Tz325 Data Sheet

DSTz325-v1.1
Novemer 2024
www.efinixinc.com

Contents

Introduction

The Topaz Tz325 FPGA features the high-density, low-power Efinix® Quantum® compute fabric wrapped with an I/O interface. This FPGA has a variety of features, such as a hardened RISC-V block, transceivers, LPDDR4 DRAM controller, and MIPI D-PHY.

The quad-core hardened RISC-V block has a 32-bit CPU featuring the ISA RISCV32I with M, A, C, F, and D extensions, and six pipeline stages. You utilize the hardened RISC-V block by instantiating the Sapphire High-Performance SoC, combining the speed and efficiency of a hardened RISC-V block with the flexibility of peripherals in soft logic.

The full-duplex transceivers support multiple protocols including PCIe® Gen3, SGMII, and 10GBase-KR as well as a PMA Direct mode with data rates from 1.25 Gbps to 12.5 Gbps.

Tz325 FPGAs include a hardened MIPI D-PHY, which you can use with Efinix® MIPI CSI-2 and DSI controller IP cores to create multi-camera, high definition vision systems, edge computing, and hardware acceleration systems. Additionally, these FPGAs have a hardened DDR DRAM controller block that supports LPDDR4 DRAM interfaces.

Together, these features enable a wide variety of applications such as PCIe cards, RF repeaters and radio units, small cell/massive MIMO, cable modems, machine vision, automotive, AV equipment and broadcast, medical imaging, and video bridges.

Features

- High-density, low-power Quantum® compute fabric
- Built on TSMC 16 nm process
- 10-kbit high-speed, embedded SRAM, configurable as single-port RAM, simple dual-port RAM, true dual-port RAM, or ROM
- High-performance DSP blocks for multiplication, addition, subtraction, accumulation, and up to 15-bit variable-right-shifting
- Versatile on-chip clocking
  - Low-skew global network supporting 32 clock or control signals
  - Regional and local clock networks
  - Up to 12 PLLs with support for fractional-N division, programmable duty cycle, spread-spectrum clocking, and dynamic reconfiguration
- FPGA interface blocks
  - 32-bit quad-core hardened RISC-V block
  - Four high-speed transceiver banks, each with 4 lanes:
    - Support data rates from 1.25 Gbps up to 12.5 Gbps per channel
    - PCIe Gen3 x4:
      - Compliant with the PCIe® 3.0, 2.1, and 1.1 specifications
      - Support x1, x2, and x4 configurations
      - Configure as Root Port (RP) or End Point (EP)
      - Single Root IO Virtualization (SRIOV)
    - Supports SGMII and 10GBase-KR protocols as well as PMA Direct
  - Two LPDDR4 PHY interfaces (supporting x32 DQ widths) with memory controller hard IP
  - Three MIPI D-PHY RX and TX interfaces with speeds up to 2.0 Gbps
  - Two varieties of general-purpose I/O (GPIO) pins:
    - High-voltage I/O (HVIO) pins support 1.8, 2.5, and 3.3 V

- Configurable high-speed I/O (HSIO) pins support
  - Single-ended and differential I/O
  - LVDS, subLVDS, Mini-LVDS, and RSDS (RX, TX, and bidirectional), up to 1.3 Gbps
  - MIPI lane (DSI and CSI) in high-speed and low-power modes, up to 1.3 Gbps
- One oscillator
- Spread-Spectrum Clocking (SSC) PLL

- Flexible device configuration
  - Standard SPI interface (active, passive, and daisy chain)
  - JTAG interface
  - Supports internal reconfiguration
- Single-event upset (SEU) detection feature
- Optional security feature
  - Asymmetric bitstream authentication using RSA-4096
  - Bitstream encryption/decryption using AES-GCM
- Fully supported by the Efinity® software, an RTL-to-bitstream compiler

Important: All specifications are preliminary and pending hardware characterization.

Table 1: Tz325 FPGA Resources

Logic Elements (LEs)	eXchangeable Logic and Routing (XLR) Cells		Global Clock and Control Signals	Embedded Memory (Mbits)	Embedded Memory Blocks (10 Kbits)	Embedded DSP Blocks
	Total	SRL8[1]				
326,080	348,749	59,200	Up to 32	19.22	1,877	1,008

[1] Number of XLR that can be configured as shift register with 8 maximum taps.

Table 2: Tz325 Package-Dependent Resources

Resource		N484	C529
Single-ended GPIO (maximum)	HVIO LVCMOS: 1.8, 2.5, 3.0, 3.3 V LVTTL: 3.0, 3.3 V	20	51
	HSIO (1.2, 1.35, 1.5, 1.8 V LVCMOS, HSTL, and SSTL)[(2)]	85	176
Differential GPIO (maximum)	HSIO (LVDS, Differential HSTL, and SSTL)[(2)]	42	88
	HSIO (MIPI D-PHY Data Lanes)	28	64
	HSIO (MIPI D-PHY Clock Lanes)	7	11
LPDDR4 PHY with memory controller	x32 DQ width	1	1
MIPI D-PHY Hard Blocks	RX	1	–
	TX or SSC PLL	1	–
Global clock or control signals from GPIO pins		17	29
Fractional PLLs		9	12
Transceiver banks	PCIe	1xGen3	–
	SGMII, 10GBase-KR, or PMA Direct	up to 2	–

Available Package Options

Table 3: Available Packages

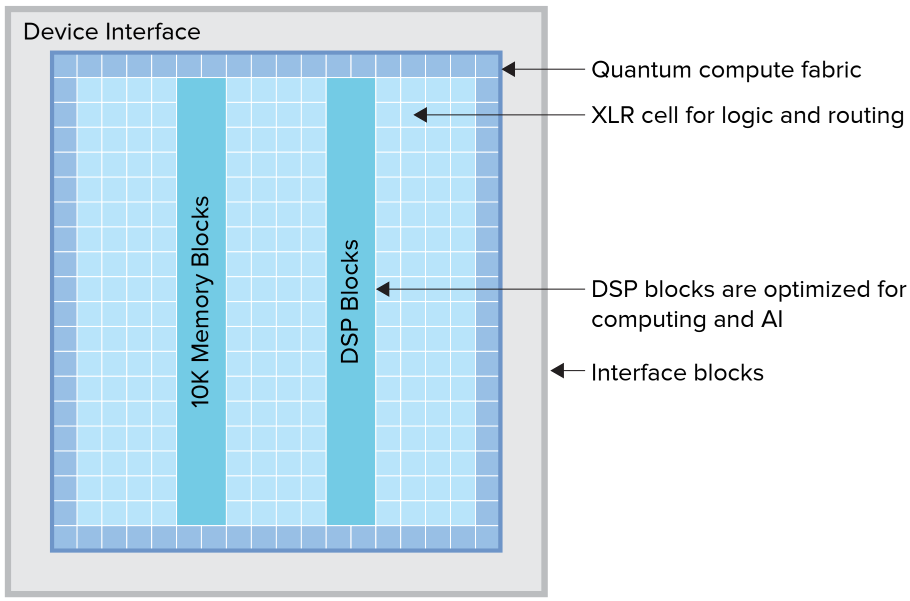
Package	Dimensions (mm x mm)	Pitch (mm)
484-ball FBGA	18 x 18	0.8
529-ball FBGA	19 x 19	0.8

(2) GPIOB_PN_00, GPIOB_PN_01, and GPIOB_PN_02 do not support the SSTL and HSTL input I/O standards.

Device Core Functional Description

Tz325 FPGAs feature an eXchangeable Logic and Routing (XLR) cell that Efinix® has optimized for a variety of applications. Topaz FPGAs contain LEs that are constructed from XLR cells. Each FPGA in the Topaz family has a custom number of building blocks to fit specific application needs. As shown in the following figure, the FPGA includes I/O ports on all four sides, as well as columns of LEs, memory, and DSP blocks. A control block within the FPGA handles configuration.

Figure 1: Tz325 FPGA Block Diagram



Interface blocks include GPIO, LVDS, PLL, MIPI lane I/O, MIPI D-PHY, DDR DRAM, RISC-V, and transceivers.

XLR Cell

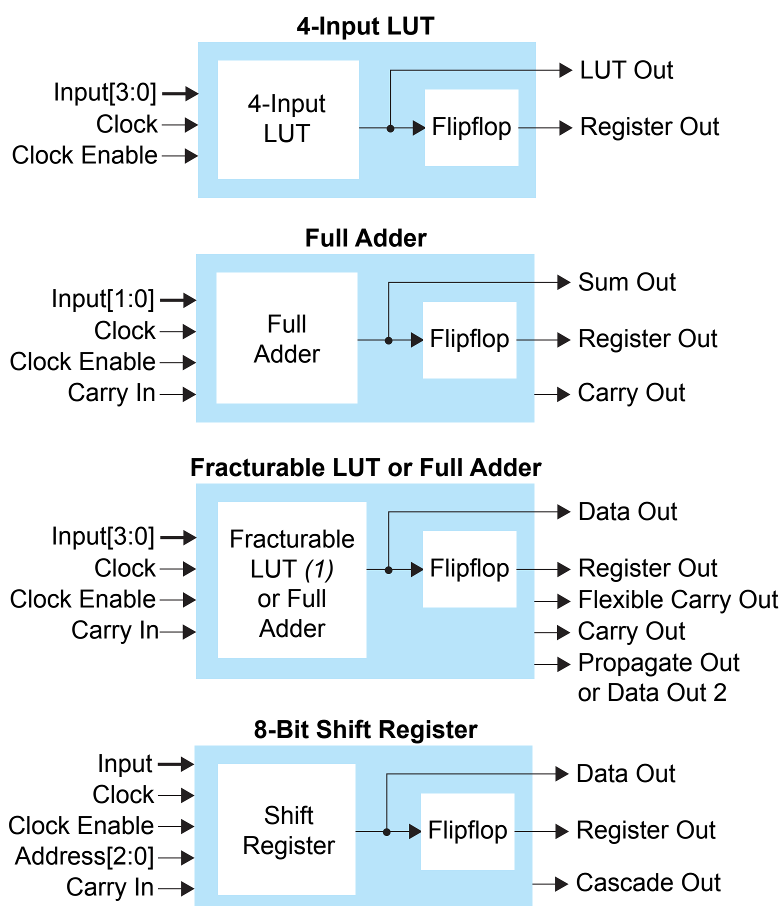
The eXchangeable Logic and Routing (XLR) cell is the basic building block of the Quantum® architecture. The Efinix® XLR cell combines logic and routing and supports both functions. This unique innovation greatly enhances the transistor flexibility and utilization rate, thereby reducing transistor counts and silicon area significantly.

Learn more: For more detailed on the advantages the XLR cell brings to Topaz FPGAs, read the **Why the XLR Cell is a Big Deal White Paper**.

The XLR cell functions as:
- A 4-input LUT that supports any combinational logic function with four inputs.
- A simple full adder.
- An 8-bit shift register that can be cascaded.
- A fracturable LUT or full adder.

The logic cell includes an optional flipflop. You can configure multiple logic cells to implement arithmetic functions such as adders, subtractors, and counters.

Figure 2: Logic Cell Functions



1. The fracturable LUT is a combination of a 3-input LUT and a 2-input LUT. They share 2 of the same inputs.

Learn more: Refer to the **Quantum® Topaz Primitives User Guide** for details on the Topaz logic cell primitves.

Embedded Memory

The core has 10-kbit high-speed, synchronous, embedded SRAM memory blocks. Memory blocks can operate as single-port RAM, simple dual-port RAM, true dual-port RAM, or ROM. You can initialize the memory content during configuration. The Efinity® software includes a memory cascading feature to connect multiple blocks automatically to form a larger array. This feature enables you to instantiate deeper or wider memory modules.

Note: The block RAM content is random and undefined if it is not initialized.

The read and write ports support independently configured data widths, an address enable, and an output register reset. The simple dual-port mode also supports a write byte enable.

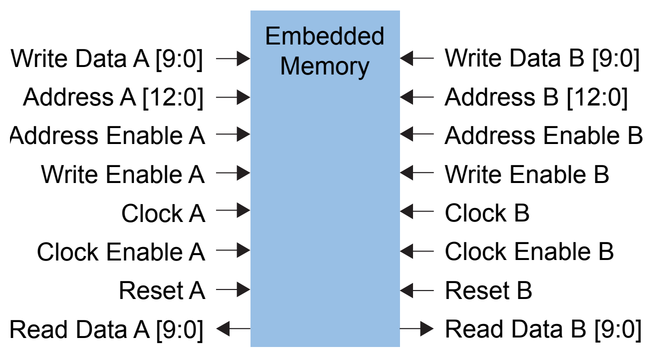
Learn more: Refer to the **Quantum® Topaz Primitives User Guide** for details on the Topaz RAM configuration.

True Dual-Port Mode

The memory read and write ports have the following modes for addressing the memory (depth x width):

1024 x 8	2048 x 4	4096 x 2
8192 x 1	1024 x 10	2048 x 5

Figure 3: RAM Block Diagram (True Dual-Port Mode)

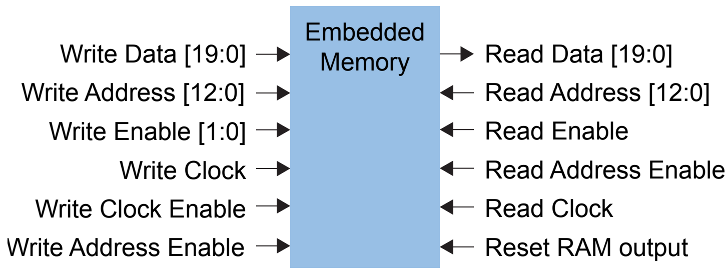


Simple Dual-Port Mode

The memory read and write ports have the following modes for addressing the memory (depth x width):

512 x 16	1024 x 8	2048 x 4	4096 x 2
8192 x 1	512 x 20	1024 x 10	2048 x 5

Figure 4: Simple Dual-Port Mode RAM Block Diagram (512 x 20 Configuration)



DSP Block

The Topaz FPGA has high-performance, complex DSP blocks that can perform multiplication, addition, subtraction, accumulation, and 4-bit variable right shifting. The 4-bit variable right shift supports one lane in normal mode, two lanes in dual mode and four lanes in quad mode. Each DSP block has four modes, which support the following multiplication operations:

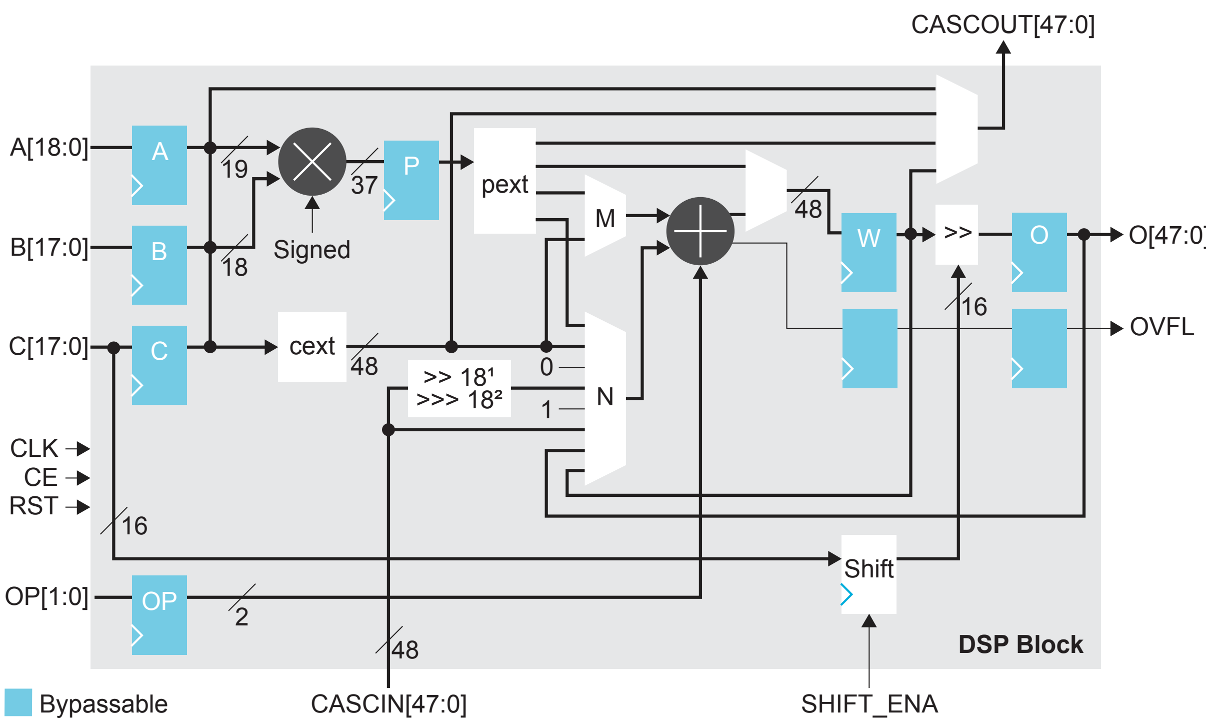
- *Normal*—One 19 x 18 integer multiplication with 48-bit addition/subtraction.
- *Dual*—One 11 x 10 integer multiplication and one 8 x 8 integer multiplication with two 24-bit additions/subtractions.
- *Quad*—One 7 x 6 integer multiplication and three 4 x 4 integer multiplications with four 12-bit additions/subtractions.

Important: The 7 x 6 Quad mode output is truncated to 12-bit.

- *Float*—One fused-multiply-add/subtract/accumulate (FMA) BFLOAT16 multiplication.

The integer multipliers can represent signed or unsigned values based on the `SIGNED` parameter. When multiple EFX_DSP12 or EFX_DSP24 primitives are mapped to the same DSP block, they must have the same `SIGNED` value. The inputs to the multiplier are the A and B data inputs. Optionally, you can use the result of the multiplier in an addition or subtraction operation.

Figure 5: DSP Block Diagram



1. Logical right-shift-by-18.
2. Arithmetic right-shift-by-18.

Learn more: Refer to the **Quantum® Topaz Primitives User Guide** for details on the Topaz DSP block primitives.

Clock and Control Network

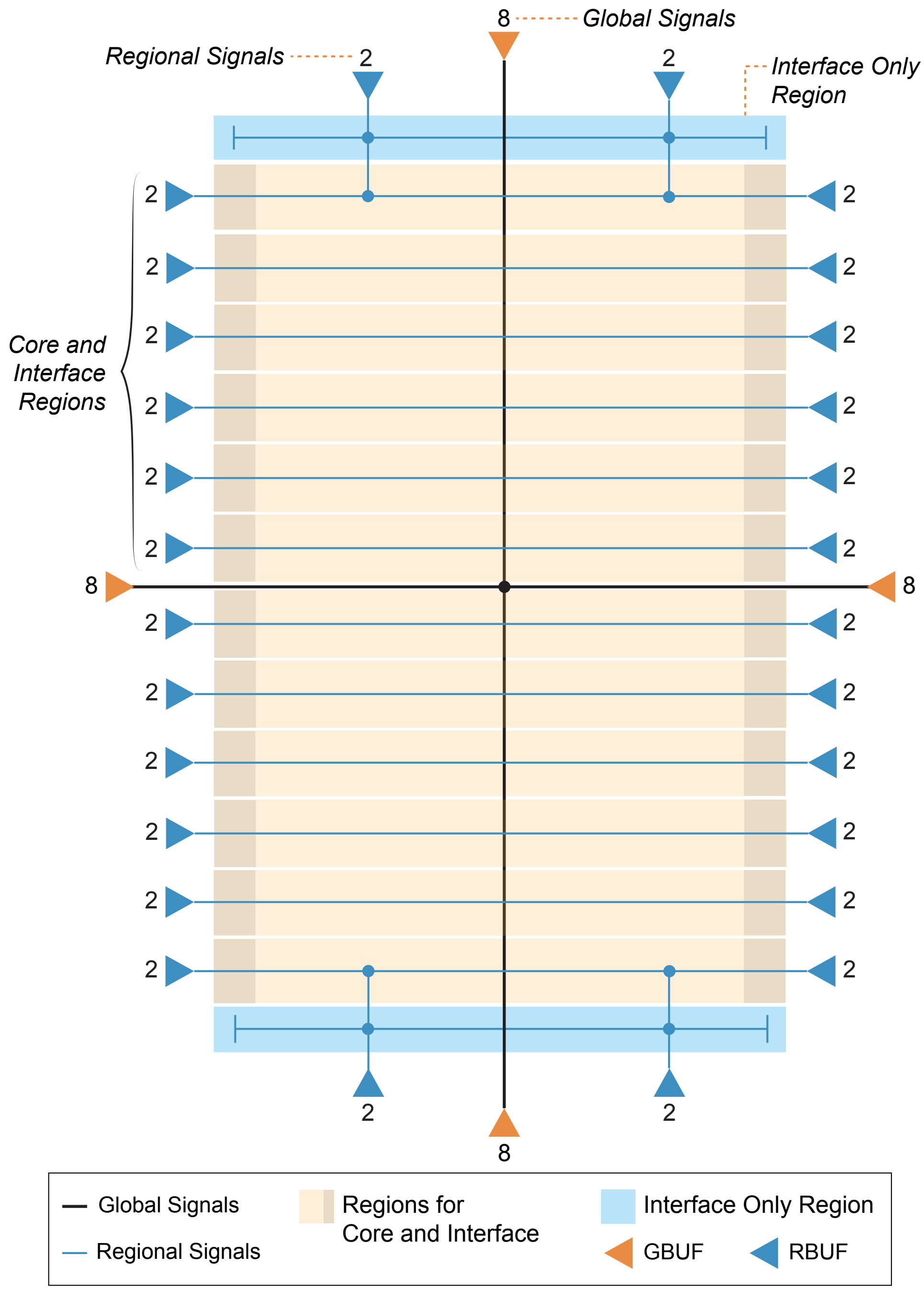
The clock and control network is distributed through the FPGA to provide clocking for the core's LEs, memory, DSP blocks, I/O blocks, and control signals. The FPGA has 32 global signals that can be used as either clocks or control signals. The global signals are balanced trees that feed the whole FPGA.

The FPGA also has regional signals that can only reach certain FPGA regions, including the top or bottom edges. The FPGA has 12 regional networks for the core, right interface, and left interface blocks. The top and bottom interface blocks have 1 regional clock network each. You can drive the right and left sides of each region independently. Each region also has a local network of clock signals that can only be used in that region.

The core's global buffer (GBUF) blocks drive the global and regional networks. Signals from the core and interface can drive the GBUF blocks.

Each network has dedicated enable logic to save power by disabling the clock tree. The logic dynamically enables/disables the network and guarantees no glitches at the output.

Figure 6: Global and Regional Clock Network Overview



The transceivers drive some of the regional buffers on the right side. See **Figure 15** on page 23 for details.

The regional clocks can also drive adjacent regions. See **Driving the Regional Network** on page 21 for details.

Clock Sources that Drive the Global and Regional Networks

The Topaz global and regional networks are highly flexible and configurable. Clock sources can come from interface blocks, such as GPIO or PLLs, or from the core fabric.

Table 4: Clock Sources that Drive the Global and Regional Networks

Source	Description
GPIO	Supports GCLK and RCLK. (Only the P resources support this connection type).
LVDS RX	Supports GCLK and RCLK.
MIPI D-PHY RX, TX, and SSC PLL	Can drive the word clock onto the global and regional clock networks.
Transceiver TX, RX, and PIPE P clocks	Can drive the global and regional clock networks on the right side.
MIPI RX Lane (configured as clock lane)	Supports GCLK (default) and RCLK. You can only use resources that are identified as clocks.
PLL	All output clocks connect to the global network. Refer to **Driving the Regional Network** on page 21 for the PLL clocks that drive the regional network.
Oscillator	Connects to global buffer.
Core	Signals from the core logic can drive the global or regional network.

Driving the Global Network

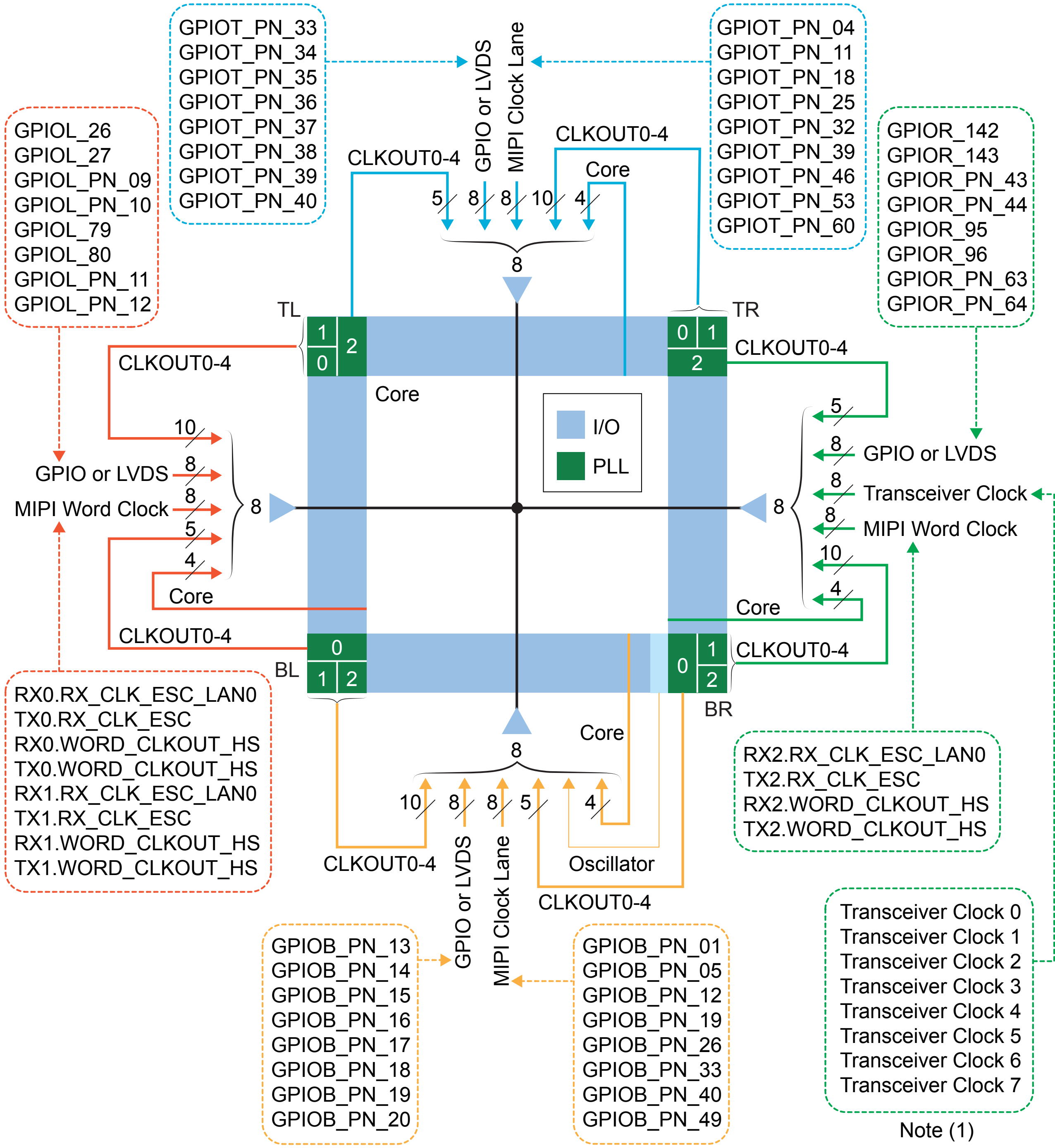
You can access the global clock network using the global clock GPIO pins, PLL outputs, oscillator output, MIPI word clocks, and core-generated clocks.

A clock multiplexing network controls which interface blocks can drive the global and regional networks. Eight of the clock multiplexers are dynamic (two on each side of the FPGA), allowing you to change which clock drives the global signal in user mode.

Learn more: Refer to the **Quantum® Topaz Primitives User Guide** for information on how to configure the global and regional clock networks.

The following figure shows the global network clock sources graphically.

