Chapter 28

Multiplexed I/O

Introduction

The features and functional description of the multiplexed I/Os (MIOs) are described in this chapter including the MIO signal routing, bank-level mapping, and pin assignment considerations for efficient use of the available MIO pins.

The basic MIO function is to multiplex access from the processing system (PS) peripheral interface pins to the appropriate peripheral interfaces, as defined in the configuration registers. An additional function is to control access from the extended multiplexed I/O interface (EMIO) block to the input signals of the peripheral interfaces, for instance, where there is a receive path. The MIO module allows you to configure the PS pin-out as required. Seventy-eight (78) of the general purpose I/Os (GPIO) are used as MIOs. They are configured by accessing the MIO control registers (detailed in this chapter) and are located in the system-level control, IOU_SLCR register set.

The 78 MIO signals are divided into three banks, and each bank includes 26 device pins. Each bank (500, 501, and 502) has its own power pins, VCCO_PSIO{0:2} for the hardware interface. The I/O logic and interface to the system are in the LPD power domain. The voltage signaling level, 1.8 or 3.3V, can be determined by reading the IOU_SLCR.bank{0:2} registers.

The boot device is assigned to a specific set of MIO pins (see Table 11-1). These assignments can drive the decisions on bank assignments for interfacing with other hardware.

Overview of the Blocks Function

The MIO module can be described as a wide multiplexer/de-multiplexer, routing a number of different peripheral interfaces to a limited number of external pins under software configuration. A number of different interfaces are routed to and from the pins by the MIO, with varying timing requirements. Therefore, a priority structure based on maximum toggle rates must be implemented to place high-speed signal interfaces (such as gigabit Ethernet RGMII or USB ULPI) closer to the pin in the multiplexer tree structure.

Control of the functionality associated with each pin is through the MIO section of the IOU_SLCR system-level control registers. Output control signals are generated from these register settings. These signals are used either directly as multiplexer selects or indirectly through multiplexer select remapping functions. There are multiple port mapping options available for peripherals (e.g., 12 for CAN and I2C) where the interface to the peripheral can be constructed using any of the following.

- Mapping of ports from a single group.
- Mapping of ports from different groups.
- A mix of PS pins and PL pins through the EMIO interface.

PS and PL Pins

The MIO is fundamental to the I/O peripheral connections due to the limited number of MIO pins (Figure 28-1). Software programs the routing of the I/O signals to the MIO pins. The I/O peripheral signals can also be routed to the PL (including PL device pins) through the EMIO interface. This is used to gain access to more device pins (PL pins) and to allow an I/O peripheral controller to interface to internal logic in the PL.

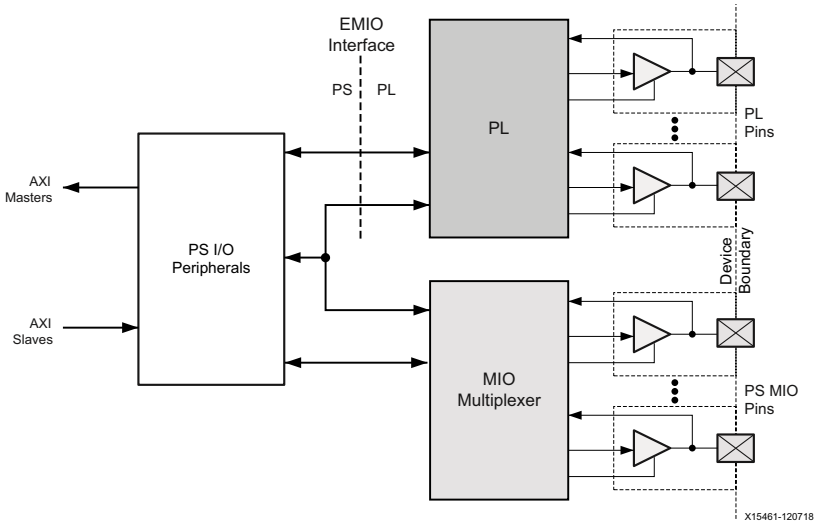

Figure 28-1: **MIO-EMIO Wiring Diagram**

Output Multiplexer

The output multiplexer example in Figure 28-2 shows a single bit cell of an output multiplexer. The I3_output_* signal name is used to denote any of the range of low-speed peripherals, where the ordering is not significant. To illustrate the general multiplexing structure, other interfaces are identified without specifying a particular signal. Interfaces of similar speed can be swapped at each multiplexer level. For instance, the fast trace-port interface can be used where neither the ULPI nor RGMII PHY interfaces are used.

Figure 28-2 shows the default multiplexer structure for the output and enable multiplexer. For most pins, only one of the high-speed interfaces (RGMII or ULPI or trace) is present. Similarly, for many signals generated or consumed by peripherals, there is no corresponding 3-state enable under the implemented protocol for its external interface. For example, because RGMII does not use 3-state enables, the diagram includes them to illustrate the concept of the output enable shadowing the output signal.

