UG0686
User Guide
PolarFire FPGA User I/O

Microsemi
Power Matters.™

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA
Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996
Email: sales.support@microsemi.com
www.microsemi.com

About Microsemi

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; enterprise storage and communication solutions, security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, Calif., and has approximately 4,800 employees globally. Learn more at www.microsemi.com.

50200686. 1.0 2/17

Contents

Figures

Tables

1 Revision History

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

1.1 Revision 1.0

Revision 1.0 is the first publication of this document.

2 I/O Overview

PolarFire™ device user I/Os support multiple I/O standards while simultaneously providing the high bandwidth needed to maximize the internal logic capabilities of the device and achieve the required system-level performance. They are specifically designed for ease of use and rapid system integration.

PolarFire devices have two types of user I/Os:

- General-purpose I/O (GPIO), which supports a wide range of I/O standards operating with supplies between 1.2 V to 3.3 V nominal. These I/Os operate at speeds of up to 1.066 Gbps for single-ended standards, and 1.25 Gbps using differential standards.
- High-speed I/O (HSIO), which supports I/O standards operating with supplies between 1.1 V to 1.8 V. These I/Os are optimized for high-speed and support operations at speeds of up to 1.6 Gbps.

GPIO and HSIO are organized in I/O banks and each I/O bank has dedicated I/O supplies. The unused supplies are connected to grounds to reduce noise leakage. In addition to GPIO and HSIO, a number of I/Os are associated with PolarFire FPGA system controller and with transceiver clocks and data pads. These I/Os are powered up independently of other user I/O banks. For more information, see *UG0677: PolarFire FPGA Transceiver User Guide, UG0714: PolarFire FPGA Programming User Guide*, and *UG0722: PolarFire FPGA Packaging and Pin Descriptions User Guide*.

This chapter describes the features and supported standards for each of these user I/O types, providing details about PolarFire FPGA I/O banks and I/O naming conventions.

2.1 GPIO and HSIO Features

PolarFire devices support different I/O features for GPIO and HSIO. The following is a summary of I/O features:

2.1.1 GPIO Features

- Supports 1.2 V to 3.3 V operation
- Single-ended input and output modes
- Flexible supply voltage for certain I/O standards
- Reference, differential, and complementary input receiver modes
- True current-based differential output driver modes and pseudo-differential complementary output modes
- Single-ended static or dynamic termination at 1.8 V and 1.5 V
- Differential static or dynamic termination of 100 Ω
- Cold-sparing and hot socketing (hot plug-in or hot-swaping) capabilities
- Process, voltage, and temperature (PVT)-compensated programmable drive strengths
- Supports full and reduced drive for SSTL18 (as defined by JEDEC standards)
- Built-in weak pull-up, pull-down, and bus-keeper circuits
- Programmable hysteresis
- DDR3 support at up to 1.066 Gbps
- Support for the low-power Flash*Freeze mode

2.1.2 HSIO Features

- Supports 1.1 V to 1.8 V operation
- Single-ended input and output modes
- Mixed single-ended input modes for LVTTL/LVCMOS, regardless of power supply level
- Reference, differential, and complementary input receiver modes
- Pseudo-differential complementary output modes
- Single-ended static or dynamic termination at 1.8 V, 1.5 V, 1.35 V, 1.2 V, and 1.1 V (only for LVSTL)
- PVT-compensated programmable drive strengths
- Supports full and reduced drives for SSTL18 as defined by JEDEC standards
- Built-in weak pull-up, pull-down, and bus-keeper circuits
- DDR3 support at up to 1333 Mbps and DDR4 support at up to 1.6 Gbps
- Support for the low-power Flash*Freeze mode

2.2 Supported I/O Standards

PolarFire FPGA GPIO and HSIO have configurable high-performance I/O drivers and receivers, supporting a wide variety of I/O standards.

The following table lists the I/O standards supported in the receiver and transmitter modes respectively.

Table 1 • **Slew Rate Control**

I/O Standards	Receiver/Transmitter Modes	V_{DDI} (Nominal) Required	Bank Types	Applications
Single-Ended Standards				
PCI	Receiver, Transmitter	3.3 V	GPIO	PC and embedded systems
LVTTL[1]	Receiver	3.3 V, 2.5 V, 1.8 V, 1.5 V, 1.2 V	GPIO	General purpose
	Transmitter	3.3 V		
LVCMOS33[1]	Receiver	3.3 V, 2.5 V, 1.8 V, 1.5 V, 1.2 V	GPIO	General purpose
	Transmitter	3.3 V		
LVCMOS25[1]	Receiver	3.3 V, 2.5 V, 1.8 V, 1.5 V, 1.2 V	GPIO	General purpose
	Transmitter	2.5 V		
LVCMOS18[1]	Receiver	3.3 V, 2.5 V, 1.8 V, 1.5 V, 1.2 V	GPIO, HSIO	General purpose
	Transmitter	1.8 V		
LVCMOS15[1]	Receiver	3.3 V, 2.5 V, 1.8 V, 1.5 V, 1.2 V	GPIO, HSIO	General purpose
	Transmitter	1.8 V		
LVCMOS12[1]	Receiver	3.3 V, 2.5 V, 1.8 V, 1.5 V, 1.2 V	GPIO, HSIO	General purpose
	Transmitter	1.2 V		
SSTL25I, SSTL25II	Receiver	2.5 V	GPIO	DDR1
	Transmitter	2.5 V	GPIO	DDR1
SSTL18I, SSTL18II	Receiver, Transmitter	1.8 V	GPIO, HSIO	DDR2/RLDRAM2
SSTL15I, SSTL15II	Receiver, Transmitter	1.5 V	GPIO, HSIO	DDR3
SSTL135I, SSTL135II	Receiver, Transmitter	1.35 V	HSIO	DDR3L
HSTL15I, HSTL15II	Receiver, Transmitter	1.5 V	GPIO, HSIO	QDRII+
HSTL135I, HSTL135II	Receiver, Transmitter	1.35 V	HSIO	RLDRAM3
HSTL12I	Receiver, Transmitter	1.2 V	HSIO	QDRII+
HSUL18I, HSUL18II	Receiver, Transmitter	1.8 V	GPIO, HSIO	LPDDR
HSUL12I, HSUL12II	Receiver, Transmitter	1.2 V	HSIO	LPDDR2, LPDDR3

Table 1 • **Slew Rate Control *(continued)***

I/O Standards	Receiver/Transmitter Modes	V_{DDI} (Nominal) Required	Bank Types	Applications
POD12I, POD12II	Receiver, Transmitter	1.2 V	HSIO	DDR4
Differential Standards				
LVDS33	Receiver	3.3 V	GPIO	General purpose
	Transmitter[2]	3.3 V	GPIO	General purpose
LVDS25	Receiver	2.5V	GPIO	General purpose
	Transmitter[2]	2.5V	GPIO	General purpose
LVDS18[3, 4]	Receiver	1.8 V	HSIO	General purpose
RSDS33	Receiver	3.3 V	GPIO	General purpose
	Transmitter[2]	3.3 V	GPIO	General purpose
RSDS25	Receiver	2.5 V	GPIO	General purpose
	Transmitter[2]	2.5 V	GPIO	General purpose
RSDS18[4]	Receiver	1.8 V	HSIO	General purpose
MINILVDS33	Receiver	3.3 V	GPIO	General purpose
	Transmitter[2]	3.3 V	GPIO	General purpose
MINILVDS25	Receiver	2.5V	GPIO	General purpose
	Transmitter[2]	2.5V	GPIO	General purpose
MINILVDS18[4]	Receiver	1.8 V	HSIO	General purpose
SUBLVDS33	Receiver	3.3 V	GPIO	General purpose
	Transmitter[2]	3.3 V	GPIO	General purpose
SUBLVDS25	Receiver	2.5V	GPIO	General purpose
	Transmitter[2]	2.5V	GPIO	General purpose
SUBLVDS18[4]	Receiver	1.8 V	HSIO	General purpose
PPDS33	Receiver	3.3 V	GPIO	General purpose
	Transmitter[2]	3.3 V	GPIO	General purpose
PPDS25	Receiver	2.5 V	GPIO	General purpose
	Transmitter[2]	2.5 V	GPIO	General purpose
PPDS18[4]	Receiver	1.8 V	HSIO	General purpose
SLVS33	Receiver	3.3 V	GPIO	General purpose
SLVS25	Receiver	2.5 V	GPIO	General purpose
SLVS18	Receiver	1.8 V	HSIO	General purpose
SLVSE15[5]	Transmitter	1.5 V	GPIO, HSIO	General purpose
HCSL33	Receiver	3.3 V	GPIO	General purpose
HCSL25	Receiver	2.5 V	GPIO	General purpose
HCSL18	Receiver	1.8 V	HSIO	General purpose
BUSLVDSE25[5]	Transmitter	2.5 V	GPIO	Multipoint backplane applications
MLVDSE25[5]	Transmitter	2.5 V	GPIO	Multipoint backplane applications
LVPECL33	Receiver	3.3 V	GPIO	Video graphics and clock distribution

Table 1 • **Slew Rate Control *(continued)***

I/O Standards	Receiver/Transmitter Modes	V_{DDI} (Nominal) Required	Bank Types	Applications
LVPECLE33[5]	Transmitter	3.3 V	GPIO	Video graphics and clock distribution
MIPI12	Receiver	1.2 V	GPIO	Consumer mobile applications
MIPIE33[5]	Transmitter	3.3 V	GPIO	Consumer mobile applications, High–speed Mode

1. Certain I/O standards are designed to support flexible V_{DDI} assignment, see Flexible VDDI, page 19.
2. Buffers configured for these standards are true-differential transmitters that do not support bidirectional operations.
3. For HSIO, native LVDS inputs are supported with a single external-differential termination resistor, and emulated LVDS outputs are supported using two single-ended LVCMOS buffers with external resistors.
4. These standards require an external voltage reference (V_{REF}) and require two single-ended drivers with biasing through external resistors.
5. Buffers are configured as emulated-differential transmitters and also support bidirectional operations. However, they require an external board termination.

2.2.1 I/O Standard Descriptions

This section provides an overview for each of the I/O standards supported by PolarFire FPGA I/Os.

2.2.1.1 3.3 V Peripheral Component Interface (PCI)

PolarFire FPGA GPIO supports the PCI I/O standards. The PCI standard uses an LVTTL input buffer and a push-pull output buffer. This standard is used for both 33 MHz and 66 MHz PCI bus applications.

2.2.1.2 Low-Voltage TTL (LVTTL)

LVTTL is a general-purpose standard (EIA/JESD8-B) for 3.3 V applications. It uses an LVTTL input buffer and a push-pull output buffer. PolarFire FPGA GPIO supports the LVTTL I/O standards, and the LVTTL output buffer can have up to six different programmable drive strengths. For more information about programmable drive strength control, see Table 6, page 14.

2.2.1.3 Low-Voltage CMOS (LVCMOS)

LVCMOS is a general-purpose standard implemented in CMOS transistors. PolarFire devices support five different LVCMOS operational modes:

- **LVCMOS33—**an extension of the LVCMOS standard (JESD8-B-compliant) is used for general-purpose 3.3 V applications.
- **LVCMOS25—**an extension of the LVCMOS standard (JESD8-5-compliant) is used for general-purpose 2.5 V applications.
- **LVCMOS18—**an extension of the LVCMOS standard (JESD8-7-compliant) is used for general-purpose 1.8 V applications.
- **LVCMOS15—**an extension of the LVCMOS standard (JESD8-11-compliant) is used for general-purpose 1.5 V applications.
- **LVCMOS12—**an extension of the LVCMOS standard (JESD8-26-compliant) is used for general-purpose 1.2 V applications.

2.2.1.4 Stub Series Terminated Logic (SSTL)

Stub series terminated logic (SSTL) is a general-purpose memory bus standard. PolarFire devices support the following SSTL operational modes:

- **SSTL25I—**SSTL Class I-standard with V_{DDI} (nominal) = 2.5 V
- **SSTL25II—**SSTL Class II-standard with V_{DDI} (nominal) = 2.5 V
- **SSTL18I—**SSTL Class I-standard with V_{DDI} (nominal) = 1.8 V
- **SSTL18II—**SSTL Class II-standard with V_{DDI} (nominal) = 1.8 V
- **SSTL15I—**SSTL Class I-standard with V_{DDI} (nominal) = 1.5 V
- **SSTL15II—**SSTL Class II-standard with V_{DDI} (nominal) = 1.5 V
- **SSTL135I—**SSTL Class I-standard with V_{DDI} (nominal) = 1.35 V
- **SSTL135II—**SSTL Class II-standard with V_{DDI} (nominal) = 1.35 V

SSTL25 is defined by the JEDEC standard, JESD8-9B, and used for DDR SDRAM and DDR1 memory interfaces. SSTL18 is defined by the JEDEC standard, JESD8, and used for DDR2 SDRAM memory interfaces. SSTL15 is used for DDR3 memory interfaces; SSTL135 is used for DDR3L memory interfaces.

For more information about signal levels for the various SSTL I/O standards, see *DS0141: PolarFire FPGA Datasheet*.

2.2.1.5 High-Speed Transceiver Logic (HSTL)

HSTL is a general-purpose, high-speed bus standard (EIA/JESD8-6) with a signaling range between 0 V and 1.5 V, and signals can either be single-ended or differential. This standard is used in memory bus interfaces with data switching capabilities of up to 1.267 GHz.

PolarFire devices support the following HSTL operational modes:

- **HSTL15I—**HSTL Class I-standard with V_{DDI} (nominal) = 1.5 V
- **HSTL15II—**HSTL Class II-standard with V_{DDI} (nominal) = 1.5 V
- **HSTL135I—**HSTL Class I-standard with V_{DDI} (nominal) = 1.35 V
- **HSTL135II—**HSTL Class II-standard with V_{DDI} (nominal) = 1.35 V
- **HSTL12I—**HSTL Class I-standard with V_{DDI} (nominal) = 1.2 V
- **HSTL12I—**HSTL Class II-standard with V_{DDI} (nominal) = 1.2 V

For more information about signal levels for the various HSTL I/O standards, see Table 1, page 2. also, see *DS0141: PolarFire FPGA Datasheet*.

Note: HSTL135 and HSTL12 are not part of the JEDEC specification; they are scaled from HSTL15. For more information about HSTL signal levels, see *DS0141: PolarFire FPGA Datasheet*.

2.2.1.6 High-Speed Unterminated Logic (HSUL)

HSUL, as specified by the JEDEC standard JESD8-22, is a standard for LPDDR2 and LPDDR3 memory buses. PolarFire devices support HSUL I/O standards in both HSIO and GPIO.

2.2.1.7 Pseudo Open Drain (POD)

POD standards are intended for DDR4, DDR4L, and LLDRAM3 applications. PolarFire FPGA HSIO supports both POD receive and transmit modes.

2.2.1.8 Low-Voltage Differential Signal (LVDS)

Low-voltage differential signaling (ANSI/TIA/EIA-644) is a high-speed, differential I/O standard. The voltage swing between two signal lines is approximately 350 mV. PolarFire FPGA GPIO supports LVDS receive and transmit modes. PolarFire FPGA HSIO supports LVDS receive mode with an external 100 Ω board termination, see I/O External Termination, page 20 for more information.

2.2.1.9 Reduced-Swing Differential Signal (RSDS)

Reduced-swing differential signaling is similar to an LVDS high-speed interface using differential signaling, but with a smaller voltage swing and requiring a parallel termination resistor. RSDS is only intended for point-to-point applications. For more information about RSDS Voltage Swing, see *DS0141: PolarFire FPGA Datasheet*.

While PolarFire devices support RSDS receive and transmit modes with GPIO, PolarFire FPGA HSIO supports RSDS receive mode with an external 100 Ω on-board termination.

2.2.1.10 Mini-LVDS

Mini-LVDS is a unidirectional interface from the timing controller to the column drivers in TFT LCD displays, and is specified in Texas Instruments standard, SLDA007A. PolarFire FPGA GPIO supports mini-LVDS in both receive and transmit modes. PolarFire FPGA HSIO supports mini-LVDS only in the receive mode and requires an external resistor..

2.2.1.11 Sub-LVDS

Sub-LVDS is a differential low-voltage standard that is a subset of LVDS, and uses a reduced-voltage swing and lower common-mode voltage compared to LVDS. For sub-LVDS, the maximum differential swing is 200 mV compared to 350 mV for LVDS. The nominal common-mode voltage for sub-LVDS is 0.9 V, while it is 1.25 V for LVDS. PolarFire FPGA GPIO supports sub-LVDS in both receive and transmit modes. PolarFire FPGA HSIO supports sub-LVDS only in the receive mode and requires an external resistor.

2.2.1.12 Point-to-Point Differential Signaling (PPDS)

PPDS is the next generation of the RSDS standards introduced by National Semiconductor Corporation, and is used to interface to next-generation LCD row and column drivers. PPDS inputs require a parallel termination resistor.

PolarFire FPGA GPIO supports PPDS in both receive and transmit modes. HSIO supports PPDS only in receive mode and requires an external resistor.

2.2.1.13 Scalable Low-Voltage Signaling (SLVS)

SLVS is a chip-to-chip signaling standard designed for maximum performance with minimum power consumption, inheriting low noise susceptibility from LVDS. The standard features a scaled-down 400 mV signal swing, versus the 700 mV swing of LVDS, and includes a ground reference. PolarFire devices support the SLVS I/O standards in GPIO and HSIO banks, but an external resistor is required for transmitter mode. For more information, see Implementing Emulated Standards for Outputs, page 21.

2.2.1.14 High-Speed Current Steering Logic (HCSL)

HCSL is a differential output standard used in PCI Express applications. Both GPIO and HSIO in PolarFire devices support the HCSL I/O standards (receive-only mode). Although, the common mode range for this standard is from 250 mV to 550 mV, PolarFire FPGA HCSL I/O receivers support a wider range of 50 mV to 2.4 V.

2.2.1.15 Bus-LVDS (B-LVDS)/Multipoint LVDS (M-LVDS)

B-LVDS refers to bus interface circuits based on the LVDS technology with the M-LVDS specification extending the LVDS standard to high-performance multipoint bus applications. Multidrop and multipoint bus configurations may contain any combination of drivers, receivers, and transceivers. LVDS drivers provide the higher drive current required by B-LVDS and M-LVDS to accommodate bus loading. These drivers require series terminations for better signal quality and voltage swing control. The drivers can be located anywhere on the bus, and therefore termination is also required at both ends of the bus.

PolarFire FPGA GPIO supports B-LVDS and M-LVDS in receive mode. For transmit mode, however, external board termination is required. For more information about various BLVDS standards, see Bus-LVDS Emulated (BLVDSE25) Output Mode, page 21, and Multipoint Low-Voltage Emulated (MLVDSE25) Output Mode, page 22.

2.2.1.16 Low-Voltage Positive Emitter-Coupled Logic (LVPECL)

LVPECL is a 3.3 V differential signal standard that transmits one data bit over a pair of signal lines, thus requiring two pins per input or output. The voltage swing between the two signal lines is approximately 850 mV. While LVPECL input is supported for PolarFire FPGA GPIO, external board termination is required for the LVPECL outputs. For more information about LVPECL33, see LVPECL Emulated (LVPECLE33) Output Mode, page 22.

2.2.1.17 Mobile Industry Processor Interface (MIPI) D-PHY

MIPI is a serial communication interface used in camera and display applications. PolarFire devices support implementing the MIPI D-PHY standards in GPIO bank using an external termination. For more information, see Implementing MIPI D-PHY, page 23.

2.3 I/O Banks

Depending upon the device size, each PolarFire device has five, six, or eight user I/O banks. The I/O banks on the north side of the device support only HSIO. Each I/O bank has dedicated I/O supplies and grounds. Each I/O within a given bank shares the same V_{DDI} power supply, and the same V_{REF} reference voltage. Only compatible I/O standards can be assigned to a given I/O bank.