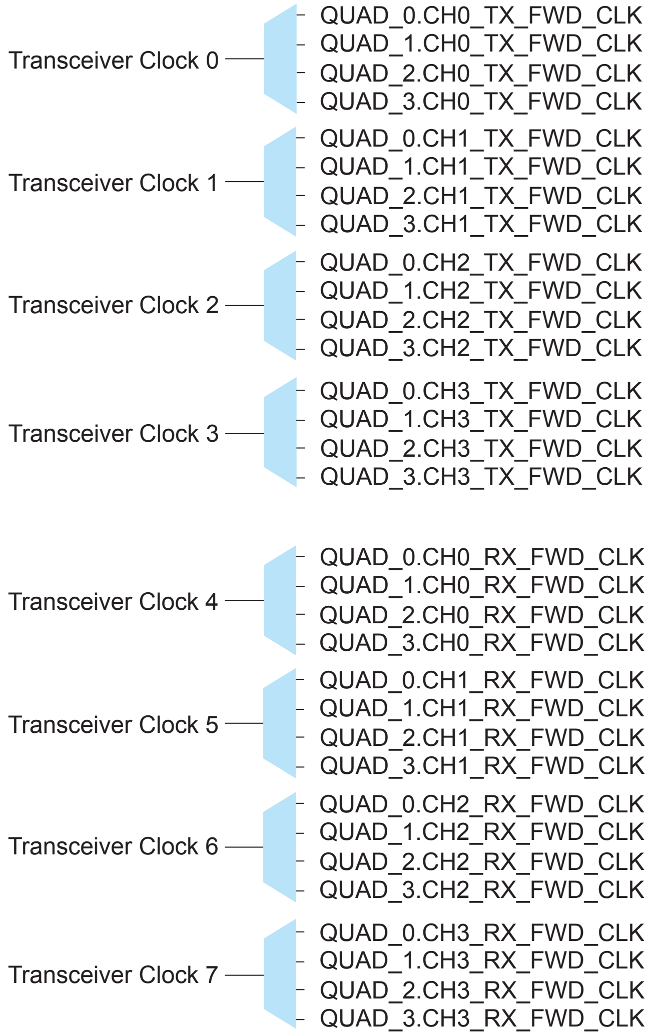
Figure 7: Clock Sources that Drive the Global Network



Note:

1. See Figure 8 on page 16 for the transceiver clocks that can drive the global network.

Figure 8: Transceiver Clock Multiplexers



Numerous clock sources feed the global network. These signals are multiplexed together with static and dynamic clock multiplexers.

The dynamic multiplexers are configurable by the user at run-time. You can choose which clock source drives which input to the dynamic multiplexer. When you enable the dynamic multiplexer, you specify a select bus to choose which clock source is active.

When dynamically switching between the clock inputs of a dynamic multiplexer, both the currently active input and the input you intend to switch to must have toggling clocks during the switching period. Additionally, upon configuration completion and when the device transitions into user mode, input `0` of the dynamic multiplexer becomes the default active input. Therefore, you must feed a toggling clock to input `0` before switching to other inputs.

The following figures show the resources that drive each multiplexer.

Figure 9: Clock Sources that Drive the Multiplexers: Top

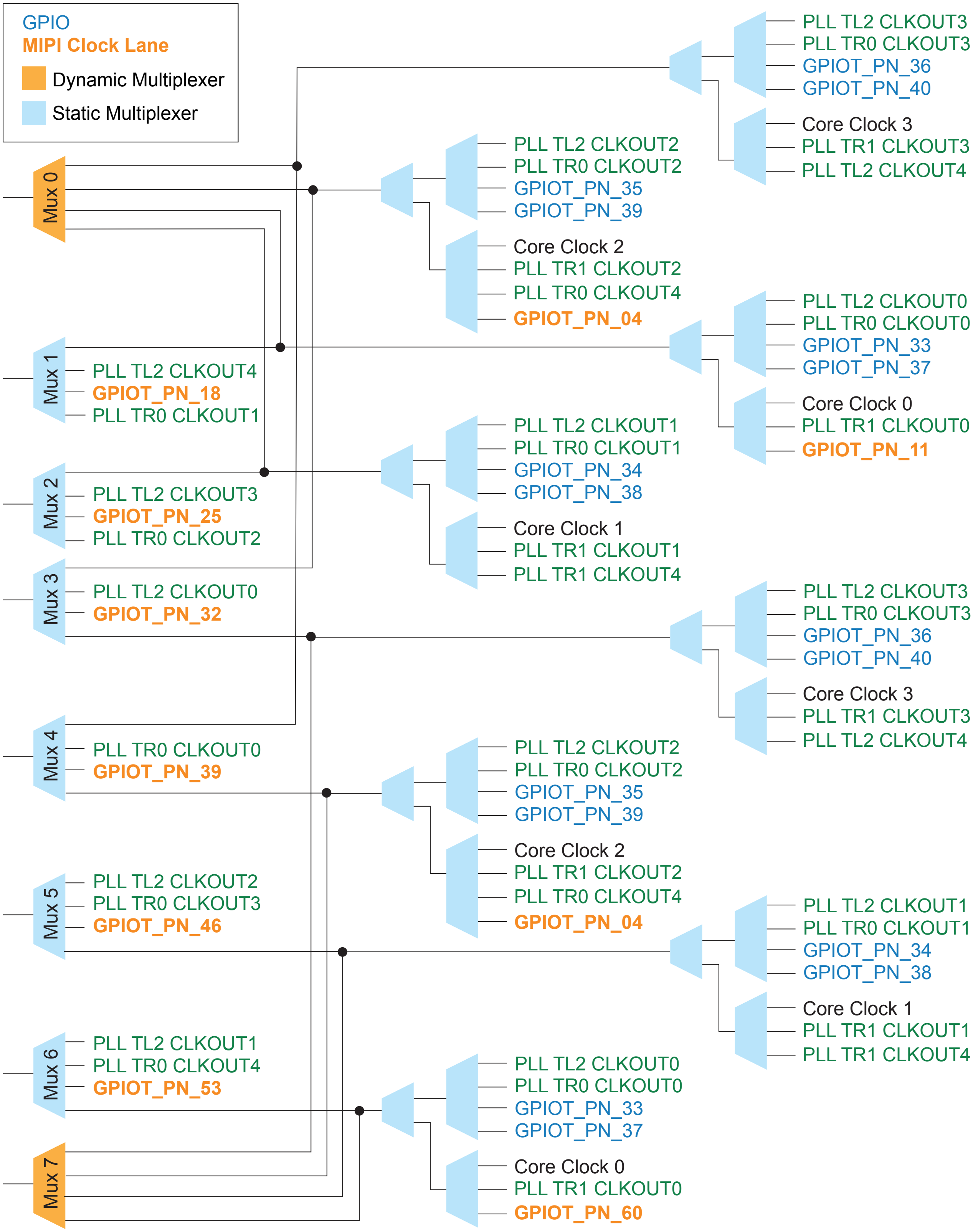


Figure 10: Clock Sources that Drive the Multiplexers: Bottom

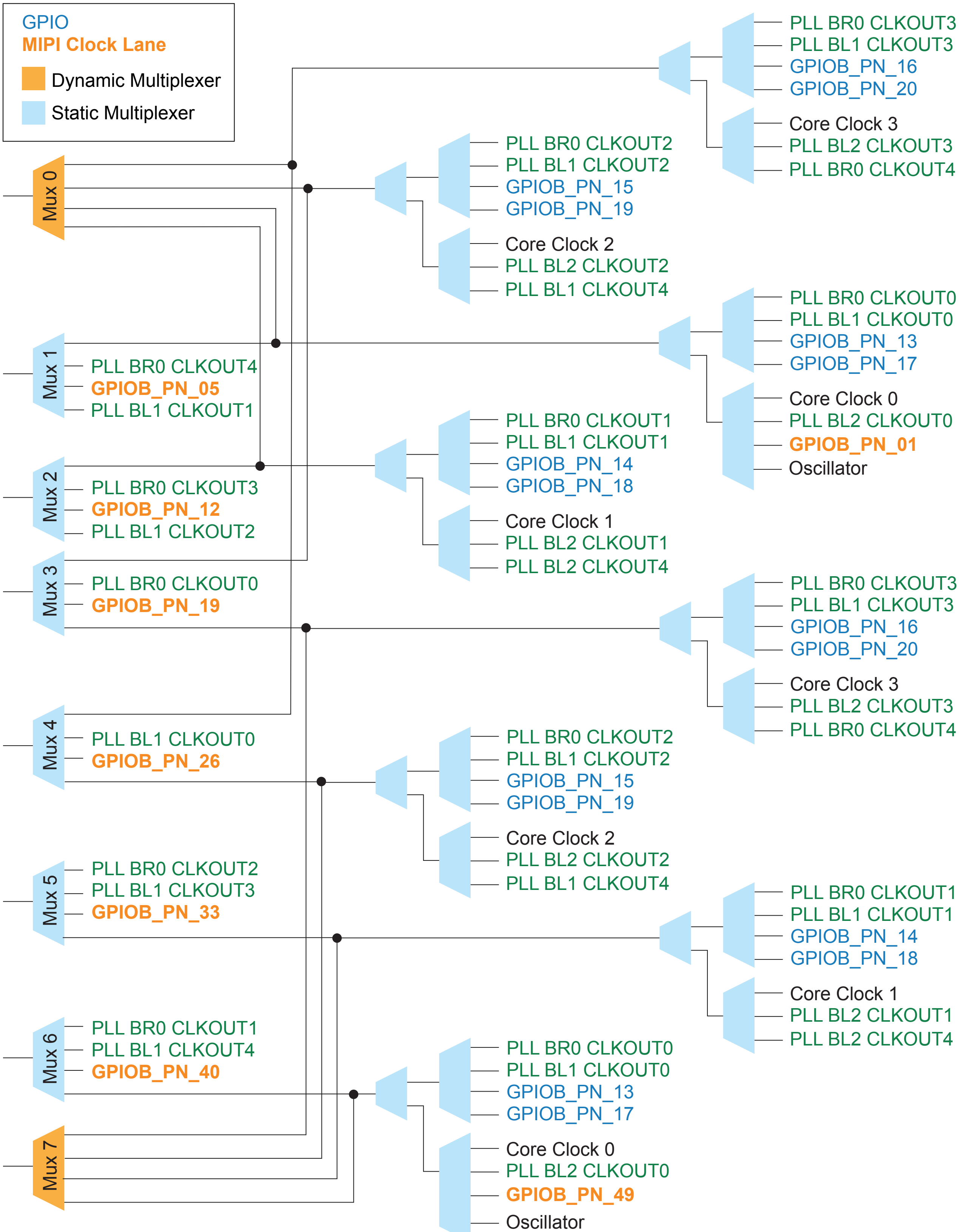


Figure 11: Clock Sources that Drive the Multiplexers: Left

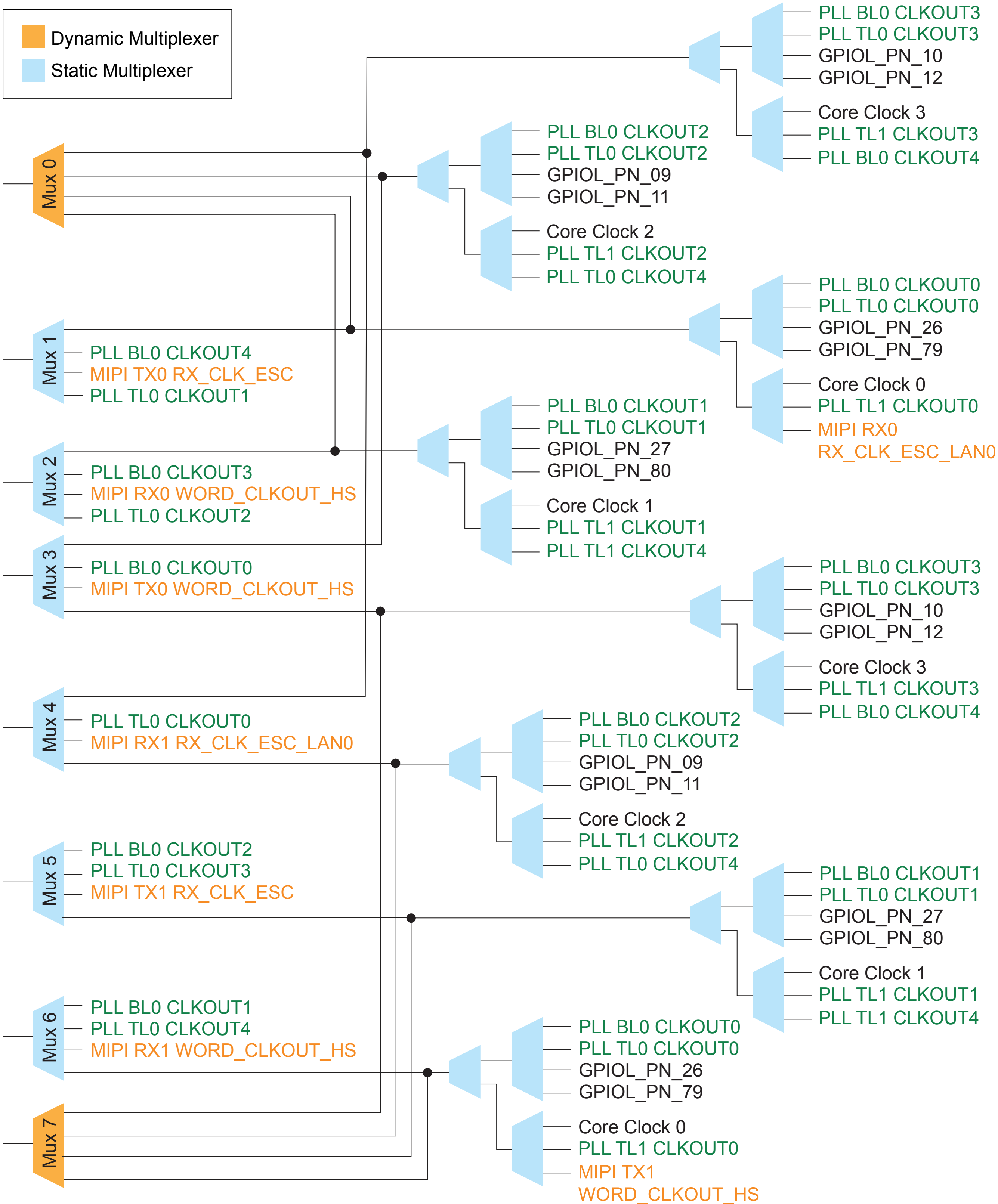
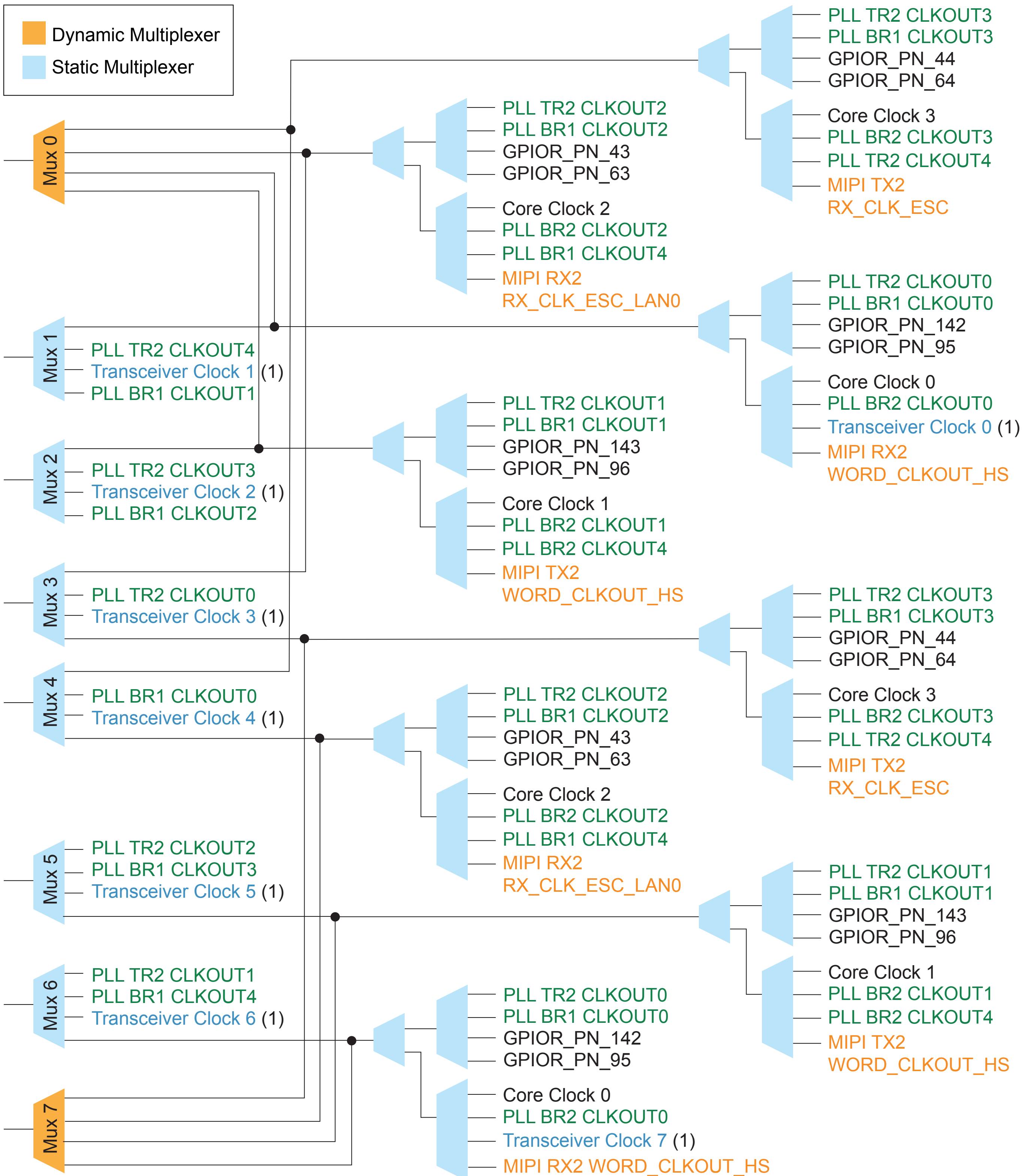


Figure 12: Clock Sources that Drive the Multiplexers: Right



Note:

See **Figure 8** on page 16 for the transceiver clocks that can drive the global network.

Driving the Regional Network

The following figures show the regional network clock sources graphically.

Figure 13: Clock Sources that Drive the Regional Network: Top and Bottom

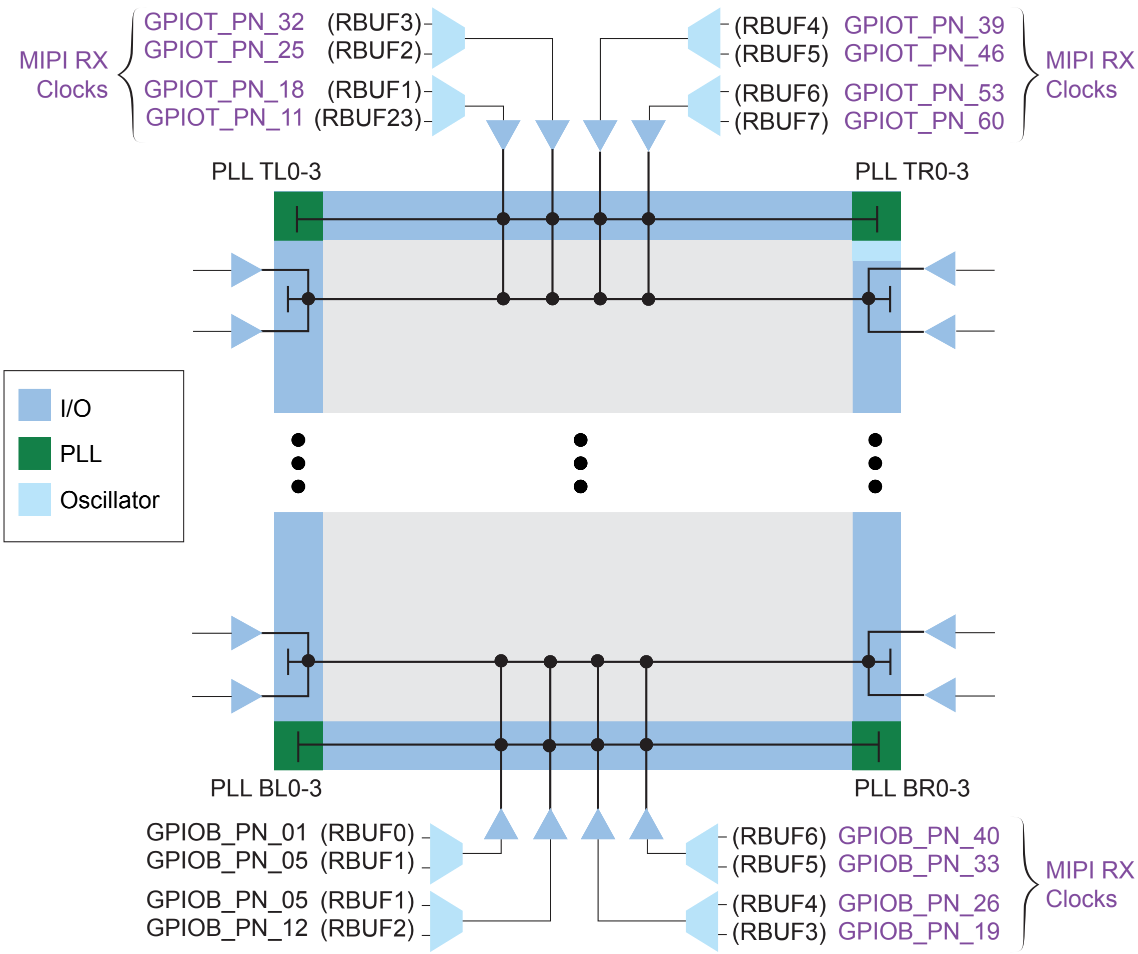


Figure 14: Clock Sources that Drive the Regional Network: Left

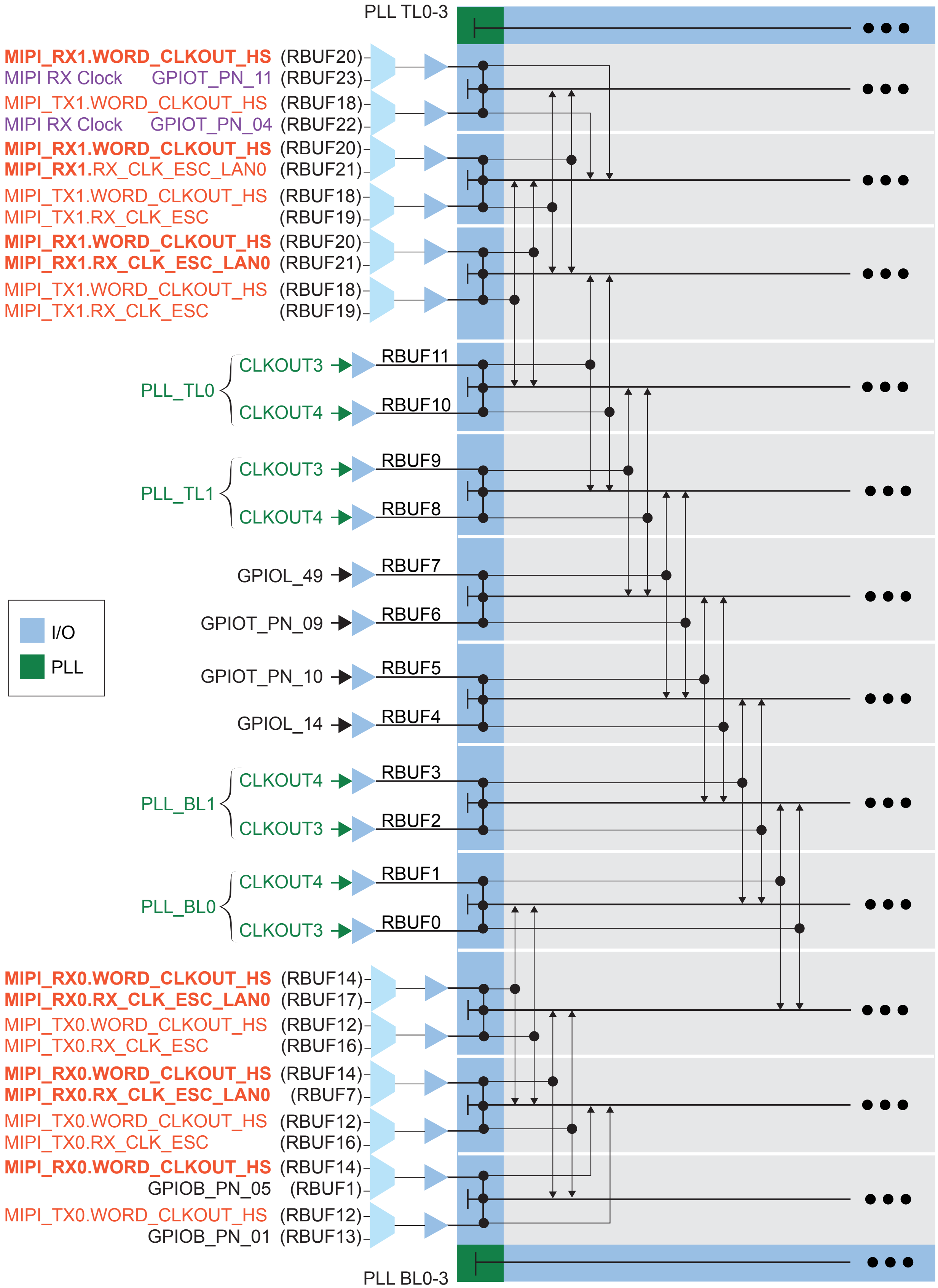
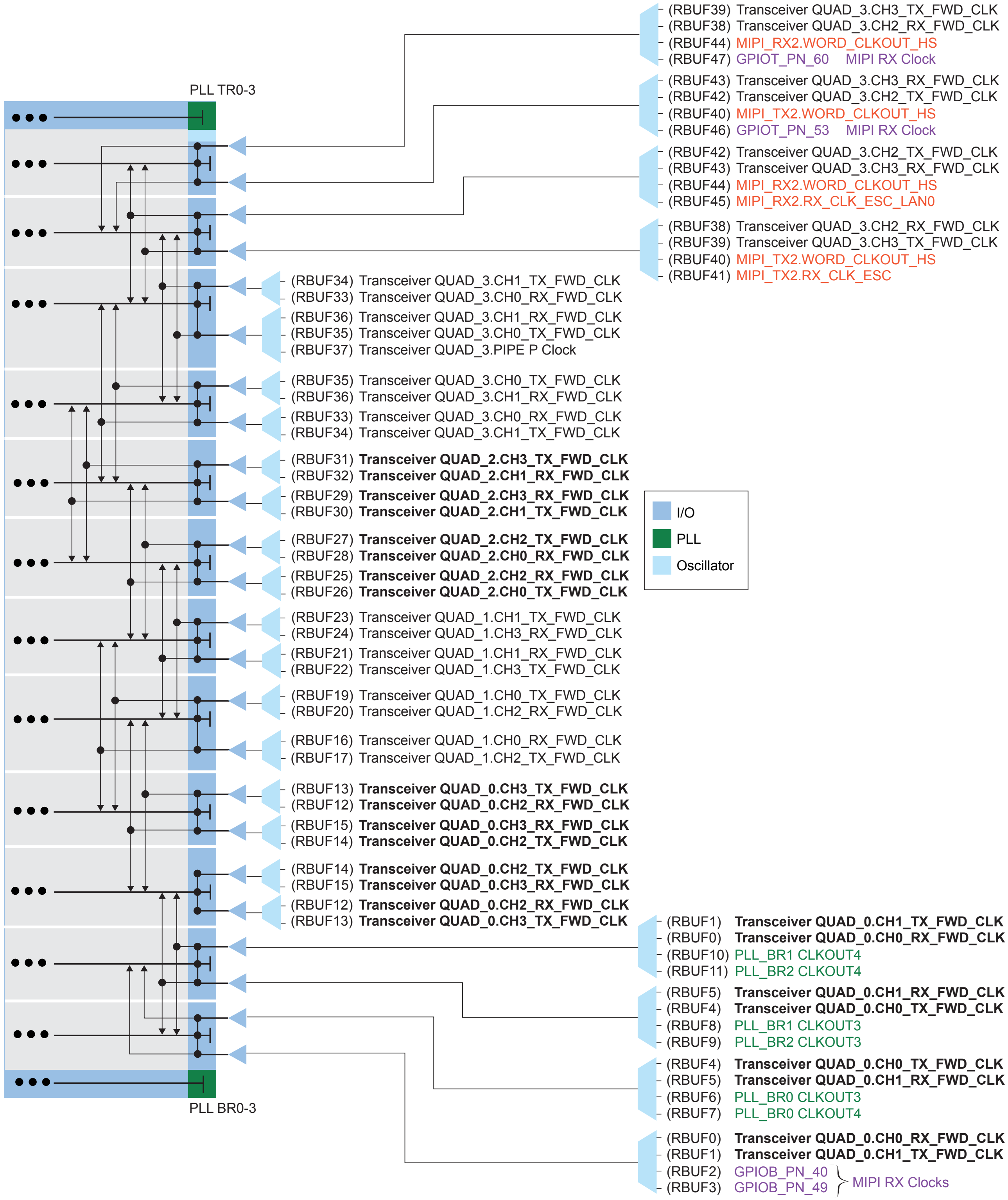


Figure 15: Clock Sources that Drive the Regional Network: Right



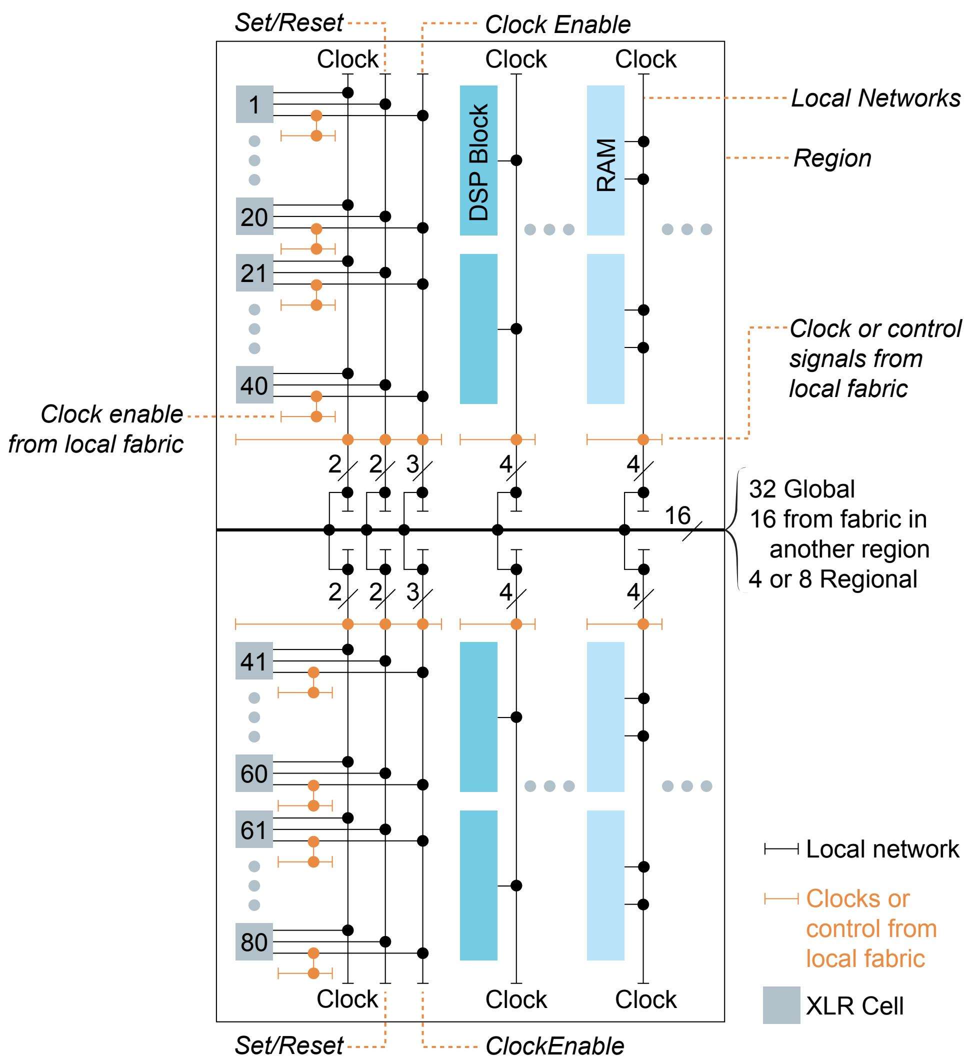
Driving the Local Network

As described previously, the FPGA has horizontal clock regions. The top and bottom regions are **only** for the top and bottom interfaces. The other regions are for the core logic (XLR cells, DSP Blocks, and RAM) and the interfaces on the sides.