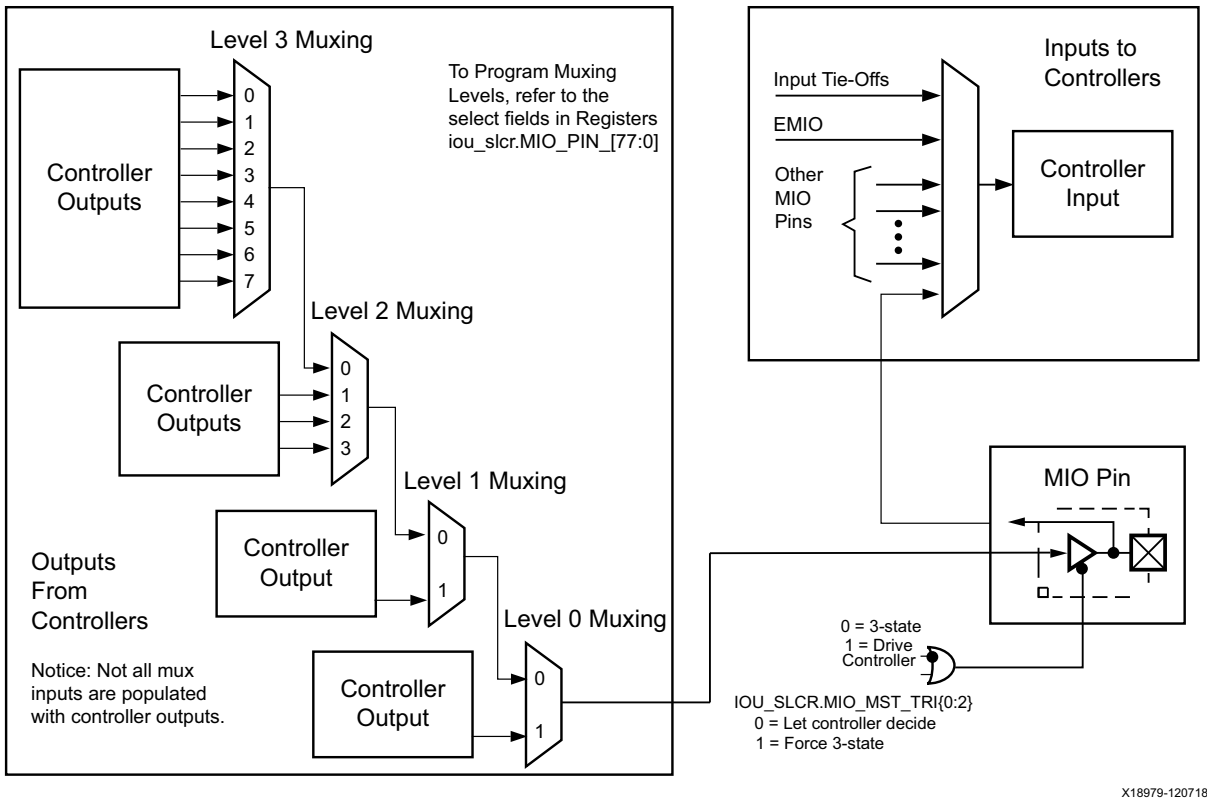

Figure 28-2: **MIO Multiplexing Stages and 3-State Output Control**

Master 3-state Enables

As shown in Figure 28-2, each pin has a master 3-state enable that overrides any interface specific output enable provided by the peripherals. The master enable is logically combined with the interface specific output enable signals (if provided) currently selected by the output enable multiplexer tree to produce a single output enable for connection to the I/O cell.

Access to the master enable control registers is on a bit-by-bit basis as the pins are configured or in parallel by accessing two 32-bit registers.

Default Logic Levels

The inputs to the I/O peripherals are driven with default values when another source is not routed to either the MIO or the EMIO. If an input is routed to EMIO, but the PL is powered down, then the same default value is driven to the I/O peripheral (see Figure 28-3.)

For MIO-only signals, the default signal input is driven when the MIO multiplexer does not route the signal to an MIO pin.

For MIO-EMIO signals, the default signal input is driven when the MIO multiplexer does not route the signal to an MIO pin (the signal defaults to the EMIO interface) and when the signal is programmed to be routed through the EMIO, but the PL either does not drive the signal (not configured) or is not able to drive it (powered down).

The default input signal logic levels are designed to be benign to the I/O peripheral. As a precaution, the related peripheral core should also be disabled when not in use. The logic levels are shown in the signal tables in each chapter for each I/O peripheral.

When PS_POR_B is asserted Low, the PS GPIO outputs connected to EMIO are forced and held High.

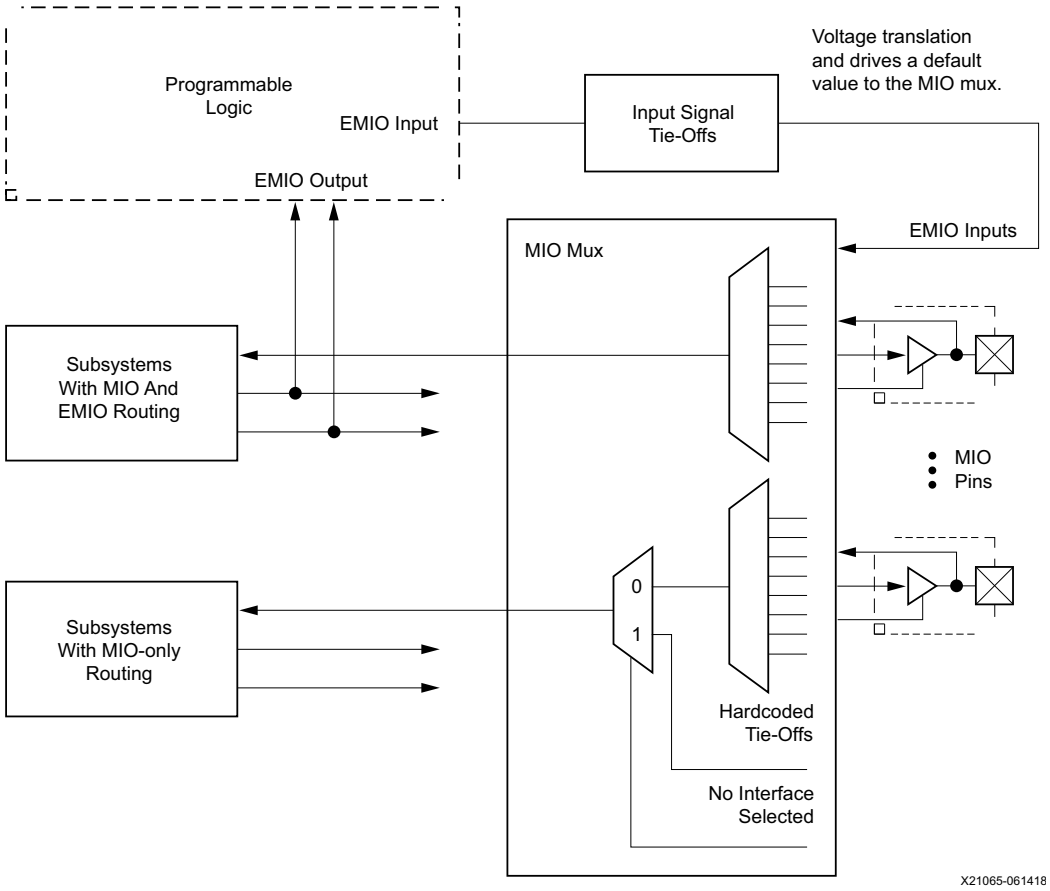

Figure 28-3: **Non-selected Controller Inputs**

The following table shows the state of each EMIO when PS only reset.

Table 28-1: **EMIO when PS only Reset**

Controller	MIO Signal	Vivado Generated Wrapper Signal	Output state when PS only reset is asserted
CAN	CAN 0 TX	emio_can0_phy_tx	1
	CAN 1 TX	emio_can1_phy_tx	1

Table 28-1: **EMIO when PS only Reset *(Cont'd)***

Controller	MIO Signal	Vivado Generated Wrapper Signal	Output state when PS only reset is asserted
GEM	N/A	[1:0] emio_enet0_dma_bus_width	0
		[1:0] emio_enet1_dma_bus_width	0
		[1:0] emio_enet2_dma_bus_width	0
		[1:0] emio_enet3_dma_bus_width	0
	N/A	emio_enet0_dma_tx_end_tog	0
		emio_enet1_dma_tx_end_tog	0
		emio_enet2_dma_tx_end_tog	0
		emio_enet3_dma_tx_end_tog	0
	Tx Data (7:0)	[7:0] emio_enet0_gmii_txd	FF
		[7:0] emio_enet1_gmii_txd	FF
		[7:0] emio_enet2_gmii_txd	FF
		[7:0] emio_enet3_gmii_txd	FF
	TX Enable	emio_enet0_gmii_tx_en	0
		emio_enet1_gmii_tx_en	0
		emio_enet2_gmii_tx_en	0
		emio_enet3_gmii_tx_en	0
	TX Error	emio_enet0_gmii_tx_er	1
		emio_enet1_gmii_tx_er	1
		emio_enet2_gmii_tx_er	1
		emio_enet3_gmii_tx_er	1