Each bank contains a bank power detector, and a bank receiver reference voltage generator to create an internally generated reference voltage, V_{REF}. Each bank also interfaces with a PVT controller to calibrate the I/O buffer output driver strengths and termination values (needed only for certain I/O standards). The PVT controller generates a set of codes to control the source driver and the sink driver, and also calibrates the HSIO output slew. Each I/O buffer has individual drive-strength programmability to multiply the PVT digital code value by a drive setting to create the desired drive, impedance, or termination settings. For more information, see I/O Analog (IOA) Buffer Programmable Features, page 12.

Figure 1, page 7 through Figure 3, page 8 show simplified PolarFire device floorplans for each device, including the bank locations. These figures also show the corner block and transceiver block. The corner block includes CCCs and two PLLs and two DLLs each, providing flexible clock management and synthesis for the FPGA fabric, external system, and I/Os. Note that all banks are not available in all devices, see I/O Lanes in Each Bank, page 36 for more information. For more information about CCC and PolarFire FPGA transceivers, see *UG0684: PolarFire FPGA Clocking Resources User Guide* and *UG0677: PolarFire FPGA Transceiver User Guide*.

Figure 1 • **MPFS300 and MPFS500 Device I/O Banks**

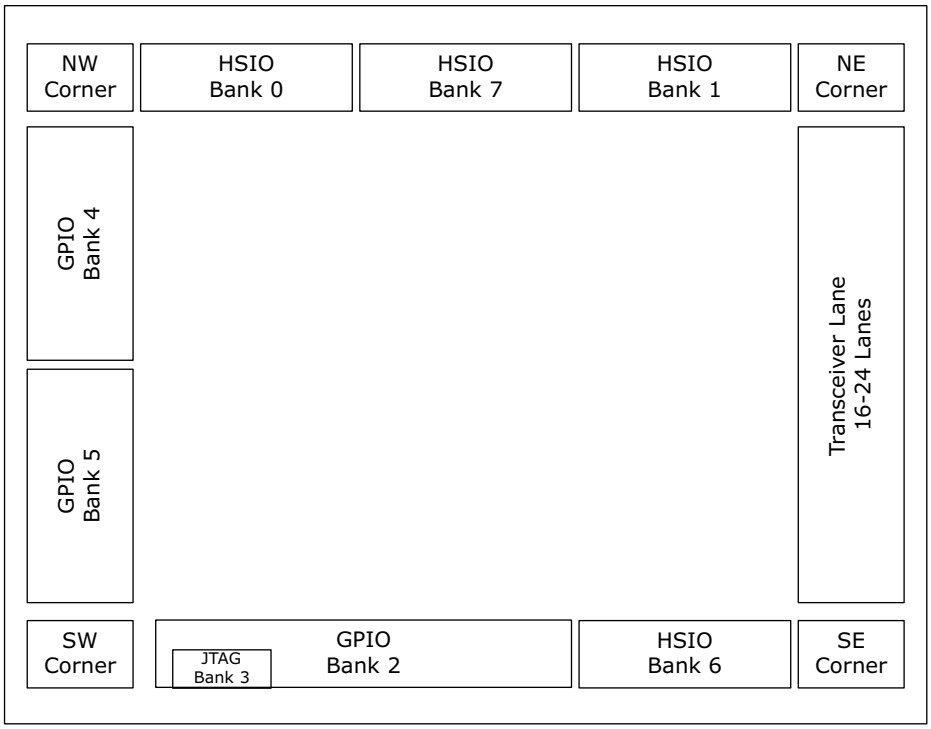


Figure 2 • **MPFS200 Device I/O Banks**

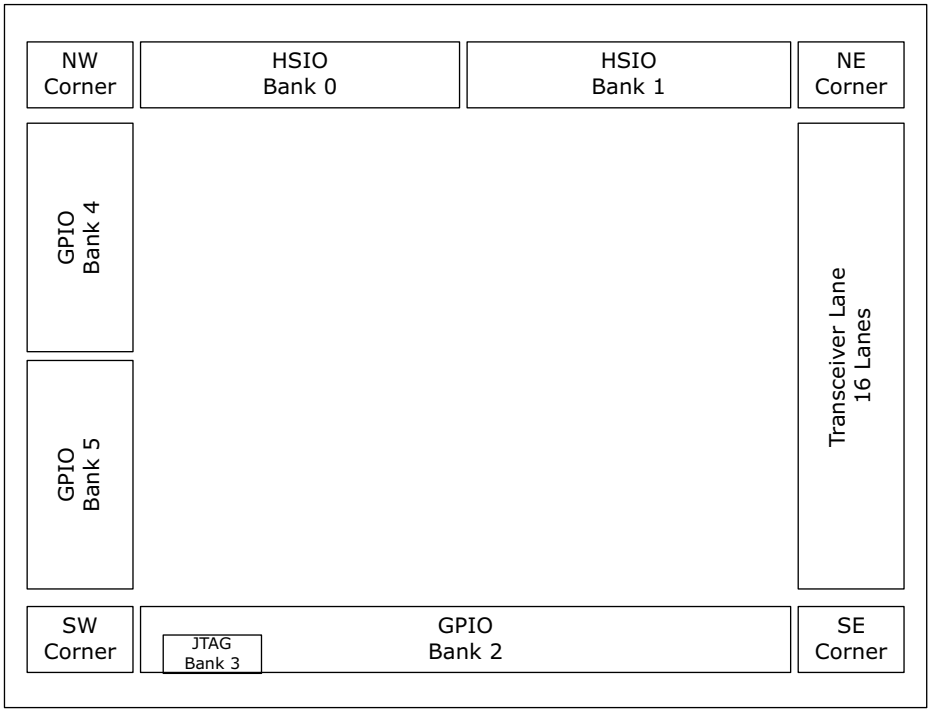
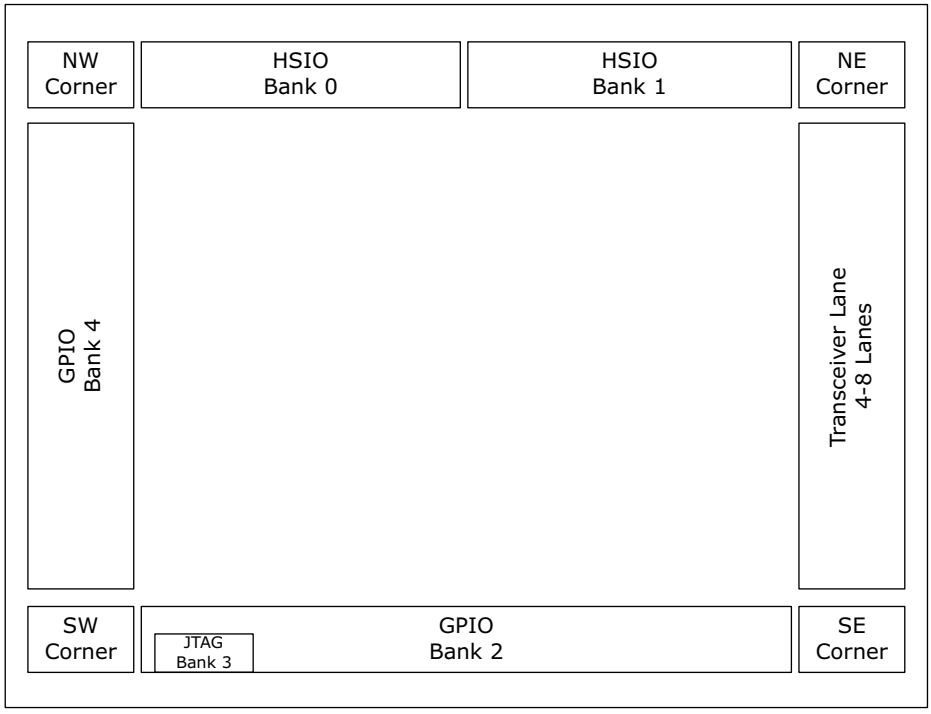


Figure 3 • **MPFS100 Device I/O Banks**



2.4 Supply Voltages for PolarFire FPGA I/O Banks

PolarFire devices have multiple I/O banks that require the following bank power supplies listed in the table:

Table 2 • **Supply Pin**

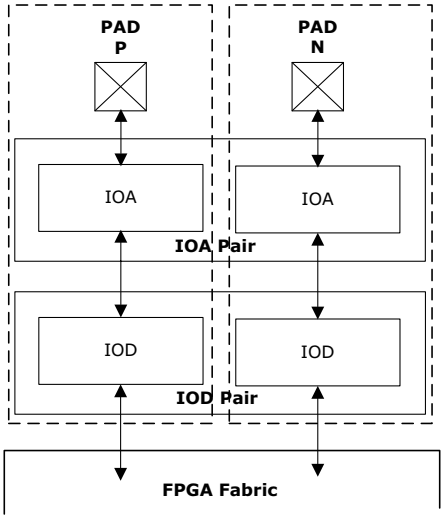
Name	Description	Operating Voltage	Unused Condition
V_{DDIx}	Supply for I/O circuits in a bank	For JTAG bank—1.8 V/2.5 V/3.3 V For GPIO bank—1.2 V/1.5 V/ 1.8 V/2.5 V/3.3 V For HSIO bank—1.2 V/1.5 V/1.8 V	The bank supply is not allowed to be left floating for any unused bank. It must be powered down to ground by direct connection
V_{DD25}	Power for corner PLLs and PNVM	2.5 V	Must connect to 2.5 V
V_{DD18}	Power for programming and HSIO receiver	1.8 V	Must connect to 1.8 V
V_{DDAUXx}	Auxiliary supply for I/O circuits. Auxiliary supply voltage must be set to 2.5 V or 3.3 V and must be always equal to or higher than V_{DDIx}	Greater than or equal to V_{DDI}	Must connect to greater than or equal to V_{DDI}
V_{REF}	V_{REF} is the supply reference voltage for reference receivers. Each bank can have only one V_{REF} value. V_{REF} can be externally supplied or internally generated, see Supply Voltages for PolarFire FPGA I/O Banks, page 9 for V_{REF} assignment use model for more information.	Depends on the I/O standards	The regular I/Os are used as V_{REF} supply. Libero® configures unused user I/Os as input buffer disabled or output buffer tri-stated with weak pull-up.

2.5 PolarFire FPGA I/O Overview

Each PolarFire FPGA I/O is composed of an analog I/O buffer (referred to as IOA) and a digital logic block (referred to as IOD). IOA blocks include analog input and output buffers, while IOD blocks include a logic that enables the IOA buffer to interface with the FPGA fabric. The IOD also includes data bus digital logic to widen the bus to and from the IOA, allowing the external pins to run at a much faster clock rate than the fabric logic.

To support a variety of I/O standards, PolarFire FPGA I/Os are organized into pairs, as shown in the following illustration. The two I/O paths in a pair, labeled as positive (P) and negative (N) respectively, can be configured as two separate single-ended I/Os, as one differential, or as a complementary I/O pair.

Figure 4 • **PolarFire FPGA I/O Pair**



The IOA buffer includes a transmit and receive buffer, on-die termination (Thévenin, differential, up, and down), a slew-rate control circuit, a bus-keeper circuit, and a programmable weak pull-up or pull-down resistor. The transmit and receive buffers transfer signals between the I/O pad and the IOD. Figure 5, page 11 shows the overview of IOA buffer.

2.5.1 Single-Ended Transmitter and Receiver Mode

An I/O buffer can be configured as either a single-ended transmitter, a single-ended receiver, or both. Both, PolarFire FPGA GPIO and HSIO support single-ended mode.

2.5.2 Differential Transmitter Mode

The I/O buffer pair allows implementing both true differential output mode and pseudo-differential output mode. The true differential output mode uses an LVDS H-bridge-type driver. The pseudo-differential output mode, also known as complementary-mode, consists of two single-ended drivers where one driver's output is inverted relative to the other. The pseudo-differential output drivers have lower signal integrity and performance, and usually require biasing by external resistors to emulate true differential signal levels. Only PolarFire FPGA GPIO bank supports true differential output modes using a differential current driver. Both, PolarFire FPGA GPIO and HSIO banks support complementary output modes.

2.5.3 Differential Receiver Mode

Both GPIO and HSIO receivers support operations in differential receiver mode, where the input data from the differential pair of pads (PAD P and PAD N) is received on both pads and is then driven to the FPGA fabric from the IOD block on the P side.

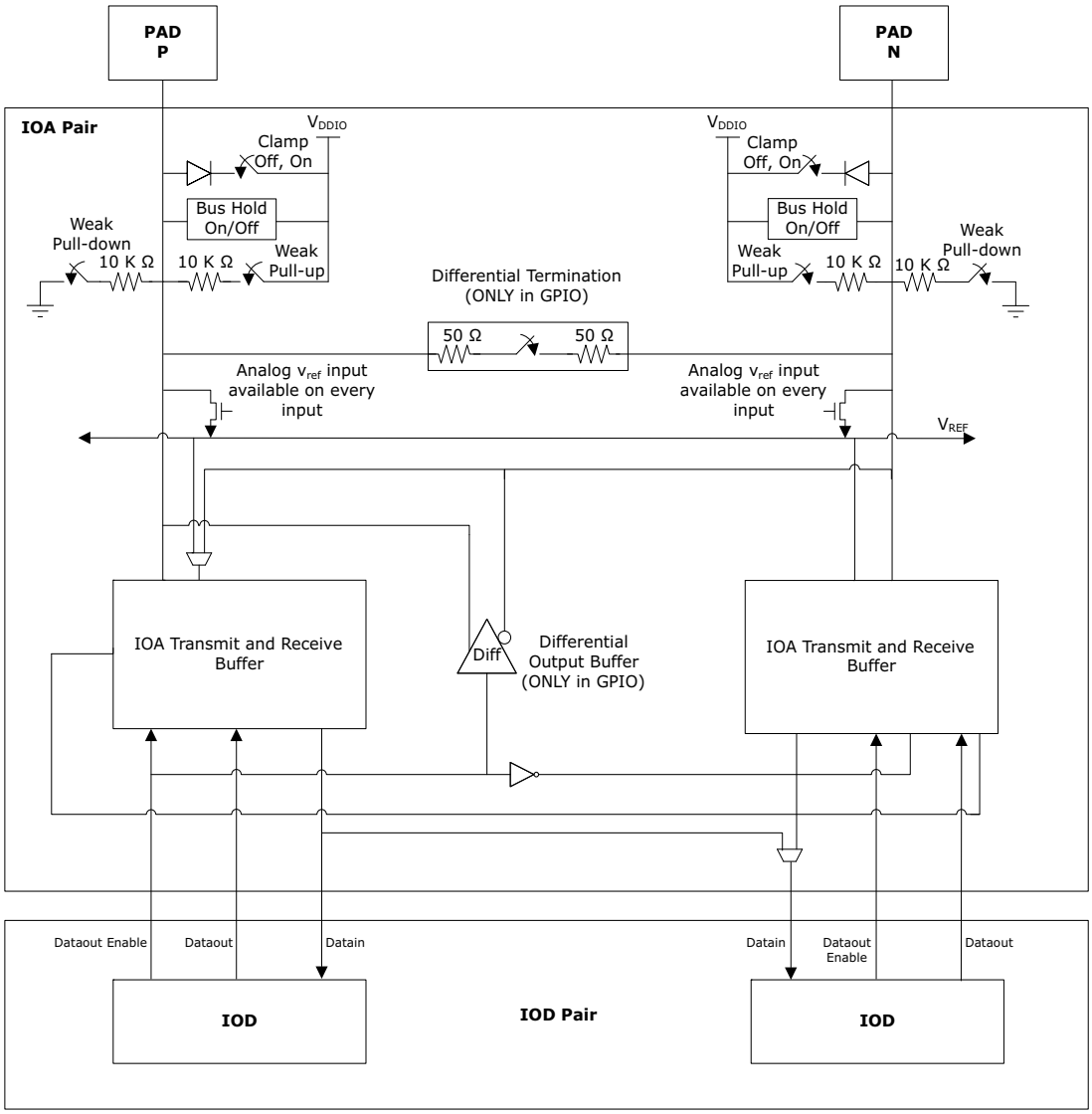
Libero SoC controls the enabling and disabling of the transmit and receive buffer based upon the selected standard and I/O mode, whether single-ended or differential. For more information about IOA buffer and its use model, see I/O Features and Implementation, page 12.

2.5.4 I/O Digital (IOD) Buffer

The IOD block interfaces with the FPGA fabric on one side and the IOA buffers on the other side, and deserializes and transfers input data to a lower core clock speed, or transfers lower-speed data from the fabric to the high-speed output clock domain, serializing it in the process. The PolarFire FPGA I/O digital block works in conjunction with fast and low-skew clock networks. It also includes special clock dividers and other support circuits to guarantee clock domain crossings. The I/O digital block deserializes high-speed DDR input data and transfers to FPGA fabric at lower speeds, and also serializes the lower speed FPGA fabric data and transfers to high-speed DDR output. For more information about IOD buffer and its use models, see IOD Features and User Modes, page 27.

The following illustration shows the PolarFire FPGA I/O pair block diagram.

Figure 5 • **PolarFire FPGA I/O Pair (Detailed View)**



2.6 I/O Primitive

The PolarFire FPGA macro library includes a list of I/O primitives to support various I/O standards. Following are the generic I/O primitives, representing most of the available I/O standards.

- **INBUF—**represents input buffer
- **INBUF_DIFF—**represents differential input buffer
- **OUTBUF—**represents output buffer
- **OUTBUF_DIFF—**represents differential output buffer
- **TRIBUFF—**represents tri-state buffer
- **TRIBUFF_DIFF—**represents differential tri-state buffer

For more information about macro library, see *PolarFire FPGA Macro Library User Guide*.

3 I/O Features and Implementation

This chapter describes PolarFire FPGA I/O features and provides details about their use. It also provides guidelines for implementing the various I/O standards using PolarFire FPGA I/Os. Note that the terms receive and input, transmit and output are used interchangeably in this document.

3.1 I/O Analog (IOA) Buffer Programmable Features

PolarFire FPGA GPIO and HSIO provide a number of programmable features. These features are set using the I/O attribute editor in Libero SoC, or through PDC commands. The following sections describe these features.

3.1.1 Slew Rate Control

PolarFire FPGA GPIO supports slew rate control in non-differential output mode. Turning the slew rate on results in faster slew rate, which improves the available timing margin, see *DS0141: PolarFire FPGA Datasheet* for the timing data. When slew rate is turned off, the device uses the default slew rate to reduce the impact of simultaneous switching noise (SSN).

The following table lists the I/O standards that support slew rate control.

Table 3 • **Slew Rate Control**

I/O Standards	Supported I/O Type	Slew Rate Control Options
PCI	GPIO (output only)	On (default), Off
LVTTL	GPIO (output only)	On (default), Off
LVCMOS25 and LVCMOS33	GPIO (output only)	On (default), Off

Slew rate settings are controlled using the I/O attribute editor in Libero SoC, or by using the following PDC command:

```
set_io –slew <value>
```

The value can be set as on or off.

Slew rate control is not available in PolarFire FPGA HSIO buffers. However, these buffers have built-in PVT-compensated slew rate controllers for optimized signal integrity.

3.1.1.1 Programmable Weak Pull- Up/Down and Bus-Keeper (Hold) Circuits

PolarFire devices have a programmable weak pull-down (10 KΩ), pull-up (10 KΩ), and bus-keeper circuit on every I/O pad when in input mode. Weak pull-up and pull-down circuits create a default setting for an input when it is not driven. The bus-keeper circuit is used to weakly hold the signal on an I/O pin at its last driven state, keeping it at a valid level with minimal power dissipation. The bus-keeper circuitry also pulls undriven pins away from the input threshold voltage where noise can cause unintended oscillation. The programmable weak pull-down, pull-up, and bus-keeper circuits are disabled when the output driver is enabled. Differential inputs do not support the programmable weak pull-down, pull-up, and bus-keeper modes.