Local Network for Core Logic

As shown in the following figure, the regions that contain the core logic are 80 XLR cells tall, and the local network connects an area that is 40 XLR cells tall. Additionally, each column has it's own local network. For example, in the first column, XLR cells 1 - 40 are in the same local network and XLR cells 41 - 80 are in another local network. DSP Blocks and RAM also have their own local networks. This pattern of block/local network is repeated for each column in the die.

Figure 16: Clock Sources for Logic, DSP Blocks, and RAM



There are 16 signals that can feed the local networks. These signals can come from several sources:

- The global network (32 possible signals)
- The core fabric in another region (16 possible signals)
- The regional network (12 possible signals)

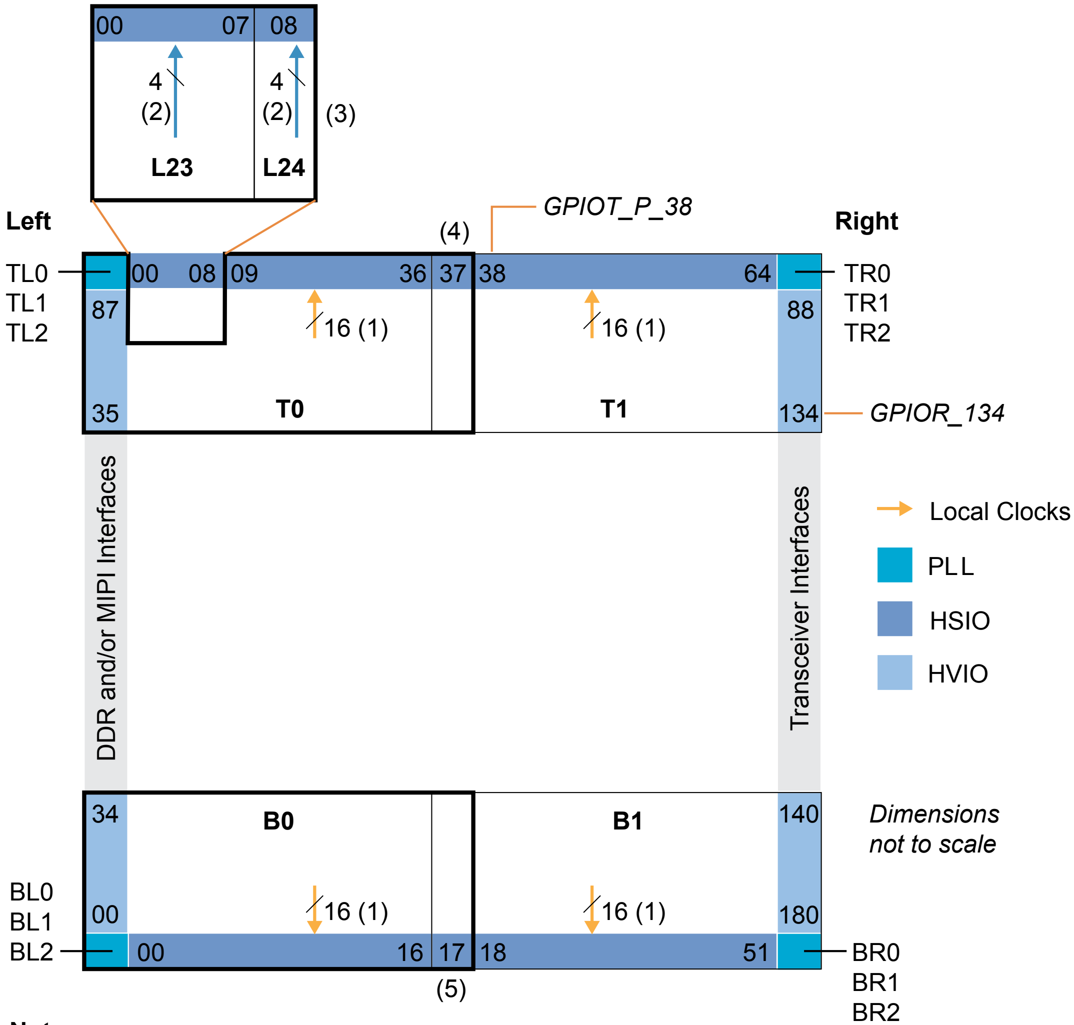
Additionally, the local fabric can generate clock and control signals for the local network. The fabric can also drive the clock enable for the XLR cell directly, allowing each XLR cell to have a unique clock enable.

Local Network for Interface Regions

The following figure shows the local clock networks for the interface blocks. There are a limited number of unique clocks per local clock region.

- The top and bottom regions can each support up to 16 unique clock signals; 14 from the global network and 2 from the fabric.
- The left and right regions can each support up to 4 unique clock signals. Up to 2 can come from the routing fabric, the rest come from the global or regional buffers. These regions are the same height as the core local regions (that is, 40 rows).

Figure 17: Clock Sources that Drive the Interfaces



Notes:
1. 14 signals come from the global network; 2 come from the routing fabric.
2. Up to 2 signals can come from the routing fabric. The rest come from the regional/global buffer.
3. GPIOT_PN_00 - 07 are in region L23. GPIOT_PN_08 is in regions L23 and L24.
4. GPIOT_PN_37 is in regions T0 and T1.
5. GPIOB_PN_17 is in regions B0 and B1.

Device Interface Functional Description

The device interface wraps the core and routes signals between the core and the device I/O pads through a signal interface. Because they use the flexible Quantum® architecture, devices in the Topaz family support a variety of interfaces to meet the needs of different applications.

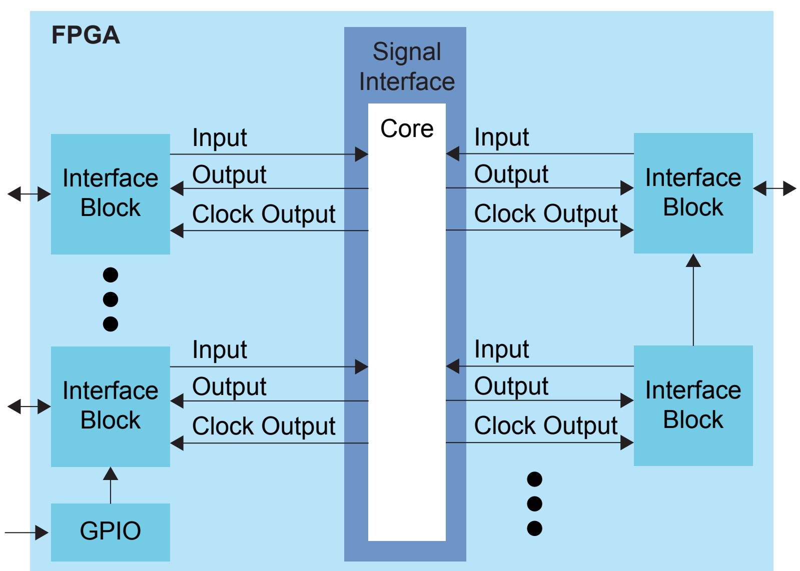
Learn more: The following sections describe the available device interface features in Tz325 FPGAs. Refer to the **Topaz Interfaces User Guide** for details on the Efinity® Interface Designer settings.

Interface Block Connectivity

The FPGA core fabric connects to the interface blocks through a signal interface. The interface blocks then connect to the package pins. The core connects to the interface blocks using three types of signals:

- *Input*—Input data or clock to the FPGA core
- *Output*—Output from the FPGA core
- *Clock output*—Clock signal from the core clock tree

Figure 18: Interface Block and Core Connectivity



GPIO blocks are a special case because they can operate in several modes. For example, in alternate mode the GPIO signal can bypass the signal interface and directly feed another interface block. So a GPIO configured as an alternate input can be used as a PLL reference clock without going through the signal interface to the core.

When designing for Topaz FPGAs, you create an RTL design for the core and also configure the interface blocks. From the perspective of the core, outputs from the core are inputs to the interface block and inputs to the core are outputs from the interface block.

The Efinity netlist always shows signals from the perspective of the core, so some signals do not appear in the netlist:

- GPIO used as reference clocks are not present in the RTL design, they are only visible in the interface block configuration of the Efinity® Interface Designer.

- The FPGA clock tree is connected to the interface blocks directly. Therefore, clock outputs from the core to the interface are not present in the RTL design, they are only part of the interface configuration (this includes GPIO configured as output clocks).

The following sections describe the different types of interface blocks. Signals and block diagrams are shown from the perspective of the interface, not the core.

GPIO

The Tz325 FPGA supports two types of GPIO:
- *High-voltage I/O (HVIO)*—Simple I/O blocks that can support single-ended I/O standards.
- *High-speed I/O (HSIO)*—Complex I/O blocks that can support single-ended and differential I/O functionality.

The I/O logic comprises three register types:
- *Input*—Capture interface signals from the I/O before being transferred to the core logic
- *Output*—Register signals from the core logic before being transferred to the I/O buffers
- *Output enable*—Enable and disable the I/O buffers when I/O used as output

The HVIO supports the following I/O standards.

Table 5: HVIO Supported Standards

Standard	VCCIO33 (V)	When Configured As
LVTTL 3.3 V	3.3	GPIO
LVTTL 3.0 V	3.0	GPIO
LVCMOS 3.3 V	3.3	GPIO
LVCMOS 3.0 V	3.0	GPIO
LVCMOS 2.5 V	2.5	GPIO
LVCMOS 1.8 V	1.8	GPIO

Important: Efinix recommends that you limit the number of 3.0/3.3 V HVIO and 2.5 V HVIO as bidirectional or output to 6 per bank to avoid switching noise. The Efinity® software issues a warning if you exceed the recommended limit.

The HSIO supports the following I/O standards.

Table 6: HSIO Supported I/O Standards

Standard	VCCIO (V)		VCCAUX (V)	VREF (V)	When Configured As
	TX	RX			
LVCMOS 1.8 V	1.8	1.8	1.8	-	GPIO
LVCMOS 1.5 V	1.5	1.5	1.8	-	GPIO
LVCMOS 1.2 V	1.2	1.2	1.8	-	GPIO
HSTL/Differential HSTL 1.8 V SSTL/Differential SSTL 1.8 V	1.8	1.8	1.8	0.9	GPIO
HSTL/Differential HSTL 1.5 V SSTL/Differential SSTL 1.5 V	1.5	1.5, 1.8[3]	1.8	0.75	GPIO

(3) To prevent pin leakage, you must ensure that the voltage at the pin does not exceed VCCIO.

Standard	VCCIO (V)		VCCAUX (V)	VREF (V)	When Configured As
	TX	RX			
SSTL/Differential SSTL 1.35 V	1.35	1.35, 1.5, 1.8(3)	1.8	0.675	GPIO
HSTL/Differential HSTL 1.2 V SSTL/Differential SSTL 1.2 V	1.2	1.2, 1.35, 1.5, 1.8(3)	1.8	0.6	GPIO
LVDS/RSDS/mini-LVDS	1.8	1.5, 1.8(3)	1.8	-	LVDS
Sub-LVDS	1.8	1.5, 1.8(3)	1.8	-	LVDS
MIPI	1.2	1.2	1.8	-	MIPI Lane
SLVS	1.2	1.2	1.8	-	LVDS

The differential receivers are powered by VCCAUX, which gives you the flexibility to choose the VCCIO you want to use. However, you must comply to the requirements stated in the previous table.

Features for HVIO and HSIO Configured as GPIO

The following table describes the features for HVIO and HSIO configured as GPIO.

Table 7: Features for HVIO and HSIO Configured as GPIO

Feature	HVIO	HSIO Configured as GPIO
Double-data I/O (DDIO)	✓	✓
Dynamic pull-up	-	✓
Pull-up/Pull-down	✓	✓
Slew-Rate Control	-	✓
Variable Drive Strength	✓	✓
Schmitt Trigger	✓	✓
1:4 Serializer/Deserializer (Full rate mode only)	-	✓
Programmable Bus Hold	-	✓
Static Programmable Delay Chains	✓	✓
Dynamic Programmable Delay Chains	-	✓

Table 8: GPIO Modes

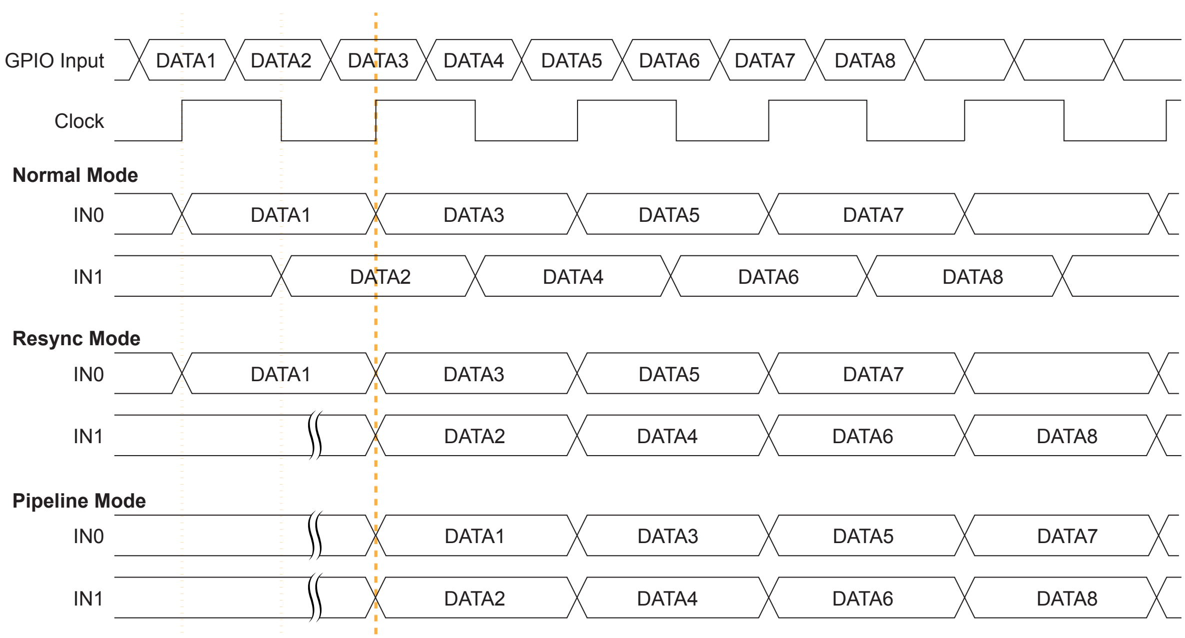
GPIO Mode	Description
Input	Only the input path is enabled; optionally registered. If registered, the input path uses the input clock to control the registers (positively or negatively triggered). Select the alternate input path to drive the alternate function of the GPIO. The alternate path cannot be registered. In DDIO mode, two registers sample the data on the positive and negative edges of the input clock, creating two data streams.
Output	Only the output path is enabled; optionally registered. If registered, the output path uses the output clock to control the registers (positively or negatively triggered). The output register can be inverted. In DDIO mode, two registers capture the data on the positive and negative edges of the output clock, multiplexing them into one data stream.
Bidirectional	The input, output, and OE paths are enabled; optionally registered. If registered, the input clock controls the input register, the output clock controls the output and OE registers. All registers can be positively or negatively triggered. Additionally, the input and output paths can be registered independently. The output register can be inverted.
Clock output	Clock output path is enabled.

Double-Data I/O

Tz325 FPGAs support double data I/O (DDIO) on input and output registers. In this mode, the DDIO register captures data on both positive and negative clock edges. The core receives 2 bit wide data from the interface.

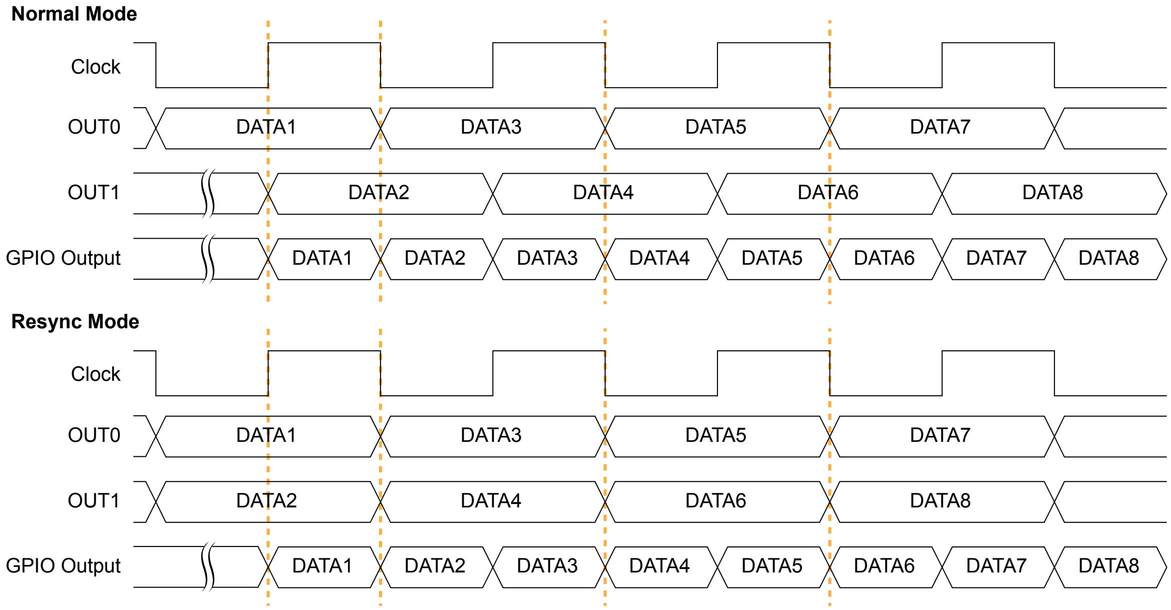
In normal mode, the interface receives or sends data directly to or from the core on the positive and negative clock edges. In resync and pipeline mode, the interface resynchronizes the data to pass both signals on the positive clock edge only.

Figure 19: DDIO Input Timing Waveform



In resync mode, the IN1 data captured on the falling clock edge is delayed one half clock cycle.
In the Interface Designer, IN0 is the HI pin name and IN1 is the LO pin name.

Figure 20: DDIO Output Timing Waveform



In the Interface Designer, OUT0 is the HI pin name and OUT1 is the LO pin name.

Programmable Delay Chains

The HVIO and HSIO configured as GPIO support programmable delay chain. In some cases you can use static and dynamic delays at the same time.

Table 9: Programmable Delay Support

GPIO Type	Delay Steps	
	Static Delay	Dynamic Delay
Single-Ended		
HVIO input	16	N/A
HVIO output	16	N/A
HSIO P pin input	16	64
HSIO P pin output	16	N/A
HSIO N pin input	16	N/A
HSIO N pin output	16	N/A
Differential		
HSIO TX	64	N/A
HSIO RX	64[4]	64[4]

Learn more: Refer to the following tables for the delay step size:
Table 91: Single-Ended I/O Programmable Delay Chain Step Size: Static on page 97
Table 92: Single-Ended I/O Programmable Delay Chain Step Size: Dynamic on page 97
Table 93: Differential I/O Programmable Delay Chain Step Size: Static and Dynamic on page 97

[4] You cannot use the static delay and dynamic delay simultaneously.

HVIO

The HVIOs are grouped into banks. Each bank has its own VCCIO33 that sets the bank voltage for the I/O standard. Each HVIO consists of I/O logic and an I/O buffer. I/O logic connects the core logic to the I/O buffers. I/O buffers are located at the periphery of the device.

Figure 21: HVIO Interface Block

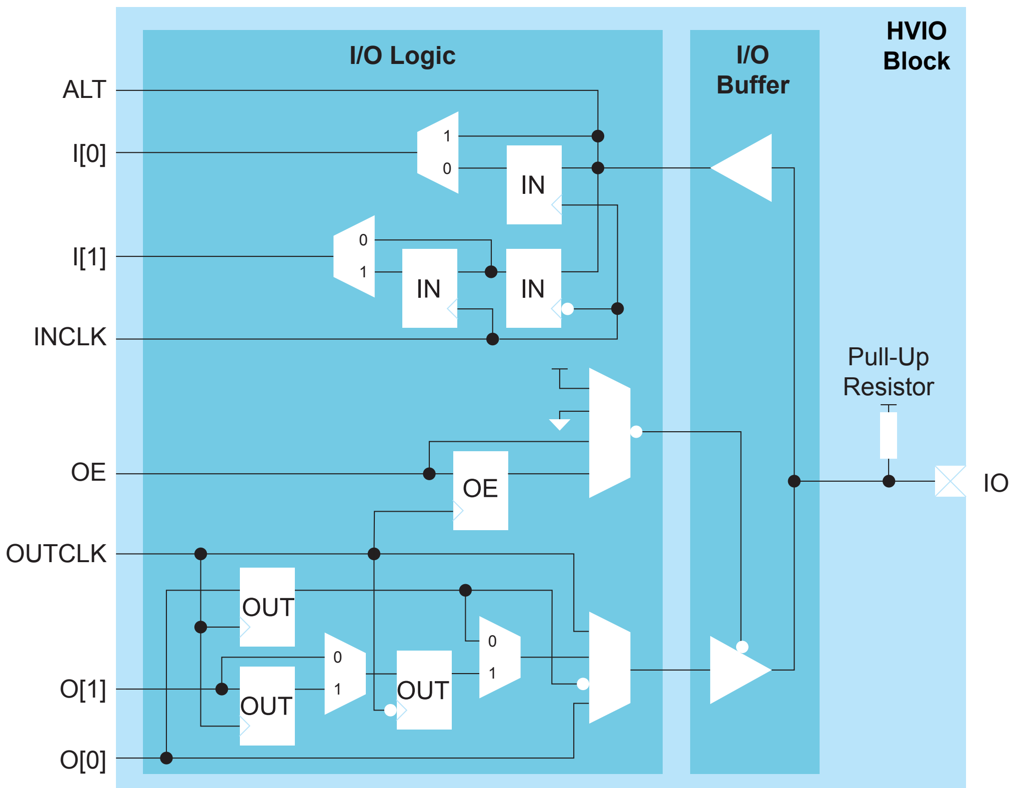


Table 10: HVIO Signals (Interface to FPGA Fabric)

Signal	Direction	Description
I[1:0]	Output	Input data from the HVIO pad to the core fabric. I[0] is the normal input to the core. In DDIO mode, I[0] is the data captured on the positive clock edge (HI pin name in the Interface Designer) and I[1] is the data captured on the negative clock edge (LO pin name in the Interface Designer).
ALT	Output	Alternative input connection (in the Interface Designer, **Register Option** is **none**). HVIO only support pll_clkin as the alternative connection.
O[1:0]	Input	Output data to HVIO pad from the core fabric. O[0] is the normal output from the core. In DDIO mode, O[0] is the data captured on the positive clock edge (HI pin name in the Interface Designer) and O[1] is the data captured on the negative clock edge (LO pin name in the Interface Designer).
OE	Input	Output enable from core fabric to the I/O block. Can be registered.
OUTCLK	Input	Core clock that controls the output and OE registers. This clock is not visible in the user netlist.
INCLK	Input	Core clock that controls the input registers. This clock is not visible in the user netlist.

Table 11: HVIO Pads

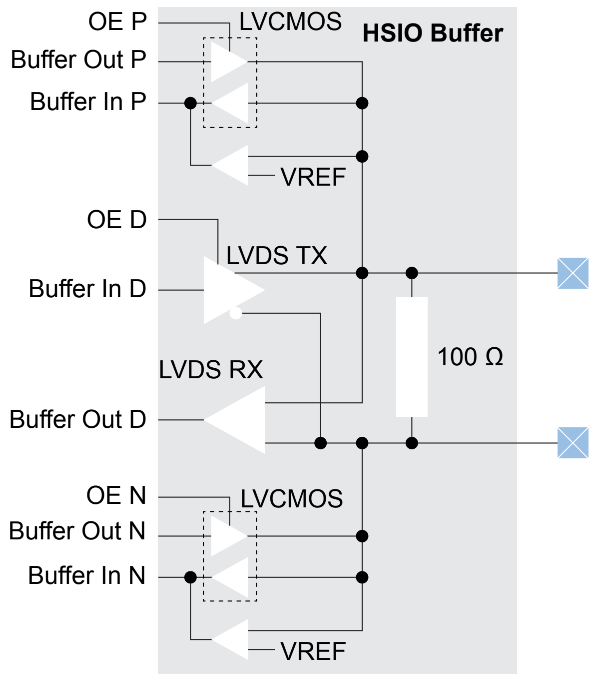
Signal	Direction	Description
IO	Bidirectional	HVIO pad.

HSIO

Each HSIO block uses a pair of I/O pins as one of the following:

- *Single-ended HSIO*—Two single-ended I/O pins (LVCMOS, SSTL, HSTL)
- *Differential HSIO*—One differential I/O pins:
  - Differential SSTL and HSTL
  - LVDS—Receiver (RX), transmitter (TX), or bidirectional (RX/TX)
  - MIPI lane I/O—Receiver (RX) or transmitter (TX)

Figure 22: HSIO Buffer Block Diagram



Important: When you are using an HSIO pin as a GPIO, make sure to leave at least 1 pair of unassigned HSIO pins between any GPIO and LVDS or MIPI lane pins. This rule applies for pins on each side of the device (top, bottom, left, right). This separation reduces noise. The Efinity software issues an error if you do not leave this separation.