Table 28-1: **EMIO when PS only Reset *(Cont'd)***

Controller	MIO Signal	Vivado Generated Wrapper Signal	Output state when PS only reset is asserted
GEM	GEM0_MDC	emio_enet0_mdio_mdc	1
	GEM1_MDC	emio_enet1_mdio_mdc	1
	GEM2_MDC	emio_enet2_mdio_mdc	1
	GEM3_MDC	emio_enet3_mdio_mdc	1
	GEM0_MDIO	emio_enet0_mdio_o	1
	GEM1_MDIO	emio_enet1_mdio_o	1
	GEM2_MDIO	emio_enet2_mdio_o	1
	GEM3_MDIO	emio_enet3_mdio_o	1
	N/A	emio_enet0_mdio_t(3)	1
		emio_enet1_mdio_t(3)	1
		emio_enet2_mdio_t(3)	1
		emio_enet3_mdio_t(3)	1
	N/A	[7:0] emio_enet0_rx_w_data	FF
		[7:0] emio_enet1_rx_w_data	FF
		[7:0] emio_enet2_rx_w_data	FF
		[7:0] emio_enet3_rx_w_data	FF
	N/A	emio_enet0_rx_w_eop	1
		emio_enet1_rx_w_eop	1
		emio_enet2_rx_w_eop	1
		emio_enet3_rx_w_eop	1
	N/A	emio_enet0_rx_w_err	0
		emio_enet1_rx_w_err	0
		emio_enet2_rx_w_err	0
		emio_enet3_rx_w_err	0

Table 28-1: **EMIO when PS only Reset *(Cont'd)***

Controller	MIO Signal	Vivado Generated Wrapper Signal	Output state when PS only reset is asserted
GEM	N/A	emio_enet0_rx_w_flush	1
		emio_enet1_rx_w_flush	1
		emio_enet2_rx_w_flush	1
		emio_enet3_rx_w_flush	1
	N/A	emio_enet0_rx_w_sop	1
		emio_enet1_rx_w_sop	1
		emio_enet2_rx_w_sop	1
		emio_enet3_rx_w_sop	1
	N/A	[44:0] emio_enet0_rx_w_status	1FFF_FFFF_FFFF
		[44:0] emio_enet1_rx_w_status	1FFF_FFFF_FFFF
		[44:0] emio_enet2_rx_w_status	1FFF_FFFF_FFFF
		[44:0] emio_enet3_rx_w_status	1FFF_FFFF_FFFF
	N/A	emio_enet0_rx_w_wr	1
		emio_enet1_rx_w_wr	1
		emio_enet2_rx_w_wr	1
		emio_enet3_rx_w_wr	1
	Speed mode (2:0)[(1)]	[2:0] emio_enet0_speed_mode	7
		[2:0] emio_enet1_speed_mode	7
		[2:0] emio_enet2_speed_mode	7
		[2:0] emio_enet3_speed_mode	7
	N/A	emio_enet0_tx_r_rd	1
		emio_enet1_tx_r_rd	1
		emio_enet2_tx_r_rd	1
		emio_enet3_tx_r_rd	1

Table 28-1: **EMIO when PS only Reset *(Cont'd)***

Controller	MIO Signal	Vivado Generated Wrapper Signal	Output state when PS only reset is asserted
GEM	N/A	[3:0] emio_enet0_tx_r_status	F
		[3:0] emio_enet1_tx_r_status	F
		[3:0] emio_enet2_tx_r_status	F
		[3:0] emio_enet3_tx_r_status	F
	N/A	[93:0]emio_enet0_enet_tsu_timer_cnt	3FFFFFFF_FFFFFFFF_FFFFFFFF
	N/A	emio_enet0_delay_req_rx	1
		emio_enet1_delay_req_rx	1
		emio_enet2_delay_req_rx	1
		emio_enet3_delay_req_rx	1
	N/A	emio_enet0_delay_req_tx	1
		emio_enet1_delay_req_tx	1
		emio_enet2_delay_req_tx	1
		emio_enet3_delay_req_tx	1
	N/A	emio_enet0_pdelay_req_rx	1
		emio_enet1_pdelay_req_rx	1
		emio_enet2_pdelay_req_rx	1
		emio_enet3_pdelay_req_rx	1
	N/A	emio_enet0_pdelay_req_tx	1
		emio_enet1_pdelay_req_tx	1
		emio_enet2_pdelay_req_tx	1
		emio_enet3_pdelay_req_tx	1
	N/A	emio_enet0_pdelay_resp_rx	1
		emio_enet1_pdelay_resp_rx	1
		emio_enet2_pdelay_resp_rx	1
		emio_enet3_pdelay_resp_rx	1
	N/A	emio_enet0_pdelay_resp_tx	1
		emio_enet1_pdelay_resp_tx	1
		emio_enet2_pdelay_resp_tx	1
		emio_enet3_pdelay_resp_tx	1
	N/A	emio_enet0_rx_sof	1
		emio_enet1_rx_sof	1
		emio_enet2_rx_sof	1
		emio_enet3_rx_sof	1

Table 28-1: **EMIO when PS only Reset *(Cont'd)***

Controller	MIO Signal	Vivado Generated Wrapper Signal	Output state when PS only reset is asserted
GEM	N/A	emio_enet0_sync_frame_rx	1
		emio_enet1_sync_frame_rx	1
		emio_enet2_sync_frame_rx	1
		emio_enet3_sync_frame_rx	1
	N/A	emio_enet0_sync_frame_tx	1
		emio_enet1_sync_frame_tx	1
		emio_enet2_sync_frame_tx	1
		emio_enet3_sync_frame_tx	1
	N/A	emio_enet0_tsu_timer_cmp_val	1
		emio_enet1_tsu_timer_cmp_val	1
		emio_enet2_tsu_timer_cmp_val	1
		emio_enet3_tsu_timer_cmp_val	1
	N/A	emio_enet0_tx_r_fixed_lat	1
		emio_enet1_tx_r_fixed_lat	1
		emio_enet2_tx_r_fixed_lat	1
		emio_enet3_tx_r_fixed_lat	1
	N/A	emio_enet0_tx_sof	1
		emio_enet1_tx_sof	1
		emio_enet2_tx_sof	1
		emio_enet3_tx_sof	1
GPIO	N/A	[95:0] emio_gpio_o_temp	FFFFFFFF_FFFFFFFF_FFFFFFFF
	N/A	[95:0] emio_gpio_t_temp	0
I2C	I2C 0 SCL	emio_i2c0_scl_o	0
	I2C 1 SCL	emio_i2c1_scl_o	0
	N/A	emio_i2c0_scl_t(3)	1
		emio_i2c1_scl_t(3)	1
	I2C 0 SDA	emio_i2c0_sda_o	0
	I2C 1 SDA	emio_i2c1_sda_o	0
	N/A	emio_i2c0_sda_t(3)	1
		emio_i2c1_sda_t(3)	1
SDIO	SDIO 0 power control	emio_sdio0_buspower	1
	SDIO 1 power control	emio_sdio1_buspower	1
	SDIO 0 bus voltage	[2:0] emio_sdio0_bus_volt	7