The following table lists the I/O standards that support weak pull- up/down and bus-keeper control.

Table 4 • **Weak Pull and Bus-Keeper Control**

I/O Standards	Supported I/O Types	Weak Pull and Bus-Keeper Control Options
LVTTL LVCMOS33, LVCMOS25, LVCMOS18, LVCMOS15, and LVCMOS12 PCI	GPIO (input only)	Off Weak pull-down Weak pull-up Bus-keeper
LVCMOS18, LVCMOS15, and LVCMOS12	HSIO (input only)	Off Weak pull-down Weak pull-up Bus-keeper

The programmable weak pull-down, pull-up, and bus-keeper settings are controlled by using the I/O attribute editor in Libero SoC, or by using the following PDC command:

```
set_io -res_pull <value>
```

The value can be set as up, down, hold, or none.

3.1.1.2 Schmitt Trigger Input Hysteresis

PolarFire FPGA GPIO and HSIO can be configured as a Schmitt Trigger input that, when enabled, exhibits a hysteresis that helps to filter out the noise at the receiver and prevents double-glitching caused by noisy input edges.

The following table lists the I/O standards that support the Schmitt Trigger feature. For more information about hysteresis values for different I/O standards when Schmitt Trigger mode is enabled, see *DS0141: PolarFire FPGA Datasheet*.

Table 5 • **Schmitt Trigger Control**

I/O Standards	Supported I/O Types	Schmitt Trigger Control Options
LVTTL LVCMOS33 LVCMOS25 PCI	GPIO (input only)	On Off
LVCMOSI15 V LVCMOSI18 V	HSIO (input only)	On Off

Schmitt Trigger mode is enabled by using the I/O attribute editor in Libero SoC, or by using the following PDC command:

```
set_io -schmitt_trigger <value>
```

The value can be set as on or off.

3.1.1.3 Programmable Output Drive Strength

For LVCMOS, LVTTL, LVDS, and PPDS I/O standards, the PolarFire FPGA I/O output buffer has programmable drive strength control to mitigate the effects of high signal attenuation caused by long transmission lines.

The following table lists the programmable drive strength support and settings in PolarFire devices.

Table 6 • **Programmable Drive Strength Control**

I/O Standards	Supported I/O Types	Drive Strength Settings (mA)
LVTTL	GPIO (output only)	2, 4, 8, 12, 16, 20
LVCMOS33	GPIO (output only)	2, 4, 8, 12, 16, 20
LVCMOS25	GPIO (output only)	2, 4, 6, 8, 12, 16
LVDS25 and LVDS33	GPIO (output only)	3, 3.5, 4, 6
RSDS33 and RSDS25	GPIO (output only)	1.5, 2, 3
MINILVDS33 and MINILVDS25	GPIO (output only)	3, 3.5, 4, 6
SUBLVDS33 and SUBLVDS25	GPIO (output only)	1, 1.5, 2
PPDS33 and PPDS25	GPIO (output only)	1.5, 2, 3
LVCMOS18	GPIO and HSIO (output only)	2, 4, 6, 8, 10, 12
LVCMOS15	GPIO and HSIO (output only)	2, 4, 6, 8, 10
LVCMOS12[1]	GPIO and HSIO (output only)	2, 4, 6, 8, 10

1. LVCMOS12 output drive strength of 10 mA is supported only for HSIO.

The programmable drive strength is set by using the I/O attribute editor in Libero SoC or by using the following PDC command:

```
set_io –output_drive <value>
```

Values can be set as listed in Table 6, page 14.

3.1.1.4 Programmable Output Impedance Control

For voltage reference I/O standards, PolarFire FPGA I/Os provide the option to control the driver impedance for certain I/O standards: SSTL, HSUL, HSTL, POD, and LVSTL.

The following table lists the programmable output impedance support and settings in PolarFire devices.

Table 7 • **Programmable Output Impedance Standards**

I/O Standards	Supported I/O Types	Impedance (Ω)
SSTL25I	GPIO	48, 60, 80, 120
SSTL25II	GPIO	34, 40, 48, 60
SSTL18I	GPIO and HSIO	40, 48, 60, 80
SSTL18II	GPIO and HSIO	30, 34, 40, 48
SSTL15I	GPIO and HSIO	40, 48
SSTL15II	GPIO and HSIO	27, 30, 34
SSTL135I	HSIO	40, 48
SSTL135II	HSIO	27, 30, 34
HSUL18I	GPIO and HSIO	34, 40, 55, 60
HSUL18II	GPIO and HSIO	22, 25, 27, 30
HSTL15I	GPIO and HSIO	34, 40, 50, 60
HSTL15II	GPIO and HSIO	22, 25, 27, 30

Table 7 • **Programmable Output Impedance Standards *(continued)***

I/O Standards	Supported I/O Types	Impedance (Ω)
HSTL135I	HSIO	34, 40, 50, 60
HSTL135II	HSIO	22, 25, 27, 30
HSUL12I	HSIO	34, 40, 48, 60, 80, 120
POD12I	HSIO	40, 48, 60
POD12II	HSIO	27, 30, 34
LVSTLI	HSIO	30, 34, 40, 48, 60, 80, 120, 240
LVSTLII	HSIO	30, 34, 40, 48, 60, 80, 120, 240

The output impedance values can be programmed by using the I/O attribute editor in Libero SoC, or by using the following PDC command:

```
set_io –impedance <value>
```

values can be set as listed in Table 7, page 14.

3.1.1.5 On-Die Termination (ODT)

ODT is used to terminate input signals, helping to maintain signal quality, saving board space, and reducing external component costs. In PolarFire FPGAs, ODT is available in receive mode and also in bidirectional mode when the I/O acts as an input. If ODT is not used or not available, the PolarFire FPGA I/O standards may require external termination for better signal integrity. For more information, see I/O External Termination, page 20.

ODT can be a pull-up, pull-down, differential, or Thévenin termination with both static and dynamic control available, and is set by using the I/O attribute editor in Libero SoC or by using a PDC command.

In addition, ODT can be controlled dynamically for individual I/Os as well as for all I/Os in a lane simultaneously on a per-lane basis. For more information about I/O lanes, see PolarFire FPGA I/O Lanes, page 32.

The following table lists ODT support in PolarFire devices.

Table 8 • **ODT Support in GPIO and HSIO**

I/O Standards	I/O Types (Input Only)	ODT Control	ODT Type	ODT (Ω)
LVDS33, LVDS25 RSDS33, RSDS25, MINILVDS33, MINILVDS25, SUBLVDS33, SUBLVDS25, LVPECL33,LVPECL25	GPIO, HSIO	Off Static Dynamic	Off Differential	100
SSTL18I, SSTL18II	GPIO, HSIO	Off Static Dynamic	Off Thévenin	50, 75, 150
SSTL15I, SST15II SSTL135I, SSTL135II	GPIO, HSIO	Off Static Dynamic	Off Thévenin	20, 30, 40, 60, 120
POD12I, POD12II	HSIO	Off Static Dynamic	Off Up	34, 40, 48, 60, 120, 240

Table 8 • **ODT Support in GPIO and HSIO** ***(continued)***

I/O Standards	I/O Types (Input Only)	ODT Control	ODT Type	ODT (Ω)
HSUL12I, HSUL12II	HSIO	Off Static Dynamic	Off Up	120, 240
HSTL15I, HSTL15II	GPIO	Off Static Dynamic	Off Differential	50
HSUL18I, HSUL18II	GPIO, HSIO	Off Static Dynamic	Off Differential	50
LVCMOS25	GPIO, HSIO	Off Static	Off Down	120, 240
LVCMOS18, LVCMOS15, LVCMOS12	GPIO, HSIO	Off Static	Off Up Down Thévenin	60, 120, 240

The static ODT setting and values can be programmed by using the I/O attribute editor in Libero SoC, or by using the following PDC command.

```
set_io -ODT <value> -ODT_VALUE <odt_value>
```

Value can be set as on or off and odt_value can be set as listed in Table 8, page 15.

3.1.1.6 Common Mode Voltage (Vcm) Settings

PolarFire FPGA GPIO and HSIO inputs allow common mode settings for differential receivers. It help preventing common-mode mismatches between devices.

The following table lists the programmable differential termination control support and settings in PolarFire devices. For more information about common mode voltage levels for various I/O standards, see *DS0141: PolarFire FPGA Datasheet*.

Table 9 • **Programmable Differential Termination Control**

I/O Standards	Supported I/O Types	Differential Termination Type[1]
SSTL18	GPIO, HSIO	Off, Low, Mid
HSUL18	GPIO, HSIO	Off, Low, Mid
SSTL15	GPIO, HSIO	Off, Low, Mid
HSTL15	GPIO, HSIO	Off, Low, Mid
SSTL135	HSIO	Off, Low, Mid
HSTL135	HSIO	Off, Low[1], Mid
HSUL12I	HSIO	Off, Low, Mid
HSTL12	HSIO	Low, Mid
POD12	HSIO	Off, Low, Mid
SSTL25	GPIO	
SLVS25	GPIO, HSIO	MID (HSIO) Low, Mid (GPIO)
HCSL25	GPIO, HSIO	MID (HSIO) Low, Mid (GPIO)

Table 9 • **Programmable Differential Termination Control *(continued)***

I/O Standards	Supported I/O Types	Differential Termination Type[1]
SLVSE	GPIO, HSIO	Off, Mid (HSIO) Off, Low, Mid (GPIO)
PPDS25	GPIO, HSIO	Mid (HSIO) Off, Low, Mid (GPIO)
MLVDSE	GPIO	Off, Low, Mid
BUSLVDS	GPIO	Off, Low, Mid
LVPECL	GPIO	Low, Mid
LVDS	GPIO, HSIO	Mid (HSIO) Off, Low, Mid (GPIO)
RSDS	GPIO, HSIO	Mid (HSIO) Off, Low, Mid (GPIO)
MINILVDS	GPIO, HSIO	Mid (HSIO) Off, Low, Mid (GPIO)

1. For more information about low and mid differential termination types, see *DS0141: PolarFire FPGA Datasheet*.

The programmable differential termination control values can be programmed by using the I/O attribute editor in Libero SoC or by using the following PDC command:

```
set_io –vcm_range <value>
```

Value can be set as listed in Table 9, page 16.

3.1.1.7 Programmable Clamp Diode

PolarFire devices have internal PCI clamp diodes for both HSIO and GPIO. PCI clamp diodes help reduce the voltage level at the input, and are mainly used when the voltage overshoot exceeds the maximum allowable limit. If signaling levels of the receiver are greater than the V_{DDIx} of the bank, the clamp diode must be off to support hot-socketing insertion, see Cold Sparing and Hot Socketing, page 26 for more information.

For GPIO, clamp diodes can be programmed to be on or off by using the I/O attribute editor in Libero SoC, or by using a PDC command. For HSIO, the internal clamp diode is always on.

The following table lists clamp diodes that are programmable in PolarFire I/O devices.

Table 10 • **Programmable Clamp Diode**

I/O Standards	Supported I/O Type	Clamp Diode Control
LVTTL, LVCMOS33, LVCMOS25, LVCMOS18, LCMOS15, LVCMOS12, SSTL25, SSTL18I, SSTL18II, SSTL15I, SSTL15II, HSTL15I, HSTL15II	GPIO	Off, On

The following PDC command is used for programmable clamp diode settings:

```
set_io – -clamp_diode <value>
```

value can be set as listed in Table 10, page 17.

Note: The clamp diode is always on for HSUL18I, HSUL18II, SLVSE15, MIPI12, PCI, SLVS33, HCSL33, MIPIE33, LVPECL33, LVPECL25, LVPECLE33, LVDS25, LVDS33, RSDS25, RSDS33, MINILVDS25, MINILVDS33, SUBLVDS25, SUBLVDS33, PPDS25, PPDS33, MLVDSE25, and BUSLVDSE25 I/O standards implemented in GPIO bank.

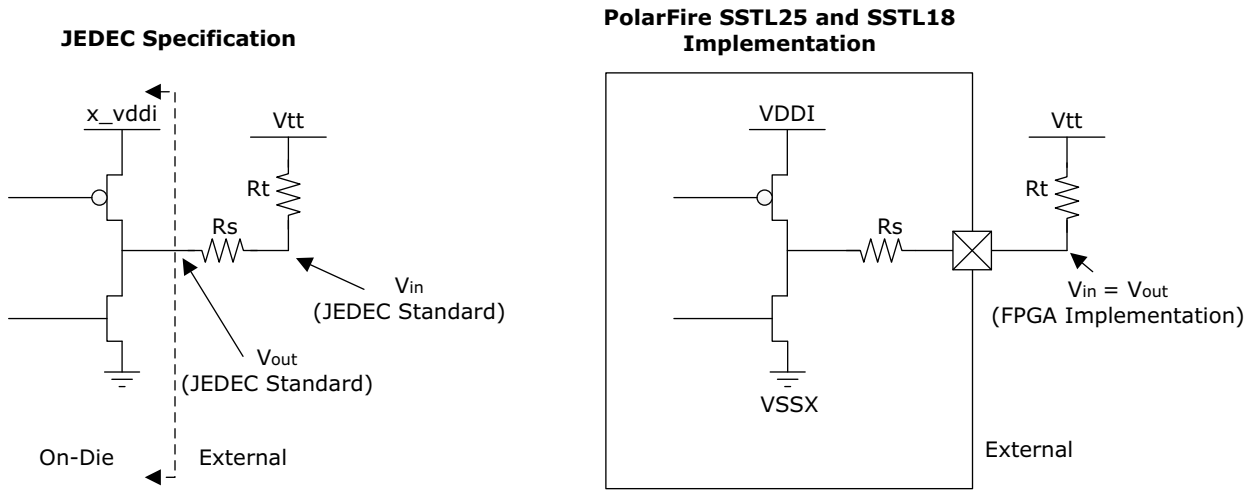
3.1.1.8 PVT-Compensated Drive Impedance and Terminations

Resistors are used to match the impedance of the trace. However, adding resistors close to device pins increases the size of the board area and component count, and can in some cases be physically impossible. To address these issues, PolarFire devices have a PVT controller between the power supply and pad signal to control the source and sink drivers between the pad and the ground. This compensation happens at power up, and on-demand by the user logic. A future version of the user guide will have a detailed user model.

3.1.1.9 SSTL25 and SSTL18 Stub Resistor

For stub-series interface standard SSTL, the output drive also includes the stub resistor. PolarFire FPGA I/Os support this stub resistor for SSTL25 and SSTL18 I/O standards (Figure 6, page 18). This feature reduces both cost and board complexity.

Figure 6 • **SSTL25 and SSTL18 Stub Resistor**



3.2 I/O Implementation Considerations

This section provides the generic guidelines when implementing various I/O standards using PolarFire devices. In addition, it also provides details of I/O states during various device operational modes such as power-up and initialization.

3.2.1 Reference Voltage for I/O Bank

Each voltage-referenced I/O standard needs a reference voltage (V_{REF}) for inputs while in operation. Each bank in a PolarFire device contains a single reference voltage bus, which can either be externally supplied through an I/O in the bank or generated internally by the bank controller.

3.2.1.1 External V_{REF} Input

Any PolarFire FPGA GPIO or HSIO pad on the device can be programmed to act as an external V_{REF} input to supply all inputs within a bank. When an I/O pad is configured as a voltage reference, all I/O buffer modes and terminations on that pad are disabled. For more information about external reference inputs, see *UG0726: PolarFire FPGA Board Design User Guide*.

3.2.1.2 Internally-Generated V_{REF}

Every bank also has an internally-generated V_{REF} available. This internally-generated V_{REF} adds more flexibility and dynamic control. This V_{REF} can be programmed in a range between 10% and 97.5% of V_{DDI} in increments of 2.5%.

3.2.2 Flexible V_{DDI}

PolarFire FPGA inputs are designed to support flexible V_{DDI} assignment for certain I/O standards, allowing I/O using compatible standards to be placed in the same I/O bank. For example, when VDDI is set to 3.3 V, the input user I/O set to 3.3 V. 2.5 V. 1.8 V and 1.2 V LVCMOS can all be placed in the same I/O bank.

The following table shows V_{DDI} and LVTTL/LVCMOS compatibility for GPIO.

Table 11 • **GPIO LVTTL/LVCMOS I/O Compatibility in Receive Mode**

V_{DDI}	LVTTL/LVCMOS33	LVCMOS25	LVCMOS18	LVCMOS15	LVCMOS12
3.3 V	Yes	Yes	Yes	No	Yes
2.5 V	Yes	Yes	Yes	Yes	Yes
1.8 V	Yes	Yes	Yes	Yes	Yes
1.5 V	Yes	Yes	Yes	Yes	Yes
1.2 V	Yes	Yes	No	Yes	Yes

The following table shows V_{DDI} and LVTTL/LVCMOS compatibility for HSIO.

Table 12 • **HSIO LVCMOS I/O Compatibility in Receive Mode**

V_{DDI}	LVCMOS18	LVCMOS15	LVCMOS12
1.8 V	Yes	Yes	Yes
1.5 V	Yes	Yes	Yes
1.2 V	No	Yes	Yes

The following table lists GPIO mixed reference receiver mode data.

Table 13 • **GPIO Mixed Reference Receiver Mode**

V_{DDI}	V_{DDAUX}	SSTL25	SSTL18, HSUL18	SSTL15, HSTL15
3.3 V	3.3 V	Yes (mid-range Vcm)	Yes (mid-range Vcm)	Yes (Low-range Vcm)
2.5 V	2.5 V	Yes (mid-range Vcm)	Yes (mid-range Vcm)	Yes (Low-range Vcm)
1.8 V	2.5 V	Yes (mid-range Vcm and clamp diode off)	Yes (mid-range Vcm)	Yes (Low-range Vcm)
1.5 V	2.5 V	Yes (mid-range Vcm and clamp diode off)	Yes (mid-range Vcm and clamp diode off)	Yes (Low-range Vcm)
1.2 V	2.5 V	Yes (mid-range Vcm and clamp diode off)	Yes (mid-range Vcm and clamp diode off)	Yes (mid-range Vcm and clamp diode off)

Table 14 • **HSIO LVCMOS I/O Compatibility in Receive Mode**

V_{DDI}	LVCMOS18	LVCMOS15	LVCMOS12
1.8 V	Yes	Yes	Yes
1.5 V	Yes	Yes	Yes

Table 14 • **HSIO LVCMOS I/O Compatibility in Receive Mode *(continued)***

V_{DDI}	LVCMOS18	LVCMOS15	LVCMOS12
1.2 V	No	Yes	Yes

Table 15 • **HSIO HSUL12/HSTL12/POD I/O Compatibility in Receive Mode**

V_{DDI}	SSTL18 HSUL18	SSTL15 HSTL15	SSTL135 HSTL135	HSUL12 HSTL12 POD	LVSTL11
1.8 V	Yes (mid-range Vcm)	Yes (mid-range Vcm)	Yes (mid-range Vcm)	Yes (mid-range Vcm)	Yes (low-range Vcm)
1.5 V	No	Yes (mid-range Vcm)	Yes (mid-range Vcm)	Yes (mid-range Vcm)	Yes (low-range Vcm)
1.2 V	No	No	Yes (mid-range Vcm)	Yes (mid-range Vcm)	Yes (low-range Vcm)

The HSIO receivers have a reduced set of compatible I/O standards when the I/O clamp-diode is set to on. For GPIO, if the signaling levels of the receiver are greater than the V_{DDI} of the bank, the clamp must be set to off.

3.2.3 I/O External Termination

If ODT is not used or not available, PolarFire FPGA I/Os require an external termination for better signal integrity. Voltage-referenced standards generally have serial (driver) and parallel (receiver) termination schemes while differential standards only require parallel (receiver) termination.

The following table lists the external termination schemes for the supported I/O standards when the ODT/driver impedance calibration feature is not used. The content will be added for PolarFire FPGA I/O termination in future releases.