HSIO Configured as GPIO

You can configure each HSIO block as two GPIO (single-ended) or one GPIO (differential).

Figure 23: I/O Interface Block

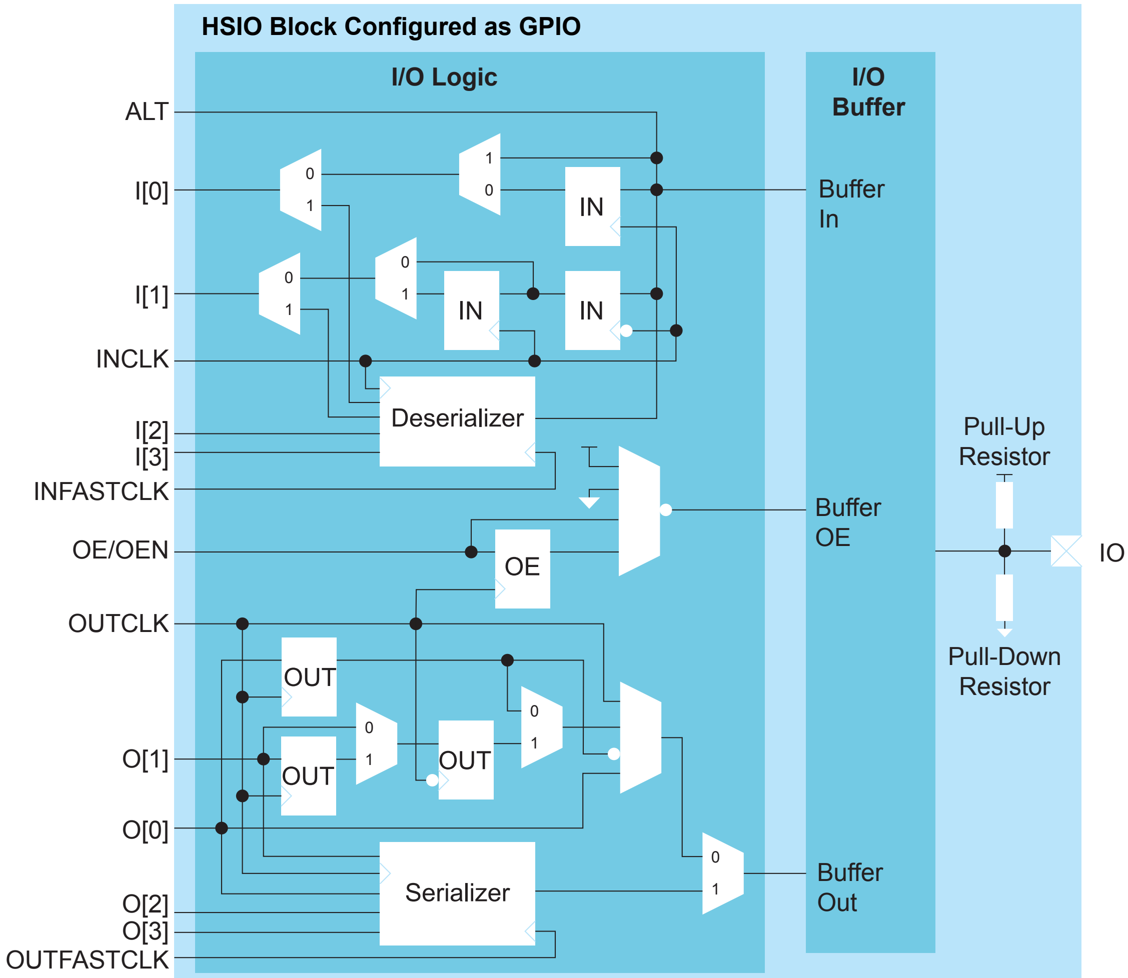


Table 12: HSIO Block Configured as GPIO Signals (Interface to FPGA Fabric)

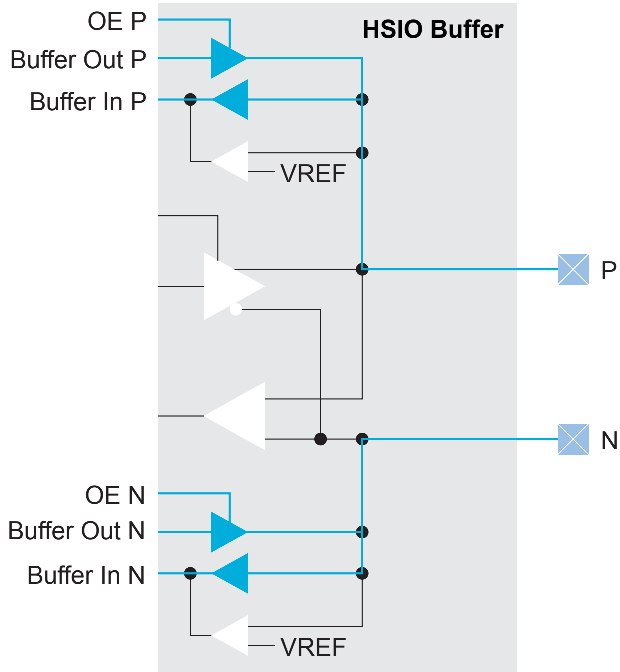
Signal	Direction	Description
I[3:0]	Output	Input data from the pad to the core fabric. I[0] is the normal input to the core. In DDIO mode, I[0] is the data captured on the positive clock edge (HI pin name in the Interface Designer) and I[1] is the data captured on the negative clock edge (LO pin name in the Interface Designer). When using the deserializer, the first bit is on I[0] and the last bit is on I[3].
ALT	Output	Alternative input connection for GCLK, PLL_CLKIN, RCLK, PLL_EXTFB, and VREF. (In the Interface Designer, **Register Option** is **none**).
O[3:0]	Input	Output data to GPIO pad from the core fabric. O[0] is the normal output from the core. In DDIO mode, O[0] is the data output on the positive clock edge (HI pin name in the Interface Designer) and O[1] is the data output on the negative clock edge (LO pin name in the Interface Designer). When using the serializer, the first bit is on O[0] and the last bit is on O[3].
OE/OEN	Input	Output enable from core fabric to the I/O block. Can be registered. OEN is used in differential mode. Drive it with the same signal as OE.
DLYCLK	Input	Delay clock for dynamic delay, sampled on the negative edge. In serializer mode, this clock must be the same clock as INCLK.
DLY_ENA	Input	(Optional) Enable the dynamic delay control.
DLY_INC	Input	(Optional) Dynamic delay control. When DLY_ENA = 1, 1: Increments 0: Decrements The updated delay count takes effect approximately 5 ns after the rising edge of the clock.
DLY_RST	Input	(Optional) Reset the delay counter.
OUTCLK	Input	Core clock that controls the output and OE registers. This clock is not visible in the user netlist.
OUTFASTCLK	Input	Core clock that controls the output serializer.
INCLK	Input	Core clock that controls the input registers. This clock is not visible in the user netlist.
INFASTCLK	Input	Core clock that controls the input serializer.

Table 13: GPIO Pads

Signal	Direction	Description
IO (P and N)	Bidirectional	GPIO pad.

The signal path from the pad through the I/O buffer changes depending on the I/O standard you are using. The following figures show the paths for the supported standards. The blue highlight indicates the path.

Figure 24: I/O Buffer Path for LVCMOS

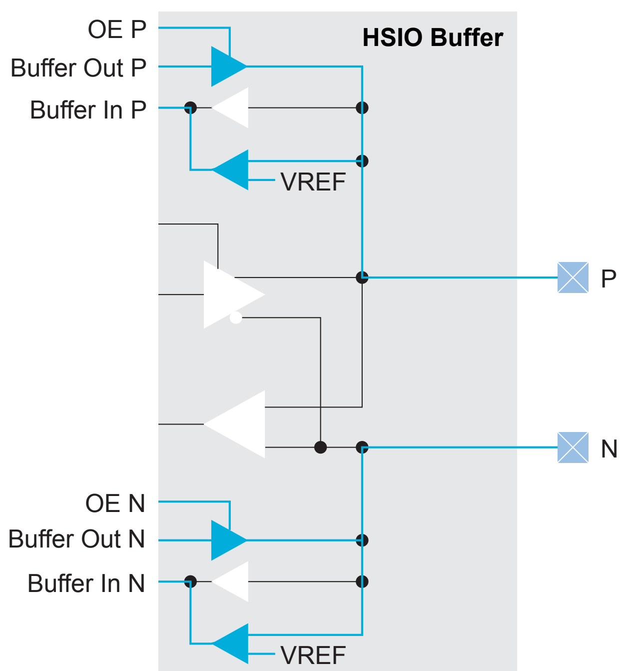


When using an HSIO with the HSTL or SSTL I/O standards, you must configure an I/O pad of the standard's input path as a `VREF` pin. There is one programmable `VREF` per I/O bank.

Important: When configuring an I/O pad of the standard's input path as a `VREF` pin, you must use the `VREF` from the same physical I/O bank even when the I/O banks are merged to share a common VCCIO pin.

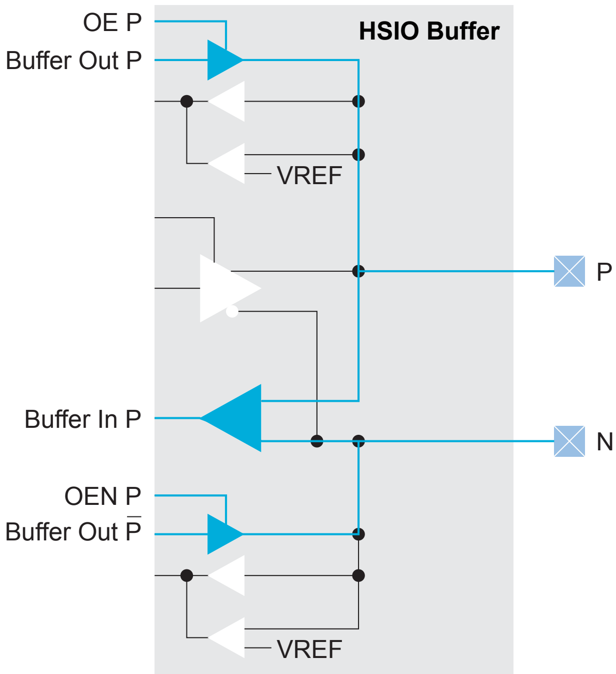
For I/O banks that do not have the `VREF` pin bonded out, the banks' GPIO pins do not support the HSTL and SSTL input I/O standards.

Figure 25: I/O Buffer Path for HSTL and SSTL



When using an HSIO with the differential HSTL or differential SSTL standard, you must use both GPIO resources in the HSIO. You use the core interface pins associated with the P resource.

Figure 26: I/O Buffer Path for Differential HSTL and SSTL



HSIO Configured as LVDS

You can configure each HSIO block in RX, TX, or bidirectional LVDS mode. As LVDS, the HSIO has these features:

- Programmable V_{OD}, depending on the I/O standard used.
- Programmable pre-emphasis.
- Up to 1.3 Gbps.
- Programmable 100 Ω termination to save power (you can enable or disable it at runtime).
- LVDS input enable to dynamically enable/disable the LVDS input.
- Support for full rate or half rate serialization.
- Up to 10-bit serialization to support protocols such as 8b10b encoding.
- Programmable delay chains.
- Optional 8-word FIFO for crossing from the parallel (slow) clock to the user's core clock to help close timing (RX only).
- Dynamic phase alignment (DPA) that automatically eliminates skew for clock to data channels and data to data channels by adjusting a delay chain setting so that data is sampled at the center of the bit period. The DPA supports full-rate serialization mode only.

Table 14: Full and Half Rate Serialization

Mode	Description	Example
Full rate clock	In full rate mode, the fast clock runs at the same frequency as the data and captures data on the positive clock edge.	Data rate: 800 Mbps Serialization/Deserialization factor: 8 Slow clock frequency: 100 Mhz (800 Mbps / 8) Fast clock frequency: 800 Mhz
Half rate clock	In half rate mode, the fast clock runs at half the speed of the data and captures data on both clock edges.	Data rate: 800 Mbps Serialization / Deserialization factor: 8 Slow clock frequency: 100 Mhz (800 Mbps / 8) Fast clock frequency: 400 Mhz (800 / 2)

You use a PLL to generate the serial (fast) and parallel (slow) clocks for the LVDS pins. The slow clock runs at the data rate divided by the serialization factor.

LVDS RX

You can configure an HSIO block as one LVDS RX signal.

Figure 27: LVDS RX Interface Block Diagram

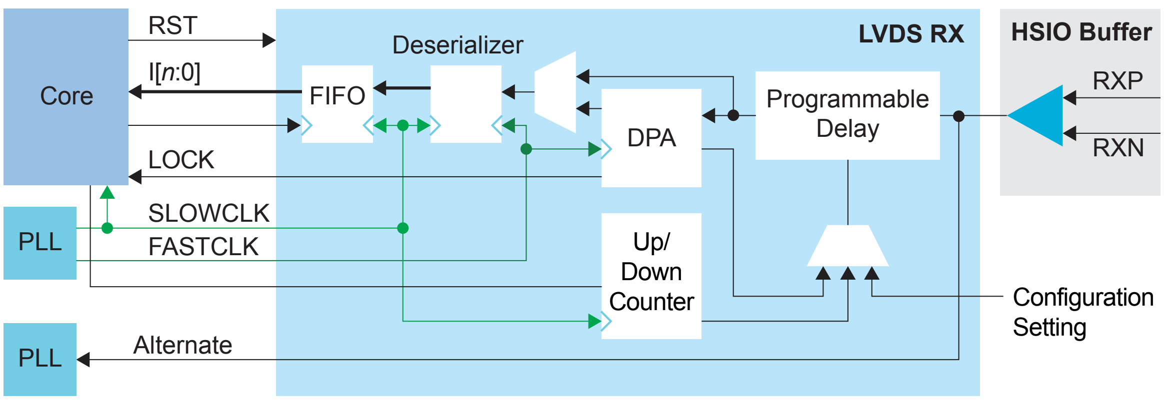
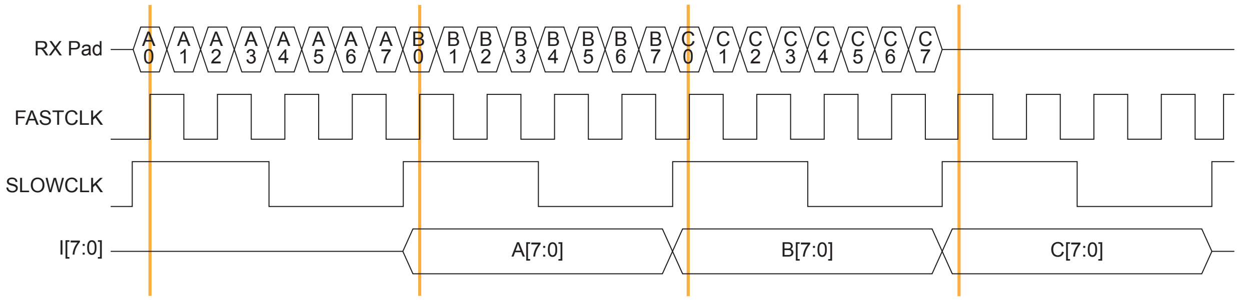


Table 15: LVDS RX Signals (Interface to FPGA Fabric)

Signal	Direction	Clock Domain	Description
I[9:0]	Output	SLOWCLK	Parallel input data to the core. The width is programmable.
ALT	Output		Alternate input, only available for an LVDS RX resource in bypass mode (deserialization width is 1; alternate connection type). Alternate connections are PLL_CLKIN, PLL_EXTFB, GCLK, and RCLK.
SLOWCLK	Input	–	Parallel (slow) clock.
FASTCLK	Input	–	Serial (fast) clock.
FIFO_EMPTY	Output	FIFOCLK	This signal is required when you turn on the **Enable Clock Crossing FIFO** option Indicates that the FIFO is empty.
FIFOCLK	Input	–	This signal is required when you turn on the **Enable Clock Crossing FIFO** option. Core clock to read from the FIFO.
FIFO_RD	Input	FIFOCLK	This signal is required when you turn on the **Enable Clock Crossing FIFO** option. Enables FIFO to read.
RST	Input	FIFOCLK SLOWCLK	(Optional) This signal is available when deserialization is enabled. Asynchronous. Resets the FIFO and deserializer. If the FIFO is enabled, it is relative to FIFOCLK; otherwise it is relative to SLOWCLK.
ENA	Input	–	Dynamically enable or disable the LVDS input buffer. Can save power when disabled. 1: Enabled 0: Disabled
TERM	Input	–	The signal is available when dynamic termination is enabled. Enables or disables termination in dynamic termination mode. 1: Enabled 0: Disabled
LOCK	Output		(Optional) This signal is available when you set **Delay Mode** to **dpa**. Indicates that the DPA has achieved training lock and data can be passed.
DLY_ENA	Input	SLOWCLK	This signal is required when you set **Delay Mode** to **dynamic** or **dpa**. Enable the dynamic delay control or the DPA circuit, depending on the LVDS RX delay settings.
DLY_INC	Input	SLOWCLK	This signal is required when you set **Delay Mode** to **dynamic**. Dynamic delay control. Cannot be used with DPA enabled. When DLY_ENA is 1: 1: Increments 0: Decrements
DLY_RST	Input	SLOWCLK	(Optional) This signal is available when you set **Delay Mode** to **dpa** or **dynamic**. Reset the delay counter or the DPA circuit, depending on the LVDS RX delay settings.
DBG[5:0]	Output	SLOWCLK	DPA debug pin. Outputs the final delay chain settings when DPA achieved lock.

The following waveform shows the relationship between the fast clock, slow clock, RX data coming in from the pad, and byte-aligned data to the core.

Figure 28: LVDS RX Timing Example Serialization Width of 8 (Half Rate)



I is byte-aligned data passed to the core on the rising edge of SLOWCLK.

Note: For LVDS RX interfaces with multiple LVDS RX lanes and an LVDS RX clock input, use the LVDS RX blocks from the same side of the FPGA to minimize skew between data lanes and RX clock input.

LVDS TX

You can configure an HSIO block as one LVDS TX signal. LVDS TX can be used in the serial data output mode or reference clock output mode.

Figure 29: LVDS TX Interface Block Diagram

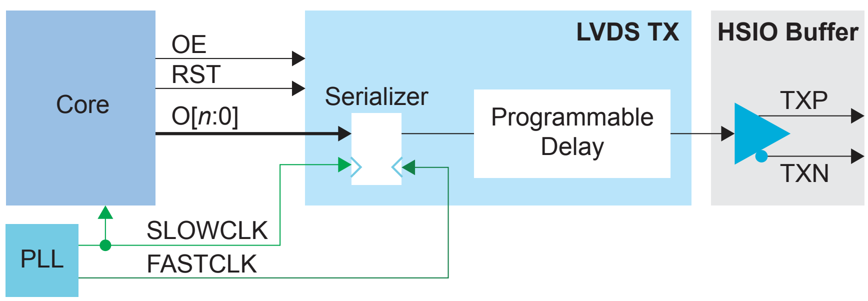
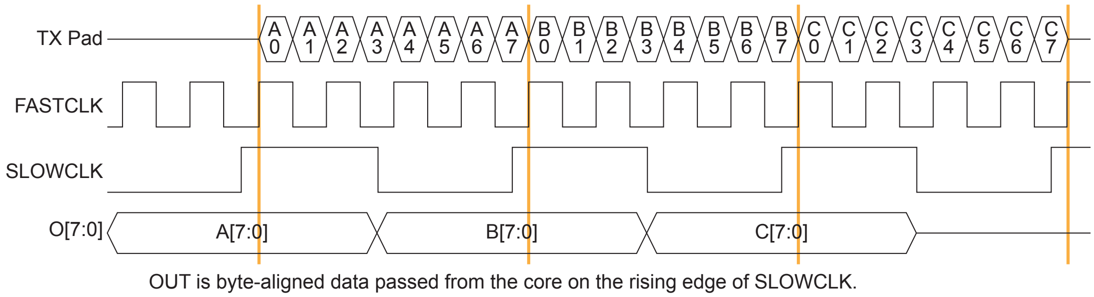


Table 16: LVDS TX Signals (Interface to FPGA Fabric)

Signal	Direction	Clock Domain	Description
O[9:0]	Input	SLOWCLK	Parallel output data from the core. The width is programmable.
SLOWCLK	Input	-	Parallel (slow) clock.
FASTCLK	Input	-	Serial (fast) clock.
RST	Input	SLOWCLK	(Optional) This signal is available when serialization is enabled. Resets the serializer.
OE	Input	-	(Optional) Output enable signal.

The following waveform shows the relationship between the fast clock, slow clock, TX data going to the pad, and byte-aligned data from the core.

Figure 30: LVDS Timing Example Serialization Width of 8 (Half Rate)



Note: For LVDS TX interfaces with multiple LVDS TX lanes and an LVDS TX reference clock output, use the LVDS TX blocks from the same side of the FPGA to minimize skew between data lanes and TX reference clock output.

LVDS Bidirectional

You can configure an HSIO block as one LVDS bidirectional signal. You must use the same serialization for the RX and TX.

Figure 31: LVDS Bidirectional Interface Block Diagram

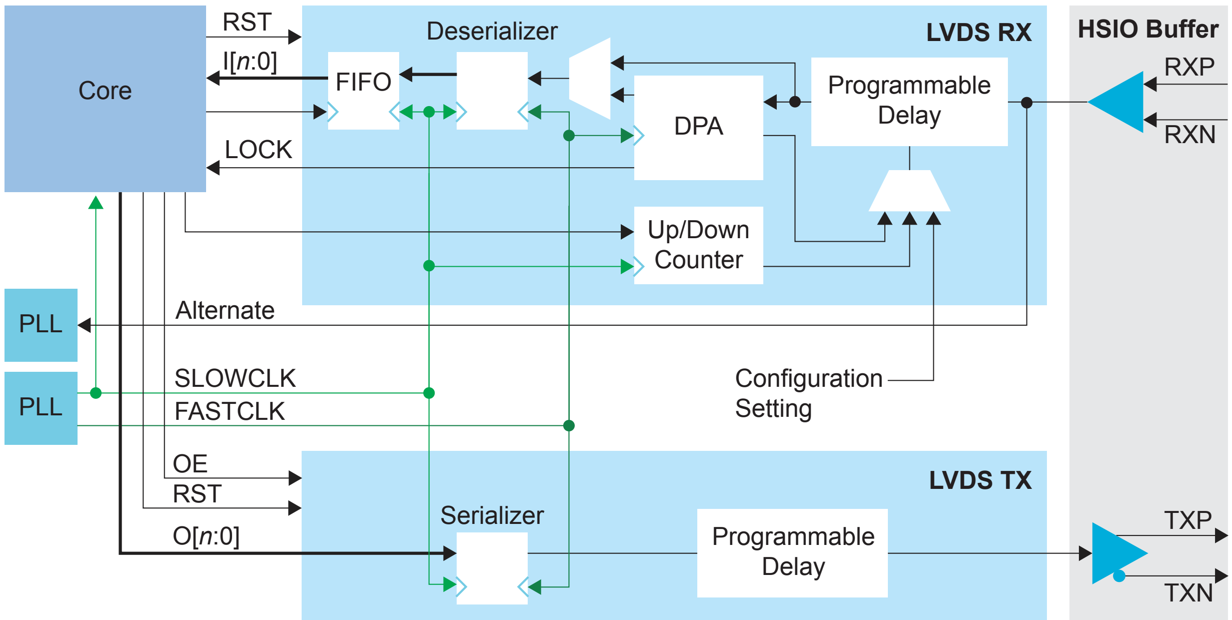


Table 17: LVDS Bidirectional Signals (Interface to FPGA Fabric)

Signal	Direction	Clock Domain	Description
I[9:0]	Output	SLOWCLK	Parallel input data to the core. The width is programmable.
INSLOWCLK	Input	-	Parallel (slow) clock for RX.
INFASTCLK	Input	-	Serial (fast) clock for RX.
FIFO_EMPTY	Output	FIFOCLK	This signal is required when you turn on the **Enable Clock Crossing FIFO** option Indicates that the FIFO is empty.
FIFOCLK	Input	-	This signal is required when you turn on the **Enable Clock Crossing FIFO** option. Core clock to read from the FIFO.
FIFO_RD	Input	FIFOCLK	This signal is required when you turn on the **Enable Clock Crossing FIFO** option. Enables FIFO to read.
INRST	Input	FIFOCLK SLOWCLK	This signal is available when deserialization is enabled. Asynchronous. Resets the FIFO and RX deserializer. If the FIFO is enabled, it is relative to FIFOCLK; otherwise it is relative to SLOWCLK.
ENA	Input	-	Dynamically enable or disable the LVDS input buffer. Can save power when disabled. 1: Enabled 0: Disabled
TERM	Input	-	The signal is available when dynamic termination is enabled. Enables or disables termination in dynamic termination mode. 1: Enabled 0: Disabled

Signal	Direction	Clock Domain	Description
LOCK	Output		(Optional) This signal is available when you set **Delay Mode** to **dpa**. Indicates that the DPA has achieved training lock and data can be passed.
DLY_ENA	Input	SLOWCLK	This signal is required when you set **Delay Mode** to **dynamic** or **dpa**. Enable the dynamic delay control or the DPA circuit, depending on the bidirectional LVDS delay settings.
DLY_INC	Input	SLOWCLK	This signal is required when you set **Delay Mode** to **dynamic**. Dynamic delay control. Cannot be used with DPA enabled. When DLY_ENA is 1, 1: Increments 0: Decrements
DLY_RST	Input	SLOWCLK	(Optional) This signal is available when you set **Delay Mode** to **dpa** or **dynamic**. Reset the delay counter or the DPA circuit, depending on the bidirectional LVDS delay settings.
DBG[5:0]	Output	SLOWCLK	DPA debug pin. Outputs the final delay chain settings when DPA achieved lock.
O[9:0]	Input	SLOWCLK	Parallel output data from the core. The width is programmable.
OUTSLOWCLK	Input	-	Parallel (slow) clock for TX.
OUTFASTCLK	Input	-	Serial (fast) clock for TX.
OUTRST	Input	SLOWCLK	This signal is available when serialization is enabled. Resets the TX serializer.
OE	Input	-	Output enable signal.

LVDS Pads

Table 18: LVDS Pads

Signal	Direction	Description
P	Output	Differential pad P.
N	Output	Differential pad N.

HSIO Configured as MIPI Lane

You can configure the HSIO block as a MIPI RX or TX lane. The block supports bidirectional data lane, unidirectional data lane, and unidirectional clock lane which can run at speeds up to 1.3 Gbps. The MIPI lane operates in high-speed (HS) and low-power (LP) modes. In HS mode, the HSIO block transmits or receives data with x8 serializer/deserializer. In LP mode, it transmits or receives data without deserializer/serializer.

The MIPI lane block does not include the MIPI D-PHY core logic. A full MIPI D-PHY solution requires:
- Multiple MIPI RX or TX lanes (at least a clock lane and a data lane)
- Soft MIPI D-PHY IP core programmed into the FPGA fabric

The MIPI D-PHY standard is a point-to-point protocol with one endpoint (TX) responsible for initiating and controlling communication. Often, the standard is unidirectional, but when implementing the MIPI DSI protocol, you can use one TX data lane for LP bidirectional communication.

The protocol is source synchronous with one clock lane and 1, 2, 4, or 8 data lanes. The number of lanes available depends on which package you are using. A dedicated HSIO block is assigned on the RX interface as a clock lane while the clock lane for TX interface can use any of the HSIO block in the group.

MIPI RX Lane

In RX mode, the HS (fast) clock comes in on the MIPI clock lane and is divided down to generate the slow clock. The fast and slow clocks are then passed to neighboring HSIO blocks to be used for the MIPI data lanes.