Table 28-1: **EMIO when PS only Reset *(Cont'd)***

Controller	MIO Signal	Vivado Generated Wrapper Signal	Output state when PS only reset is asserted
SDIO	SDIO 1 bus voltage	[2:0] emio_sdio1_bus_volt	7
	SDIO 0 clock	emio_sdio0_clkout	1
	SDIO 1 clock	emio_sdio1_clkout	1
	SDIO 0 command	emio_sdio0_cmdena(3)	0
	SDIO 1 command	emio_sdio1_cmdena(3)	0
	SDIO 0 command	emio_sdio0_cmdout	1
	SDIO 1 command	emio_sdio1_cmdout	1
	SDIO 0 data [7:0]	[7 : 0] emio_sdio0_dataena(3)	1
	SDIO 1 data [7:0]	[7 : 0] emio_sdio1_dataena(3)	1
	SDIO 0 data{7:0}	[7 : 0] emio_sdio0_dataout	FF
	SDIO 1 data{7:0}	[7 : 0] emio_sdio1_dataout	FF
	SDIO 0 LED control	emio_sdio0_ledcontrol	1
	SDIO 1 LED control	emio_sdio1_ledcontrol	1
SPI	N/A	emio_spi0_mo_t(3)	1
		emio_spi1_mo_t(3)	1
	SPI 0 MOSI	emio_spi0_m_o	1
	SPI 1 MOSI	emio_spi1_m_o	1
	SPI 0 Clock	emio_spi0_sclk_o	1
	SPI 1 Clock	emio_spi1_sclk_o	1
	SPI 0 Clock	emio_spi0_sclk_t_n	0
	SPI 1 Clock	emio_spi1_sclk_t_n	0
	SPI 0 MISO	emio_spi0_s_o	1
	SPI 1 MISO	emio_spi1_s_o	1
	SPI 0 SS 3-state	emio_spi0_ss_n_t(3)	1
	SPI 1 SS 3-state	emio_spi1_ss_n_t(3)	1
	SPI 0 Slave Select 0	emio_spi0_ss_o_n	7
	SPI 0 Slave Select 1	emio_spi0_ss1_o_n	7
	SPI 0 Slave Select 2	emio_spi0_ss2_o_n	7
	SPI 1 Slave Select 0	emio_spi1_ss_o_n	7
	SPI 1 Slave Select 1	emio_spi1_ss1_o_n	7
	SPI 1 Slave Select 2	emio_spi1_ss2_o_n	7
	–	EMIOSPI0STN	0
	–	EMIOSPI1STN	0

Table 28-1: **EMIO when PS only Reset *(Cont'd)***

Controller	MIO Signal	Vivado Generated Wrapper Signal	Output state when PS only reset is asserted
TTC	ttc0_wave_out	[2:0]emio_ttc0_wave_o	0
	ttc1_wave_out	[2:0]emio_ttc1_wave_o	0
	ttc2_wave_out	[2:0]emio_ttc2_wave_o	0
	ttc3_wave_out	[2:0]emio_ttc3_wave_o	0
USB2	N/A	emio_u2dsport_vbus_ctrl_usb2_0	1
	N/A	emio_u2dsport_vbus_ctrl_usb2_1	1
USB3	N/A	emio_u3dsport_vbus_ctrl_usb3_0	1
	N/A	emio_u3dsport_vbus_ctrl_usb3_1	1
UART0	UART 0 Data Terminal Ready	emio_uart0_dtrn	1
	UART 0 Transmit	emio_uart0_txd	1
	UART 0 Ready to Send	emio_uart0_rtsn	1
UART1	UART 1 Data Terminal Ready	emio_uart1_dtrn	1
	UART 1 Transmit	emio_uart1_txd	1
	UART 1 Ready to Send	emio_uart1_rtsn	1
WDT	wdt0_rst_o(2)	emio_wdt0_rst_o	0
	wdt1_rst_o(2)	emio_wdt1_rst_o	0

Notes:

1. See Table 34-15 for more information.
2. wdt0_rst_o and wdt1_rst_o are active high.
3. Signal is inverted in Vivado wrapper.

MIO Pin Assignment Considerations

IMPORTANT: *There are several important MIO pin assignment considerations. The MIO-at-a-Glance table and these pin assignment considerations are helpful for pin planning. There are individual MIO signal tables for each controller/unit that uses the MIO pins.*

Interface Frequencies

The clocking frequency for an interface usually depends on the device speed grade and whether the interface is routed through the MIO or EMIO.

I/O Buffer Output Enable Control

The output enable for each MIO I/O buffer is controlled by a combination of the setting of the three-state override control bit, the selected signal type (input-only or not), and the state of the peripheral controller. The three-state override bit can be controlled from either of two places: the iou_slcr.MIO_PIN_xx register bit or the iou_slcr.MIO_MST_TRIx register bits. These bits control the same flip-flop to help control the three-state signal of the I/O buffer. The I/O buffer output is enabled when the three-state override control bit equals 0 and either the signal is an output-only or the I/O peripheral is driving a signal that is configured as I/O.

Boot from SD Card

The BootROM expects the SD card to be connected to MIO pins 13 through 25 for SD0 and MIO pins 39 through 51 for SD1.

eMMC Mapping

The SD1/eMMC can only operate in 4-bit mode when it is mapped to MIO bank 3.

Quad-SPI Interface

The lower memory Quad-SPI interface (QSPI_0) must be used when using the Quad-SPI memory subsystem. The upper interface (QSPI_1) is optional and is only used for a two-memory arrangement (parallel or stacked). Do not use the Quad-SPI 1 interface alone.

Drive Strength

After power up, the default I/O setting of the MIO banks 0, 1, and 2 is 8 mA.

MIO Table at a Glance

For pin planning, see Table 28-2. MIO signals are also listed in each controller chapter along with their function, direction, and presence in EMIO.

Table 28-2: **MIO Interfaces**

Interface	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
gem0																										
gem1																										
gem2																										
gem3																										
gem_tsu																										
qspi[2]	4	1	2	3	0	5	12	6	8	9	10	11	7													
nand										2	13	14	4	1	3	0	5	6	7	8	9	10	16	11	12	15
pcie[3]																										
usb0																										
usb1																										
pmu																										
sd0[1]														4	5	6	7	8	9	10	11	3	2	13	1	0
sd1[1]																										
CSU tamper																			0	0	0	0	0	0	0	0
DisplayPort aux																										
gpio0	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
gpio1																										
gpio2																										
can0			0	1			0	1			0	1			0	1			0	1			0	1		
can1	1	0			1	0			1	0			1	0			1	0			1	0			1	0
i2c0			0	1			0	1			0	1			0	1			0	1			0	1		

Interface	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51
gem0	0	1	2	3	4	5	6	7	8	9	10	11														
gem1													0	1	2	3	4	5	6	7	8	9	10	11		
gem2																										
gem3																										
gem_tsu																									0	0
qspi[2]																										
nand	2	13	14				4																			
pcie[3]				0	0	0		0	0	0	0	0														
usb0																										
usb1																										
pmu	0	1	2	3	4	5	6	7	8	9	10	11														
sd0[1]													2	1	3	4	5	6	7	8	9	10	11	13	0	
sd1[1]														8	9	10	11	13	0	1	4	5	6	7	3	2
CSU tamper	0					0	0	0																		
DisplayPort aux		0	1	2	3				0	1	2	3														
gpio0																										
gpio1	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
gpio2																										
can0	0	1			0	1			0	1			0	1			0	1			0	1			0	1
can1			1	0			1	0			1	0			1	0			1	0			1	0		
i2c0	0	1			0	1			0	1			0	1			0	1			0	1			0	1