Table 16 • **I/O External Termination with ODT Off**

I/O Standards	External Termination Schemes
SSTL15, SSTL18, SSTL2 single-ended	Single-ended SSTL I/O standard termination
HSTL15	Single-ended HSTL I/O standard termination
SSTL15, SSTL18, SSTL2 differential	Differential SSTL I/O standard termination
HSTL15	Differential HSTL I/O standard termination
LVCMOS12, LVCMOS15, LVCMOS18, LVCMOS25	No external termination required
LVDS	100 Ω, parallel termination (HSIO only)
MLVDS	100 Ω, parallel termination (HSIO only)
BLVDS	100 Ω, parallel termination (HSIO only)
RLVDS	100 Ω, parallel termination (HSIO only)
Mini-LVDS	100 Ω, parallel termination (HSIO only)
LVPECL	100 Ω, parallel termination (HSIO only)

3.2.4 Implementing Emulated Standards for Outputs

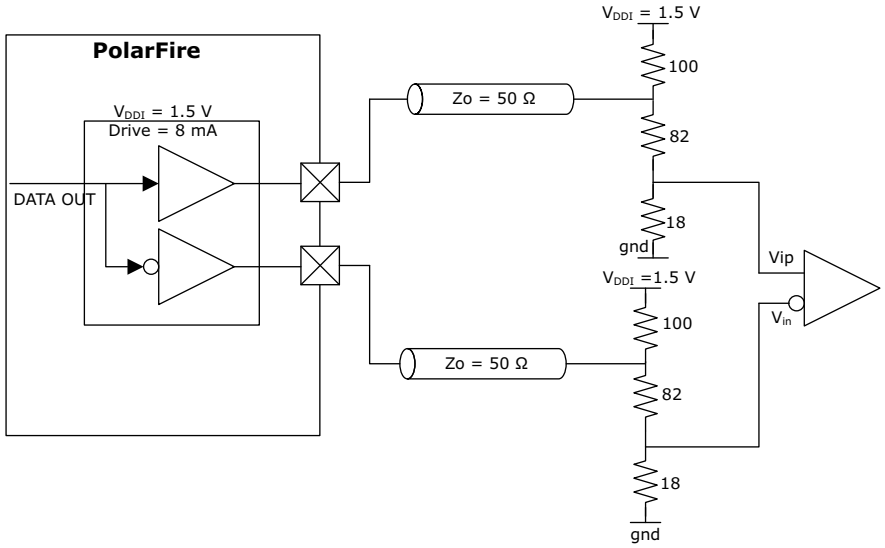
PolarFire devices require external terminations to implement SLVSE, BLVDSE, MLVDSE, and LVPECLE output modes. These outputs, referred to as emulated differential outputs, are noted in Table 5, page 13.

Emulated differential standards use PVT-compensated push-pull drivers in complementary output mode, and require external terminations on the board. This section provides example implementations for the emulated standards.

3.2.4.1 Scalable Low-Voltage Signaling Emulated (SLVSE15) Output Mode

PolarFire FPGA GPIO and HSIO support SLVS transmitter with external terminations. The following illustration shows an example of SLVSE implementation. This implementation requires 100 Ω, 82 Ω, and 18 Ω external termination. Additionally, all driver output levels in the implementation are level-shifted by approximately 18%.

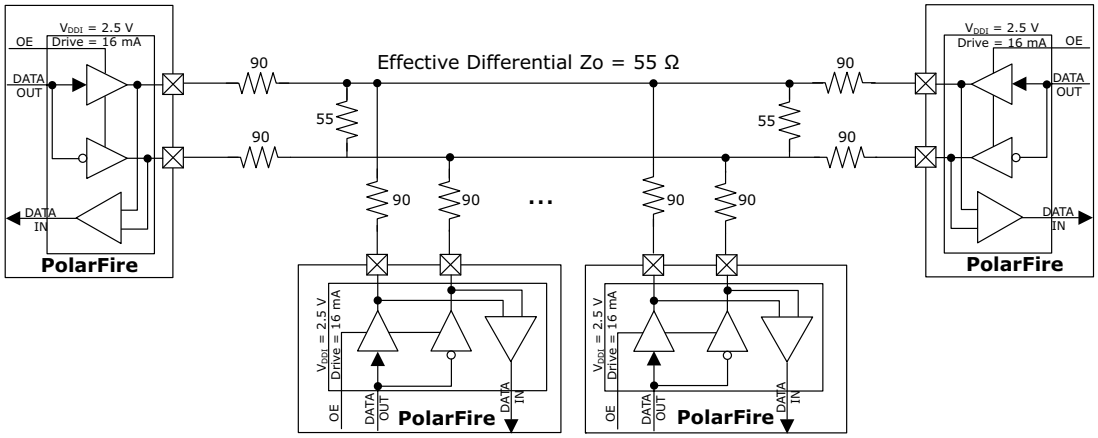
Figure 7 • **SLVSE System Diagram**



3.2.4.2 Bus-LVDS Emulated (BLVDSE25) Output Mode

BLVDS is used in multipoint, bidirectional, and heavily-loaded backplane applications. The effective impedance of these systems is lower than a typical pair of PCB traces due to the backplane capacitance, the connectors on the backplane, and the line stubs. The following illustration shows an example of PolarFire FPGA BLVDS implementation using 90 Ω stub resistors at every drop and 55 Ω stub resistors on either side of the bus. The termination values at the ends of the bus, which can range anywhere between 45 Ω and 90 Ω, must be optimized to match the effective differential impedance of the bus. In this example, the two parallel 55 Ω stub resistors yield an effective 27 Ω differential termination.

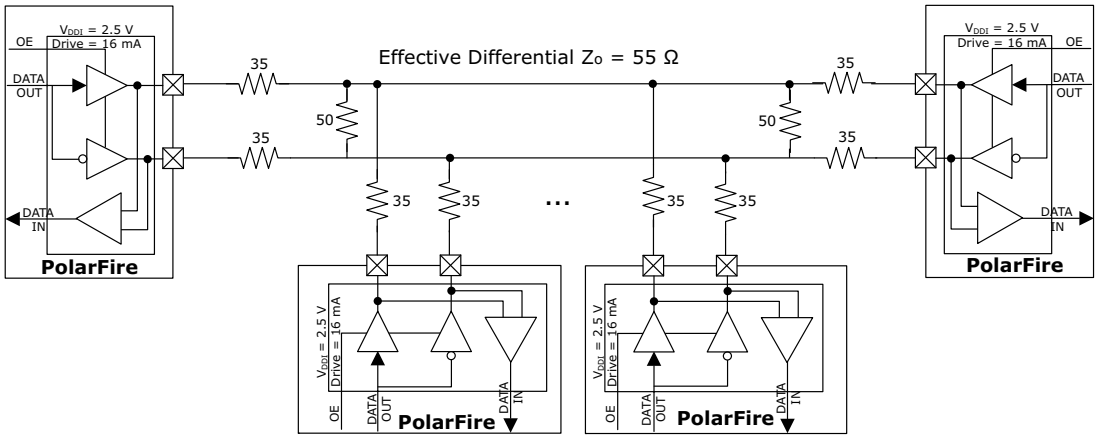
Figure 8 • **Bus-LVDSE System Diagram**



3.2.4.3 Multipoint Low-Voltage Emulated (MLVDSE25) Output Mode

MLVDS has larger signaling amplitude when compared to BLVDS, and therefore, requires more drive current. Similar to BLVDS, the effective impedance of these systems is lower than a typical pair of PCB traces due to backplane capacitance, the connectors on the backplane, and the line stubs. The following illustration shows an example implementation using 35 Ω stub resistors at every drop and 50 Ω stub resistors on either side of the bus. The termination values at the ends of the bus, which can range anywhere between 50 Ω and 70 Ω, must be optimized to match the effective differential impedance of the bus.

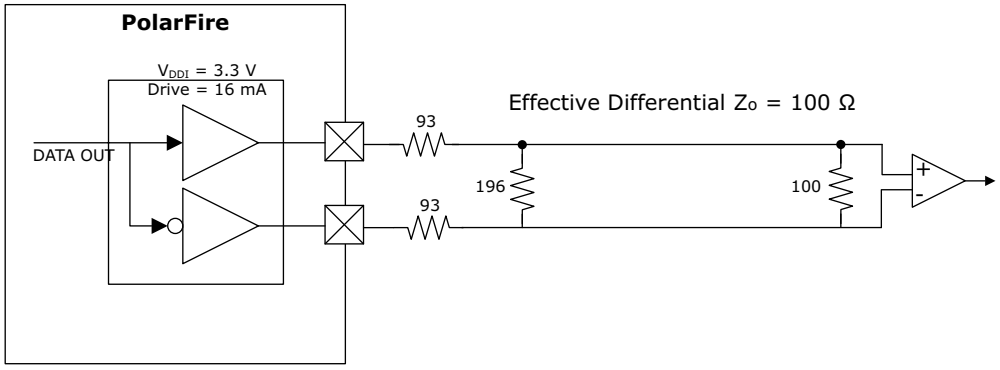
Figure 9 • **MLVDSE System Diagram**



3.2.4.4 LVPECL Emulated (LVPECLE33) Output Mode

LVPECL is derived from ECL and PECL and uses 3.3 V supply voltage. The following illustration shows an example of PolarFire FPGA implementation using 93 Ω stub resistors with a 196 Ω parallel/differential termination at the driver and a 100 Ω differential termination at the receiver. The termination values at the driver should be optimized to match the effective differential impedance of the bus. In this example, the effective parallel differential termination at the receiver is around 66 Ω. However, the series 93 Ω resistors are always seen by the driver yielding an effective differential impedance of 252 Ω. The receivers see an attenuated signal.

Figure 10 • **LVPECLE System Diagram**



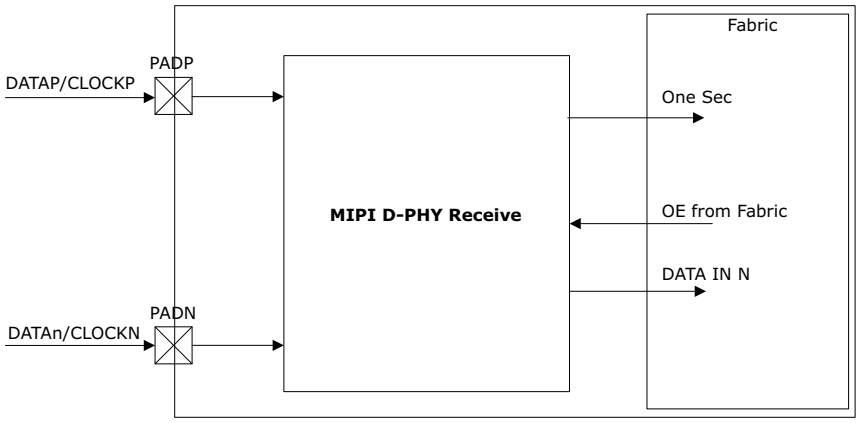
3.2.5 Implementing MIPI D-PHY

PolarFire devices support implementation of the MIPI D-PHY standard used in camera and display applications.

3.2.5.1 MIPI D-PHY Receive Interface

PolarFire FPGA GPIO supports unidirectional MIPI D-PHY I/O in the receive direction, as shown in the following illustration. The MIPI D-PHY I/O receiver interface supports both high-speed (HS) and low-power (LP) receiver modes, which can be selected from the fabric.

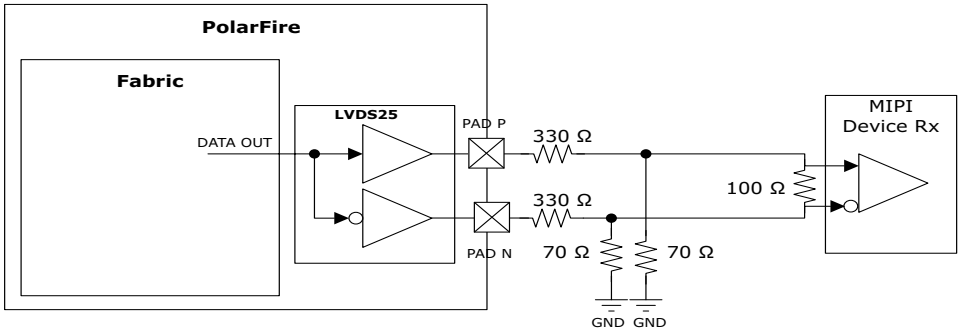
Figure 11 • **MIPI D-PHY I/O Receive Interface Diagram (High-Speed and Low-Power Modes)**



3.2.5.2 MIPI D-PHY Transmitting Interface (High-speed Only)

PolarFire FPGA GPIO supports unidirectional MIPI D-PHY transmit interface with the external resistors, as shown in the following illustration. Every GPIO P and N pair can be configured as a MIPI D-PHY transmit interface.

Figure 12 • **MIPI D-PHY Transmit Interface (High-speed only)**

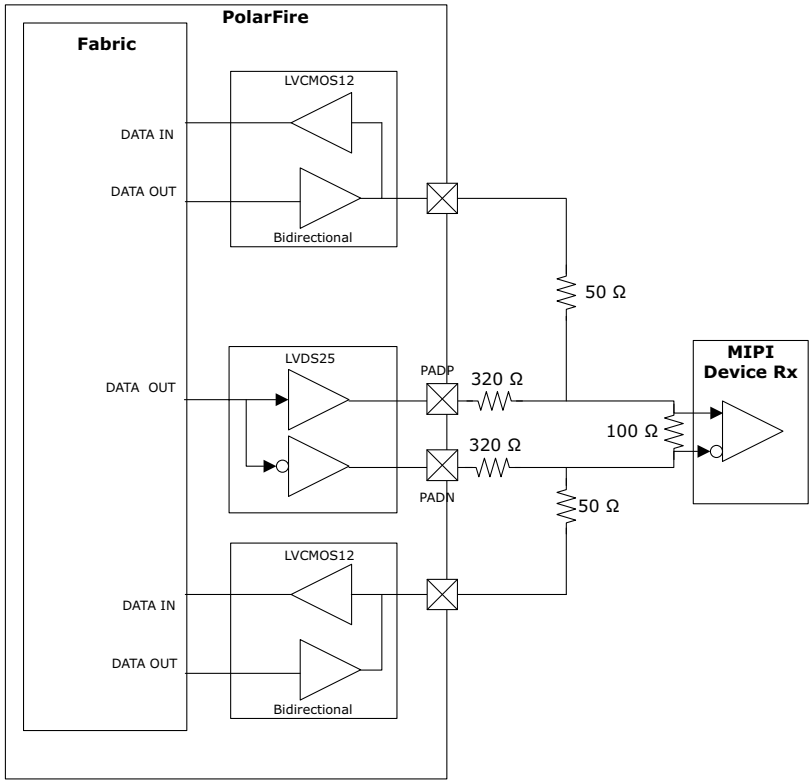


Note: The external resistor values are determined after the final board characterization.

3.2.5.3 MIPI D-PHY Transmit Interface (High-speed Only) with Bidirectional Low-Power Mode

PolarFire FPGA GPIO also supports a bidirectional MIPI D-PHY lane with external resistors, as shown in the following illustration. Microsemi provides a macro that can be instantiated in the user design to implement the MIPI transmit interface (high-speed only) with bidirectional low-power mode, see PolarFire FPGA Generic I/O Interfaces, page 36 for more information.

Figure 13 • **MIPIE High-Speed Transmit with Bidirectional Low-Power Mode**



Note: The external resistor values are determined after the final board characterization.

3.2.6 I/O States During Various Operational Modes

The state of an I/O at any given point in time depends on the operational mode of the device at that point. This section describes the I/O state during various operational modes so that users can design their boards accordingly.

3.2.6.1 Power-Up and Initialization

The following table lists the I/O states during power-up and initialization modes.

Table 17 • **I/O States during Power-Up and Initialization**

Device State	I/O State
Power-up start/powering up	Tri-state I/O buffers are disabled. Output drivers are disabled (tri-stated). Receivers are disabled (input signals are not passed to the FPGA fabric). All terminations, PCI clamp diodes, and weak pull-up/down modes are off. All I/O bank power detectors and PVT controllers are disabled.
User mode	The buffer is programmed based on Libero I/O settings. Data and output enable signals are based on user settings.

For more information about I/O states, see *UG0714: PolarFire FPGA Programming User Guide*.

3.2.6.2 Flash*Freeze Mode

For more information about PolarFire FPGA I/O states during Flash*Freeze mode, see *UG0725: PolarFire FPGA Device Power-Up and Resets User Guide*.

3.2.6.3 Device Programming Modes

The following table lists PolarFire FPGA user I/O states during various programming modes. For more information about programming modes, see *UG0714: PolarFire FPGA Programming User Guide*.

Table 18 • **GPIO and HSIO States During Programming Modes**

Programming Modes	I/O States
JTAG	Set during JTAG programming in Libero SoC
SPI slave programming	Tri-state with weak pull-up/pull-down
IAP	Tri-state with weak pull-up/pull-down
Auto-programming	Tri-state
IAP recovery	Tri-state with weak pull-up/pull-down

3.2.7 Cold Sparing and Hot Socketing

This section describes cold sparing and hot socketing capabilities of PolarFire FPGA user I/Os.

3.2.7.1 Cold Sparing

In cold-sparing applications, voltage can be applied to device I/Os before and during power-up. For cold-sparing applications, the device must support the following characteristics:

- I/Os must be tri-stated before and during power-up
- Voltage applied to an I/O must not power up any part of the device
- Device reliability must not be compromised if voltage is applied to I/Os before or during power-up

Cold Sparing is supported by both GPIO and HSIO—any I/O of an unpowered PolarFire FPGA can be safely driven with very minimal leakage current. When the device is powered off, both V_{DD} and the V_{DDI} are clamped to ground, preventing these supplies from powering up when a voltage is applied to the inputs. It is a good design practice to not rely on the outputs of an unpowered or partially powered PolarFire device to drive other components in the system. Note that both GPIO and HSIO support the cold sparing feature.

3.2.7.1.1 Hot Socketing

Hot socketing allows a voltage to be applied to the inputs of PolarFire devices before power is present on the V_{DDI} pins. PolarFire FPGA GPIO supports hot socketing, but HSIO does not support hot socketing.

When the FPGA is not powered, GPIO is in a high-impedance state (hi-Z), also known as disabled state. For GPIO configured for I/O standards requiring a $V_{REF,}$ the amount of current flowing into or out should be minimized for the GPIO pin so that the external V_{REF} signal is not affected.

4 IOD Features and User Modes

Each PolarFire FPGA I/O (both GPIO and HSIO) has a digital block, called IOD, that interfaces with the FPGA fabric on one side and the IOA buffers on the other (Figure 14, page 28). The IOD block includes several digital features, including I/O digital. The I/O digital allows for easy data transfer between the high-speed IOA buffers and the lower-speed FPGA core.

The IOD block can be configured for both input and output SDR and DDR modes. It also allows the gearing-up of the output data rate and gearing-down of the input data rate. These options are configured in Libero SoC PolarFire.

This chapter provides information about the IOD block and the various I/O user modes, including various SDR, DDR, and digital modes.