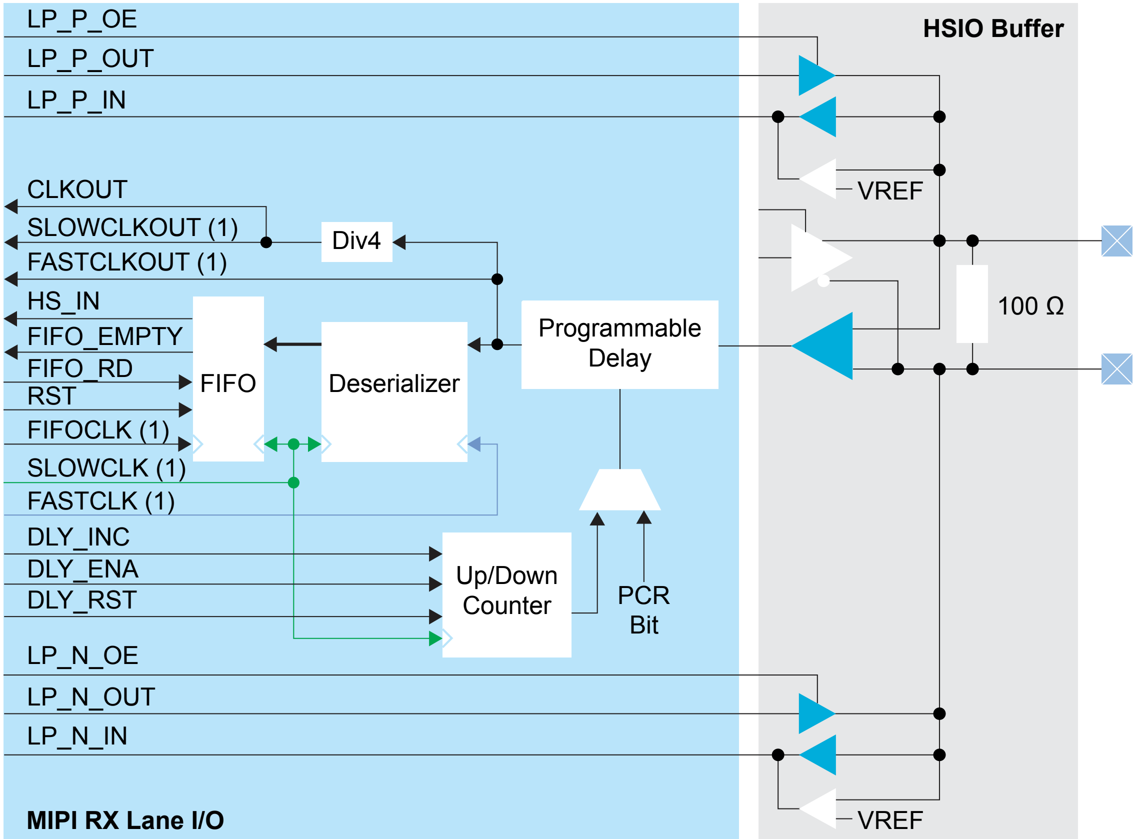
The data lane fast and slow clocks must be driven by a clock lane in the same MIPI group (dedicated buses drive from the clock lane to the neighboring data lanes).

The MIPI RX function is defined as:

Table 19: MIPI RX Function

MIPI RX Function	Description
RX_DATA_*xy*_*zz*	MIPI RX Data Lane. You can use any data lanes within the same group to form multiple lanes of MIPI RX channel. *x* = P or N *y* = 0 to 7 data lanes (Up to 8 data lanes per channel) *zz* = I0 to I17 MIPI RX channel (Up to 18 MIPI RX channels)
RX_CLK_*x*_*zz*	MIPI RX Clock Lane. One clock lane is required for each MIPI RX channel. *x* = P or N *zz* = I0 to I17 MIPI RX channel (Up to 18 MIPI RX channels)

Figure 32: MIPI RX Lane Block Diagram



1. These signals are in the primitive, but the software automatically connects them for you.

Table 20: MIPI RX Lane Signals

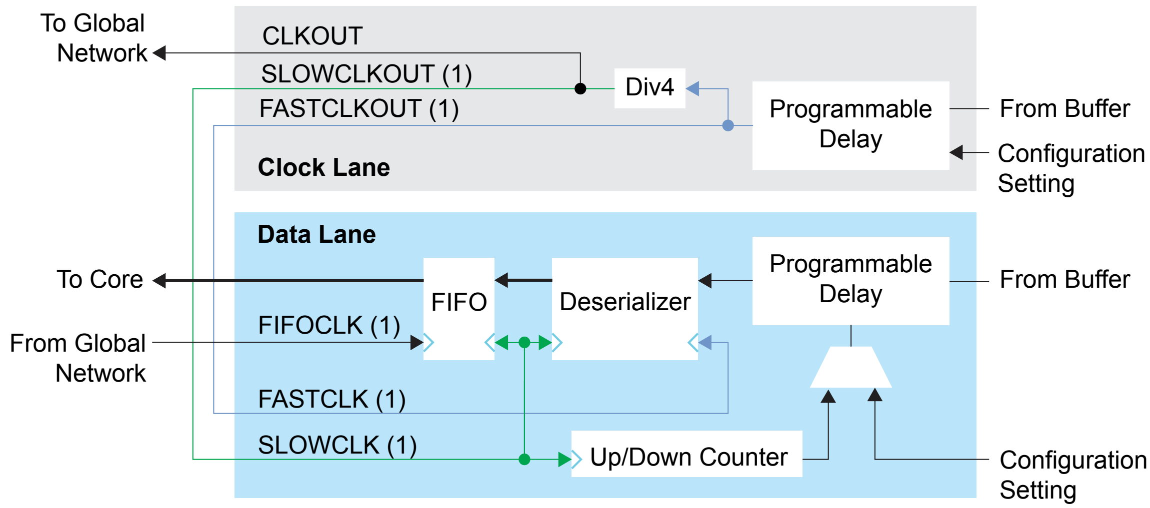
Interface to MIPI soft CSI/DSI controller with D-PHY in FPGA Fabric

Signal	Direction	Clock Domain	Description
LP_P_OE	Input	-	(Optional) LP output enable signal for P pad.
LP_P_OUT	Input	-	(Optional) LP output data from the core for the P pad. Used if the data lane is reversible.
LP_P_IN	Output	-	LP input data from the P pad.
CLKOUT	Output	-	Divided down parallel (slow) clock from the pads that can drive the core clock tree. Used to drive the core logic implementing the rest of the D-PHY protocol. It should also connect to the FIFOCLK of the data lanes.
SLOWCLKOUT(5)	Output	-	Divided down parallel (slow) clock from the pads. Can only drive RX DATA lanes.
FASTCLKOUT(5)	Output	-	Serial (fast) clock from the pads. Can only drive RX DATA lanes.
HS_IN[7:0]	Output	SLOWCLK	High-speed parallel data input.
FIFO_EMPTY	Output	FIFOCLK	(Optional) When the FIFO is enabled, this signal indicates that the FIFO is empty.
FIFO_RD	Input	FIFOCLK	(Optional) Enables FIFO to read.
RST	Input	FIFOCLK SLOWCLK	(Optional) Asynchronous. Resets the FIFO and serializer. If the FIFO is enabled, it is relative to FIFOCLK; otherwise it is relative to SLOWCLK.
FIFOCLK(5)	Input	-	(Optional) Core clock to read from the FIFO.
SLOWCLK(5)	Input	-	Parallel (slow) clock.
FASTCLK(5)	Input	-	Serial (fast) clock.
DLY_INC	Input	SLOWCLK	(Optional) Dynamic delay control. When DLY_ENA is 1, 1: Increments 0: Decrements
DLY_ENA	Input	SLOWCLK	(Optional) Enable the dynamic delay control.
DLY_RST	Input	SLOWCLK	(Optional) Reset the delay counter.
LP_N_OE	Input	-	(Optional) LP output enable signal for N pad.
LP_N_OUT	Input	-	(Optional) LP output data from the core for the N pad. Used if the data lane is reversible.
LP_N_IN	Output	-	LP input data from the N pad.
HS_ENA	Input	-	Dynamically enable the differential input buffer when in high-speed mode.
HS_TERM	Input	-	Dynamically enables input termination high-speed mode.

(5) These signals are in the primitive, but the software automatically connects them for you.

The clock lane generates the fast clock and slow clock for the RX data lanes within the interface group. It also generates a clock which is divided by 4 that feeds the global network. The following figure shows the clock connections between the clock and data lanes.

Figure 33: Connections for Clock and RX Data Lane in the Same MIPI RX Channel



1. The software automatically connects this signal for you.

MIPI TX Lane

In TX mode, a PLL generates the parallel and serial clocks and passes them to the clock and data lanes.

Figure 34: MIPI TX Lane Block Diagram

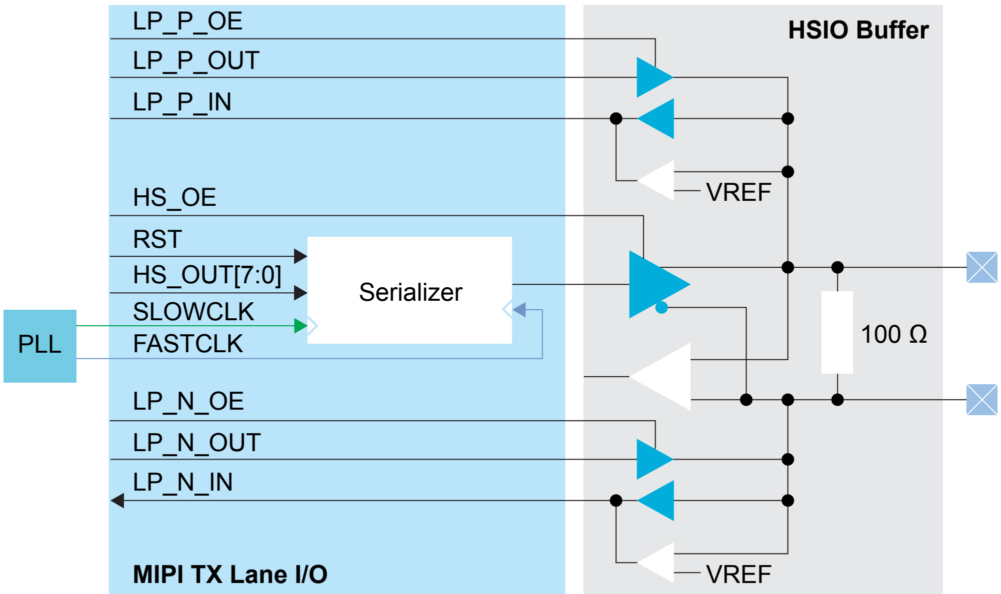


Table 21: MIPI TX Lane Signals

Interface to MIPI soft CSI/DSI controller with D-PHY in FPGA fabric

Signal	Direction	Clock Domain	Description
LP_P_OE	Input	-	LP output enable signal for P pad.
LP_P_OUT	Input	-	LP output data from the core for the P pad.
LP_P_IN	Output	-	(Optional) LP input data from the P pad. Used if data lane is reversible.
HS_OE	Input	-	High-speed output enable signal.
RST	Input	SLOWCLK	(Optional) Resets the serializer.
HS_OUT[7:0]	Input	SLOWCLK	High-speed output data from the core. Always 8-bits wide.
SLOWCLK	Input	-	Parallel (slow) clock.
FASTCLK	Input	-	Serial (fast) clock.
LP_N_OE	Input	-	LP output enable signal for N pad.
LP_N_OUT	Input	-	LP output data from the core for the N pad.
LP_N_IN	Output	-	(Optional) LP input data from the N pad. Used if data lane is reversible.

MIPI Lane Pads

Table 22: MIPI Lane Pads

Signal	Direction	Description
P	Output	Differential pad P.
N	Output	Differential pad N.

I/O Banks

Efinix FPGAs have input/output (I/O) banks for general-purpose usage. Each I/O bank has independent power pins. The number and voltages supported vary by FPGA and package.

Some I/O banks are merged at the package level by sharing VCCIO pins, these are called merged banks. Merged banks have underscores (_) between banks in the VCCIO name (e.g., 1B_1C means VCCIO for bank 1B and 1C are connected). Some of the banks in a merged bank may not have available user I/Os in the package. The following table lists banks that have available user I/Os in a package.

Table 23: I/O Banks by Package

Package	GPIO Type	I/O Banks	Voltage (V)	Dynamic Voltage Support	DDIO Support	Merged Banks
N484	HSIO	2B, 2C, 2D, 2E, 4B, 4C, 4D	1.2, 1.35, 1.5, 1.8	-	All	2A_2B_2C, 4A_4B
	HVIO	BR0, BR3, TR1	1.8. 2.5, 3.0, 3.3	Yes	All	BR3_BR4
C529	HSIO	2A, 2B, 2C, 2D, 2E, 4B, 4C, 4D	1.2, 1.35, 1.5, 1.8	-	All	4A_4B
	HVIO	BL2, BL3, BR0, BR3, TL1, TL5, TR0, TR1, TR2	1.8. 2.5, 3.0, 3.3	Yes	All	BR3_BR4, TL1_TL5 BL2_BL3

DDR DRAM Interface

Important: All information is preliminary and pending definition.

The DDR PHY interface supports LPDDR4 memories with x32 DQ widths and a memory controller hard IP block. The memory controller provides two full-duplex AXI4 buses to communicate with the FPGA core.

Note: The DDR PHY and controller are hard blocks; you cannot bypass the DDR DRAM memory controller to access the PHY directly for non-DDR memory controller applications.

Figure 35: DDR DRAM Block Diagram

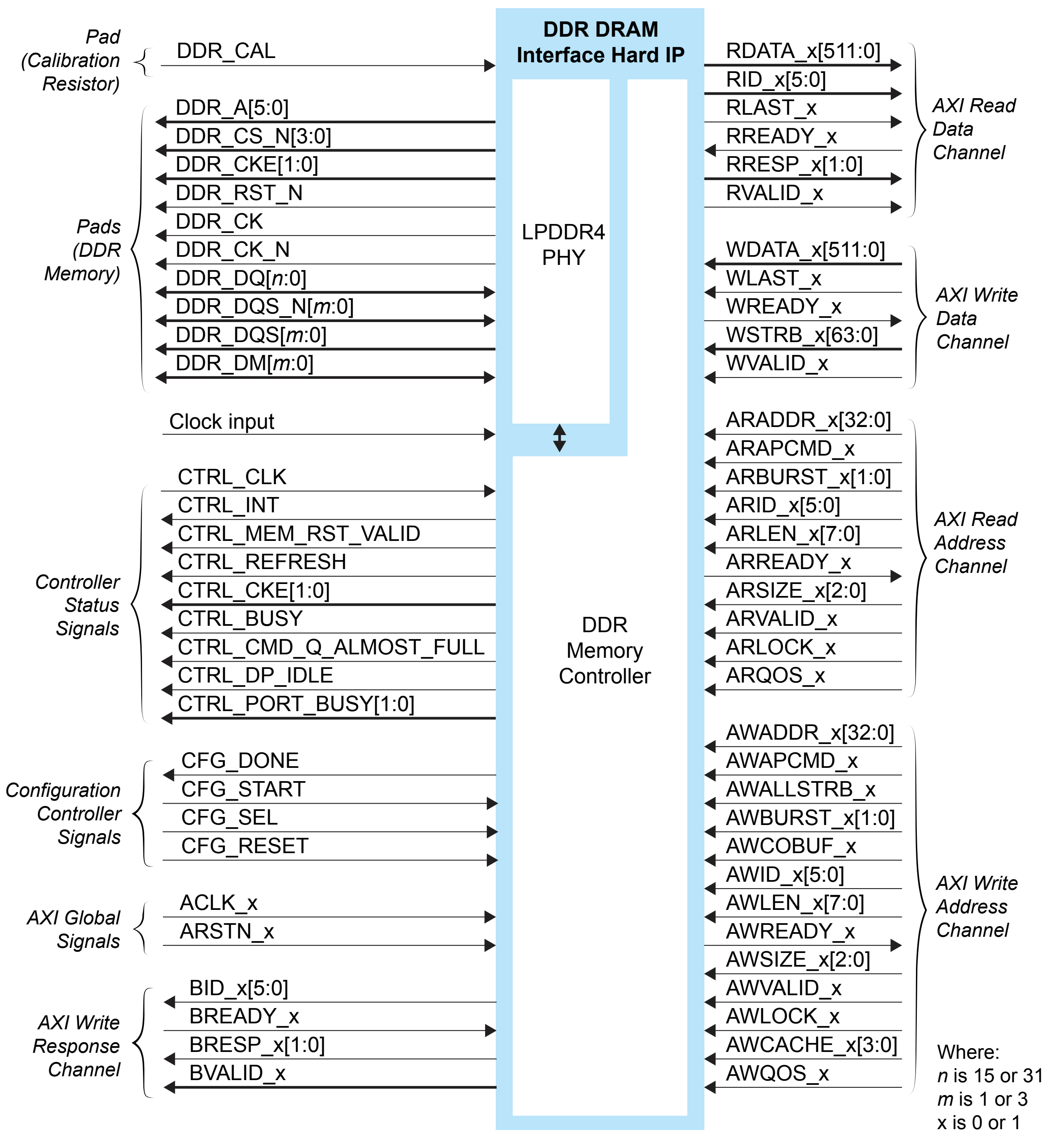
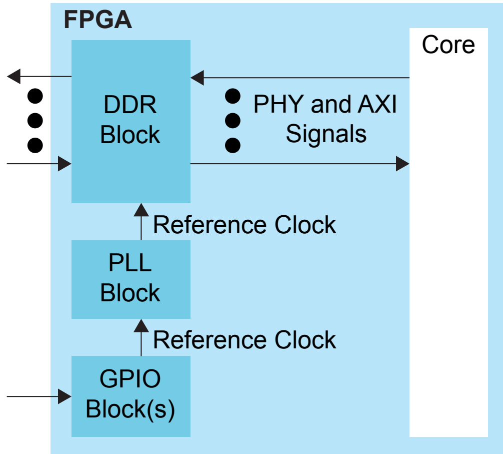


Figure 36: DDR DRAM Interface Block Diagram



Note: The PLL reference clock must be driven by I/O pads. The Efinity software issues a warning if you do not connect the reference clock to an I/O pad. (Using the clock tree may induce additional jitter and degrade the DDR performance.)

Additionally, the PLL that clocks the DDR DRAM interface should not use programmable duty cycle, fractional output, or spread-spectrum clocking because these features increase jitter.

Refer to **About the PLL Interface** for more information about the PLL block.

Table 24: DDR DRAM Pads

Signal	Direction	Description
DDR_A[5:0]	Output	Address signals to the DRAM.
DDR_CS_N[3:0]	Output	Chip select to the DRAM.
DDR_CKE[1:0]	Output	Active-high clock enable signals to the DRAM.
DDR_RST_N	Output	Active-low reset signal to the DRAM.
DDR_CK	Output	Differential clock signals to the DRAM.
DDR_CK_N	Output	
DDR_DQ[*n*:0]	Bidirectional	Data bus to/from the memories. For writes, the FPGA drives these signals. For reads, the memory drives these signals. These signals are connected to the DQ pins on the memories. *n* is 15 or 31 depending on the Data Width setting. If unused, can be left floating on the board.
DDR_DQS_N[*m*:0]	Bidirectional	Differential data strobes to/from the memories. For writes, the FPGA drives these signals. For reads, the memory drives these signals. These signals are connected to the DQS pins on the memories. *m* is 1or 3 depending on the DQ width. If unused, can be left floating on the board.
DDR_DQS[*m*:0]	Bidirectional	
DDR_DM[*m*:0]	Bidirectional	Signals used as active-high data-mask and data bus inversion indicator. *m* is 1 or 3 depending on the DQ width. If data bus inversion is enabled for a write operation, the DDR controller will drive the signal high if the write data byte is inverted. Similarly, if data bus inversion is enabled for a read operation, the memory device will drive the signal high if the read data byte is inverted. If unused, can be left floating on the board.

Table 25: Calibration Resistor Pad

Signal	Direction	Description
DDR_CAL	Input	Calibration resistor connection. Connect to the ground through a 240 Ω resistor on your board.

Table 26: Controller Status Signals

Signal	Direction	Clock Domain	Description
CTRL_CLK	Input	N/A	Clock for controller status signals.
CTRL_INT	Output	N/A	Controller detects Interrupt.
CTRL_MEM_RST_VALID	Output	N/A	Controller has been reset.
CTRL_REFRESH	Output	CTRL_CLK	Indicate controller is executing refresh command.
CTRL_CKE[1:0]	Output	CTRL_CLK	Delayed 'control_cke' from the controller, indicating that the memory is in self-refresh or power down mode.
CTRL_BUSY	Output	CTRL_CLK	Controller is busy reading data.
CTRL_CMD_Q_ALMOST_FULL	Output	CTRL_CLK	Command queue reached 'q_fullness' parameter.
CTRL_DP_IDLE	Output	CTRL_CLK	Datapath is idle.
CTRL_PORT_BUSY[1:0]	Output	CTRL_CLK	Indicate if port is reading data.

Table 27: Configuration Controller Signals

Signal	Direction	Description
CFG_RESET	Input	Active-high configuration controller reset. Asserting this signal also resets the DDR controller, PHY and the DRAM device.
CFG_START	Input	Start the configuration controller.
CFG_DONE	Output	Indicates the configuration controller is done
CFG_SEL	Input	Tie this input to low to enable the configuration controller.

Table 28: AXI4 Global Signals (Interface to FPGA Core Logic)

Signal	Direction	Clock Domain	Description
ACLK_*x*	Input	N/A	AXI4 clock inputs.
ARSTN_*x*	Input	ACLK_x	Active-low reset signal to the AXI interface.

Table 29: AXI4 Write Response Channel Signals (Interface to FPGA Core Logic)

Signal x is 0 or 1	Direction	Clock Domain	Description
BID_*x*[5:0]	Output	ACLK_x	Response ID tag. This signal is the ID tag of the write response.
BREADY_*x*	Input	ACLK_x	Response ready. This signal indicates that the master can accept a write response.
BRESP_*x*[1:0]	Output	ACLK_x	Read response. This signal indicates the status of the read transfer.
BVALID_*x*	Output	ACLK_x	Write response valid. This signal indicates that the channel is signaling a valid write response.

Table 30: AXI4 Read Data Channel Signals (Interface to FPGA Core Logic)

Signal x is 0 or 1	Direction	Clock Domain	Description
RDATA_*x*[511:0]	Output	ACLK_x	Read data.
RID_*x*[5:0]	Output	ACLK_x	Read ID tag. This signal is the identification tag for the read data group of signals generated by the slave.
RLAST_*x*	Output	ACLK_x	Read last. This signal indicates the last transfer in a read burst.
RREADY_*x*	Input	ACLK_x	Read ready. This signal indicates that the master can accept the read data and response information.
RRESP_*x*[1:0]	Output	ACLK_x	Read response. This signal indicates the status of the read transfer.
RVALID_*x*	Output	ACLK_x	Read valid. This signal indicates that the channel is signaling the required read data.

Table 31: AXI4 Write Data Channel Signals (Interface to FPGA Core Logic)

Signal x is 0 or 1	Direction	Clock Domain	Description
WDATA_*x*[511:0]	Input	ACLK_x	Write data.
WLAST_*x*	Input	ACLK_x	Write last. This signal indicates the last transfer in a write burst.
WREADY_*x*	Output	ACLK_x	Write ready. This signal indicates that the slave can accept the write data.
WSTRB_*x*[63:0]	Input	ACLK_x	Write strobes. This signal indicates which byte lanes hold valid data. There is one write strobe bit for each eight bits of the write data bus.
WVALID_*x*	Input	ACLK_x	Write valid. This signal indicates that valid write data and strobes are available.

Table 32: AXI4 Read Address Signals (Interface to FPGA Core Logic)

Signal x is 0 or 1	Direction	Clock Domain	Description
ARADDR_*x*[32:0]	Input	ACLK_x	Read address. It gives the address of the first transfer in a burst transaction.
ARBURST_*x*[1:0]	Input	ACLK_x	Burst type. The burst type and the size determine how the address for each transfer within the burst is calculated. 'b01 = INCR 'b10 = WRAP
ARID_*x*[5:0]	Input	ACLK_x	Address ID. This signal identifies the group of address signals.
ARLEN_*x*[7:0]	Input	ACLK_x	Burst length. This signal indicates the number of transfers in a burst.
ARREADY_*x*	Output	ACLK_x	Address ready. This signal indicates that the slave is ready to accept an address and associated control signals.
ARSIZE_*x*[2:0]	Input	ACLK_x	Burst size. This signal indicates the size of each transfer in the burst.
ARVALID_*x*	Input	ACLK_x	Address valid. This signal indicates that the channel is signaling valid address and control information.
ARLOCK_*x*	Input	ACLK_x	Lock type. This signal provides additional information about the atomic characteristics of the transfer.
ARAPCMD_*x*	Input	ACLK_x	Read auto-precharge.
ARQOS_*x*	Input	ACLK_x	QoS identifier for read transaction.

Table 33: AXI4 Write Address Signals (Interface to FPGA Core Logic)

Signal x is 0 or 1	Direction	Clock Domain	Description
AWADDR_*x*[32:0]	Input	ACLK_x	Write address. It gives the address of the first transfer in a burst transaction.
AWBURST_*x*[1:0]	Input	ACLK_x	Burst type. The burst type and the size determine how the address for each transfer within the burst is calculated.
AWID_*x*[5:0]	Input	ACLK_x	Address ID. This signal identifies the group of address signals.
AWLEN_*x*[7:0]	Input	ACLK_x	Burst length. This signal indicates the number of transfers in a burst.
AWREADY_*x*	Output	ACLK_x	Address ready. This signal indicates that the slave is ready to accept an address and associated control signals.
AWSIZE_*x*[2:0]	Input	ACLK_x	Burst size. This signal indicates the size of each transfer in the burst.
AWVALID_*x*	Input	ACLK_x	Address valid. This signal indicates that the channel is signaling valid address and control information.
AWLOCK_*x*	Input	ACLK_x	Lock type. This signal provides additional information about the atomic characteristics of the transfer.
AWAPCMD_*x*	Input	ACLK_x	Write auto-precharge.
AWQOS_*x*	Input	ACLK_x	QoS identifier for write transaction.
AWCACHE_*x*[3:0]	Input	ACLK_x	Memory type. This signal indicates how transactions are required to progress through a system.
AWALLSTRB_*x*	Input	ACLK_x	Write all strobes asserted. The DDR controller only supports a maximum of 16 AXI beats for write commands using this signal.
AWCOBUF_*x*	Input	ACLK_x	Write coherent bufferable selection.

DDR DRAM Interface Input Clocks

You only need one clock to drive the DDR DRAM interface block. To select the PLL as the clock source for the DDR DRAM block, choose **CLKIN 0**, **CLKIN 1**, or **CLKIN 2** as the **Clock Source** in the Interface Designer.

Table 34: Input Clocks

Refer to **Tz325 Interface Floorplan** on page 117 for which blocks are available in the package you are using.

DDR Block	PLL Resource			PLL CLKOUT
	CLKIN 0	CLKIN 1	CLKIN 2	
DDR 0	BL0	BL1	BL2	CLKOUT3
DDR 1	TL0	TL1	TL2	CLKOUT3

`CLKOUT3` of the selected PLL drives the DDR DRAM interface block. The clock runs at a quarter of the PHY data rate (for example, 2,000 Mbps requires a 500 MHz clock). You need to instantiate the selected PLL in the Interface Designer and configure the PLL's `CLKOUT3` with the required frequency. The Efinity software then connects the PLL's `CLKOUT3` signal to the DDR DRAM interface block automatically.

MIPI D-PHY

Important: All information is preliminary and pending definition.

In addition to the HSIO, which you can configure as MIPI RX or TX lanes, Tz325 FPGAs have hardened MIPI D-PHY blocks, each with 4 data lanes and 1 clock lane. The MIPI D-PHY RX and MIPI D-PHY TX can operate independently with dedicated I/O banks.

You can use the hardened MIPI D-PHY blocks along with the HSIO configured as MIPI D-PHY lanes to create systems that aggregate data from many cameras or sensors.

