode configuration.		
	Section 8.4, "SPI Host or Client Interface"	Check if the connected device supports 1.8V IO, or if a voltage converter is needed.		
Section 9.0, "USB Device and Host ULPI Interface"		External USB PHY needed.		
Section 10.0, "Serial GPIO Controller (SGPIO)"		Check switch port to SGPIO port mapping for the link/active LED output and RX_LOS signal input.		
Section 11.0, "MII Management Interface"		Pull MDIO high with 1.5 kΩ end termination for MDC if connected to multiple PHYs. Follow layout scheme shown in Figure 11-1 .		
Section 12.0, "GPIO Usage"		Check if the connected device supports 1.8V IO, and if a voltage converter is needed.		
Section 13.0, "JTAG Interface"		Not 3.3V or 5V compatible; 1.8V signal levels only.		
Section 14.0, "Layout Considerations"		Follow layout rules described in the section.		
Section 15.0, "Miscellaneous"	Section 15.1, "CAN FD"	External transceiver is required.		
	Section 15.2, "FAN Controller"	It is recommended to use a FAN compatible with 20 kΩ PWM control.		
	Section 15.3, "SyncE Recovered Clocks"	Check if an external DPLL is needed for SYNCE.		
	Section 15.4, "Functional Safety"	Test point is recommended for board bring-up and debugging.		
	Section 15.5, "Clock Monitor Output"	Test point is recommended for board bring-up and debugging.		

TABLE 16-1: HARDWARE DESIGN CHECKLIST (CONTINUED)

Section	Check	Explanation	√	Notes
	Section 15.6, "PTP Engines"	Check customer's PTP accuracy requirement, and decide if an external DPLL is needed. Retain the reference design's use of 1PPS as much as possible.		
	Section 15.7, "Interrupt Controller"	Make sure the intended interrupt pins are pulled.		

APPENDIX A: REVISION HISTORY

TABLE A-1: REVISION HISTORY

Revision Level & Date	Section/Figure/Entry	Correction
DS00005358B (02-20-25)	Table 2-1, Table 4-1, Table 5-3, Table 5-4, Table 6-2, Table 6-3	Updated details.
	Figure 5-3	Update figure.
	Figure 5-4	Updated figure.
	Figure 11-1	Updated figure.
	Figure 11-2	New figure
	Figure 15-1	New figure
	Section 14.0 CAT5 Media Interface	Removed section.
	All	Changed device family number from LAN969x to LAN9691/2/3 to reflect SKUs covered in this document. Made minor text changes throughout the document.
DS00005358A (04-05-24)	Initial release	

NOTES:

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: 979-8-3371-0708-0

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 product is 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.