4.1 IOD Block Features

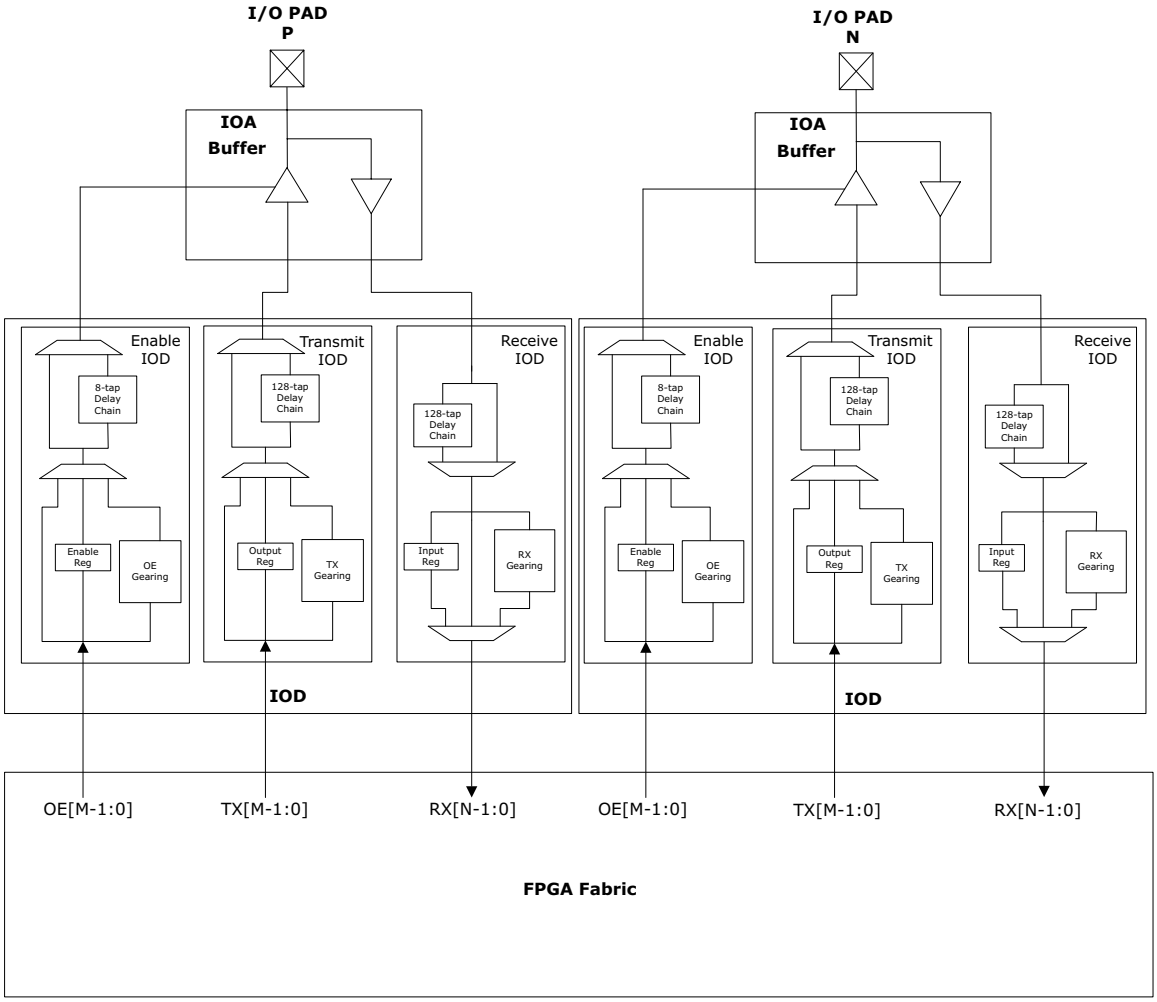
- Programmable input and/or output delay chain
- I/O register for data-in, data-out, and output enable signals
- Autonomous data eye monitor to optimize the relationship between the clock and data at the input
- Up to 1:10 input deserialization (input digital)
- Up to 10:1 output serialization (output digital)
- Support for DDR and SDR interfaces
- Word alignment with a slip control
- High-speed and low-skew I/O clock networks
- Clock recovery for serial protocols and other similar interfaces
- Low-power mode support to latch state of input or output data

4.2 IOD Block Overview

The IOD block includes the input and output delay functions, I/O registers, and digital logic blocks. The digital logic blocks are receive digital (Rx digital) for input, transmit digital (Tx digital) for output, and enable digital (OE digital) for the enable signals. The IOD block also includes several high-speed, low-skew clock networks. Figure 14, page 28 shows an overview of the IOD block. Various I/O features are set mainly by the protocol configurator or the Libero configurator within Libero SoC PolarFire. However, some of the I/O features such as I/O register and programmable delay can be controlled automatically or manually by Libero SoC PolarFire.

The following illustration shows an overview of the IOD block.

Figure 14 • **IOD Configured for I/O Registers**



Note: The values of M and N depend on the digital ratio.

The following sections describe functions of the IOD subblocks:

4.2.1 Programmable I/O Delay

The IOD block includes programmable delay chains for both input and output data paths. These programmable delay chains on input or output data paths allow 128-tap delay of either 25 ps or 50 ps clock skew. The enable path also includes a 8-tap programmable delay chain. These programmable delays are PVT compensated. The programmable delay chain can be set statically by using the I/O attribute editor or by using a PDC command in Libero SoC PolarFire. The delay value corresponding to each tap setting and the PDC command details will be available in a future version of this user guide.

The programmable delay chain is used to:

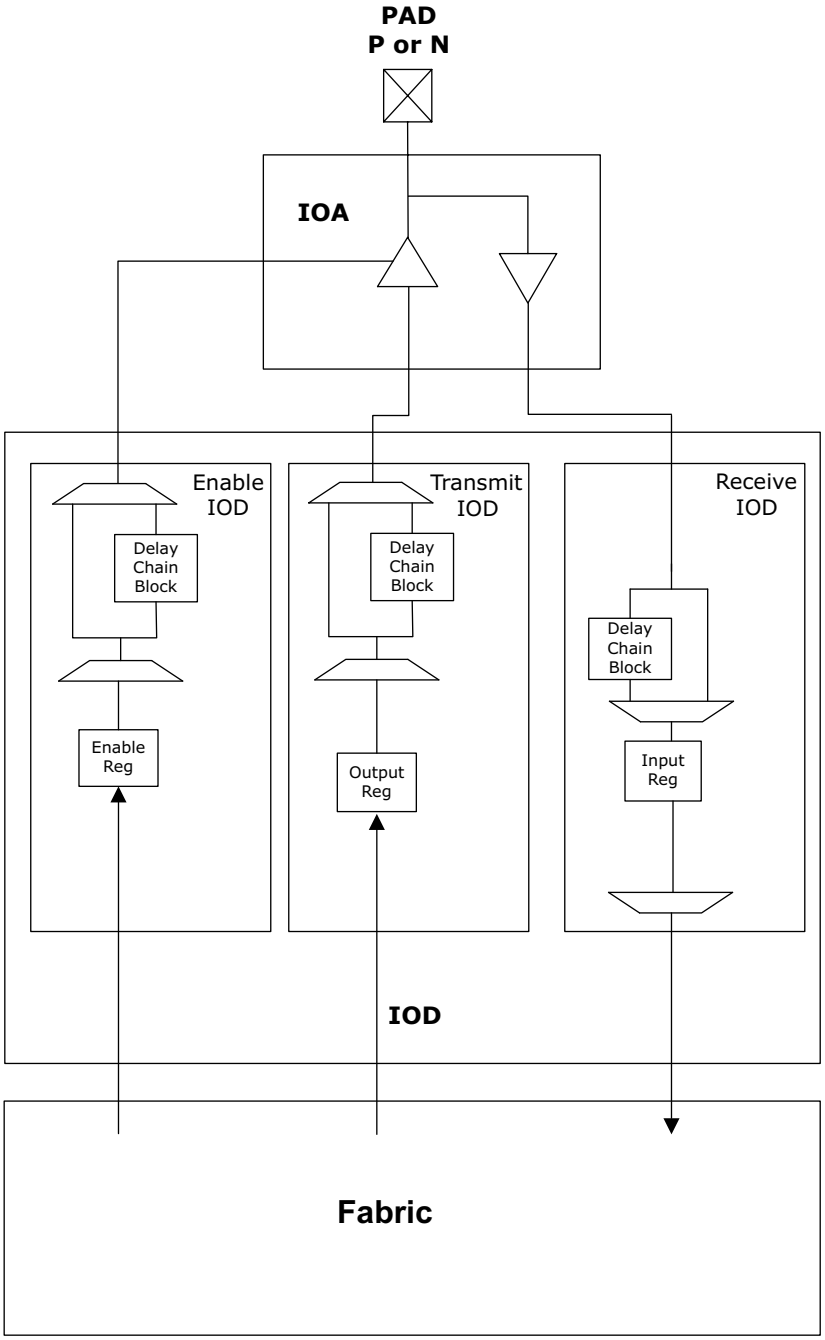
- Ensure zero hold time for the input registers
- Cancel the skew between the input data path and clock injection path
- Spread out I/O buffer timing along with an edge of the device for SSO noise control

The programmable delay chain can also be controlled via dynamic control signals from the FPGA fabric. Dynamic delay control is useful for high-speed interfaces that require per-bit training.The dynamic control is only available for certain PolarFire I/O interfaces, see PolarFire FPGA Generic I/O Interfaces, page 36 for more information.

4.2.2 I/O Registers

The IOD block includes registers for data-in, data-out, and output enable signals. The input registers (IOINFF) provide the registered version of the input signals from the IOA to the FPGA fabric. The output registers (IOUTFF) provide the registered version of the output signals from the FPGA fabric to the IOA. The output enable register (IOENFF) acts as a control signal for the output if the I/O is configured as tri-stated or bidirectional. Figure 15, page 29 shows the I/O registers. These registers in IOD blocks are similar to the D-type flip-flops available in fabric logic elements.

Figure 15 • **I/O Registers in IOD**



The I/O register is used for:

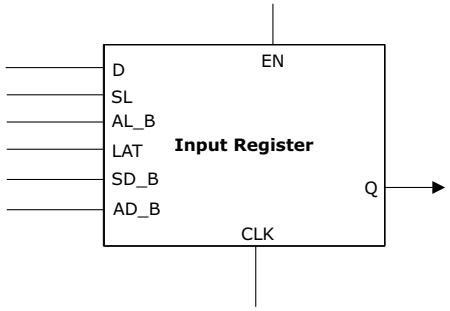
- Better I/O interface performance, as the registers are placed close to the I/O pads.
- Synchronizing the transmit and receive bus signals. For example, the I/O registers ensure that all the bits of the bus are synchronized to the clock signal when they are transmitted or received.

The I/O registers are used by default during place-and-route if the register can be mapped to the I/O register. The Libero I/O attribute editor or a PDC command can also control this behavior. A future version of the user guide will have the PDC command that controls the I/O register uses.

4.2.2.1 Input Register

The following illustration shows the input register.

Figure 16 • **Input Register**



The following table lists the input register pins and descriptions.

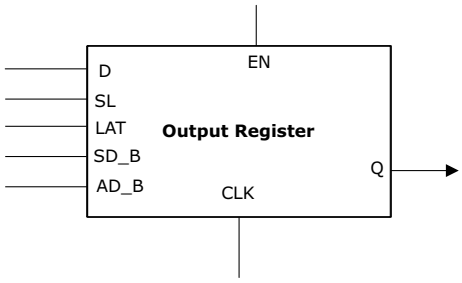
Table 19 • **I/O Input Register Ports**

Ports	Types	Descriptions
CLK	Input	Clock input
D	Input	Data input
Q	Output	Data output
EN	Input	Clock enable (active high)
SL	Input	Synchronous load (active high)
AL_B	Input	Active low asynchronous load (active low)
LAT	Input	Latch enable (active high)
SD_B	Input	Synchronous data
AD_B	Input	Asynchronous data (active low)

4.2.2.2 Output Register

The following illustration shows the output register.

Figure 17 • **Output Register**



The following table lists the output register pins and their descriptions.

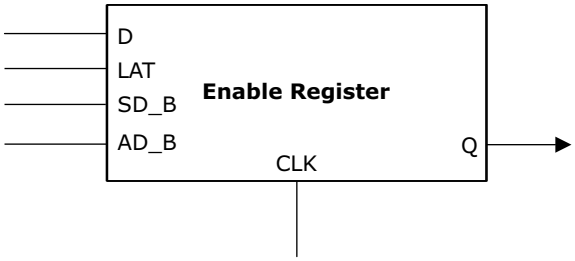
Table 20 • **I/O Output Register Ports**

Ports	Types	Descriptions
CLK	Input	Clock input
D	Input	Data input
Q	Output	Data output
EN	Input	Clock enable (active high)
SL	Input	Synchronous load (active high)
LAT	Input	Latch enable (active high)
SD_B	Input	Synchronous data
AD_B	Input	Asynchronous data (active low)

4.2.2.3 Enable Register

The following illustration shows enable register.

Figure 18 • **Enable Register**



The following table lists the enable register pins and their descriptions.

Table 21 • **I/O Register Ports**

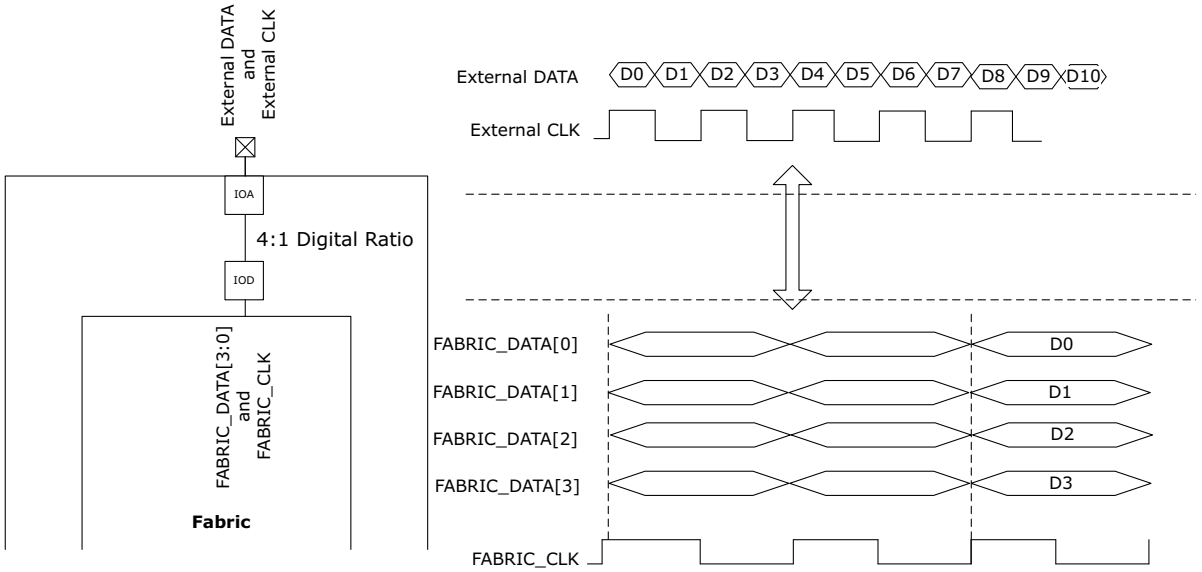
Ports	Types	Descriptions
CLK	Input	Clock input
D	Input	Data input
Q	Output	Data output
LAT	Input	Latch enable (active high)
SD_B	Input	Synchronous data
AD_B	Input	Asynchronous data (active Low)

4.2.3 I/O Digital

I/O digital handles serial-to-parallel and parallel-to-serial conversion of multiple FPGA fabric signals to and from a single device I/O based on user clock settings, as shown in the following illustration. The gearbox either deserializes and transfers input data to a lower core clock speed, or transfers lower-speed data from the fabric to the high-speed output clock domain, and serializes it in the process. Libero SoC PolarFire automatically configures these gearboxes based on the application settings.

The following illustration shows the I/O digital example, where high speed serial data is passed from I/O to fabric via four signals at lower speed.

Figure 19 • **I/O Digital**



4.2.4 Data Eye Monitor

The IOD block has a data eye monitor to optimize the relationship between the clock and input data so that the data can be easily captured at the center. The data eye monitor includes positive edge-triggered and negative edge-triggered register sets to analyze the data-to-clock relationship. The input signal is passed to these registers with the user configurable delay settings. By changing the delay settings and monitoring the register sets, the data edges can be checked if they are closer to the clock edges than the required settings or not. The data eye monitor is used for various protocols such as DDR, QDR, RLDRAM, SPI4.2, and MIPI; it is unavailable for a generic use model. When protocol configurators are generated, they add a training soft IP core that uses a data eye monitor to optimize the data edge with respect to the clock.

The data eye monitor is capable of operating in SDR mode or DDR mode. In DDR mode, it monitors data edges relative to both phases of the clock. In SDR mode, it monitors relative to only one edge of the clock.

4.2.5 I/O FIFO

The IOD block contains an I/O FIFO for clock domain transfers. In DDR applications, the I/O FIFO is used for high-speed transfer data from the external DQS domain to the internal data clock domain. Libero SoC PolarFire and Microsemi memory controller cores configure the I/O FIFO based on the application settings.

4.3 PolarFire FPGA I/O Lanes

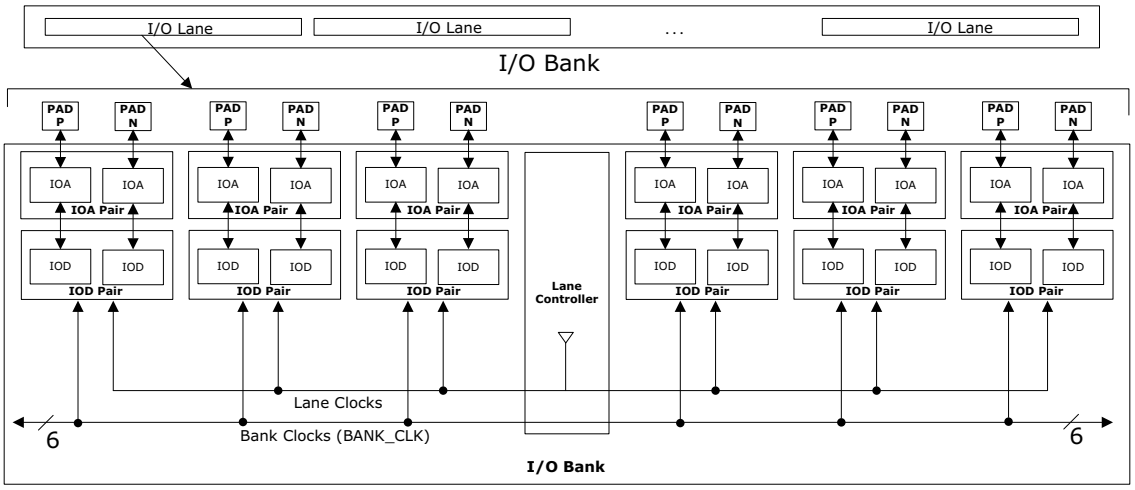
To support memory interfaces, PolarFire FPGA I/O pairs are grouped into lanes, with multiple lanes per bank. Each lane consists of twelve I/Os (six I/O pairs), a lane controller, and a set of high-speed, low-skew clock resources. The uppermost lane on the western side of devices has less than six I/O pairs in each lane. The high-speed and low-skew clock resources in the I/O lane include a global clock network, regional clock networks, bank clock networks, and lane controller clock networks, see I/O Clock Networks, page 33 for more information.

The PolarFire FPGA I/O lane is used for easy implementation of integrated PHY for memory. For example, a 32-bit SDRAM interface requires four I/O data lanes. Each data lane uses one PolarFire

FPGA I/O lane—two I/O pads are used for DQS, eight I/O pads are used for DQ bits, one pad is used for data mask (DM), and one I/O pad is used as a spare.

The following illustration shows the PolarFire FPGA I/O lanes diagram.

Figure 20 • **PolarFire FPGA I/O Lanes**



4.3.1 Lane Controller

The lane controller handles the complex operations necessary for the high-speed interfaces, such as DDR memory interfaces and CDR interfaces. To bridge the lane clock to the bank clock, the lane controller is used to control an I/O FIFO in each IOD. This I/O FIFO interfaces with DDR memory by utilizing the DQS strobe on the lane clock. The lane controller can also delay the lane clock using a PVT-calculated delay code from the DLL to provide a 90° shift. Certain I/O interfaces require a lane controller to handle the clock-domain that results with higher gear ratios. For more information, see PolarFire FPGA Generic I/O Interfaces, page 36.