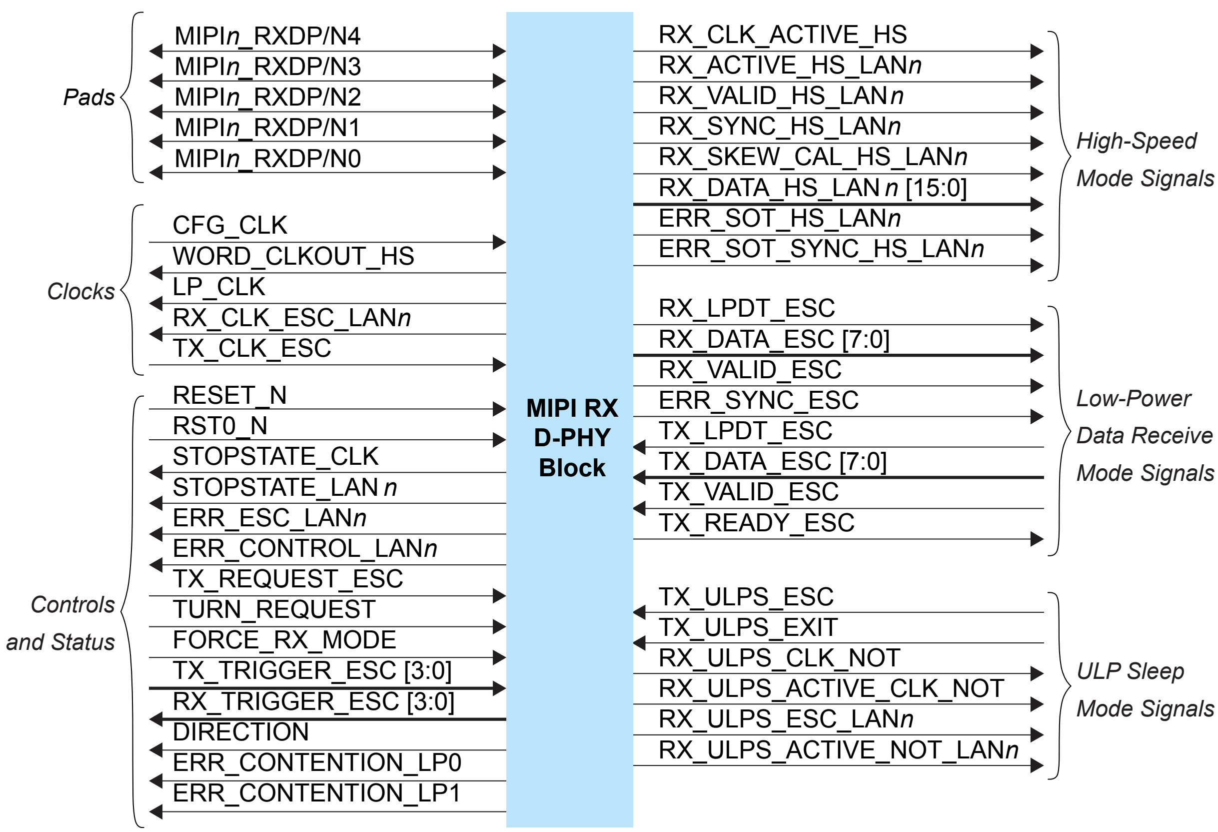
The MIPI TX/RX interface supports the MIPI D-PHY specification v1.1. It has the following features:

- Programmable data lane configuration supporting up to 4 lanes
- High-speed mode supports up to 2.0 Gbps data rates per lane
- Operates in continuous and non-continuous clock modes
- Supports Ultra-Low Power State (ULPS)

MIPI RX D-PHY

The MIPI RX D-PHY is a receiver interface designed to receive data and the control information of MIPI CSI, DSI, or other associated protocols. The MIPI RX D-PHY comprises of one clock lane and up to four data lanes for a single-channel configuration. The MIPI RX D-PHY also interfaces with MIPI-associated protocol controllers via a standard MIPI D-PHY PHY Protocol Interface (PPI) that supports the 8- or 16-bit high-speed receiving data bus.

Figure 37: MIPI RX D-PHY x4 Block Diagram



The status signals provide optional status and error information about the MIPI RX D-PHY interface operation.

Figure 38: MIPI RX D-PHY Interface Block Diagram

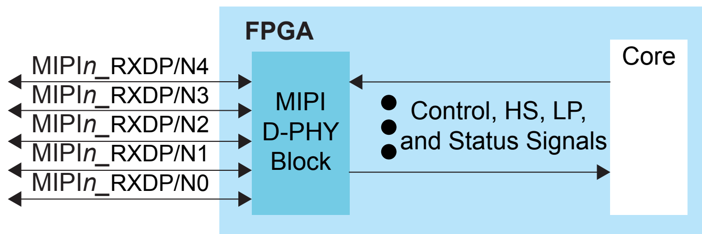


Table 35: MIPI RX D-PHY Clocks Signals (Interface to FPGA Fabric)

Signal	Direction	Clock Domain	Notes
CFG_CLK	Input	N/A	Configuration Clock (used for time counter and EQ calibration). The clock must be between 80 MHz to 120 MHz.
WORD_CLKOUT_HS	Output	N/A	HS Receive Byte/Word clock.
LP_CLK	Output	N/A	Low Power State clock.
RX_CLK_ESC_LAN*n*	Output	N/A	Escape Mode Receive clock.
TX_CLK_ESC	Input	N/A	Escape Mode Transmit clock. The clock must be lower than 20 MHz.

Table 36: MIPI RX D-PHY Control and Status Signals (Interface to FPGA Fabric)

Signal	Direction	Clock Domain	Notes
RESET_N	Input	N/A	Reset. Disables PHY and reset the digital logic.
RST0_N	Input	N/A	Asynchronous FIFO reset and synchronous out of reset.
STOPSTATE_CLK	Output	N/A	Lane in Stop State.
STOPSTATE_LAN*n*	Output	N/A	Data Lane in Stop State (Lane N).
ERR_ESC_LAN*n*	Output	N/A	Lane *n* Escape Command Error.
ERR_CONTROL_LAN*n*	Output	N/A	Lane *n* Has Line State Error.
TX_REQUEST_ESC	Input	TX_CLK_ESC	Lane 0 Request TX Escape Mode.
TURN_REQUEST	Input	TX_CLK_ESC	Lane 0 Request Turnaround.
FORCE_RX_MODE	Input	N/A	Lane 0 Force Lane into Receive Mode/Wait for Stop State.
TX_TRIGGER_ESC [3:0]	Input	TX_CLK_ESC	Lane 0 Send a Trigger Event.
RX_TRIGGER_ESC [3:0]	Output	RX_CLK_ESC_LAN0	Lane 0 Received a Trigger Event.
DIRECTION	Output	N/A	Lane 0 Transmit/Receive Direction (0 = TX, 1 = RX).
ERR_CONTENTION_LP0	Output	N/A	Lane 0 Contention Error when driving 0.
ERR_CONTENTION_LP1	Output	N/A	Lane 0 Contention Error when driving 1.

Table 37: MIPI RX D-PHY High-Speed Mode Signals (Interface to FPGA Fabric)

Signal	Direction	Clock Domain	Notes
RX_CLK_ACTIVE_HS	Output	N/A	HS Clock Lane Active.
RX_ACTIVE_HS_LAN*n*	Output	WORD_CLKOUT_HS	HS Reception Active.
RX_VALID_HS_LAN*n*	Output	WORD_CLKOUT_HS	HS Data Receive Valid.
RX_SYNC_HS_LAN*n*	Output	WORD_CLKOUT_HS	HS Reveiver Sync. Observed.
RX_SKEW_CAL_HS_LAN*n*	Output	WORD_CLKOUT_HS	HS Reveiver DeSkew Burst Received.
RX_DATA_HS_LAN*n* [15:0]	Output	WORD_CLKOUT_HS	HS Receive Data.
ERR_SOT_HS_LAN*n*	Output	WORD_CLKOUT_HS	State-of-Transmission (SOT) Error.
ERR_SOT_SYNC_HS_LAN*n*	Output	WORD_CLKOUT_HS	SOT Sync. Error.

Table 38: MIPI RX D-PHY Low-Power Data Receive Mode Signals (Interface to FPGA Fabric)

Signal	Direction	Clock Domain	Notes
RX_LPDT_ESC	Output	RX_CLK_ESC_LAN0	Lane 0 enter LPDT RX Mode.
RX_DATA_ESC [7:0]	Output	RX_CLK_ESC_LAN0	Lane 0 LPDT RX Data.
RX_VALID_ESC	Output	RX_CLK_ESC_LAN0	Lane 0 LPDT RX Data Valid.
ERR_SYNC_ESC	Output	N/A	Lane 0 LPDT RX Data Sync. Error.
TX_LPDT_ESC	Input	TX_CLK_ESC	Lane 0 Enter LPDT TX Mode.
TX_DATA_ESC [7:0]	Input	TX_CLK_ESC	Lane 0 LPDT TX Data.
TX_VALID_ESC	Input	TX_CLK_ESC	Lane 0 LPDT TX Data Valid.
TX_READY_ESC	Output	TX_CLK_ESC	Lane 0 LDPT TX Data Ready.

Table 39: MIPI RX D-PHY ULP Sleep Mode Signals (Interface to FPGA Fabric)

Signal	Direction	Clock Domain	Notes
TX_ULPS_ESC	Input	TX_CLK_ESC	Lane 0 Enter ULPS Mode.
TX_ULPS_EXIT	Input	TX_CLK_ESC	Lane 0 Exit ULPS Mode.
RX_ULPS_CLK_NOT	Output	N/A	CLK0 Enter ULPS Mode.
RX_ULPS_ACTIVE_CLK_NOT	Output	N/A	CLK0 is in ULPS (Active Low).
RX_ULPS_ESC_LAN*n*	Output	RX_CLK_ESC_LAN*n*	Lane *n* Enter ULPS Mode.
RX_ULPS_ACTIVE_NOT_LAN*n*	Output	N/A	Lane *n* is in ULPS (Active Low).

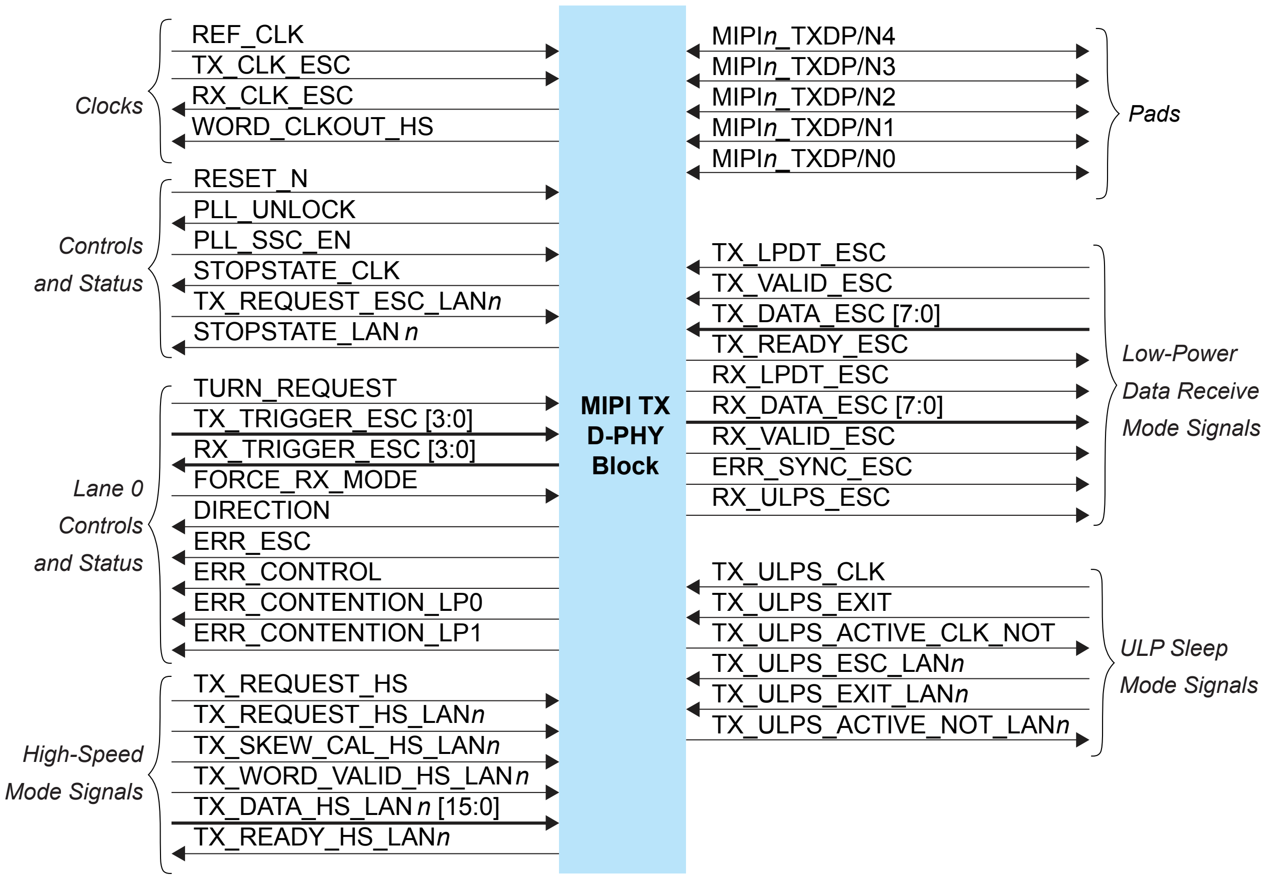
Table 40: MIPI RX D-PHY Pads

Pad	Direction	Description
MIPI*n*_RXDP[4:0]	Bidirectional	MIPI transceiver P pads.
MIPI*n*_RXDN[4:0]	Bidirectional	MIPI transceiver N pads.

MIPI TX D-PHY

The MIPI TX D-PHY is a transmitter interface designed to transmit data and the control information of MIPI CSI, DSI, or other associated protocols. The MIPI TX D-PHY comprises of one clock lane and up to four data lanes for a single-channel configuration. The MIPI TX D-PHY also interfaces with MIPI-associated protocol controllers via a standard MIPI D-PHY PPI that supports the 8- or 16-bit high-speed receiving data bus.

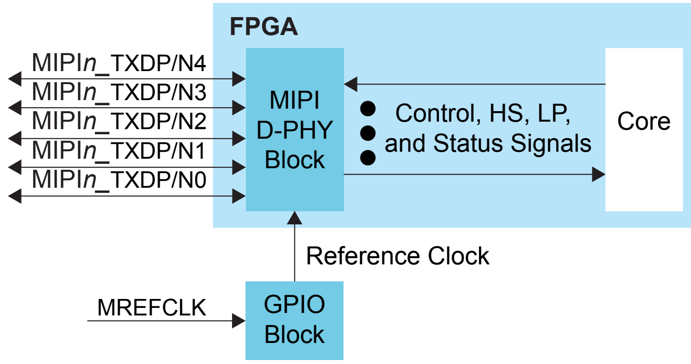
Figure 39: MIPI TX D-PHY x4 Block Diagram



The MIPI TX D-PHY block requires an escape clock (`TX_CLK_ESC`) for use when the MIPI interface is in escape (low-power) mode, which runs up to 20 MHz.

Note: Efinix recommends that you set the escape clock frequency as close to 20 MHz as possible.

Figure 40: MIPI TX D-PHY Interface Block Diagram



Note: GPIO block is the default reference clock source. However, the PLL and core clock out can also be set as the reference clock source.

Table 41: MIPI TX D-PHY Clocks Signals (Interface to FPGA Fabric)

Signal	Direction	Clock Domain	Notes
REF_CLK	Input	N/A	Reference Clock. The clock must be between 12 MHz to 52 MHz.
TX_CLK_ESC	Input	N/A	Escape Mode Transmit Clock, used to generate escape sequence. The clock must be less than 20 MHz.
RX_CLK_ESC	Output	N/A	Escape Mode Receive Clock (lane 0 only)
WORD_CLKOUT_HS	Output	N/A	HS Transmit Byte/Word Clock. This signal must be 1/8 of the bit-rate in normal 8-bit HS-PPI D-PHY mode, or 1/16 of the bit-rate in 16-bit PHY mode.

Table 42: MIPI TX D-PHY Control and Status Signals (Interface to FPGA Fabric)

Signal	Direction	Clock Domain	Notes
RESET_N	Input	N/A	Reset. Disables PHY and reset the digital logic.
PLL_UNLOCK	Output	N/A	PLL is in unlock state.
PLL_SSC_EN	Input	N/A	(Optional) PLL SSC Enable: Always enable: 1 Disable: 0 Driven by active signal for dynamic enable
STOPSTATE_CLK	Output	N/A	Clock Lane in Stop State (Clk 0).
TX_REQUEST_ESC_LAN*n*	Input	TX_CLK_ESC	Escape Mode Transmit Request (Lane N).
STOPSTATE_LAN*n*	Output	N/A	Data Lane in Stop State (Lane N).

Table 43: MIPI TX D-PHY Lane 0 Control and Status Signals (Interface to FPGA Fabric)

Signal	Direction	Clock Domain	Notes
TURN_REQUEST	Input	TX_CLK_ESC	Lane 0 Turnaround Request.
TX_TRIGGER_ESC [3:0]	Input	TX_CLK_ESC	Lane 0 Send an Escape Mode Trigger Event.
RX_TRIGGER_ESC [3:0]	Output	RX_CLK_ESC	Lane 0 Received an Escape Mode Trigger Event.
FORCE_RX_MODE	Input	N/A	Lane 0 Force into Receive Mode/Wait for Stop.
DIRECTION	Output	N/A	Lane 0 Transmit/Receive Direction: 0: TX, 1: RX
ERR_ESC	Output	N/A	Lane 0 Escape Command Error.
ERR_CONTROL	Output	N/A	Lane 0 Line State Error.
ERR_CONTENTION_LP0	Output	N/A	Lane 0 Line Contention Detected (when driving 0).
ERR_CONTENTION_LP1	Output	N/A	Lane 0 Line Contention Detected (when driving 1).

Table 44: MIPI TX D-PHY High Speed Mode Signals (Interface to FPGA Fabric)

Signal	Direction	Clock Domain	Notes
TX_REQUEST_HS	Input	WORD_CLKOUT_HS	HS Clock Request (Clk 0).
TX_REQUEST_HS_LAN*n*	Input	WORD_CLKOUT_HS	HS Transmit Request and Data Valid (Lane 0-3).
TX_SKEW_CAL_HS_LAN*n*	Input	WORD_CLKOUT_HS	HS Skew Calibration (Lane N).
TX_WORD_VALID_HS_LAN*n*	Input	WORD_CLKOUT_HS	HS High Byte Valid (Lane N) for 16-bit mode.
TX_DATA_HS_LAN*n* [15:0]	Input	WORD_CLKOUT_HS	HS Transmit Data (Lane N).
TX_READY_HS_LAN*n*	Output	WORD_CLKOUT_HS	HS Transmit Ready (Lane N).

Table 45: MIPI TX D-PHY Low-Power Data Receive Mode Signals (Interface to FPGA Fabric)

Signal	Direction	Clock Domain	Notes
TX_LPDT_ESC	Input	TX_CLK_ESC	Lane 0 Enter LPDT Mode.
TX_VALID_ESC	Input	TX_CLK_ESC	Lane 0 LPDT Data Valid .
TX_DATA_ESC [7:0]	Input	TX_CLK_ESC	Lane 0 LPDT Data Bus.
TX_READY_ESC	Output	TX_CLK_ESC	Lane 0 LPDT Data Ready.
RX_LDPT_ESC	Output	RX_CLK_ESC	Escape LP Data Receive Mode.
RX_DATA _ESC[7:0]	Output	RX_CLK_ESC	Escape Mode Receive Data.
RX_VALID_ESC	Output	RX_CLK_ESC	Escape Mode Receive Data Valid.
ERR_SYNC_ESC	Output	N/A	LPDT Data Sync Error.
RX_ULPS_ESC	Output	RX_CLK_ESC	Lane 0 entered ULPS mode.

Table 46: MIPI TX D-PHY ULP Sleep Mode (Interface to FPGA Fabric)

Signal	Direction	Clock Domain	Notes
TX_ULPS_CLK	Input	TX_CLK_ESC	CLK0 to enter Ultra-Low Power State.
TX_ULPS_EXIT	Input	TX_CLK_ESC	CLK0 to exit Ultra-Low Power State.
TX_ULPS_ACTIVE_CLK_NOT	Output	N/A	Clock Lane in ULP State - Active Low (Clk 0).
TX_ULPS_ESC_LAN*n*	Input	TX_CLK_ESC	Lane *n* to enter Ultra-Low Power State.
TX_ULPS_EXIT_LAN*n*	Input	TX_CLK_ESC	Lane *n* to exit Ultra-Low Power State.
TX_ULPS_ACTIVE_NOT_LAN*n*	Output	N/A	Data Lane in ULP State - Active Low (Lane N).

Table 47: MIPI TX D-PHY Pads

Pad	Direction	Description
MIPI*n*_TXDP[4:0]	Bidirectional	MIPI transceiver P pads.
MIPI*n*_TXDN[4:0]	Bidirectional	MIPI transceiver N pads.

Oscillator

The Tz325 has One low-frequency oscillator tailored for low-power operation. The oscillator runs at a nominal frequency of 10, 20, 40, or 80 MHz. You can use the oscillator to perform always-on functions with the lowest power possible. It's output clock is available to the core. You can enable or disable the oscillator to allow power savings when not in use. The oscillator has:

- An output duty cycle of 45% to 55%.
- A ±20% frequency variation from device to device.

Fractional PLL

Tz325 FPGAs have 12 PLLs to synthesize clock frequencies. The PLLs are located in the corners of the FPGA. You can use the PLL to compensate for clock skew/delay via external or internal feedback to meet timing requirements in advanced applications. The PLL reference clock has up to four sources. You can dynamically select the PLL reference clock with the `CLKSEL` port. (Hold the PLL in reset when dynamically selecting the reference clock source.)

Tz325 FPGAs also support dynamic reconfiguration, programmable duty cycle, a fractional output divider, and spread-spectrum clocking. These features are described in later sections. The PLL consists of a pre-divider counter (N counter), a feedback multiplier counter (M counter), a post-divider counter (O counter), and output dividers (C). A delta sigma modulator supports the fractional output divider features.

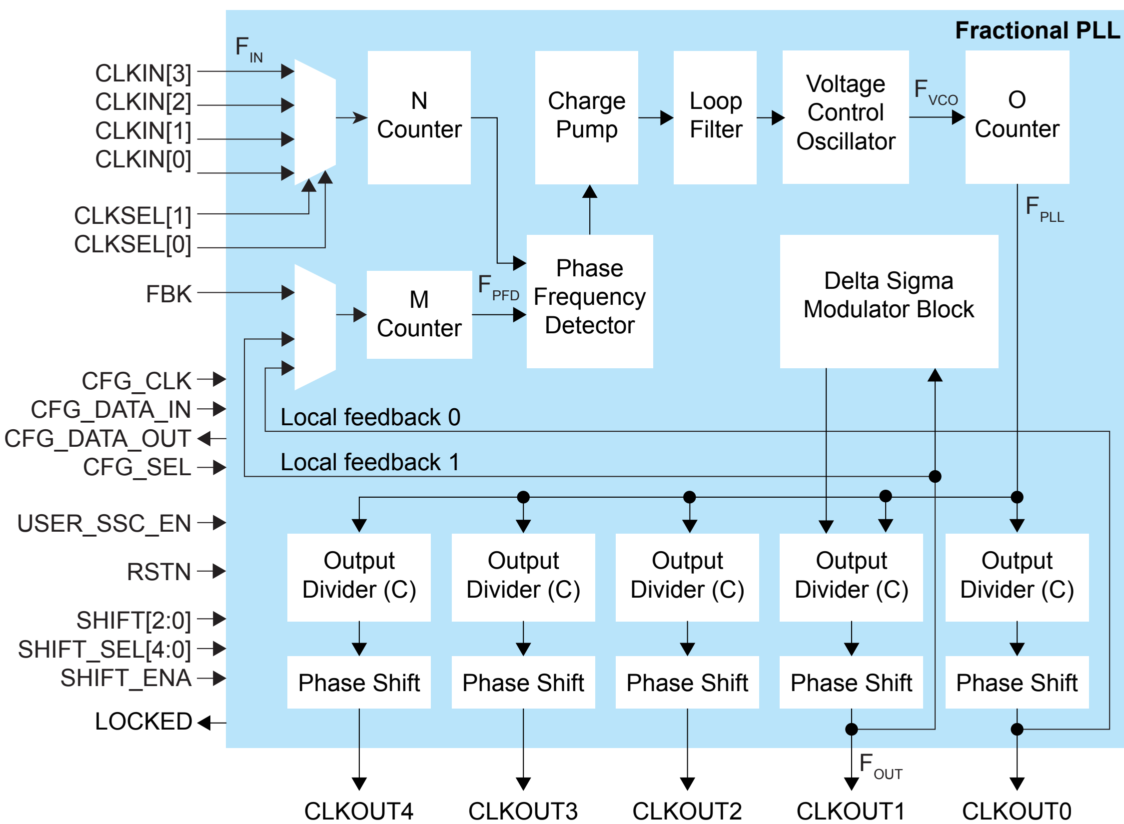
At startup, Efinix recommends that you hold the PLL in reset until the PLL's reference clock source is stable.

Note: You can cascade the PLLs in Tz325 FPGAs. To avoid the PLL losing lock, Efinix recommends that you do not cascade more than two PLLs.

At startup, Efinix recommends resetting all cascaded PLLs. Hold the first PLL in reset until the PLL's reference clock source is stable. Hold the cascaded PLLs in reset until the previous PLL is locked.

Cascaded PLLs do not need a 50% duty cycle on the reference clock. However, the clock needs to meet the PLL minimum pulse width as specified in the data sheet.

Figure 41: PLL Block Diagram



The counter settings define the PLL output frequency:

Feedback Mode	Where:
$F_{PFD} = F_{IN} / N$ $F_{VCO} = (F_{PFD} \times M \times O \times C_{FBK})$ $F_{PLL} = F_{VCO} / O$ $F_{OUT} = (F_{IN} \times M \times C_{FBK}) / (N \times C)$	F_{VCO} is the voltage control oscillator frequency F_{PLL} is the post-divider PLL VCO frequency F_{OUT} is the output clock frequency F_{IN} is the reference clock frequency F_{PFD} is the phase frequency detector input frequency O is the post-divider counter C is the output divider

Note: Refer to the PLL Timing and AC Characteristics on page 104 for F_{VCO}, F_{OUT}, F_{IN}, F_{PLL}, and F_{PFD} values.

Figure 42: PLL Interface Block Diagram

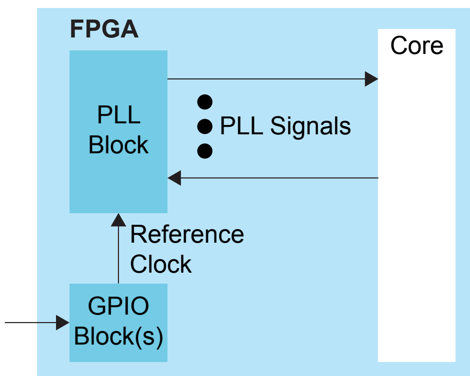


Table 48: PLL Signals (Interface to FPGA Fabric)

Signal	Direction	Description
CLKIN[3:0]	Input	Reference clocks driven by I/O pads or core clock tree. In dynamic mode, the CLKSEL pin chooses which of these inputs to use.
CLKSEL[1:0]	Input	You can dynamically select the reference clock from one of the clock in pins.
RSTN	Input	(Optional) Active-low PLL reset signal. When asserted, this signal resets the PLL; when de-asserted, it enables the PLL. De-assert only when the CLKIN signal is stable. Connect this signal in your design to power-up or reset the PLL. Assert the RSTN pin for a minimum pulse of 10 ns to reset the PLL. Assert RSTN when dynamically changing the selected PLL reference clock.
FBK	Input	Connect to a clock out interface pin when the PLL is not in internal feedback mode. Required when any output is using dynamic phase shift.
CLKOUT0 CLKOUT1 CLKOUT2 CLKOUT3 CLKOUT4	Output	PLL output. You can route these signals as input clocks to the core's GCLK network. The PLL output clock used as the feedback clock can have a maximum frequency of 4x (integer) of the reference clock. If all your system clocks do not fall within this range, you should dedicate one unused PLL output clock for feedback.
LOCKED	Output	(Optional) Goes high when PLL achieves lock; goes low when a loss of lock is detected. Connect this signal in your design to monitor the lock status. This signal is not synchronized to any clock and the minimum high or low pulse width of the lock signal may be smaller than the CLKOUT's period.

Signal	Direction	Description
SHIFT[2:0]	Input	(Optional) Dynamically change the phase shift of the output selected to the value set with this signal. Possible values from 000 (no phase shift) to 111 (3.5 F_{PLL} cycle delay). Each increment adds 0.5 cycle delay. Required when any output is using dynamic phase shift.
SHIFT_SEL[4:0]	Input	(Optional) Choose the output(s) affected by the dynamic phase shift. Required when any output is using dynamic phase shift.
SHIFT_ENA	Input	(Optional) When high, changes the phase shift of the selected PLL(s) to the new value. Required when any output is using dynamic phase shift.
CFG_CLK	Input	Configuration clock pin name; used with dynamic configuration.
CFG_DATA_IN	Input	Configuration data input pin name; used with dynamic configuration.
CFG_DATA_OUT	Output	Configuration data output pin name; used with dynamic configuration.
CFG_SEL	Input	Configuration select pin name; used with dynamic configuration.
USER_SSC_EN	Input	User spread-spectrum clocking enable pin name.