Interface	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77
gem0																										
gem1																										
gem2	0	1	2	3	4	5	6	7	8	9	10	11														
gem3													0	1	2	3	4	5	6	7	8	9	10	11		
gem_tsu																										
qspi[2]																										
nand																										
pcie[3]																										
usb0	0	1	6	3	4	5	2	7	8	9	10	11														
usb1													0	1	6	3	4	5	2	7	8	9	10	11		
pmu																										
sd0[1]													2	1	3	4	5	6	7	8	9	10	11	13	0	
sd1[1]																		0	13	4	5	6	7	3	2	1
CSU tamper																										
DisplayPort aux																										
gpio0																										
gpio1																										
gpio2	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
can0			0	1			0	1			0	1			0	1			0	1			0	1		
can1	1	0			1	0			1	0			1	0			1	0			1	0			1	0
i2c0			0	1			0	1			0	1			0	1			0	1			0	1		

Table 28-2: **MIO Interfaces *(Cont'd)***

Interface	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	32	33	34	35	36	37
i2c1	0	1			0	1			0	1			0	1			0	1			0	1			0	1			0	1			0	1			0	1
pjtag	3	0	1	2									3	0	1	2											3	0	1	2								
lpd_swdt							0	1			0	1							0	1			0	1							0	1			0	1		
fpd_swdt					0	1			0	1							0	1			0	1			0	1							0	1			0	1
spi0	5	4	3	2	1	0							5	4	3	2	1	0									5	4	3	2	1	0						
spi1							5	4	3	2	1	0							1	4	3	2	5	0									5	4	3	2	1	0
ttc0							0	1							0	1							0	1							0	1						
ttc1					0	1							0	1							0	1							0	1							0	1
ttc2			0	1							0	1							0	1							0	1							0	1		
ttc3	0	1							0	1							0	1							0	1							0	1				
mdio{0:3}																																						
ua0			0	1			0	1			0	1			0	1			0	1			0	1			0	1			0	1			0	1		
ua1	1	0			1	0			1	0			1	0			1	0			1	0			1	0			1	0			1	0			1	0
trace	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17									6	7	8	9	10	11	12	13	14	15	16	17

Interface	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77
i2c1			0	1			0	1			0	1			0	1			0	1			0	1			0	1			0	1			0	1			0	1
pjtag	3	0	1	2											3	0	1	2			3	0	1	2																
lpd_swdt					0	1			0	1			0	1											0	1			0	1			0	1			0	1		
fpd_swdt							0	1			0	1							0	1							0	1			0	1			0	1				
spi0	5	4	3	2	1	0									5	4	3	2	1	0							5	4	3	2	1	0								
spi1							5	4	3	2	1	0									5	4	3	2	1	0							5	4	3	2	1	0		
ttc0	0	1							0	1							0	1							0	1							0	1						
ttc1							0	1							0	1							0	1							0	1								
ttc2					0	1							0	1							0	1							0	1										
ttc3			0	1							0	1							0	1							0	1												
mdio{0:3}													0	1																									0	1
ua0	0	1			0	1			0	1			0	1			0	1			0	1			0	1			0	1			0	1			0	1		
ua1			1	0			1	0			1	0			1	0			1	0			1	0			1	0			1	0			1	0				
trace	0	1	2	3	4	5									0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17								

Notes:

1. SD0/1 peripheral pins can also be configured as eMMC 0/1, respectively. The difference between SD and eMMC configuration is as follows.
   - The **Card Detect** and **Write Protect** signals are only available in SD mode.
   - The BUS_POW pin in SD mode is treated as a reset pin in eMMC mode.
   - In SD mode, data transfers in 1-bit and 4-bit modes. In eMMC mode, data transfers in 1-bit, 4-bit, and 8-bit modes.
   - If the SD interface is configured for SD 3.0, the signals SEL, DIR_CMD, DIR_0, and DIR_1_3 are mapped to sdio{0,1}_data_out [4], [5], [6], and [7], respectively.
2. In Quad-SPI loopback mode, leave the clk_for_lpbk signal floating. In Quad-SPI non-loopback mode, the clk_for_lpbk signal is not used by the Quad-SPI and can be used as a peripheral I/O (such as GPIO, CAN, or I2C).
3. The PCIe Root Port mode reset signals are routed to specific MIO pins as listed in Table 30-10.

Register Overview

Some MIO pins are programmed by the PMU ROM pre-boot. Some might also be programmed by the PMU user firmware, the CSU for the boot device, the FSBL, or other low-level code. The affected registers are listed in Table 28-3. All MIO registers use the IOU_SLCR register set and can be programmed in any order.

Table 28-3: **MIO Control Registers**

Description	Register Name	Type
Route I/O signals of IOP peripherals to MIO pins {0:77}.	MIO_PIN_{0:77}	R/W
Disable 3-state output buffers on MIO pins {0:77}.	MIO_MST_TRI{0:2}	R/W
Select input type (CMOS or Schmitt with hysteresis).	BANK{0:2}_CTRL3	R/W
Select internal pull-up or pull-down.	BANK{0:2}_CTRL4	R/W
Enable or disable internal resister.	BANK{0:2}_CTRL5	R/W
Select slew rate output (fast or slow).	BANK{0:2}_CTRL6	R/W
Select output drive strength (2 bits; 2, 4, 8, and 12 mA).	BANK{0:2}_CTRL{0, 1}	R /W
Read the voltage applied to PSIO bank.	BANK{0:2}_STATUS	R
Enable loopback function with MIO for SPI, UART, CAN, and I2C I/O interfaces.	MIO_LOOPBACK	R/W

Note: Setting MIO_MST_TRIx [PIN_xx_TRI] to `0` enables the GPIO to control the 3-state mode of the I/O. If the 3-state control is set to `1` in the MIO, then the output driver will be set to 3-state regardless of the GPIO settings.

Programming Model

Typically, the MIO configuration code is generated as part of the FSBL from the hardware project. An SDK export of a Vivado Design Suite project carries the PCW configuration information for the MIO pins. The SDK tools process the MIO configuration during FSBL creation.

I2C Interface Programming Example

The MIO can be configured to route the I2C interface signals to MIO pins 2 and 3. To route the I2C SCL signal to MIO pin 2, write `'h40` to the IOU_SLCR.MIO_PIN_2 register. To route the I2C SDA signal to MIO pin 3, write `'h40` to the IOU_SLCR.MIO_PIN_3 register.

The MIO pins and their signal names for each interface are listed in Table 28-4.