The lane controller also provides the functionality for the IOD CDR. Using the four phases from the CCC PLL, the lane controller creates eight phases and selects the proper phase for the current input condition with the input data, see PF_IOD_CDR, page 42 for more information. A divided-down version of the recovered clock is provided to the fabric (DIVCLK).

4.4 I/O Clock Networks

Each PolarFire FPGA I/O contains a fabric, a bank clock resource, and a lane controller clock resource for efficient clock distribution. All of these four clock networks can be used to interface with the IOD block. For more information about global clock network, see *UG0684: PolarFire FPGA Clocking Resources User Guide*.

4.4.1 Global Clock Resource

Each IOD has two global clock inputs from the fabric: one for the receive block (Receive IOD) and the other for the transmit block (Transmit IOD and Enable IOD). Libero SoC PolarFire automatically routes the clock signals through the global clock network and connects to the two global clock inputs of the IOD block, if they are driven from the specified resources. The global clock network can be driven by any of the following:

- Preferred clock inputs (CLKIN_z_w)
- Oscillator clocks
- CCC (PLL/DLL)
- Fabric routing
- Clock dividers
- NGMUXs
- Transceiver reference clock inputs

For more information about global clock architecture, see *UG0684: PolarFire FPGA Clocking Resources User Guide*.

4.4.2 Regional Clock Networks

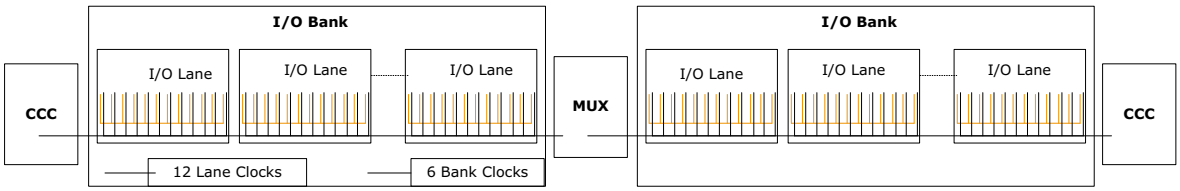
The regional clock networks are low-latency networks; they can only distribute clocks to a certain area of the device with low skew; they can be driven from the divided CDR clock, the divided bank clock, and the IOA placed closer to the DQS function. PolarFire FPGAs offer one regional clock buffer per I/O lane on the northern, southern, and western edges. Note that the size of the region depends on the regional clock buffer location and does not overlap. For more information about regional lock buffer location, see *UG0684: PolarFire FPGA Clocking Resources User Guide*.

The PolarFire FPGA I/O interface modes (DDRxn) are designed to use the regional clocks. Libero SoC PolarFire automatically routes the clocks coming from I/O lanes on the regional clock networks.

4.4.3 Lane Clock Resources

Each lane has several clock networks in the I/O lane. The lane clock resources are distributed from each lane controller to each of the 12 IODs within a lane. The lane clock resource is not controllable as Libero SoC PolarFire automatically uses the lane clock resource based on the I/O configuration.

Figure 21 • **Distribution of the Lane Clock**



4.4.4 High-Speed I/O Clock Resource (HS_IO_CLK)

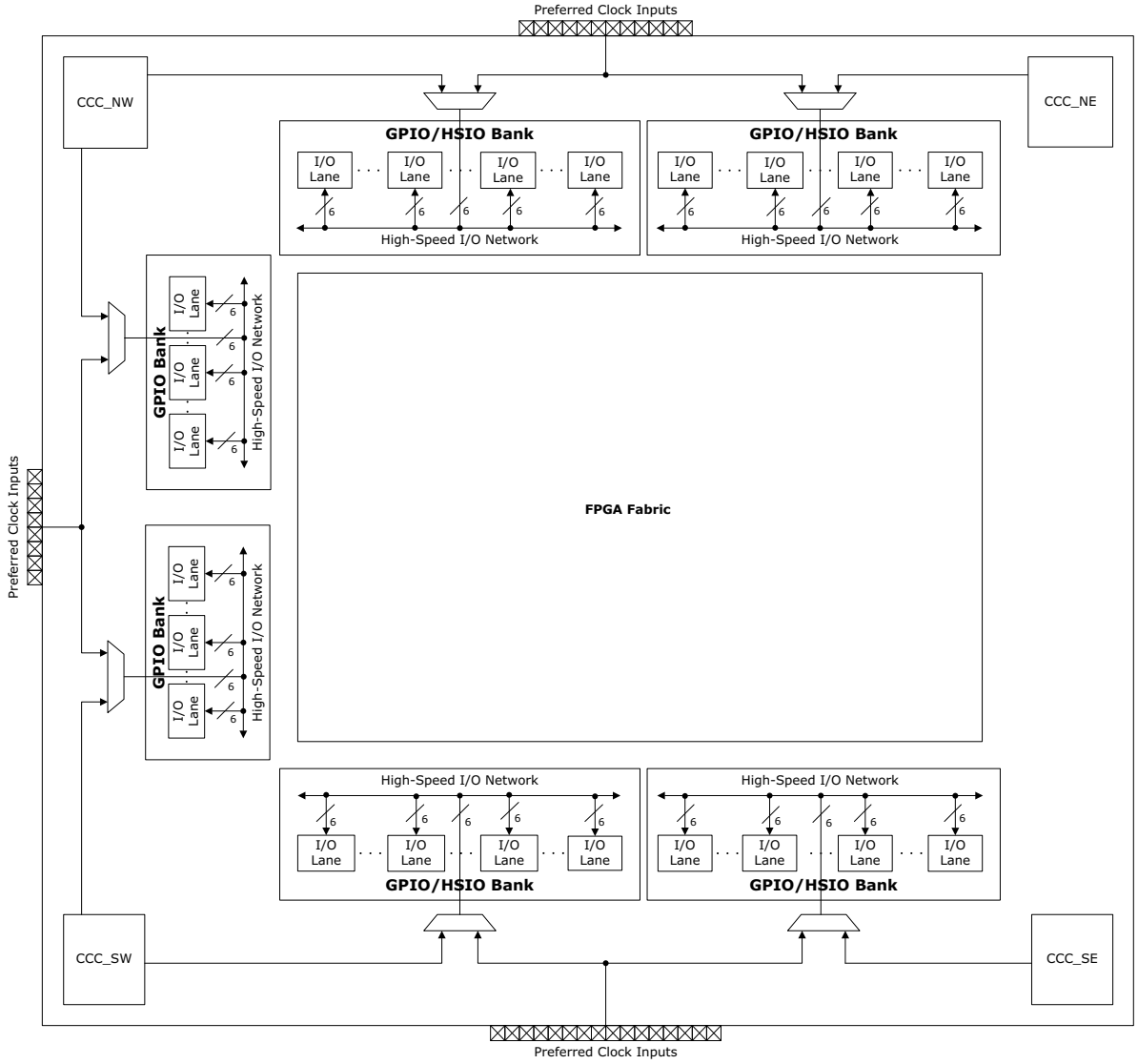
High-speed I/O clock networks are integrated into I/O banks and distribute clocks along the entire I/O bank with low-skew. They are used to clock data in and out of the I/O logic when implementing the high-speed interfaces. The high-speed I/O clock networks are located on the east corner of the FPGA fabric. Each I/O bank can have six high-speed I/O clocks. High-speed I/O clocks from adjacent banks on the same edge can be bridged to build large I/O interfaces.

High-speed I/O clock networks are driven either from I/Os or CCCs. The high-speed clocks can be configured to feed reference clock inputs of adjacent CCCs.

4.4.4.1 Bank Clock Bridging

Bank clocks can be extended beyond its bank limitation. They can be shared with other banks on the same side of the device. However, this increases the skew on the clock. To minimize the skew, assign the I/Os carefully and place the I/Os on the same bank.

Figure 22 • **Distribution of the Bank Clock**



4.5 I/O Lanes in Each Bank

The following table lists the number of I/Os and lanes in each bank for each device and package option.

Table 22 • **I/O Lanes in Each Bank**

		North Corner I/Os						South Corner I/Os				West Corner I/Os			
		Bank 0		Bank 7		Bank 1		Bank 3	Bank 2		Bank 6		Bank 4		Bank 5
Devices	**Packages**	HSIO	Lanes	HSIO	Lanes	HSIO	Lanes	JTAG	GPIO	Lanes	HSIO	Lanes	GPIO	Lanes	GPIO	Lanes
MPF100, MPF200	FCSG325	36	3	0	0	48	4	13	48	4	0	0	38	4	0	0
MPF100. MPF200, MPF300	FCSG536	60	5	0	0	60	5	13	96	8	0	0	84	7	0	0
MPF100, MPF200, MPF300	FCVG484	60	5	0	0	60	5	13	96	8	0	0	68	6	0	0
MPF100, MPF200, MPF300	FCG484	48	4	0	0	48	4	13	84	7	0	0	64	6	0	0
MPF200, MPF300, MPF500	FCG784	72	6	24	2	60	5	13	96	8	0	0	92	8	44	4
MPF300, MPF500	FCG1152	72	6	96	8	60	5	13	96	8	96	8	92	8	72	6

4.6 PolarFire FPGA Generic I/O Interfaces

The PolarFire FPGA I/O supports a number of interface modes that can be selected to build the required data interface. The following table lists the supported I/O interfaces.

Table 23 • **PolarFire FPGA Generic I/O Interfaces**

Interface Name	Topology	Clock to Data	Clock Source
RX_SDR_G	Rx SDR	Aligned	Global
RX_SDR_R		Aligned	Regional
RX_SDR_G_DLL		Aligned with DLL	Global
RX_SDR_R_DLL		Aligned with DLL	Regional
RX_DDR_G_A	Rx DDR	Aligned with DLL delay code	Global
RX_DDR_R_A		Aligned with DLL delay code	Regional
RX_DDR_L_A		Aligned with DLL delay code	Lane
RX_DDR_G_C		Centered	Global
RX_DDR_R_C		Centered	Regional
RX_DDR_L_C		Centered	Lane
RX_DDR_B_A		Aligned with DLL delay code	Bank
RX_DDR_B_C		Centered	Bank

Table 23 • **PolarFire FPGA Generic I/O Interfaces *(continued)***

Interface Name	Topology	Clock to Data	Clock Source
RX_DDRX_B_A	Rx DDR digital	Aligned with DLL delay code	Bank
RX_DDRX_B_DYN		Automated dynamic clock versus data	Bank
TX_SDR_G	Tx SDR	Centered	Global
TX_SDR_R		Centered	Regional
TX_DDR_G_A	Tx DDR	Aligned with DLL delay code	Global
TX_DDR_R_A		Aligned with DLL delay code	Regional
TX_DDRX_B_A	Tx DDR digital	Aligned with DLL	Bank
TX_DDRX_B_DYN		Dynamic clock versus data	Bank

Libero SoC PolarFire supports the interface modes such as RX_DDR_L_A, TX_DDR_G_A, RX_DDRX_L_DYN, TX_DDRX_B_DYN, RX_DDRX_L_DYN_MIPI, and TX_DDRX_B_DYN_MIPI. A future version of this user guide and the Libero software will have the remaining interfaces and their uses. Note that for the simple source-synchronous RX_SDR_G, RX_SDR_R, TX_SDR_G, and TX_SDR_R interfaces, the proper I/O mode can be inferred. For the memory interfaces and the remaining IOD interfaces, the DDR memory configurator and the IOD generic interface configurator in Libero are used to produce the intended interface.

4.6.1 RX_DDR_L_A

The PolarFire FPGA IOD block can be used to implement input and output DDR registers. It supports both GPIO and HSIO banks. The RX_DDR_L_A interface is used when the DDR data and clock are aligned with the input of the PolarFire device, as shown in the following illustration.

Figure 23 • **RX_DDR_L_A Mode Data and Clock Waveform**

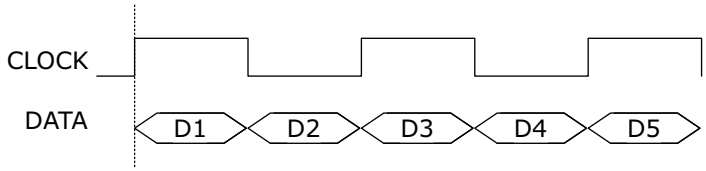
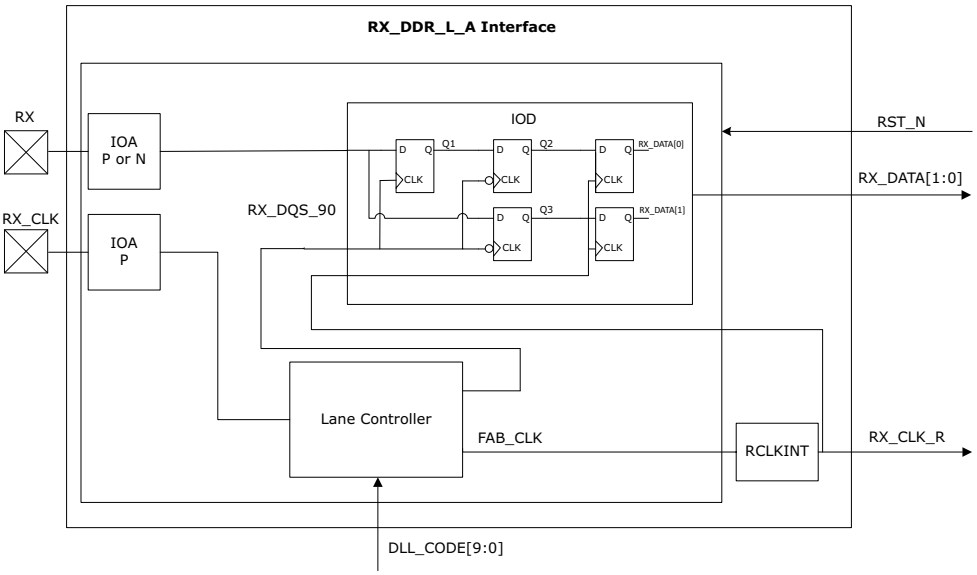


Figure 24, page 38 shows RX_DDR_L_A block diagram. The RX_DDR_L_A interface receives RX data and RX_CLK clock through I/Os and passes RX_DATA and RX_CLK_R to the fabric. The lane controller uses the receive clock input (RX_CLK) and generate two clocks:

- **RX_DQS_90—**90 degree shifted clock
- **RX_CLK_R—**receive clock for fabric interface

The input data is captured using both the rising and falling edges of RX_DQS_90 clock. The captured data is transferred from the RX_DQS_90 domain to RX_CLK_R domain in the IOD, and is then passed to the fabric interface through RX_DATA ports. The receive clock for fabric interface RX_CLK_R, is driven by RCLKINT, drives the regional clock resource.

Figure 24 • **Block Diagram of the RX_DDR_L_A Interface**



The following table lists the RX_DDR_L_A interface mode ports.

Table 24 • **RX_DDR_L_A Interface Mode Ports**

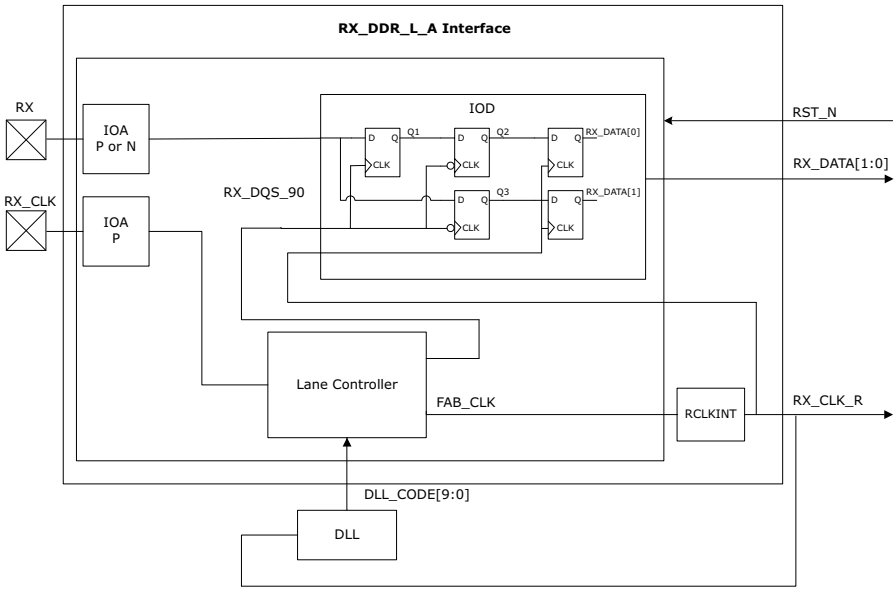
Port	I/O	Description
RX	Input	Input DDR data. Supports up to 11 bits wide and all bits must fit within a lane
RX_CLK	Input	Input DDR clock. Must use the DQS pin of the lane of the RX data
RST_N	Input	Asynchronous reset to IOD and lane controller
DLL_CODE[7:0]	Input	Delay code to generate 90 degree clock shift. This is sourced from a DLL of RX and RX_CLK
RX_DATA[1:0]	Output	DDR output to FPGA fabric RX_DATA[0]—rising edge data RX_DATA[1]—falling edge data
RX_CLK_R	Output	Receive clock to FPGA fabric using a regional clock

4.6.1.1 Generate Bank Clock DLL_CODE Using DLL Configurator

The lane controller uses the delay code (DLL_CODE) to generate a 90 degree delayed clock (RX_DQS_90) from the receive clock. The received clock (RX_CLK_R) needs to be connected to a DLL reference clock and the DLL then generates a delay code (DLL_CODE) representing a 90 degree phase shift. The lane controller also sends the received clock on a local clock region in the fabric (RX_CLK_R). This clock is delayed to compensate for the regional clock insertion delay.

The following illustration shows the RX_DDR_L_A interface with DLL.

Figure 25 • **RX_DDR_L_A Interface with DLL**

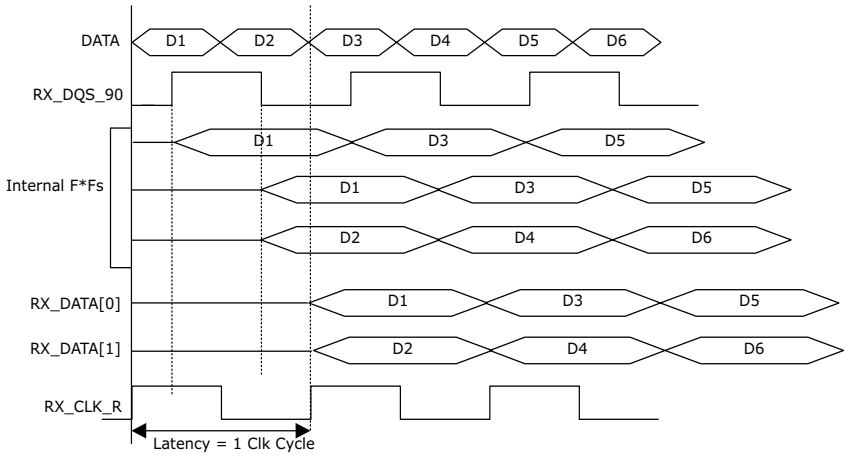


The DLL is used to produce DLL_CODE and can be shared for several RX_DDR_L_A interfaces. When sharing the same DLL, each interface must be placed within the reach of the DLL. This feature allows many RX_DDR_L_A interfaces to share the same DLL resulting in an overall lower power solution.

4.6.1.2 Interface Waveform

The following illustration shows the RX_DDR_L_A interface signal waveform.

Figure 26 • **RX_DDR_L_A Interface Signal Waveform**



4.6.1.3 Resource Utilization

Each RX_DDR_L_A interface uses the following I/Os:

- Eleven single-ended I/O and five differential I/O
- One LANECTRL
- One local clock region

One DLL can be shared with another if with the same data rate.