Table 49: PLL Reference Clock Resource Assignments (N484)

PLL	REFCLK0	REFCLK1	External Feedback I/O
BL0	Differential: GPIOB_P_03_PLLIN0, GPIOB_N_03_CDI31 Single-ended: GPIOB_P_03_PLLIN0	–	Differential: GPIOB_P_04_CDI30_EXTFB, GPIOB_N_04_CDI29 Single-ended: GPIOB_P_04_CDI30_EXTFB
BL1	Differential: GPIOB_P_05_CDI28_PLLIN0, GPIOB_N_05_CDI27 Single-ended: GPIOB_P_05_CDI28_PLLIN0	–	Differential: GPIOB_P_06_CDI26_EXTFB, GPIOB_N_06_CDI25 Single-ended: GPIOB_P_06_CDI26_EXTFB
BL2	Differential: GPIOB_P_07_CDI24_PLLIN0, GPIOB_N_07_CDI23 Single-ended: GPIOB_P_07_CDI24_PLLIN0	–	Differential: GPIOB_P_08_CDI22_EXTFB, GPIOB_N_08_CDI69 Single-ended: GPIOB_P_08_CDI22_EXTFB
TR0	Differential: GPIOT_P_50_CDO1_PLLIN0, GPIOT_N_50_CDO0 Single-ended: GPIOT_P_50_CDO1_PLLIN0	Single-ended: GPIOR_93_PLLIN1	Differential: GPIOT_P_49_CDO3_EXTFB, GPIOT_N_49_CDO2 Single-ended: GPIOT_P_49_CDO3_EXTFB
TR1	Differential: GPIOT_P_59_CDI79_PLLIN0, GPIOT_N_59_CDI78 Single-ended: GPIOT_P_59_CDI79_PLLIN0	–	–
TR2	Differential: GPIOT_P_62_CDI73_PLLIN0, GPIOT_N_62_CDI72 Single-ended: GPIOT_P_62_CDI73_PLLIN0	–	–

PLL	REFCLK0	REFCLK1	External Feedback I/O
BR0	Differential: GPIOB_P_30_CDI9_PLLIN0, GPIOB_N_30_CDI8 Single-ended: GPIOB_P_30_CDI9_PLLIN0	Single-ended: GPIOR_140_PLLIN1	Differential: GPIOB_P_29_CDI11_EXTFB, GPIOB_N_29_CDI10 Single-ended: GPIOB_P_29_CDI11_EXTFB
BR1	Differential: GPIOB_P_39_CCK_PLLIN0, GPIOB_N_39_CDI2 Single-ended: GPIOB_P_39_CCK_PLLIN0	Single-ended: GPIOR_145_PLLIN1	Differential: GPIOB_P_40_CDI1_EXTFB, GPIOB_N_40_CDI0 Single-ended: GPIOB_P_40_CDI1_EXTFB
BR2		Single-ended: GPIOR_165_PLLIN1	

Table 50: PLL Reference Clock Resource Assignments (C529)

PLL	REFCLK0	REFCLK1	External Feedback I/O
BL0	Differential: GPIOB_P_03_PLLIN0, GPIOB_N_03_CDI31 Single-ended: GPIOB_P_03_PLLIN0	Single-ended: GPIOL_25_PLLIN1	Differential: GPIOB_P_04_CDI30_EXTFB, GPIOB_N_04_CDI29 Single-ended: GPIOB_P_04_CDI30_EXTFB
BL1	Differential: GPIOB_P_05_CDI28_PLLIN0, GPIOB_N_05_CDI27 Single-ended: GPIOB_P_05_CDI28_PLLIN0	Single-ended: GPIOL_32_PLLIN1	Differential: GPIOB_P_06_CDI26_EXTFB, GPIOB_N_06_CDI25 Single-ended: GPIOB_P_06_CDI26_EXTFB
BL2	Differential: GPIOB_P_07_CDI24_PLLIN0, GPIOB_N_07_CDI23 Single-ended: GPIOB_P_07_CDI24_PLLIN0	Single-ended: GPIOL_34_PLLIN1	Differential: GPIOB_P_08_CDI22_EXTFB, GPIOB_N_08_CDI69 Single-ended: GPIOB_P_08_CDI22_EXTFB
TL0	Differential: GPIOT_P_08_PLLIN0, GPIOT_N_08 Single-ended: GPIOT_P_08_PLLIN0	Single-ended: GPIOL_52_PLLIN1	Differential: GPIOT_P_07_EXTFB, GPIOT_N_07 Single-ended: GPIOT_P_07_EXTFB
TL1	Differential: GPIOT_P_13_CDI119_PLLIN0, GPIOT_N_13_CDI118 Single-ended: GPIOT_P_13_CDI119_PLLIN0	Single-ended: GPIOL_85_PLLIN1	Differential: GPIOT_P_14_CDI117_EXTFB, GPIOT_N_14 Single-ended: GPIOT_P_14_CDI117_EXTFB
TL2	Differential: GPIOT_P_23_CDO31_PLLIN0, GPIOT_N_23_CDO30 Single-ended: GPIOT_P_23_CDO31_PLLIN0	Single-ended: GPIOL_87_PLLIN1	Differential: GPIOT_P_24_CDO29_EXTFB, GPIOT_N_24_CDO28 Single-ended: GPIOT_P_24_CDO29_EXTFB
TR0	Differential: GPIOT_P_50_CDO1_PLLIN0, GPIOT_N_50_CDO0 Single-ended: GPIOT_P_50_CDO1_PLLIN0	Single-ended: GPIOR_93_PLLIN1	Differential: GPIOT_P_49_CDO3_EXTFB, GPIOT_N_49_CDO2 Single-ended: GPIOT_P_49_CDO3_EXTFB

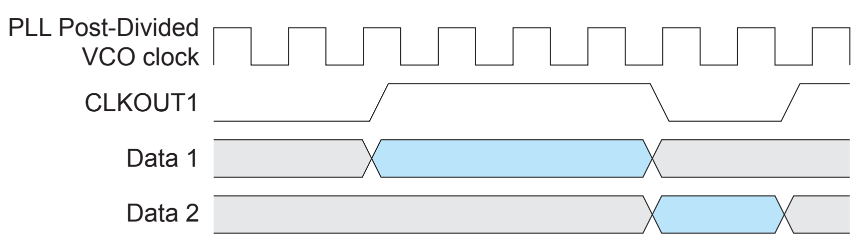
PLL	REFCLK0	REFCLK1	External Feedback I/O
TR1	Differential: GPIOT_P_59_CDI79_PLLIN0, GPIOT_N_59_CDI78 Single-ended: GPIOT_P_59_CDI79_PLLIN0	Single-ended: GPIOR_88_PLLIN1	Differential: GPIOT_P_60_CDI77_EXTFB, GPIOT_N_60_CDI76 Single-ended: GPIOT_P_60_CDI77_EXTFB
TR2	Differential: GPIOT_P_62_CDI73_PLLIN0, GPIOT_N_62_CDI72 Single-ended: GPIOT_P_62_CDI73_PLLIN0	Single-ended: GPIOR_101_PLLIN1	Differential: GPIOT_P_61_CDI75_EXTFB, GPIOT_N_61_CDI74 Single-ended: GPIOT_P_61_CDI75_EXTFB
BR0	Differential: GPIOB_P_30_CDI9_PLLIN0, GPIOB_N_30_CDI8 Single-ended: GPIOB_P_30_CDI9_PLLIN0	Single-ended: GPIOR_140_PLLIN1	Differential: GPIOB_P_29_CDI11_EXTFB, GPIOB_N_29_CDI10 Single-ended: GPIOB_P_29_CDI11_EXTFB
BR1	Differential: GPIOB_P_39_CCK_PLLIN0, GPIOB_N_39_CDI2 Single-ended: GPIOB_P_39_CCK_PLLIN0	Single-ended: GPIOR_145_PLLIN1	Differential: GPIOB_P_40_CDI1_EXTFB, GPIOB_N_40_CDI0 Single-ended: GPIOB_P_40_CDI1_EXTFB
BR2		Single-ended: GPIOR_165_PLLIN1	

Programmable Duty Cycle

Tz325 FPGAs support a programmable duty cycle on the CLKOUT1 signal. A programmable duty cycle means that the clock's highs and lows can be different lengths (see Figure 43 on page 68).

If you turn on output clock inversion, the duty cycle setting is applied before the clock is inverted.

Figure 43: Programmable Duty Cycle Example



Important: You cannot use the programmable duty cycle at the same time as the fractional output divider.

Fractional Output Divider

Tz325 FPGAs have a fractional output divider, i.e., you can use an output divider that has a fractional part and an integer part. The advantage of the fractional part is that you can potentially get the clock output signals closer to a desired frequency.

Important: You cannot use the fractional output divider at the same time as the programmable duty cycle.

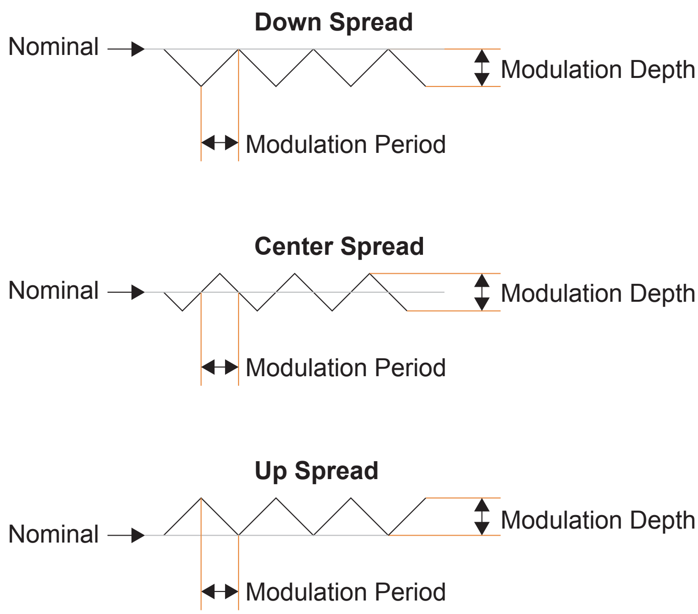
To use this feature you choose local feedback mode in the Interface Designer and specify the fractional options. The PLL feeds `CLKOUT1` back into the M counter, which causes the fractional part to propagate to all of the clock output signals.

Important: When using `CLKOUT1` for fractional feedback, you cannot use the output from `CLKOUT1` as a clock source for your design.

Spread-Spectrum Clocking

Tz325 FPGAs feature spread-spectrum clocking (SSC) with a modulation frequency from 30 kHz to 33 kHz and a modulation amplitude up to 0.5%.

Figure 44: Supported Modulation Types



Important: To use SSC, you must also enable fractional feedback mode. Refer to "Programmable Duty Cycle and Fractional Feedback" in the **Topaz Interfaces User Guide** .

Dynamic PLL Reconfiguration

Tz325 FPGAs support dynamic reconfiguration via signals from the core. You can reconfigure most of the PLL's internal blocks, including:

- N, M, and O counters
- Delta sigma modulator
- Output dividers
- Phase shift
- Output delay
- Inversion on the clock outputs

Dynamic Phase Shift

Tz325 FPGAs support a dynamic phase shift where you can adjust the phase shift of each output dynamically in user mode by up to 3.5 F_{PLL} cycles. For example, to phase shift a 400 MHz clock by 90-degree, configure the PLL to have a F_{PLL} frequency of 800 MHz, set the output counter division to 2, and set `SHIFT[2:0]` to `001`.

Implementing Dynamic Phase Shift

Use these steps to implement the dynamic phase shift:

1. Write the new phase setting into `SHIFT[2:0]`.
2. After 1 clock cycle of the targeted output clock that you want to shift, assert the `SHIFT_SEL[n]` and `SHIFT_ENA` signals.
3. Hold `SHIFT_ENA` and `SHIFT_SEL[n]` high for a minimum period of 4 clock cycles of the targeted output clock.
4. De-assert `SHIFT_ENA` and `SHIFT_SEL[n]`. Wait for at least 4 clock cycles of the targeted output clock before asserting `SHIFT_ENA` and `SHIFT_SEL[n]` again.

Note: *n* in `SHIFT_SEL[n]` represents the output clock that you intend to add phase shift.

The following waveforms describe the signals for a single phase shift and consecutive multiple phase shifts.

Figure 45: Single Dynamic Phase Shift Waveform Example for CLKOUT1

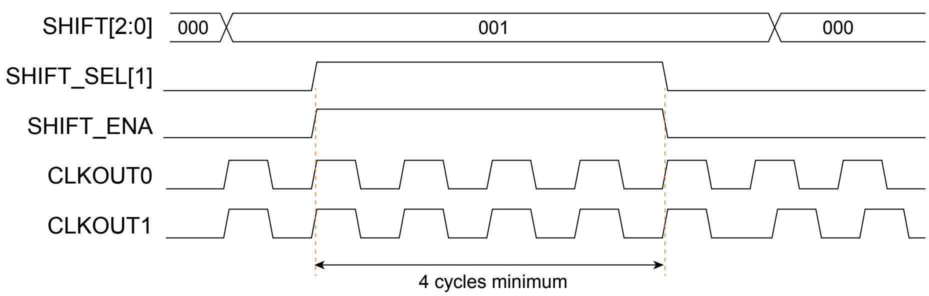
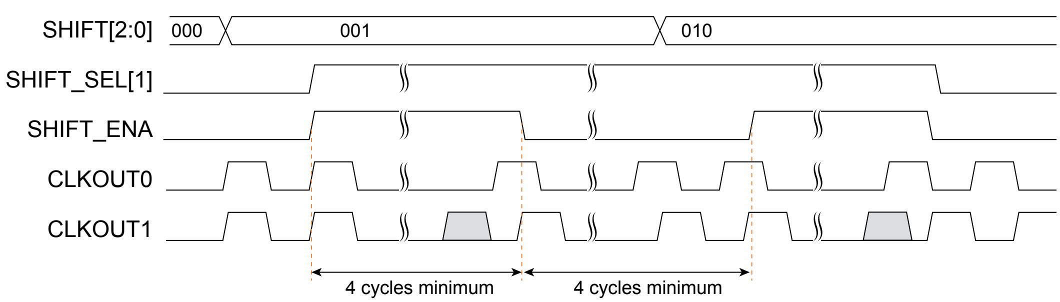


Figure 46: Consecutive Dynamic Phase Shift Waveform Example for CLKOUT1



Spread-Spectrum Clocking PLL

The Tz325 MIPI D-PHY interface includes a spread-spectrum clocking (SSC) PLL that spreads or varies the signal spectrum around the ideal clock frequency. If you are not using the MIPI D-PHY TX interface for MIPI signals, you can use the SSC PLL as another clock source.

The PLL consists of a pre-divider counter (N counter), a feedback multiplier counter (M counter), a post-divider counter (O counter), and output divider (C). You cannot modify the counter settings. Instead, you specify the output frequency you want and the reference clock frequency. If the SSC PLL cannot exactly match the output frequency, it displays (and uses) the frequency that is closest to your setting.

By default, the SSC PLL acts as a regular PLL. You enable the spread-spectrum clocking by turning on the **Enable Spread Spectrum Clock (SSC)** option in the Interface Designer.

Figure 47: SSC PLL Block Diagram

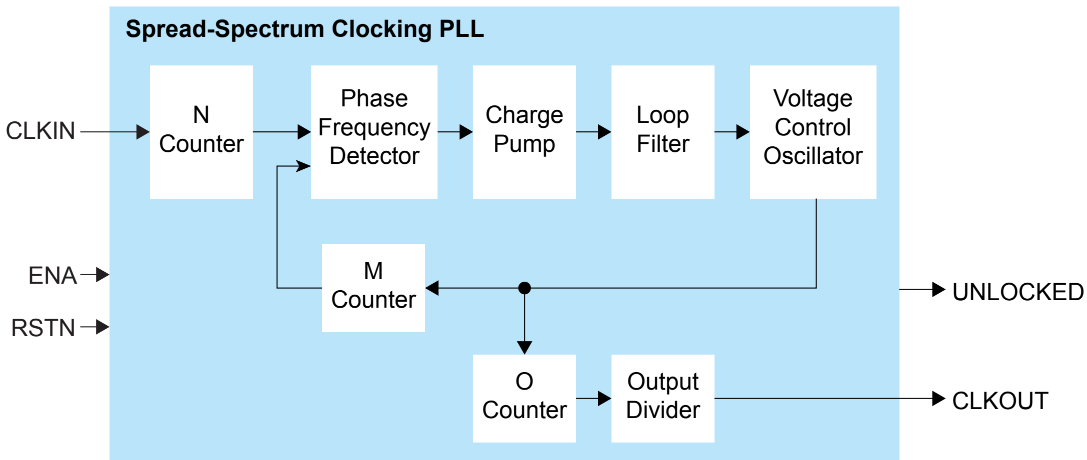


Figure 48: SSC PLL Interface Block Diagram

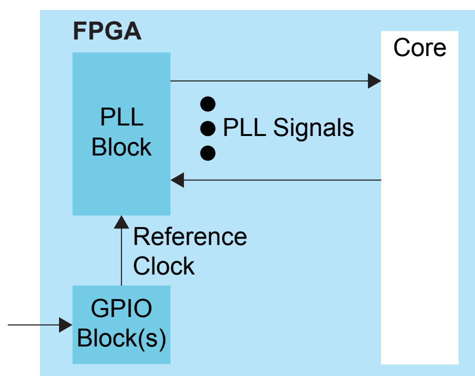


Table 51: SSC PLL Signals (Interface to FPGA Fabric)

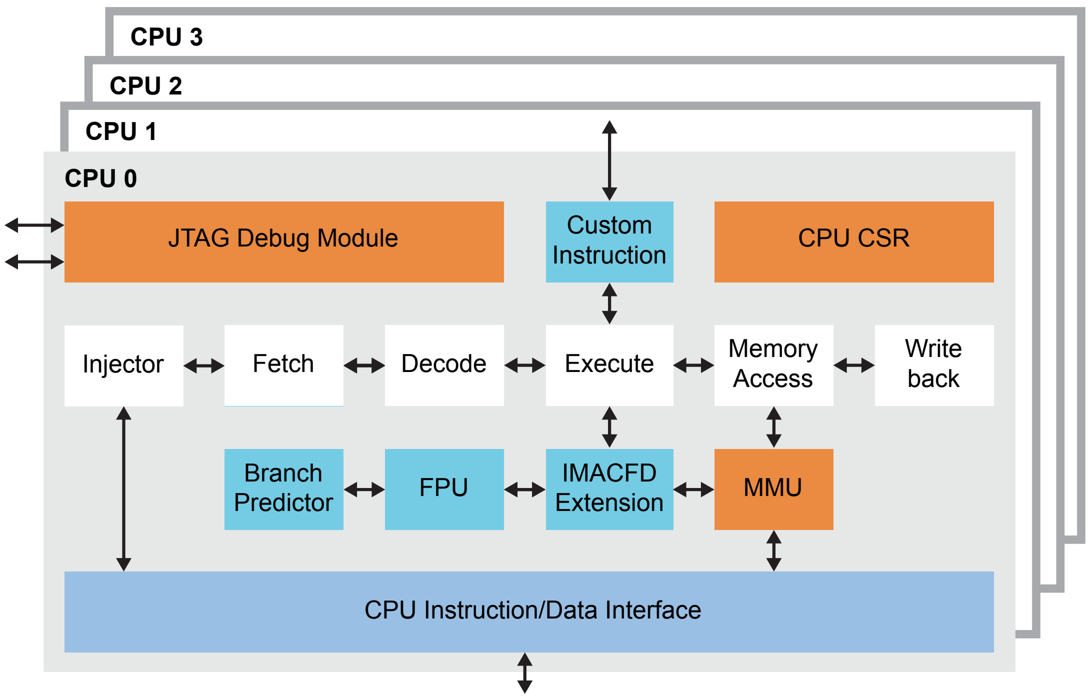
Signal	Direction	Description
CLKIN	Input	Reference clocks from core, PLL, or GPIO.
CLKOUT	Output	PLL SSC Clock Out Pin Name.
RSTN	Input	Active-low PLL SSC reset signal.
UNLOCKED	Output	PLL Unlock State Pin Name. Goes high when PLL SSC is in unlock state. Connect this signal in your design to monitor the lock status.
ENA	Input	(Optional) PLL SSC Enable Pin Name: Always enable: 1 Disable: 0 Can be driven by an active signal for dynamic enable.

Hardened RISC-V Block Interface

Important: All information is preliminary and pending definition.

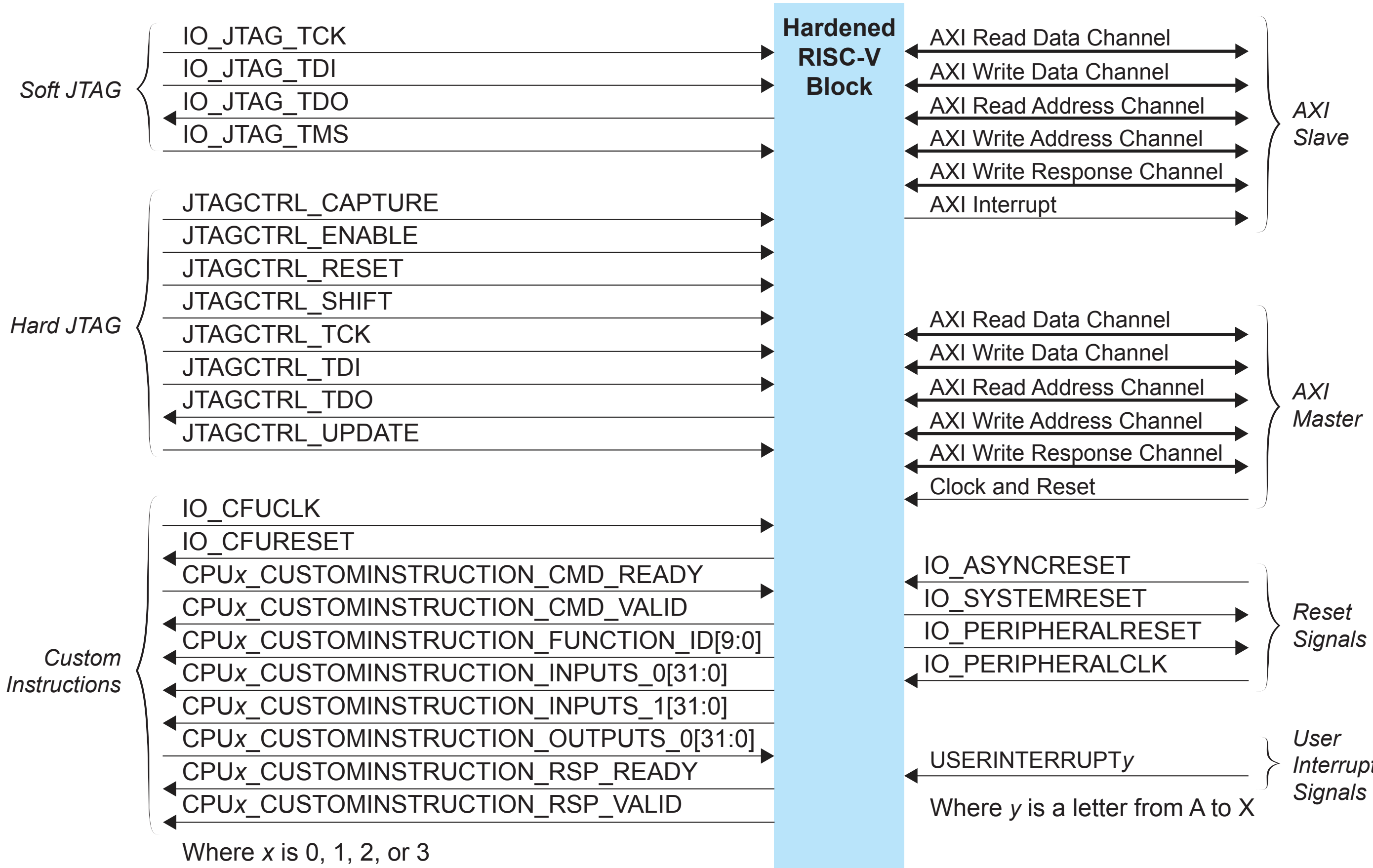
Tz325 FPGAs have a hardened RISC-V block with a 32-bit CPU featuring the ISA RISCV32I with M, A, C, F, and D extensions, and six pipeline stages (fetch, injector, decode, execute, memory, and writeback). The hard processor has 4 CPUs each with a dedicated FPU and custom instructions. The processor supports the standard RISC-V debug specification with 8 hardware breakpoints as well as machine and supervisor privileged mode, and Linux MMU SV32 page-based virtual memory.

Figure 49: Hardened RISC-V Block Overview



This topic provides an overview of the hardened RISC-V block and the signals that connect to the Tz325's core fabric and interfaces. For complete details on the processor and it's specifications, refer to the Sapphire High-Performance SoC Data Sheet.

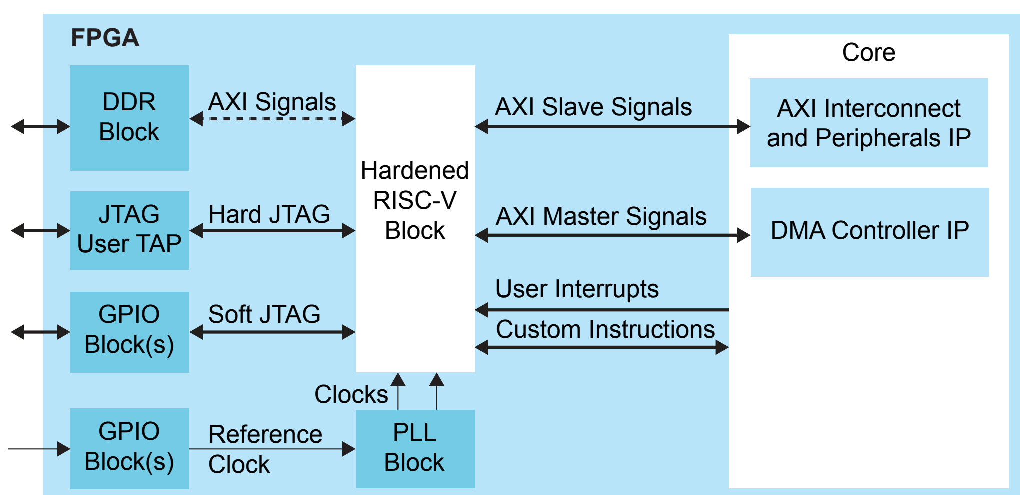
Figure 50: Hardened RISC-V Block Diagram



The hardened RISC-V block is connected directly to port 1 on the hard LPDDR4 controller; therefore, you do not need to implement those connections. Your design should implement the other interface blocks as needed.

Note: See the Topaz Interfaces User Guide for more details.

Figure 51: Hardened RISC-V Block Interface Block Diagram



The PLLs that can feed the RISC-V system clock are BL0 CLOCKOUT1, BL1 CLOCKOUT2, or BL2 CLOCKOUT1.

The PLLs that can feed the RISC-V memory clock are BL0 CLKOUT2, BL1 CLKOUT1, or BL2 CLKOUT2.

Note: The PLL that clocks the hardened RISC-V block should not use fractional output or spread-spectrum clocking because these features increase jitter.

AXI4 Slave Interface for Peripherals

The AXI slave interface connects to user-defined peripherals through an AXI interconnect bus. You use the IP Manager to build the AXI interconnect and peripherals.

Table 52: AXI Interrupt

Port	Direction	Clock Domain
AXIAINTERRUPT	Output	io_peripheralClk

Table 53: AXI Slave Read Address Channel

Port	Direction	Clock Domain	Description
AXIA_ARADDR[31:0]	Input	io_peripheralClk	Read address. It gives the address of the first transfer in a burst transaction.
AXIA_ARBURST[1:0]	Input	io_peripheralClk	Burst type. The burst type and the size determine how the address for each transfer within the burst is calculated. 'b01 = INCR 'b10 = WRAP
AXIA_ARCACHE[3:0]	Input	io_peripheralClk	Memory type. This signal indicates how transactions are required to progress through a system.
AXIA_ARLEN[7:0]	Input	io_peripheralClk	Burst length. This signal indicates the number of transfers in a burst.
AXIA_ARLOCK	Input	io_peripheralClk	Lock type. This signal provides additional information about the atomic characteristics of the transfer.
AXIA_ARPROT[2:0]	Input	io_peripheralClk	Defines the access permissions for read accesses.
AXIA_ARQOS[3:0]	Input	io_peripheralClk	QoS identifier for read transaction.
AXIA_ARREADY	Output	io_peripheralClk	Address ready. This signal indicates that the slave is ready to accept an address and associated control signals.
AXIA_ARREGION[3:0]	Input	io_peripheralClk	Region identifier. Permits a single physical interface on a slave to be used for multiple logical interfaces.
AXIA_ARSIZE[2:0]	Input	io_peripheralClk	Burst size. This signal indicates the size of each transfer in the burst.
AXIA_ARVALID	Input	io_peripheralClk	Address valid. This signal indicates that the channel is signaling valid address and control information.

Table 54: AXI Slave Write Address Channel

Port	Direction	Clock Domain	Description
AXIA_AWADDR[31:0]	Input	io_peripheralClk	Write address. It gives the address of the first transfer in a burst transaction.