Table 28-4: **MIO Interfaces**

	Interface Type																																					
	gem0	gem1	gem2	gem 3	gem_ tsu	qspi(2)	nand	pcie(3)	usb0	usb1	pmu	sd0(1)	sd1(1)	test_ scan	csu	dpaux	gpio0	gpio1	gpio2	can0	can1	i2c0	i2c1	mdio0	pjtag	lpd_ swdt	fpd_ swdt	mdio1	spi0	spi1	mdio2	ttc0	ttc1	ttc2	ttc3	mdio3	ua0	ua1	trace
Size	12	12	12	12	1	13	17	1	12	12	12	13	13	38	1	4	26	26	26	2	2	2	2	2	4	2	2	2	6	6	2	2	2	2	2	2	2	2	18
Pin																																							
0						sclk_out 4								io[0] 0			io[0] 0				phy_tx 1		scl 0		tck 3				sclk 5						clk 0			txd 1	clk 0
1						io[1] 1								io[1] 1			io[1] 1				phy_rx 0		sda 1		tdi 0				n_ss_ out[2] 4					wave_ out 1			rxd 0	clk 1
2						io[2] 2								io[2] 2			io[2] 2			phy_rx 0		scl 0			tdo 1				n_ss_ out[1] 3				clk 0			rxd 0		dq_0 2
3						io[3] 3								io[3] 3			io[3] 3			phy_tx 1		sda 1			tms 2				n_ss_ out[0] 2				wave_ out 1			txd 1		dq_1 3
4						si_mio[0] 0								io[4] 4			io[4] 4				phy_tx 1		scl 0				clk_in 0		miso 1				clk 0				txd 1	dq_2 4
5						n_ss_ou t 5								io[5] 5			io[5] 5				phy_rx 0		sda 1				rst_out 1		mosi 0				wave_ out 1				rxd 0	dq_3 5
6						clk_for_l pbk 12								io[6] 6			io[6] 6			phy_rx 0		scl 0				clk_in 0			sclk 5		clk 0					rxd 0		dq_4 6
7						n_ss_ou t_ upper 6								io[7] 7			io[7] 7			phy_tx 1		sda 1				rst_out 1			n_ss_ out[2] 4		wave_ out 1					txd 1		dq_5 7
8						upper _io[0] 8								io[8] 8			io[8] 8				phy_tx 1		scl 0				clk_in 0			n_ss_ out[1] 3					clk 0			txd 1	dq_6 8
9						upper _io[1] 9	ce[1] 2							io[9] 9			io[9] 9				phy_rx 0		sda 1				rst_out 1			n_ss_ out[0] 2					wave_ out 1			rxd 0	dq_7 9
10						upper _io[2] 10	rb_n[0] 13							io[10] 10			io[10] 10			phy_rx 0		scl 0				clk_in 0			miso 1					clk 0			rxd 0		dq_8 10
11						upper _io[3] 11	rb_n[1] 14							io[11] 11			io[11] 11			phy_tx 1		sda 1				rst_out 1			mosi 0					wave_ out 1			txd 1		dq_9 11
12						sclk_ out_ upper 7	dqs 4							io[12] 12			io[12] 12				phy_tx 1		scl 0		tck 3				sclk 5				clk 0					txd 1	dq_10 12

The MIO pins and their signal names for each interface are listed in Table 28-4.

Table 28-4: **MIO Interfaces *(Cont'd)***

	Interface Type																																						
	gem0	gem1	gem2	gem 3	gem_ tsu	qspi(2)	nand	pcie(3)	usb0	usb1	pmu	sd0(1)	sd1(1)	test_ scan	csu	dpaux	gpio0	gpio1	gpio2	can0	can1	i2c0	i2c1	mdio0	pjtag	lpd_ swdt	fpd_ swdt	mdio1	spi0	spi1	mdio2	ttc0	ttc1	ttc2	ttc3	mdio3	ua0	ua1	trace
13							ce[0] 1					data_io [0] 4		io[13] 13			io[13] 13				phy_rx 0		sda 1		tdi 0				n_ss_ out[2] 4				wave_ out 1					rxd 0	dq_11 13
14							cle 3					data_io [1] 5		io[14] 14			io[14] 14			phy_rx 0		scl 0			tdo 1				n_ss_ out[1] 3			clk 0					rxd 0		dq_12 14
15							ale 0					data_io [2] 6		io[15] 15			io[15] 15			phy_tx 1		sda 1			tms 2				n_ss_ out[0] 2			wave_ out 1					txd 1		dq_13 15
16							dq[0] 5					data_io [3] 7		io[16] 16			io[16] 16				phy_tx 1		scl 0				clk_in 0		miso 1						clk 0			txd 1	dq_14 16
17							dq[1] 6					data_io [4] 8		io[17] 17			io[17] 17				phy_rx 0		sda 1				rst_out 1		mosi 0					wave_ out 1				rxd 0	dq_15 17
18							dq[2] 7					data_io [5] 9		io[18] 18	ext_ tamper 0		io[18] 18			phy_rx 0		scl 0				clk_in 0			miso 1				clk 0				rxd 0		
19							dq[3] 8					data_io [6] 10		io[19] 19	ext_ tamper 0		io[19] 19			phy_tx 1		sda 1				rst_out 1			n_ss_ out[2] 4				wave_ out 1				txd 1		
20							dq[4] 9					data_io [7] 11		io[20] 20	ext_ tamper 0		io[20] 20				phy_tx 1		scl 0				clk_in 0			n_ss_ out[1] 3			clk 0					txd 1	
21							dq[5] 10					cmd_io 3		io[21] 21	ext_ tamper 0		io[21] 21				phy_rx 0		sda 1				rst_out 1			n_ss_ out[0] 2				wave_ out 1				rxd 0	
22							we_b 16					clk_out 2		io[22] 22	ext_ tamper 0		io[22] 22			phy_rx 0		scl 0				clk_in 0			sclk 5			clk 0					rxd 0		
23							dq[6] 11					bus_po w 13		io[23] 23	ext_ tamper 0		io[23] 23			phy_tx 1		sda 1				rst_out 1			mosi 0			wave_ out 1					txd 1		
24							dq[7] 12					cd_n 1		io[24] 24	ext_ tamper 0		io[24] 24				phy_tx 1		scl 0				clk_in 0								clk 0			txd 1	
25							re_n 15					wp 0		io[25] 25	ext_ tamper 0		io[25] 25				phy_rx 0		sda 1				rst_out 1								wave_ out 1			rxd 0	
26	rgmii_ tx_clk 0				gem_ tsu_clk 0		ce[1] 2				gpi[0] 0			io[26] 26	ext_ tamper 0			io[0] 0		phy_rx 0		scl 0			tck 3				sclk 5				clk 0				rxd 0		dq_4 6
27	rgmii_ txd[0] 1						rb_n[0] 13				gpi[1] 1			io[27] 27		data_ out 0		io[1] 1		phy_tx 1		sda 1			tdi 0				n_ss_ out[2] 4				wave_ out 1			txd 1		dq_5 7

The MIO pins and their signal names for each interface are listed in Table 28-4.