4.6.1.4 Pin Selection Rules

The following rules apply when assigning a pin to the RX_DDR_L_A interface:

- RX and RX_CLK I/Os must be placed in the same I/O lane. RX_CLK I/Os must be placed in the I/Os with the DQS function in the lane.
- Multiple IOD generic interfaces can share the same DLL. When sharing the same DLL, each interface must be placed within the reach of the DLL.
- Each DLL can drive IOD interfaces on the adjacent vertical and horizontal edges. For example, DLL_SW_0 can drive the IOD generic interface on the southern and western edges.

4.6.1.5 Latency

The following table shows the latency for the RX_DDR_L_A interface.

Table 25 • **Latency for the RX_DDR_L_A Interface**

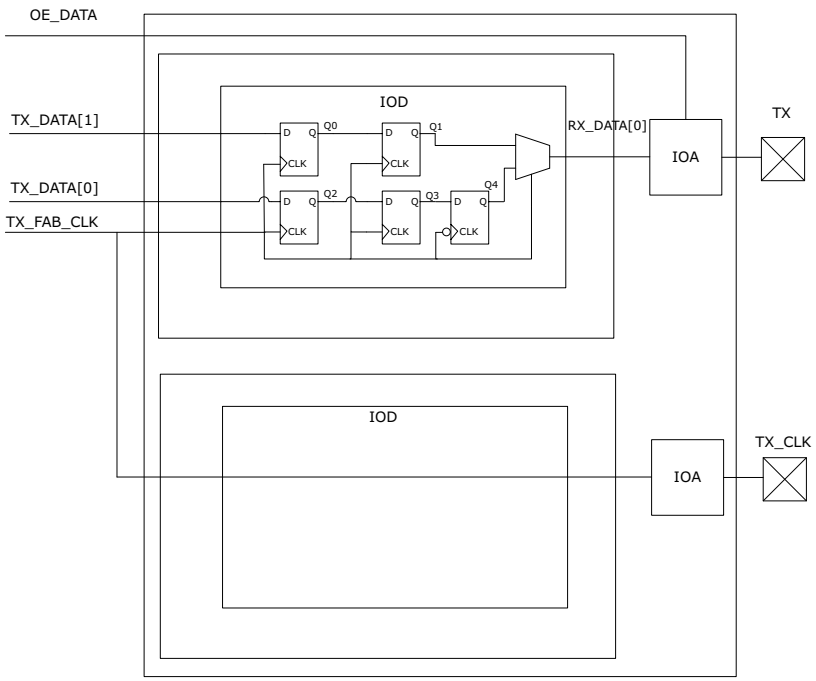
I/O Interface	Gear Ratio	Latency Cycle (RX_CLOCK cycle)
RX_DDR_L_A	1:2	1

4.6.2 TX_DDR_G_A

The TX_DDR_G_A interface implements the DDR transmit interface where clock edges are aligned with the DDR data. The IOD block uses the fabric clock (TX_FAB_CLK) to capture the transmitted data from fabric and transmit it via the TX pin. The fabric clock (TX_FAB_CLK) is also transmitted through the TX_CLK pin. Figure 24, page 38 shows the block diagram of the TX_DDR_G_A interface.

The following illustration shows the block diagram of the TX_DDR_G_A interface.

Figure 27 • **Block Diagram of the TX_DDR_G_A Interface**



The following table shows the TX_DDR_G_A interface mode ports.

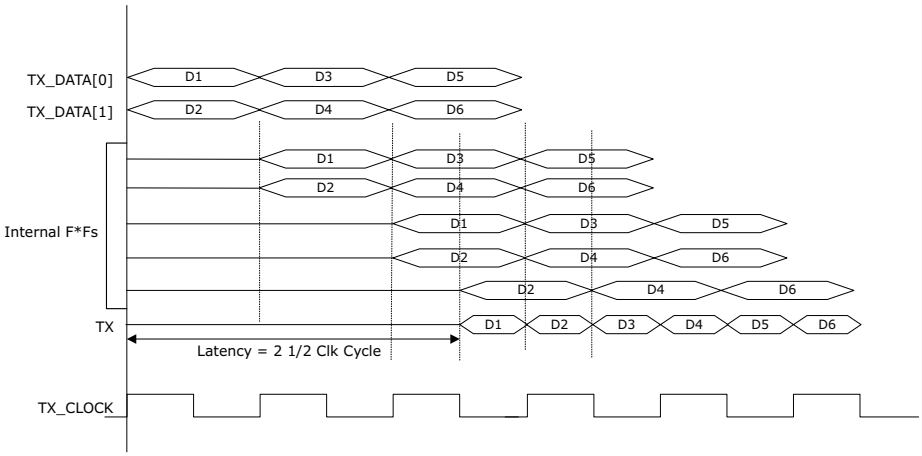
Table 26 • **TX_DDR_G_A Interface Mode Ports**

Port	I/O	Description
TX_DATA[1:0]	Input	DDR transmit data from fabric TX_DATA[0]—data transmitted on the falling edge of the TX_CLK TX_DATA[1]—data transmitted on the rising edge of the TX_CLK
TX_FAB_CLK	Input	DDR transmit clock from fabric, and can be routed via global clock network
OE_DATA	Input	DDR transmit data enable (active high)
TX	Output	DDR output to I/Os
TX_CLK	Input	DDR clock to I/Os

4.6.2.1 Interface Waveform

The following illustration shows the TX_DDR_G_A interface signal waveform.

Figure 28 • **TX_DDR_G_A Interface Signal Waveform**



4.6.2.2 Resource Utilization

The following are the number of I/Os used in the TX_DDR_G_A interface:

- One IOD per data I/Os
- One IOA per data I/Os
- One IOD for clock
- One IOA for clock

4.6.2.3 Pin Selection Rules

For the TX_DDR_G_A interface, TX and TX_CLK I/Os can be placed freely. The TX data and TX_CLK will have a skew that is equal to the global clock network.

4.6.2.4 Latency

The following table shows the latency for the TX_DDR_G_A interface.

Table 27 • **Latency for the TX_DDR_G_A Interface**

I/O Interface	Gear Ratio	Latency Cycle (RX_CLOCK cycle)
RX_DDR_L_A	2:1	2.5

4.7 PolarFire FPGA Protocol-Specific I/O Interfaces

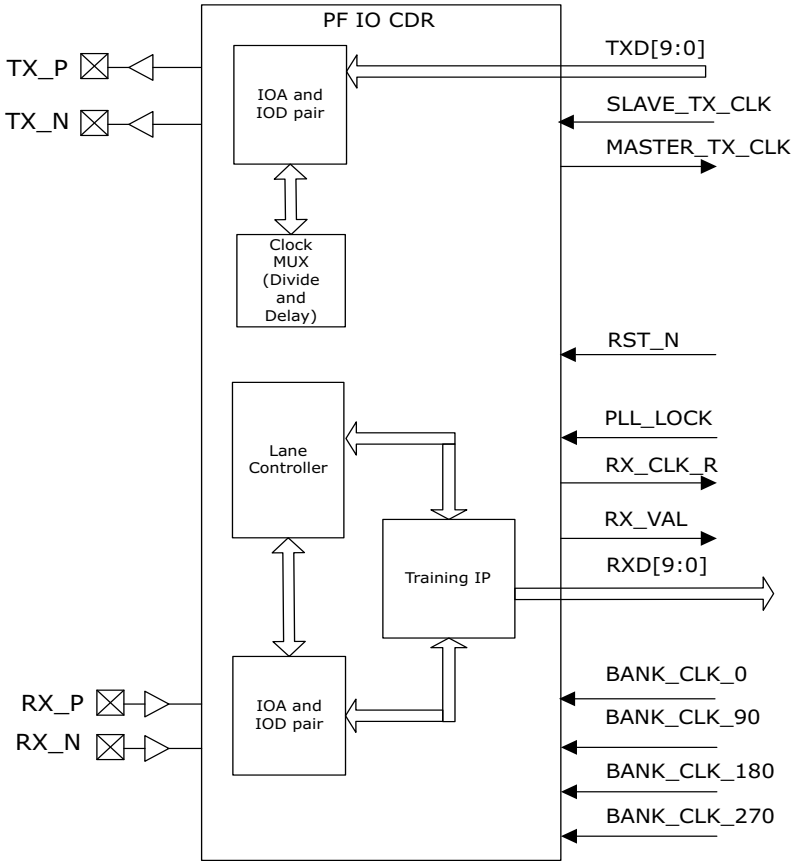
The following sections describe the PolarFire FPGA protocol-specific I/O interfaces.

4.7.1 PF_IOD_CDR

The PF_IOD_CDR interface provides an asynchronous receiver and a transmit interface for serial data transfers. This interface can support up to 1 GbE transfers. It supports serial protocols and other similar encoded serial protocols. PF_IOD_CDR uses a 10:1 digital ratio to provide a 10-bit data and clock interface for both transmit and receive modes. In the receive mode, the clock recovery circuit is used in the lane controller to generate the recovered clock. The PF_IOD_CDR interface is compatible with CoreTSE, CoreTSE_AHB, and CoreSGMII configured in TBI mode.

The following illustration shows the PF_IOD_CDR transmit and receive interface.

Figure 29 • **PF_IOD_CDR Transmit and Receive Interface Modes**



4.7.2 Receive Interface

The PF_IOD_CDR receive interface uses four high-speed bank clocks and generates the recovered clock. The lane controller in the IOD includes a clock recovery block. It uses the incoming data and the four bank clocks and generates RX_CLK_R, also known as DIVCLK. The downstream IP or logic uses this clock. The serial data is received on an IOA pair and sent to the associated IOD block. The IOD block uses a 10:1 digital ratio. The IOD block uses the recovered clock to capture the serial data stream to the core.

The CDR requires four phases of the HS_IO_CLK running at half the frequency of the serial data rate. The lane controller uses the four phases of the HS_IO_CLK and inverts them internally to create eight phases for the CDR. The soft training IP performs the clock phase selection dynamically on the eight phases to the clock, which best matches the current data phase. The RX_CLK_R into the fabric includes jitter from the switching of the phase which creates this clock.

4.7.3 Transmit Interface

The PF_IOD_CDR transmit interface converts the parallel data into a serial data stream using the IOD interface. It receives the parallel data TXD[9:0] and transmits it via the I/O ports such as TX_P and TX_N. The PF_IOD_CDR transmit interface uses the same PLL used in the receive interface. The source clock is connected to HS_IO_CLK_0. The PF_IOD_CDR interface generates the master transmit clock (MASTER_TX_CLK) from HS_IO_CLK_0 and the MASTER_TX_CLK is used by the transmit logic.

The following table shows the PF_IOD_CDR interface associated ports.

Table 28 • **PF_IOD_CDR Interface Associated Ports**

Port	I/O	Description
RX_P	Input	Serial data input (P side)
RX_N	Input	Serial data input (N side)
HS_IO_CLK_0	Input	Bank clock with phase 0 is used for both receive and transmit interface. Frequency must be half the rate of the serial data input
HS_IO_CLK_90	Input	Bank clock with phase 90 is used for the I/O clock recovery. Frequency must be half the rate of the serial data input
HS_IO_CLK_180	Input	Bank clock with phase 180 is used for the I/O clock recovery. Frequency must be half the rate of the serial data input
HS_IO_CLK_270	Input	Bank clock with phase 270 is used for the I/O clock recovery. Frequency must be half the rate of the serial data input
RX_CLK_R	Output	Recovered clock for the fabric interface is divided by five from the HS_IO_CLK. This clock is routed using a regional clock
RX_VAL	Output	The CDR is locked to the incoming serial data
RXD[9:0]	Output	Recovered data clocked on RX_CLK_R
RST_N	Input	Active low reset input
TX_P	Input	Serial data output (P side)
TX_N	Input	Serial data output (N side)
MASTER_TX_CLK	Output	Master transmit clock (125 MHz) is trained by the PF_IOD_CDR interface, and is used by the transmit logic. This is the clock source of SLAVE_TX_CLK and can be used for all the PF_IOD_CDR interfaces using the same HS_IO_CLKs.
SLAVE_TX_CLK	Input	Transmit clock for IOD. Must be connected to the MASTER_TX_CLK from one PF_IOD_CDR that uses the same HS_IO_CLKs.

Table 28 • **PF_IOD_CDR Interface Associated Ports *(continued)***

Port	I/O	Description
TXD[9:0]	Input	Transmit data clocked on SLAVE_TX_CLK

4.7.3.1 Bank Clock Generation Using CCC Configurator

The PF_IOD_CDR receive interface is sourced by a single PLL driving four bank clocks of 0, 90, 180, and 270 degrees running at the data rate. Configure the PLL using CCC SgCore available in the Libero SoC PolarFire IP catalog. The PF_IOD_CDR transmit interface uses HS_IO_CLK_0 and generates the transmit clock.

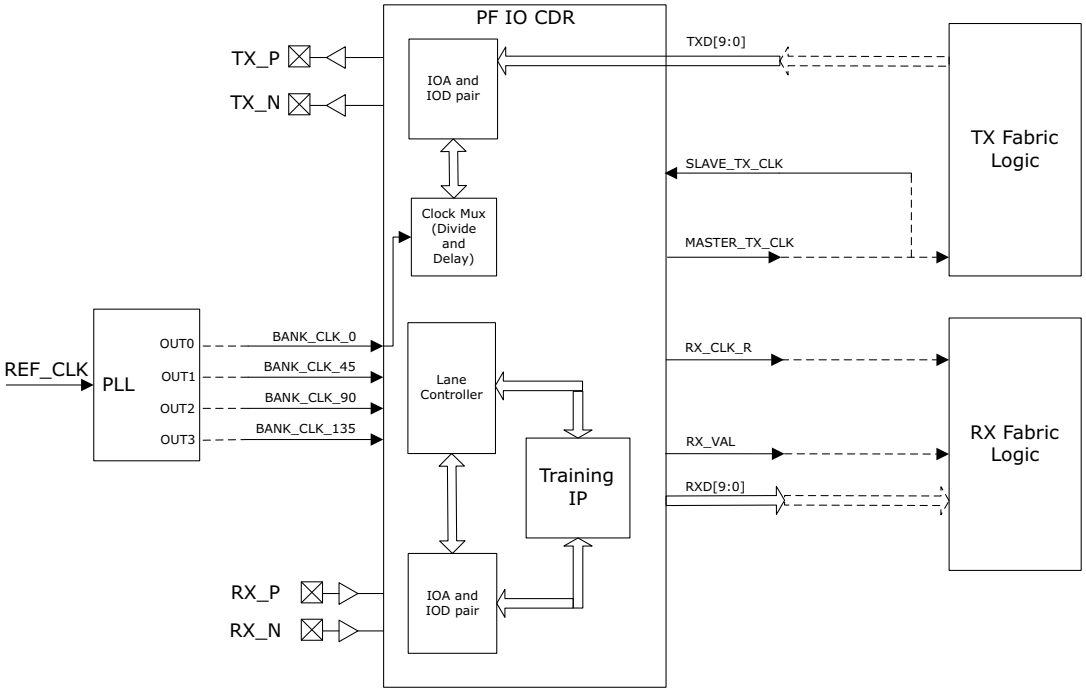
The following table lists the configuration of the instantiated PLL.

Table 29 • **Configuration of the Instantiated PLL**

Output	Frequency	Phase	Fabric	Bank Clock	Dedicated
OUT0	Data Rate/2	0	No	Yes	No
OUT1	Data Rate/2	90	No	Yes	No
OUT2	Data Rate/2	180	No	Yes	No
OUT3	Data Rate/2	270	No	Yes	No

The following illustration shows the PF_IOD_CDR T interface connected to the CCC and fabric logic.

Figure 30 • **Using PF_IOD_CDR T Interfaces**



4.7.4 RX and TX PLL Sharing

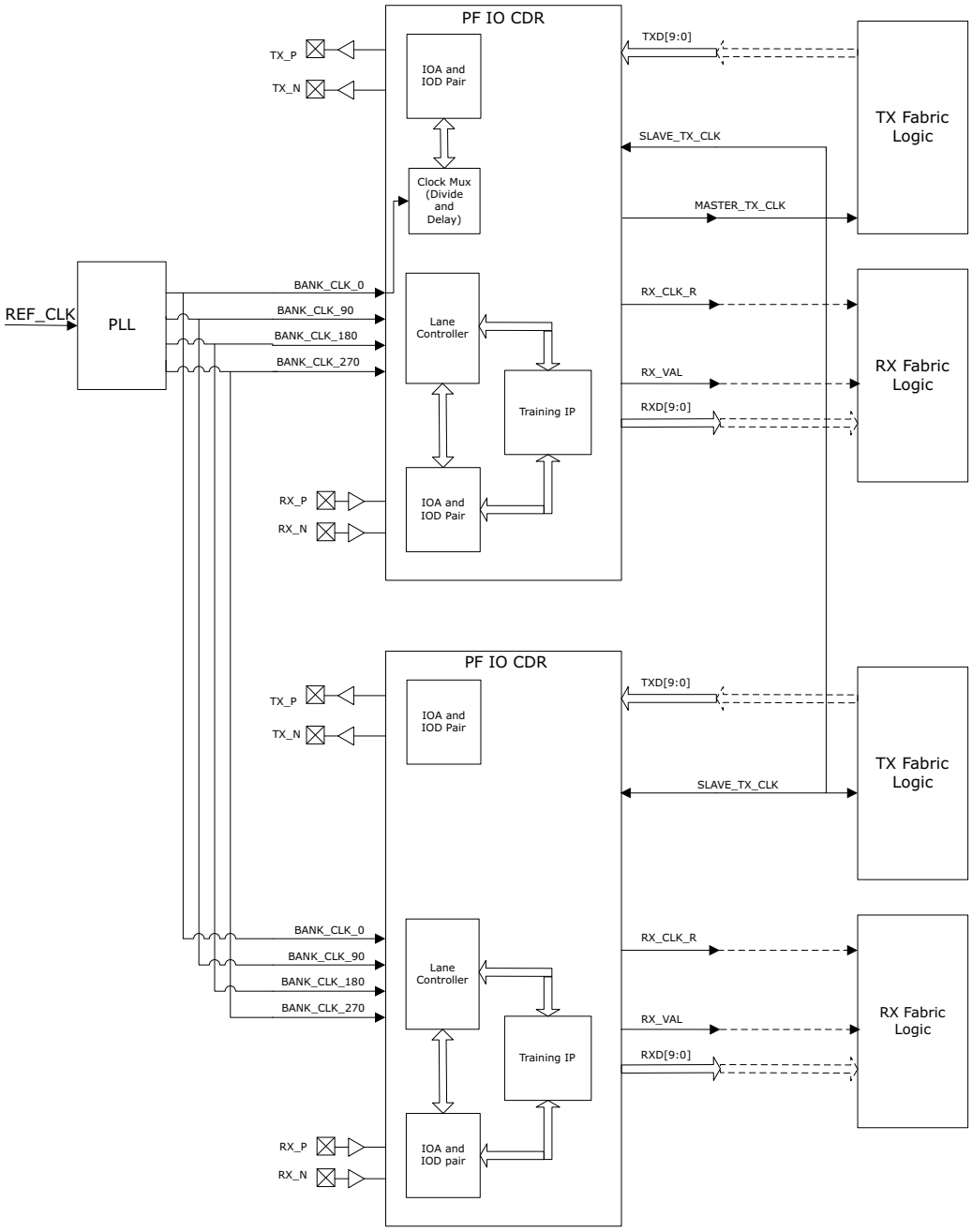
The same PLL is shared between the PF_IOD_CDR receive and transmit interfaces, as shown in Figure 31, page 45. In addition, multiple PF_IOD_CDR interfaces can share the same PLL on the adjacent vertical and horizontal edges. For instance, the PLL_SW_0 interface can drive the PF_IOD_CDR interface on the southern and western edges.

4.7.5 TX Clock Sharing

Many IOD CDR interfaces can share the same MASTER_TX_CLK. One of the interfaces must instantiate the clock generation logic, also known as the master interface. The clock generation clock generates the MASTER_TX_CLK. The other interfaces can also reuse MASTER_TX_CLK by connecting it to SLAVE_TX_CLK.

The following illustration shows multiple PF_IOD_CDR transmit and receive interfaces.

Figure 31 • **Multiple PF_IOD_CDR Transmit and Receive Interfaces**



4.7.5.1 Resource Utilization

A future version of this user guide will have this information.

4.7.5.2 Pin Selection Rules

Follow these rules when assigning a pin for the PF_IOD_CDR interface:

- RX_N, RX_P, TX_N, and TX_P must be placed on the same lane.
- Multiple IOD CDR interfaces can share the same PLL. Each interface sharing the same PLL must be placed within the reach of PLL.
- Each PLL can drive IOD interfaces on the adjacent vertical and horizontal edges. For instance, PLL_SW_0 can drive IOD CDR interface on the southern and western edges.

4.7.5.3 Latency

The following table shows the latency for the PF_IOD_CDR interface.

Table 30 • **Latency for the PF_IOD_CDR Interface**

I/O Interface	Gear Ratio	Latency Cycle (RX_CLOCK Cycle)
PF_IOD_CDR Rx	1:10	TBD
PF_IOD_CDR Tx	10:1	TBD

A future version of the user guide will have a waveform diagram and a use model for the PF_IO_CDR interface.

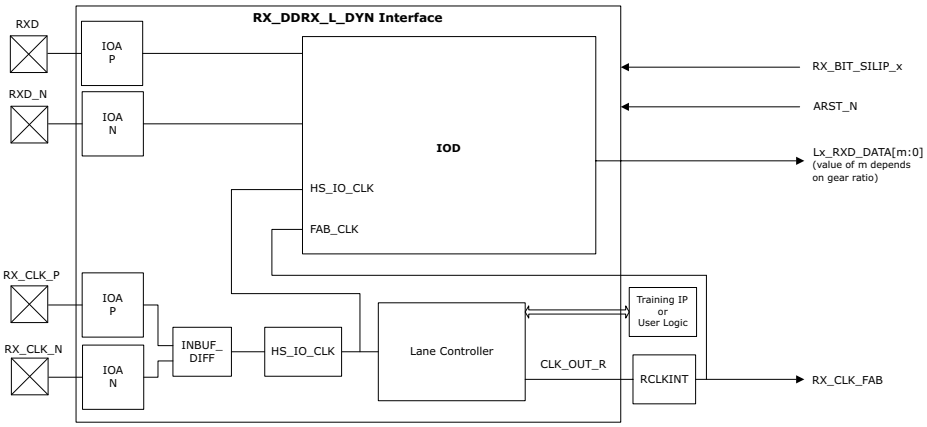
4.7.6 RX_DDRX_L_DYN

The RX_DDRX_L_DYN interface is used to capture differential DDR data using dynamic control. The clock and data relationship can be adjusted dynamically when the device receives the differential DDR data. The RX_DDRX_L_DYN interface is used for the maximum data rate of 1600 Mbps and uses the digital ratio of 2, 3.5, 4, and 5.

The RX_DDRX_L_DYN interface receives the differential data RXD/RXD_N and the differential clock RX_CLK_P/RX_CLK_N via I/O and passed the data Lx_RXD_DATA and fabric clock (RX_CLK_FAB) to the fabric. The receive clock input (RX_CLK_P/RX_CLK_N) is passed through the lane controller to generate RX_CLK_FAB, which is driven by RCLKINT.

The following illustration shows the block diagram of RX_DDRX_L_DYN interface.

Figure 32 • **Block Diagram of the RX_DDRX_L_DYN Interface**



The following table lists the RX_DDRX_L_DYN interface mode ports.

Table 31 • **RX_DDRX_L_DYN Ports**

Port	I/O	Description
RXD/RXD_N	Input	Differential input DDR data
RX_CLK_P/RX_CLK_N	Input	Differential input clock
ARST_N	Input	Asynchronous reset to IOD and lane controller
RX_BIT_SILIP_x	Input	Bit slip input from fabric is initiated by a rising edge of the slip signal from the core fabric
Lx_RXD_DATA[m:0]	Output	DDR output to FPGA fabric, value depends on digital ratio
RX_CLK_FAB	Output	Receive clock to FPGA fabric

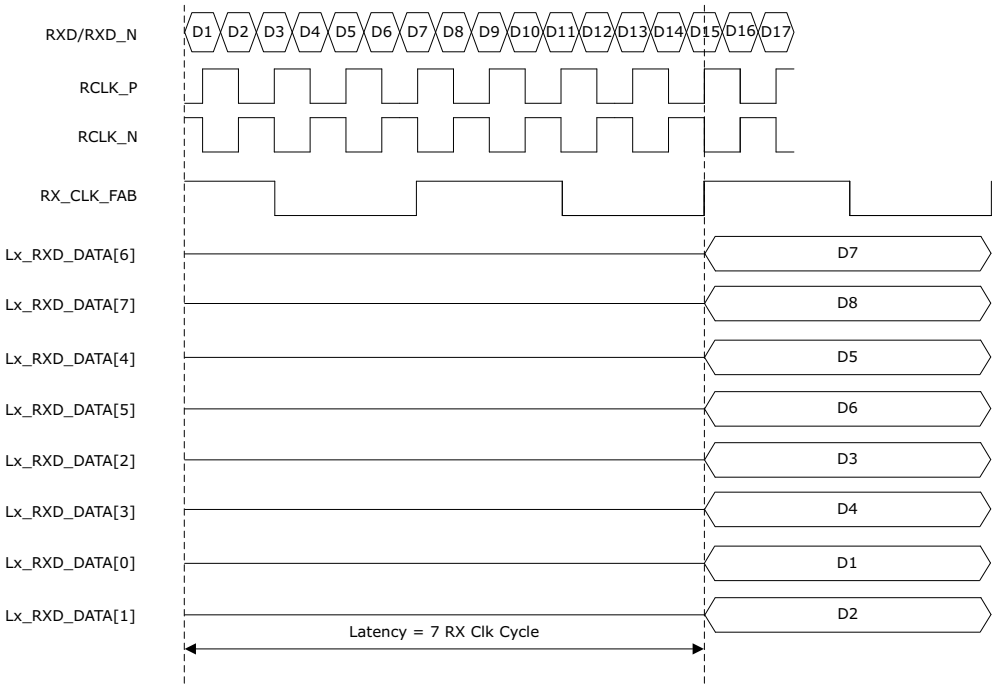
The RX_DDRX_L_DYN interface has bit slip input from fabric, called RX_BIT_SILIP_x. The slip input pin is used for word alignment. The slip function is used in one of two ways:

- 3.5 Digital Mode—slips 2 bits at a time
- 2, 4, and 5 Digital Modes—slip 1 bit at a time

4.7.6.1 Interface Waveform

The following illustration shows the signal waveform of RX_DDRX_L_DYN interface when slip input is not used.

Figure 33 • **RX_DDRX_L_DYN Waveform**



4.7.6.2 Resource Utilization

TBD

4.7.6.3 Pin selection rules

The following conditions are applicable when assigning pins to the RX_DDRX_L_DYN interface:

- RX_CLK_P/RX_CLK_N must be on a P I/O pad.
- RXD/RXD_N and RX_CLK_P/ RX_CLK_N must be in the high-speed I/O clock (HS_IO_CLK) network region.

4.7.6.4 Latency

The following table lists the latency for the RX_DDR_L_A interface.

Table 32 • **RX_DDR_L_A Latency**

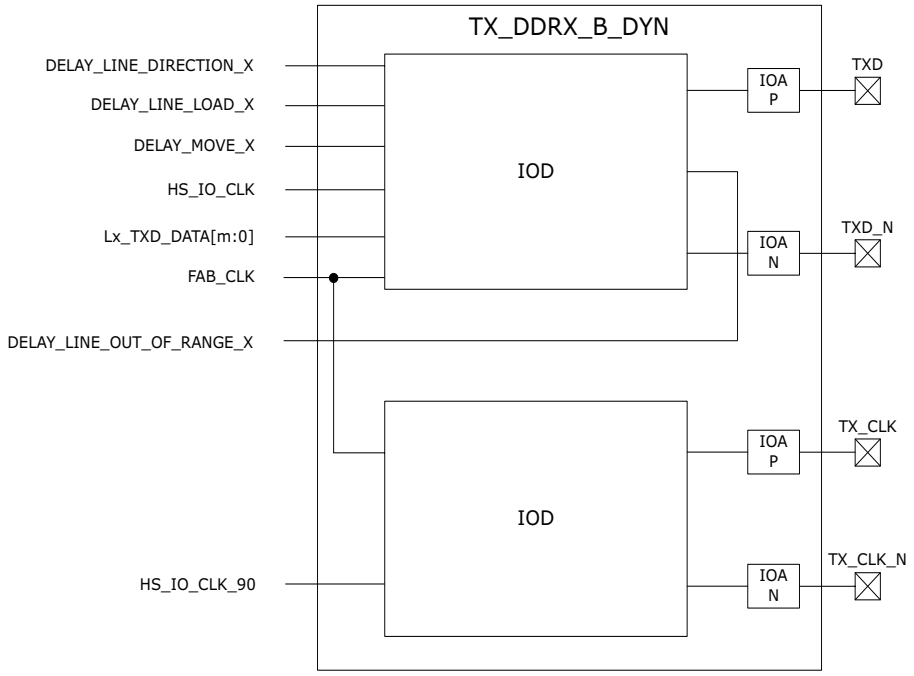
I/O Interface	Gear Ratio	Latency Cycle (RX_CLOCK)
RX_DDRX_L_DYN	2, 3.5, 4, 5	Depends on digital ratio

4.7.7 TX_DDRX_B_DYN

The TX_DDRX_B_DYN interface implements the DDR transmit interface using dynamic control, where the user has the ability to dynamically adjust the clock and data relationship as it leaves the device. The IOD block uses TX_CLK_Xx (same as the fabric clock, FAB_CLK) to capture the transmitted data from fabric and transmits data using TX_CLK/TX_CLK_N via TXD/TXD_N differential pads. The transmit clock is also transmitted through TX_CLK/TX_CLK_N differential pads. The TX_DDRX_B_DYN interface is used for the maximum data rate of 1600 Mbps and uses digital ratio of 2, 3.5, 4 and 5.

The following illustration shows the block diagram of the TX_DDRX_B_DYN interface.

Figure 34 • **Block Diagram of the TX_DDRX_B_DYN Interface**



The following table lists the TX_DDRX_B_DYN Interface mode ports.

Table 33 • **TX_DDRX_B_DYN Ports**

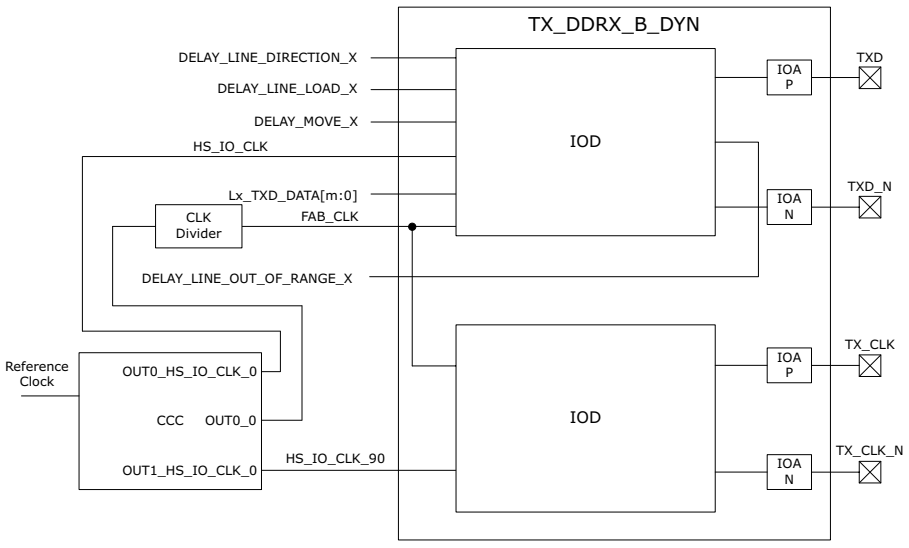
Port	I/O	Description
Lx_TXD_DATA[m:0]	Input	DDR transmit data from fabric, value of m depends on digital ratio
FAB_CLK	Input	DDR transmit clock from fabric, must be routed via the global clock network
HS_IO_CLK	Input	DDR transmit clock, normally generated using the CCC
HS_IO_CLK_90	Input	90° shifted version of DDR transmit clock
DELAY_LINE_LOAD_x	Input	Loads the static delay. Also, asynchronously reloads the initial static delay settings
DELAY_LINE_MOVE_x	Input	Changes the delay settings. Each pulse on DELAY_LINE_MOVE increases or decreases the delay setting by 1 tap delay.
DELAY_LINE_DIRECTION_x	Input	Increments or decrements the delay settings. When set to 0, pulsing DELAY_LINE_MOVE _x increases the delay settings by 1. When set to 1, pulsing DELAY_LINE_MOVE_x decreases the delay settings by 1.
DELAY_LINE_OUT_OF_RANGE	Output	Out-of-range output flag. When the delay setting reaches either the minimum or maximum value of the delay chain, the delay chain controller asserts this signal to indicate that it has reached the end of the delay chain. Note that the delay setting does not wrap around and does not exceed the maximum or fall below the minimum value when this condition is reached.
TXD/TXD_N	Output	Differential DDR transmit data
TX_CLK/TX_CLK_N	Input	Differential DDR clock

4.7.8 Generate HS_IO_CLK/HS_IO_CLK_90 and FAB_CLK using CCC

HS_IO_CLK, HS_IO_CLK _90, and FAB_CLK can be generated using the same CCC and clock divider.

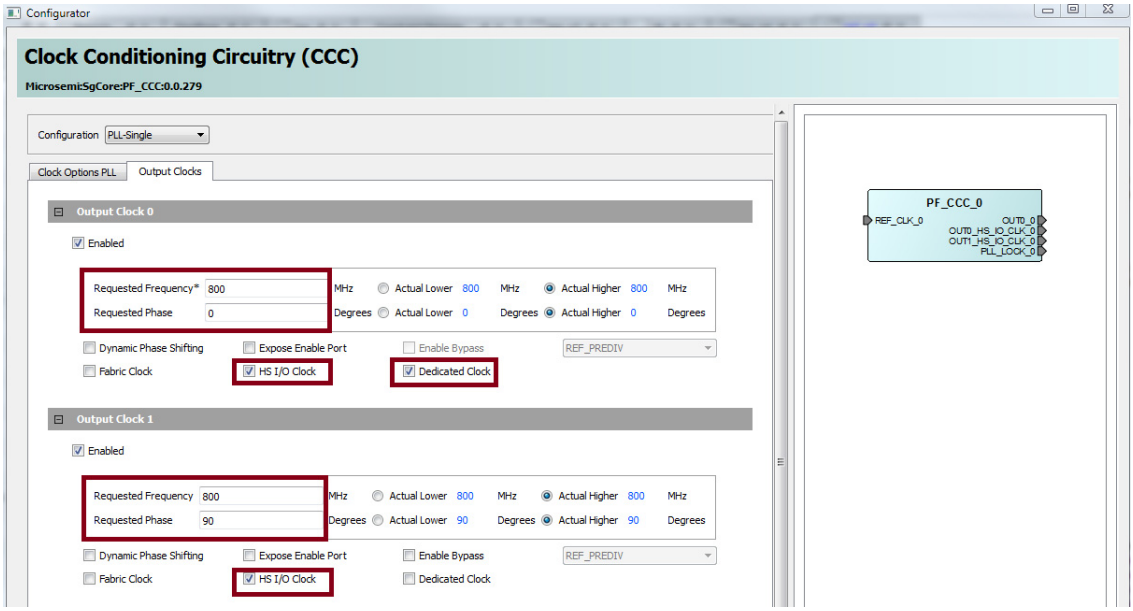
The following illustration shows the clock generation scheme.

Figure 35 • **TX_DDRX_B_DYN Interface with CCC Block**



The following figure shows the CCC configuration setting for 1600 Mbps.

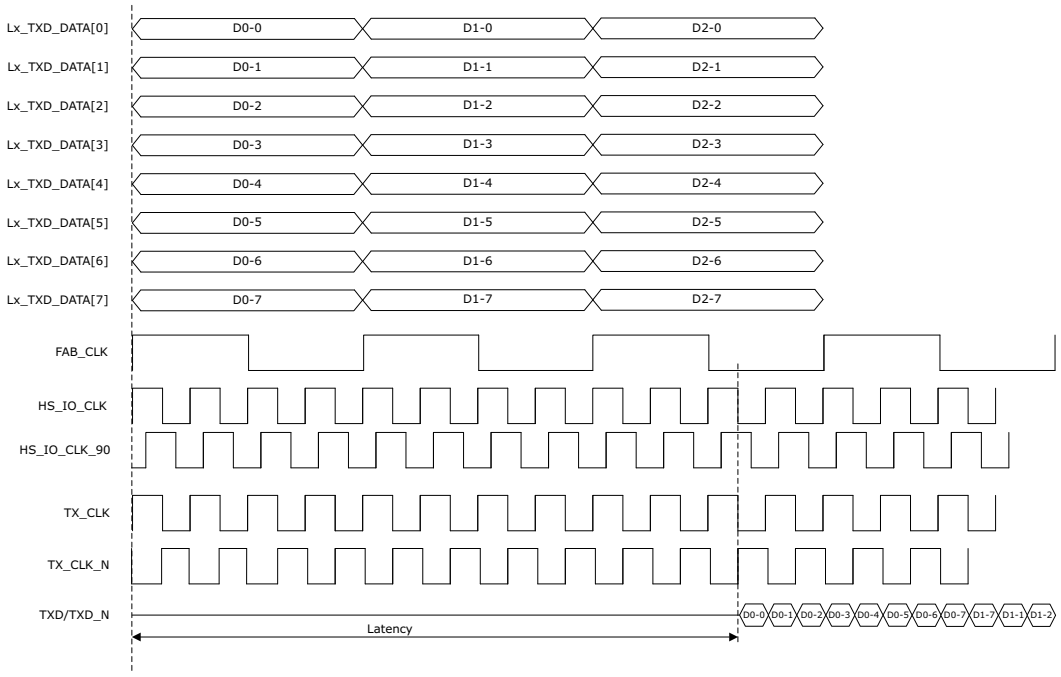
Figure 36 • **Configuring CCC with 1600 Mbps**



4.7.8.1 Interface Waveform

The following illustration shows the signal waveform of TX_DDRX_B_DYN interface.

Figure 37 • **TX_DDRX_B_DYN Interface Signal Waveform**



4.7.8.2 Resource Utilization

TBD

4.7.8.3 Pin Selection Rules

When assigning a pin to the TX_DDRX_B_DYN interface, TXD/TXD_N and TX_CLK_P/ TX_CLK_N must be placed in the same high-speed I/O clock (HS_IO_CLK) network region.

4.7.9 RX_DDRX_L_DYN_MIPI/ TX_DDRX_B_DYN_MIPI

The RX_DDRX_L_DYN_MIPI interface is similar to the RX_DDRX_L_DYN interface with support for MIPI I/O standard. Similarly, the TX_DDRX_B_DYN_MIPI interface is similar to the TX_DDRX_B_DYN interface with support for MIPI I/O standard. RX_DDRX_L_DYN_MIPI uses a lane controller and a local clock region to set clock rate on the fabric side. So, it's limited to an I/O lane on the data side (6 differential pair or 12 single ended I/O). The future version of the user guide will have detailed information.