Table 28-4: **MIO Interfaces *(Cont'd)***

	Interface Type																																				
	gem0	gem1	gem2	gem3	gem_tsu	qspi(2)	nand	pcie(3)	usb0	usb1	pmu	sd0(1)	sd1(1)	test_scan	csu	dpaux	gpio0	gpio1	gpio2	can0	can1	i2c0	i2c1	mdio0	pjtag	lpd_swdt	fpd_swdt	mdio1	spi0	spi1	mdio2	ttc0	ttc1	ttc2	ttc3	mdio3	ua0	ua1	trace
28	rgmii_txd[1] 2						rb_n[1] 14				gpi[2] 2			io[28] 28		hot_plug_detect 1		io[2] 2			phy_tx 1		scl 0		tdo 1				n_ss_out[1] 3				clk 0					txd 1	dq_6 8
29	rgmii_txd[2] 3							reset_n 0			gpi[3] 3			io[29] 29		data_oe 2		io[3] 3			phy_rx 0		sda 1		tms 2				n_ss_out[0] 2				wave_out 1					rxd 0	dq_7 9
30	rgmii_txd[3] 4							reset_n 0			gpi[4] 4			io[30] 30		data_in 3		io[4] 4		phy_rx 0		scl 0			clk_in 0			miso 1			clk 0					rxd 0		dq_8 10
31	rgmii_tx_ctl 5							reset_n 0			gpi[5] 5			io[31] 31	ext_tamper 0			io[5] 5		phy_tx 1		sda 1			rst_out 1			mosi 0			wave_out 1					txd 1		dq_9 11
32	rgmii_rx_clk 6						dqs 4				gpo[0] 6			io[32] 32	ext_tamper 0			io[6] 6			phy_tx 1		scl 0				clk_in 0			sclk 5					clk 0			txd 1	dq_10 12
33	rgmii_rxd[0] 7							reset_n 0			gpo[1] 7			io[33] 33	ext_tamper 0			io[7] 7			phy_rx 0		sda 1				rst_out 1			n_ss_out[2] 4					wave_out 1			rxd 0	dq_11 13
34	rgmii_rxd[1] 8							reset_n 0			gpo[2] 8			io[34] 34		data_out 0		io[8] 8		phy_rx 0		scl 0			clk_in 0				n_ss_out[1] 3				clk 0			rxd 0		dq_12 14
35	rgmii_rxd[2] 9							reset_n 0			gpo[3] 9			io[35] 35		hot_plug_detect 1		io[9] 9			phy_tx 1		sda 1			rst_out 1				n_ss_out[0] 2				wave_out 1			txd 1		dq_13 15
36	rgmii_rxd[3] 10							reset_n 0			gpo[4] 10			io[36] 36		data_oe 2		io[10] 10			phy_tx 1		scl 0				clk_in 0			miso 1			clk 0					txd 1	dq_14 16
37	rgmii_rx_ctl 11							reset_n 0			gpo[5] 11			io[37] 37		data_in 3		io[11] 11			phy_rx 0		sda 1				rst_out 1			mosi 0			wave_out 1					rxd 0	dq_15 17
38		rgmii_tx_clk 0										clk_out 2						io[12] 12		phy_rx 0		scl 0			tck 3				sclk 5			clk 0					rxd 0		clk 0
39		rgmii_txd[0] 1										cd_n 1	data_io[4] 8					io[13] 13		phy_tx 1		sda 1			tdi 0				n_ss_out[2] 4			wave_out 1					txd 1		clk 1
40		rgmii_txd[1] 2										cmd_io 3	data_io[5] 9					io[14] 14		phy_tx 1		scl 0			tdo 1				n_ss_out[1] 3					clk 0			txd 1	dq_0 2
41		rgmii_txd[2] 3										data_io[0] 4	data_io[6] 10					io[15] 15		phy_rx 0		sda 1			tms 2				n_ss_out[0] 2					wave_out 1			rxd 0	dq_1 3

The MIO pins and their signal names for each interface are listed in Table 28-4.

Table 28-4: **MIO Interfaces *(Cont'd)***

	Interface Type																																							
	gem0	gem1	gem2	gem 3	gem_ tsu	qspi(2)	nand	pcie(3)	usb0	usb1	pmu	sd0(1)	sd1(1)	test_ scan	csu	dpaux	gpio0	gpio1	gpio2	can0	can1	i2c0	i2c1	mdio0	pjtag	lpd_ swdt	fpd_ swdt	mdio1	spi0	spi1	mdio2	ttc0	ttc1	ttc2	ttc3	mdio3	ua0	ua1	trace
42		rgmii_t xd[3] 4										data_io [1] 5	data_io[7] 11					io[16] 16		phy_rx 0		scl 0				clk_in 0			miso 1					clk 0			rxd 0		dq_2 4
43		rgmii_t x_ctl 5										data_io [2] 6	bus_pow 13					io[17] 17		phy_tx 1		sda 1				rst_out 1			mosi 0					wave_ out 1			txd 1		dq_3 5
44		rgmii_r x_clk 6										data_io [3] 7	wp 0					io[18] 18			phy_tx 1		scl 0				clk_in 0			sclk 5			clk 0					txd 1	
45		rgmii_r xd[0] 7										data_io [4] 8	cd_n 1					io[19] 19			phy_rx 0		sda 1				rst_out 1			n_ss_ out[2] 4			wave_ out 1					rxd 0	
46		rgmii_r xd[1] 8										data_io [5] 9	data_io[0] 4					io[20] 20		phy_rx 0		scl 0				clk_in 0			n_ss_ out[1] 3		clk 0					rxd 0		
47		rgmii_r xd[2] 9										data_io [6] 10	data_io[1] 5					io[21] 21		phy_tx 1		sda 1				rst_out 1			n_ss_ out[0] 2		wave_ out 1					txd 1		
48		rgmii_r xd[3] 10										data_io [7] 11	data_io[2] 6					io[22] 22			phy_tx 1		scl 0				clk_in 0			miso 1				clk 0			txd 1	
49		rgmii_r x_ctl 11										bus_pow 13	data_io[3] 7					io[23] 23			phy_rx 0		sda 1				rst_out 1			mosi 0				wave_ out 1			rxd 0	
50					gem_ts u_clk 0							wp 0	cmd_io 3					io[24] 24		phy_rx 0		scl 0				clk_in 0		gem1_ mdc 0					clk 0			rxd 0		
51					gem_ts u_clk 0								clk_out 2					io[25] 25		phy_tx 1		sda 1				rst_out 1		gem1_ mdio 1					wave_ out 1			txd 1		
52			rgmii_tx _clk 0						ulpi_cl k_in 0										io[0] 0		phy_tx 1		scl 0		tck 3				sclk 5				clk 0				txd 1	clk 0
53			rgmii_tx d[0] 1						ulpi_di r 1										io[1] 1		phy_rx 0		sda 1		tdi 0				n_ss_ out[2] 4			wave_ out 1				rxd 0	clk 1
54			rgmii_tx d[1] 2						ulpi_rx _data[2] 6										io[2] 2	phy_rx 0		scl 0			tdo 1				n_ss_ out[1] 3		clk 0				rxd 0		dq_0 2
55			rgmii_tx d[2] 3						ulpi_n xt 3										io[3] 3	phy_tx 1		sda 1			tms 2				n_ss_ out[0] 2		wave_ out 1				txd 1		dq_1 3

The MIO pins and their signal names for each interface are listed in Table 28-4.

Table 28-4: **MIO Interfaces *(Cont'd)***

	Interface Type																																						
	gem0	gem1	gem2	gem 3	gem_ tsu	qspi(2)	nand	pcie(3)	usb0	usb1	pmu	sd0(1)	sd1(1)	test_ scan	csu	dpaux	gpio0	gpio1	gpio2	can0	can1	i2c0	i2c1	mdio0	pjtag	lpd_ swdt	fpd_ swdt	mdio1	spi0	spi1	mdio2	ttc0	ttc1	ttc2	ttc3	mdio3	ua0	ua1	trace
56			rgmii_tx_d[3] 4						ulpi_rx_data[0] 4										io[4] 4		phy_tx 1		scl 0				clk_in 0		miso 1						clk 0			txd 1	dq_2 4
57			rgmii_tx_ctl 5						ulpi_rx_data[1] 5										io[5] 5		phy_rx 0		sda 1				rst_out 1		mosi 0						wave_out 1			rxd 0	dq_3 5
58			rgmii_rx_clk 6						ulpi_stp 2										io[6] 6	phy_rx 0		scl 0			tck 3					sclk 5				clk 0			rxd 0		dq_4 6
59			rgmii_rx_d[0] 7						ulpi_rx_data[3] 7										io[7] 7	phy_tx 1		sda 1			tdi 0					n_ss_out[2] 4				wave_out 1			txd 1		dq_5 7
60			rgmii_rx_d[1] 8						ulpi_rx_data[4] 8										io[8] 8		phy_tx 1		scl 0		tdo 1					n_ss_out[1] 3				clk 0			txd 1	dq_6 8
61			rgmii_rx_d[2] 9						ulpi_rx_data[5] 9										io[9] 9		phy_rx 0		sda 1		tms 2					n_ss_out[0] 2				wave_out 1			rxd 0	dq_7 9
62			rgmii_rx_d[3] 10						ulpi_rx_data[6] 10										io[10] 10	phy_rx 0		scl 0				clk_in 0			miso 1		clk 0					rxd 0	dq_8 10
63			rgmii_rx_ctl 11						ulpi_rx_data[7] 11										io[11] 11	phy_tx 1		sda 1				rst_out 1			mosi 0		wave_out 1					txd 1		dq_9 11
64				rgmii_tx_clk 0						ulpi_clk_in 0		clk_out 2							io[12] 12		phy_tx 1		scl 0				clk_in 0		sclk 5						clk 0			txd 1	dq_10 12
65				rgmii_txd[0] 1						ulpi_dir 1		cd_n 1							io[13] 13		phy_rx 0		sda 1				rst_out 1		n_ss_out[2] 4						wave_out 1			rxd 0	dq_11 13
66				rgmii_txd[1] 2						ulpi_rx_data[2] 6		cmd_io 3							io[14] 14	phy_rx 0		scl 0				clk_in 0			n_ss_out[1] 3				clk 0			rxd 0		dq_12 14
67				rgmii_txd[2] 3						ulpi_nxt 3		data_io[0] 4							io[15] 15	phy_tx 1		sda 1				rst_out 1			n_ss_out[0] 2				wave_out 1			txd 1		dq_13 15
68				rgmii_txd[3] 4						ulpi_rx_data[0] 4		data_io.[1] 5							io[16] 16		phy_tx 1		scl 0				clk_in 0		miso 1				clk 0					txd 1	dq_14 16

The MIO pins and their signal names for each interface are listed in Table 28-4.

Table 28-4: **MIO Interfaces *(Cont'd)***

	Interface Type																																								
	gem0	gem1	gem2	gem 3	gem_ tsu	qspi(2)	nand	pcie(3)	usb0	usb1	pmu	sd0(1)	sd1(1)	test_ scan	csu	dpaux	gpio0	gpio1	gpio2	can0	can1	i2c0	i2c1	mdio0	pjtag	lpd_ swdt	fpd_ swdt	mdio1	spi0	spi1	mdio2	ttc0	ttc1	ttc2	ttc3	mdio3	ua0	ua1	trace
69				rgmii_ tx_ctl 5						ulpi_rx_ data[1] 5		data_io [2] 6	wp 0						io[17] 17		phy_rx 0		sda 1				rst_out 1		mosi 0					wave_ out 1					rxd 0	dq_15 17
70				rgmii_ rx_clk 6						ulpi_stp 2		data_io [3] 7	bus_pow 13						io[18] 18	phy_rx 0		scl 0				clk_in 0				sclk 5		clk 0					rxd 0		
71				rgmii_ rxd[0] 7						ulpi_rx_ data[3] 7		data_io [4] 8	data_io [0] 4						io[19] 19	phy_tx 1		sda 1				rst_out 1				n_ss_ out[2] 4		wave_ out 1					txd 1		
72				rgmii_ rxd[1] 8						ulpi_rx_ data[4] 8		data_io [5] 9	data_io [1] 5						io[20] 20		phy_tx 1		scl 0					clk_in 0			n_ss_ out[1] 3							txd 1		
73				rgmii_ rxd[2] 9						ulpi_rx_ data[5] 9		data_io [6] 10	data_io[2] 6						io[21] 21		phy_rx 0		sda 1				rst_out 1				n_ss_ out[0] 2							rxd 0		
74				rgmii_ rxd[3] 10						ulpi_rx_ data[6] 10		data_io [7] 11	data_io[3] 7						io[22] 22	phy_rx 0		scl 0				clk_in 0				miso 1							rxd 0		
75				rgmii_ rx_ctl 11						ulpi_rx_ data[7] 11		bus_po w 13	cmd_io 3						io[23] 23	phy_tx 1		sda 1				rst_out 1				mosi 0							txd 1		
76												wp 0	clk_out 2						io[24] 24		phy_tx 1		scl 0	gem0_ mdc 0				gem1_ mdc 0			gem2_ mdc 0				gem3_ mdc 0			
77													cd_n 1						io[25] 25		phy_rx 0		sda 1	gem0_ mdio 1				gem1_ mdio 1			gem2_ mdio 1				gem3_ mdio 1			

Notes:

1. SD0/1 peripheral pins can also be configured as eMMC 0/1, respectively. The difference between SD and eMMC configuration is as follows.
   - The **Card Detect** and **Write Protect** signals are only available in SD mode.
   - The BUS_POW pin in SD mode is treated as a reset pin in eMMC mode.
   - In SD mode, data transfers in 1-bit and 4-bit modes. In eMMC mode, data transfers in 1-bit, 4-bit, and 8-bit modes.
   - If the SD interface is configured for SD 3.0, the signals SEL, DIR_CMD, DIR_0, and DIR_1_3 are mapped to data_io[4], data_io[5], data_io[6], and data_io[7], respectively.
2. In Quad-SPI loopback mode, leave the clk_for_lpbk signal floating. In Quad-SPI non-loopback mode, the clk_for_lpbk signal is not used by the Quad-SPI and can be used as a peripheral I/O (such as GPIO, CAN, or I2C).
3. The PCIe Root Port mode reset signals are routed to specific MIO pins as listed in Table 30-10.