Port	Direction	Clock Domain	Description
AXIA_AWBURST[1:0]	Input	io_peripheralClk	Burst type. The burst type and the size determine how the address for each transfer within the burst is calculated.
AXIA_AWCACHE[3:0]	Input	io_peripheralClk	Memory type. This signal indicates how transactions are required to progress through a system.
AXIA_AWLEN[7:0]	Input	io_peripheralClk	Burst length. This signal indicates the number of transfers in a burst.
AXIA_AWLOCK	Input	io_peripheralClk	Lock type. This signal provides additional information about the atomic characteristics of the transfer.
AXIA_AWPROT[2:0]	Input	io_peripheralClk	Defines the access permissions for write accesses.
AXIA_AWQOS[3:0]	Input	io_peripheralClk	QoS identifier for write transaction.
AXIA_AWREADY	Output	io_peripheralClk	Address ready. This signal indicates that the slave is ready to accept an address and associated control signals.
AXIA_AWREGION[3:0]	Input	io_peripheralClk	Region identifier. Permits a single physical interface on a slave to be used for multiple logical interfaces.
AXIA_AWSIZE[2:0]	Input	io_peripheralClk	Burst size. This signal indicates the size of each transfer in the burst.
AXIA_AWVALID	Input	io_peripheralClk	Address valid. This signal indicates that the channel is signaling valid address and control information.

Table 55: AXI Slave Write Response Channel

Port	Direction	Clock Domain	Description
AXIA_BREADY	Input	io_peripheralClk	Response ready. This signal indicates that the master can accept a write response.
AXIA_BRESP[1:0]	Output	io_peripheralClk	Read response. This signal indicates the status of the read transfer.
AXIA_BVALID	Output	io_peripheralClk	Write response valid. This signal indicates that the channel is signaling a valid write response.

Table 56: AXI Slave Read Data Channel

Port	Direction	Clock Domain	Description
AXIA_RDATA[31:0]	Output	io_peripheralClk	Read data.
AXIA_RLAST	Output	io_peripheralClk	Read last. This signal indicates the last transfer in a read burst.
AXIA_RREADY	Input	io_peripheralClk	Read ready. This signal indicates that the master can accept the read data and response information.
AXIA_RRESP[1:0]	Output	io_peripheralClk	Read response. This signal indicates the status of the read transfer.
AXIA_RVALID	Output	io_peripheralClk	Read valid. This signal indicates that the channel is signaling the required read data.

Table 57: AXI Slave Write Data Channel

Port	Direction	Clock Domain	Description
AXIA_WDATA[31:0]	Input	io_peripheralClk	Write data.
AXIA_WLAST	Input	io_peripheralClk	Write last. This signal indicates the last transfer in a write burst.
AXIA_WREADY	Output	io_peripheralClk	Write ready. This signal indicates that the slave can accept the write data.
AXIA_WSTRB[3:0]	Input	io_peripheralClk	Write strobes. This signal indicates which byte lanes hold valid data. There is one write strobe bit for each eight bits of the write data bus.
AXIA_WVALID	Input	io_peripheralClk	Write valid. This signal indicates that valid write data and strobes are available.

AXI Interface to DMA

This AXI master interface has a 128-bit data channel to connect to the DMA controller IP core.

Table 58: Clock and Reset

Port	Direction	Clock Domain	Description
IO_DDRMASTERS_0_CLK	Input		
IO_DDRMASTERS_0_RESET	Input	IO_DDRMASTERS_0_CLK	

Table 59: AXI Master Read Address Channel

Port	Direction	Clock Domain	Description
IO_DDRMASTERS_0_ AR_PAYLOAD_ADDR[31:0]	Output	IO_DDRMASTERS_0_CLK	Read address. It gives the address of the first transfer in a burst transaction.
IO_DDRMASTERS_0 _AR_PAYLOAD_BURST[1:0]	Output	IO_DDRMASTERS_0_CLK	Burst type. The burst type and the size determine how the address for each transfer within the burst is calculated. 'b01 = INCR 'b10 = WRAP
IO_DDRMASTERS_0 _AR_PAYLOAD_CACHE[3:0]	Output	IO_DDRMASTERS_0_CLK	Memory type. This signal indicates how transactions are required to progress through a system.
IO_DDRMASTERS_0 _AR_PAYLOAD_ID[3:0]	Output	IO_DDRMASTERS_0_CLK	Address ID. This signal identifies the group of address signals.
IO_DDRMASTERS_0 _AR_PAYLOAD_LEN[7:0]	Output	IO_DDRMASTERS_0_CLK	Burst length. This signal indicates the number of transfers in a burst.
IO_DDRMASTERS_0 _AR_PAYLOAD_LOCK	Output	IO_DDRMASTERS_0_CLK	Lock type. This signal provides additional information about the atomic characteristics of the transfer.
IO_DDRMASTERS_0 _AR_PAYLOAD_PROT[2:0]	Output	IO_DDRMASTERS_0_CLK	Defines the access permissions for read accesses.
IO_DDRMASTERS_0 _AR_PAYLOAD_QOS[3:0]	Output	IO_DDRMASTERS_0_CLK	QoS identifier for read transaction.

Port	Direction	Clock Domain	Description
IO_DDRMASTERS_0 _AR_PAYLOAD_REGION[3:0]	Output	IO_DDRMASTERS_0_CLK	Region identifier. Permits a single physical interface to be used for multiple logical interfaces.
IO_DDRMASTERS_0 _AR_PAYLOAD_SIZE[2:0]	Output	IO_DDRMASTERS_0_CLK	Burst size. This signal indicates the size of each transfer in the burst.
IO_DDRMASTERS_0 _AR_READY	Input	IO_DDRMASTERS_0_CLK	Address ready. This signal indicates that the slave is ready to accept an address and associated control signals.
IO_DDRMASTERS_0_AR_VALID	Output	IO_DDRMASTERS_0_CLK	Address valid. This signal indicates that the channel is signaling valid address and control information.

Table 60: AXI Master Write Address Channel

Port	Direction	Clock Domain	Description
IO_DDRMASTERS_0 _AW_VALID	Output	IO_DDRMASTERS_0_CLK	Address valid. This signal indicates that the channel is signaling valid address and control information.
IO_DDRMASTERS_0 _AW_READY	Input	IO_DDRMASTERS_0_CLK	Address ready. This signal indicates that the slave is ready to accept an address and associated control signals.
IO_DDRMASTERS_0 _AW_PAYLOAD_ADDR[31:0]	Output	IO_DDRMASTERS_0_CLK	Write address. It gives the address of the first transfer in a burst transaction.
IO_DDRMASTERS_0 _AW_PAYLOAD_ID[3:0]	Output	IO_DDRMASTERS_0_CLK	Address ID. This signal identifies the group of address signals.
IO_DDRMASTERS_0 _AW_PAYLOAD_REGION[3:0]	Output	IO_DDRMASTERS_0_CLK	Region identifier. Permits a single physical interface to be used for multiple logical interfaces.
IO_DDRMASTERS_0 _AW_PAYLOAD_LEN[7:0]	Output	IO_DDRMASTERS_0_CLK	Burst length. This signal indicates the number of transfers in a burst.
IO_DDRMASTERS_0 _AW_PAYLOAD_SIZE[2:0]	Output	IO_DDRMASTERS_0_CLK	Burst size. This signal indicates the size of each transfer in the burst.
IO_DDRMASTERS_0 _AW_PAYLOAD_BURST[1:0]	Output	IO_DDRMASTERS_0_CLK	Burst type. The burst type and the size determine how the address for each transfer within the burst is calculated.
IO_DDRMASTERS_0 _AW_PAYLOAD_LOCK	Output	IO_DDRMASTERS_0_CLK	Lock type. This signal provides additional information about the atomic characteristics of the transfer.
IO_DDRMASTERS_0 _AW_PAYLOAD_CACHE[3:0]	Output	IO_DDRMASTERS_0_CLK	Memory type. This signal indicates how transactions are required to progress through a system.
IO_DDRMASTERS_0 _AW_PAYLOAD_QOS[3:0]	Output	IO_DDRMASTERS_0_CLK	QoS identifier for write transaction.
IO_DDRMASTERS_0 _AW_PAYLOAD_PROT[2:0]	Output	IO_DDRMASTERS_0_CLK	Defines the access permissions for write accesses.

Port	Direction	Clock Domain	Description
IO_DDRMASTERS_0 _AW_PAYLOAD_ALLSTRB	Output	IO_DDRMASTERS_0_CLK	Write all strobes asserted. The DDR controller only supports a maximum of 16 AXI beats for write commands using this signal.

Table 61: AXI Master Write Response Channel

Port	Direction	Clock Domain	Description
IO_DDRMASTERS_0 _B_PAYLOAD_ID[3:0]	Input	IO_DDRMASTERS_0_CLK	Response ID tag. This signal is the ID tag of the write response.
IO_DDRMASTERS_0 _B_PAYLOAD_RESP[1:0]	Input	IO_DDRMASTERS_0_CLK	Read response. This signal indicates the status of the read transfer.
IO_DDRMASTERS_0_B_READY	Output	IO_DDRMASTERS_0_CLK	Response ready. This signal indicates that the master can accept a write response.
IO_DDRMASTERS_0_B_VALID	Input	IO_DDRMASTERS_0_CLK	Write response valid. This signal indicates that the channel is signaling a valid write response.

Table 62: AXI Master Read Data Channel

Port	Direction	Clock Domain	Description
IO_DDRMASTERS_0 _R_PAYLOAD_DATA[127:0]	Input	IO_DDRMASTERS_0_CLK	Read data.
IO_DDRMASTERS_0 _R_PAYLOAD_ID[3:0]	Input	IO_DDRMASTERS_0_CLK	Read ID tag. This signal is the identification tag for the read data group of signals generated by the slave.
IO_DDRMASTERS_0 _R_PAYLOAD_LAST	Input	IO_DDRMASTERS_0_CLK	Read last. This signal indicates the last transfer in a read burst.
IO_DDRMASTERS_0 _R_PAYLOAD_RESP[1:0]	Input	IO_DDRMASTERS_0_CLK	Read response. This signal indicates the status of the read transfer.
IO_DDRMASTERS_0_R_READY	Output	IO_DDRMASTERS_0_CLK	Read ready. This signal indicates that the master can accept the read data and response information.
IO_DDRMASTERS_0_R_VALID	Input	IO_DDRMASTERS_0_CLK	Read valid. This signal indicates that the channel is signaling the required read data.

Table 63: AXI Master Write Data Channel

Port	Direction	Clock Domain	Description
IO_DDRMASTERS_0_W_VALID	Output	IO_DDRMASTERS_0_CLK	Write valid. This signal indicates that valid write data and strobes are available.
IO_DDRMASTERS_0 _W_READY	Input	IO_DDRMASTERS_0_CLK	Write ready. This signal indicates that the slave can accept the write data.
IO_DDRMASTERS_0 _W_PAYLOAD_DATA[127:0]	Output	IO_DDRMASTERS_0_CLK	Write data.

Port	Direction	Clock Domain	Description
IO_DDRMASTERS_0 _W_PAYLOAD_LAST	Output	IO_DDRMASTERS_0_CLK	Write last. This signal indicates the last transfer in a write burst.
IO_DDRMASTERS_0 _W_PAYLOAD_STRB[15:0]	Output	IO_DDRMASTERS_0_CLK	Write strobes. This signal indicates which byte lanes hold valid data. There is one write strobe bit for each eight bits of the write data bus.

JTAG Signals

The hardened RISC-V block includes two sets of JTAG signals. The soft JTAG connects to I/O blocks while the hard JTAG connects to the JTAG User TAP interface block.

Table 64: Soft JTAG Ports

Port	Direction	Clock Domain	
IO_JTAG_TCK	Input	IO_JTAG_TCK	JTAG test clock pin.
IO_JTAG_TDI	Input	IO_JTAG_TCK	JTAG test data in pin.
IO_JTAG_TDO	Output	IO_JTAG_TCK	JTAG test data out pin.
IO_JTAG_TMS	Input	IO_JTAG_TCK	JTAG mode select pin.

Table 65: Hard JTAG Ports

Port	Direction	Clock Domain	Description
JTAGCTRL_CAPTURE	Input	JTAGCTRL_TCK	Capture pin.
JTAGCTRL_ENABLE	Input	JTAGCTRL_TCK	Enable the JTAG user TAP interface.
JTAGCTRL_RESET	Input	JTAGCTRL_TCK	Reset.
JTAGCTRL_SHIFT	Input	JTAGCTRL_TCK	Shift pin.
JTAGCTRL_TCK	Input	JTAGCTRL_TCK	JTAG test clock pin.
JTAGCTRL_TDI	Input	JTAGCTRL_TCK	JTAG test data in pin.
JTAGCTRL_TDO	Output	JTAGCTRL_TCK	JTAG test data out pin.
JTAGCTRL_UPDATE	Input	JTAGCTRL_TCK	Update pin.

Custom Instruction Signals

The hardened RISC-V interface has two 32-bit custom instruction interfaces for each CPU. The custom instructions use a type R opcode.

Table 66: Custom Instructions

Where *n* is 0, 1, 2, or 3 for the CPU number.

Port	Direction	Clock Domain	Description
IO_CFUCLK	Input		Clock.
IO_CFURESET	Output	IO_CFUCLK	Reset.
CPU*n*_CUSTOMINSTRUCTION _CMD_READY	Input	IO_CFUCLK	Indicates that the custom processing logic is ready to process register rs1 and rs2 from the CPU.

Port	Direction	Clock Domain	Description
CPU*n*_CUSTOMINSTRUCTION_CMD_VALID	Output	IO_CFUCLK	Indicates that registers rs1 and rs2 are present and ready for processing.
CPU*n*_CUSTOMINSTRUCTION_FUNCTION_ID[9:0]	Output	IO_CFUCLK	Function id for the custom instruction.
CPU*n*_CUSTOMINSTRUCTION_INPUTS_0[31:0]	Output	IO_CFUCLK	Register rs1 for the custom instruction.
CPU*n*_CUSTOMINSTRUCTION_INPUTS_1[31:0]	Output	IO_CFUCLK	Register rs2 for the custom instruction.
CPU*n*_CUSTOMINSTRUCTION_OUTPUTS_0[31:0]	Input	IO_CFUCLK	Result of the custom instruction.
CPU*n*_CUSTOMINSTRUCTION_RSP_READY	Output	IO_CFUCLK	Indicates that the CPU is ready to accept the custom instruction result.
CPU*n*_CUSTOMINSTRUCTION_RSP_VALID	Output	IO_CFUCLK	Indicates that the custom instruction result is available.

User Interrupt Signals

Table 67: User Interrupts

Where *n* is a letter A-X.

Port	Direction	Description
USERINTERRUPT*n*	Input	Interrupt signal for a peripheral.

Clock Signals

Table 68: Reset

Port	Direction	Description
IO_PERIPHERALCLK	Input	Provides a clock for the peripherals and AXI slave interface.

Reset Signals

Table 69: Reset

Port	Direction	Description
IO_ASYNCRESET	Input	Active-high asynchronous reset for the entire system.
IO_SYSTEMRESET	Output	Synchronous active-high reset for the system clock.
IO_PERIPHERALRESET	Output	Synchronous active-high reset for the peripheral clock (io_peripheralClock).

Transceiver Interface

The Tz325 high-speed transceiver interface is a multi-protocol, full duplex transceiver that supports data rates from 1.25 Gbps to 12.5 Gbps. It supports the PCIe 3.0, SGMII, and 10GBase-KR protocols as well as a PMA Direct mode. You can use it in x1, x2, or x4 configuration.

Note: The Tz325 FPGA has 2 transceiver quads, and each quad has 4 lanes. All quads support SGMII, 10GBase-KR, and PMA Direct. 1 quad supports PCIe.

Table 70: Supported Protocols

Standard	Data Rate (Gbps) per Lane	Number of Lanes	Specification
PCIe Gen 1	2.5	x1, x2, x4	PCI Express® Base Specification Revision 1.1
PCIe Gen 2	5	x1, x2, x4	PCI Express® Base Specification Revision 2.1
PCIe Gen 3	8	x1, x2, x4	PCI Express® Base Specification Revision 3.0
SGMII	1.25	x1	IEEE std 802.3 clause 36
10GBase-KR	10.3125	x1	IEEE std 802.3ap-2007
PMA Direct	1.25 - 12.5	x1, x2, x4	-

Figure 52: Transceiver Used as PCIe

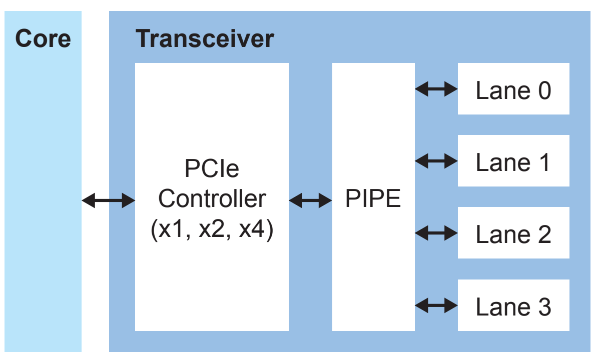
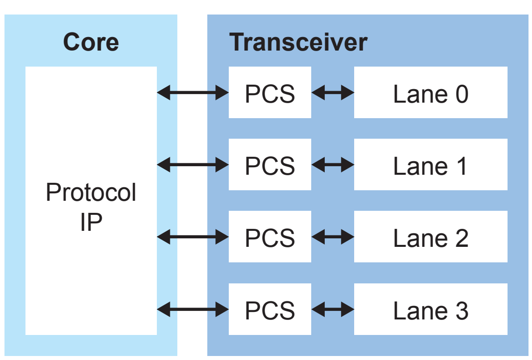


Figure 53: Transceiver Used as SGMII, 10GBase-KR, or PMA Direct



The PCS supports SGMII, 10GBase-KR, or PMA Direct for each lane

Single-Event Upset Detection

The Tz325 FPGA has a hard block for detecting single-event upset (SEU). The SEU detection feature has two modes:

- *Auto mode*—The Tz325 control block periodically runs SEU error checks and flags if it detects an error. You can configure the interval time between SEU checks.
- *Manual mode*—The user design runs the check.

In both modes, the user design is responsible for deciding whether to reconfigure the Tz325 when an error is detected.

Learn more: For more information on using the SEU detection feature, refer to the **Topaz Interfaces User Guide**.

Internal Reconfiguration Block

The Tz325 FPGAs have built-in hardware that supports an internal reconfiguration feature. The Tz325 can reconfigure itself from a bitstream image stored in flash memory.

Security Feature

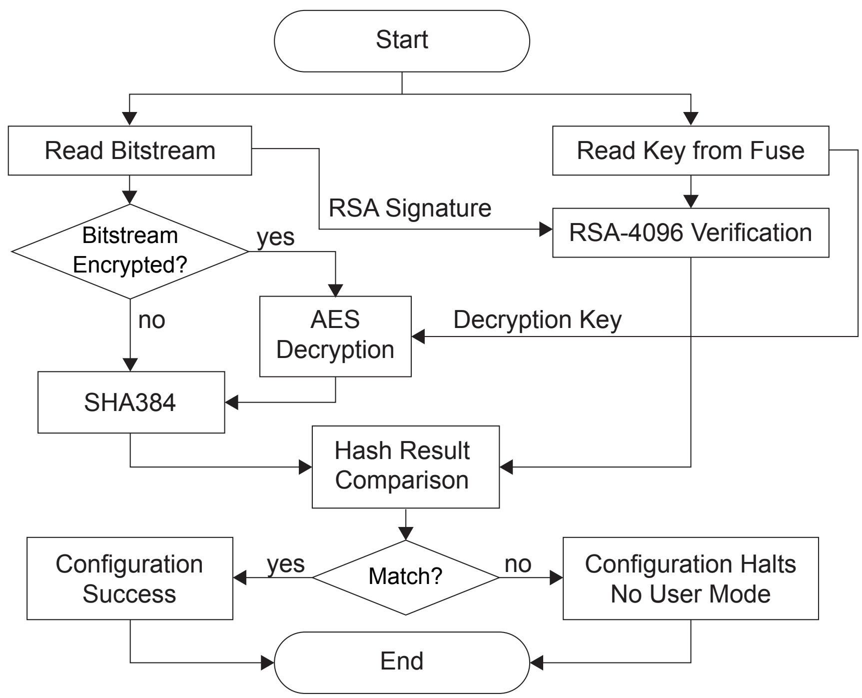
The FPGA security feature includes:

- Intellectual property protection using bitstream encryption with the AES-GCM-256 algorithm
- Anti-tampering support using asymmetric bitstream authentication with the RSA-4096 algorithm

Important: You cannot enable the FPGA security features when using compressed bitstreams.

You can enable encryption, authentication, or both. You enable the security features at the project level.

Figure 54: Security Flow



Bitstream Encryption

Symmetric bitstream encryption uses a 256-bit key and the AES-GCM-256 algorithm. You create the key and then use it to encrypt the bitstream. You also need to store the key into the FPGA's fuses. During configuration, the built-in AES-GCM-256 engine decrypts the encrypted configuration bitstream using the stored key. Without the correct key, the bitstream decryption process cannot recover the original bitstream.

Bitstream Authentication

For bitstream authentication, you use a public/private key pair and the RSA-4096 algorithm. You create a public/private key pair and sign the bitstream with the private key. Then, you save a hashed version of the public key into fuses in the FPGA. During configuration, the FPGA validates the signature on the bitstream using the public key.

If the signature is valid, the FPGA knows that the bitstream came from a trusted source and has not been altered by a third party. The FPGA continues configuring normally and goes into user mode. If the signature is invalid, the FPGA stops configuration and does not go into user mode.

The private key remains on your computer and is not shared with anyone. The FPGA only has the public key: the bitstream contains the public key data and a signature, while the fuses contain a hashed public key. You can only sign the bitstream with the private key. An attacker cannot re-sign a tampered bitstream without the private key.

Disabling JTAG Access

Tz325 FPGA's support JTAG blocking, which disables JTAG access to the FPGA by blowing a fuse. Once the fuse is blown, you cannot perform any JTAG operation except for reading the FPGA `IDCODE`, reading `DEVICE_STATUS`, using `SAMPLE/PRELOAD`, and enabling `BYPASS` mode. To fully secure the FPGA, you **must** blow the JTAG fuse.

If you still want to use the JTAG interface for debugging, you can use the **DISABLE_EFUSE_ONLY** option, which permanently disables the JTAG efuse instructions only. Other JTAG instructions are not affected, for example, you can still perform debugging. Refer to "Using the Efinity Bitstream Security Key Generator" in the **Efinity Software User Guide** for more information.

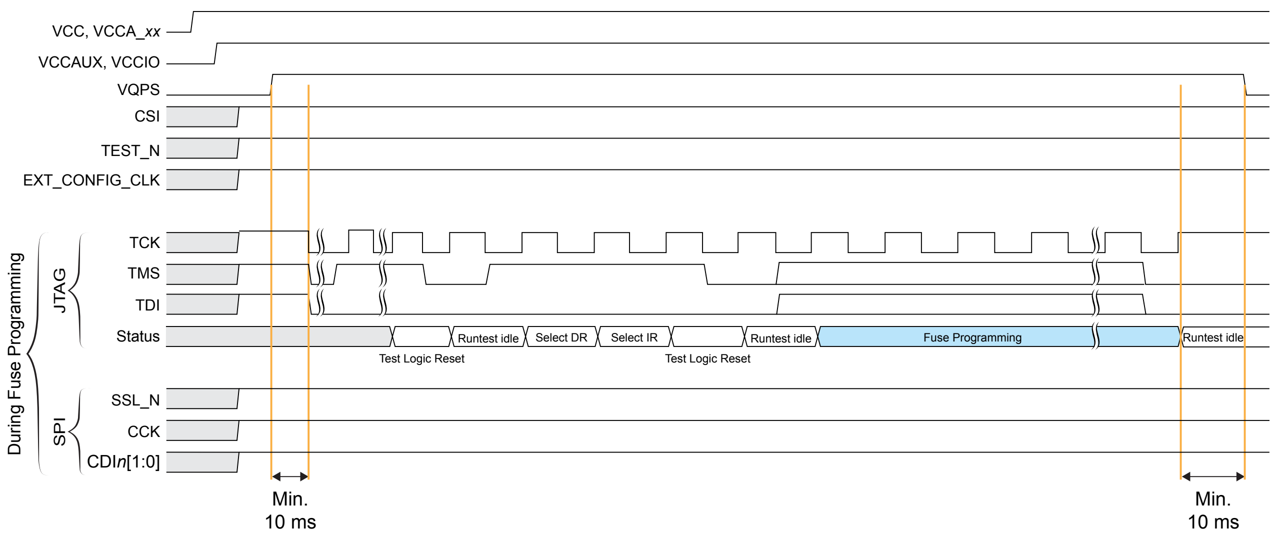
Important: Once you disable JTAG by blowing the fuse, however, you cannot use JTAG ever again in that FPGA (except for `IDCODE`, `DEVICE_STATUS`, `SAMPLE/PRELOAD`, and `BYPASS`). So blowing this fuse should be the very last step in your manufacturing process.

Fuse Programming Requirements

Important: The VQPS supply current requires a minimum of 100 mA.

To program the security fuses in FPGA, follow these requirements:
- During fuse programming, avoid device configuration and other JTAG operations that are not related to fuse programming.
- Ramp up the `VQPS` pin only after all other power supplies have ramped to their nominal voltages. The `VQPS` ramp rate follows the requirements shown in Table 78: Power Supply Ramp Rates on page 93.
- After powering up the `VQPS` pin, wait for a minimum of 10 ms before issuing JTAG instructions for fuse programming.
- After completing fuse programming through JTAG, wait for a minimum of 10 ms before powering down the `VQPS` pin.
- If required, other power supplies can be powered down only after the `VQPS` pin has been powered down below 25 % of its nominal voltage level.

Figure 55: Fuse Programming Waveform



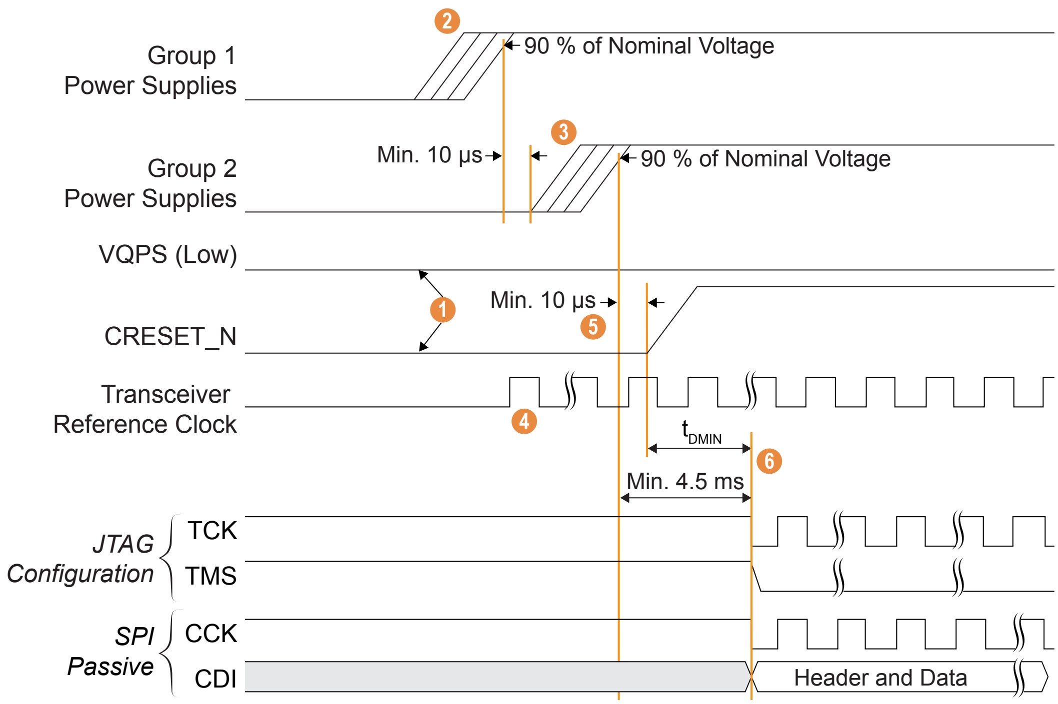
Important: The SPI bus must be inactive during fuse programming.
The `EXT_CONFIG_CLK` pin must be inactive during fuse programming.

Power Sequence

Important: You **must** follow the power-up and power-down sequence when powering Topaz FPGAs.

Power-Up Sequence

Figure 56: Power-Up Sequence



Important: You can only use one configuration channel at a time. Using SPI passive and JTAG at the same time can result in configuration failure.

1. The `CRESET_N` input must stay **low** until all power supplies are powered up. Additionally, `VQPS` must **always** stay **low** unless you are blowing the Tz325 security fuses.

   Note: Refer to Fuse Programming Requirements on page 85 if you need to blow the security fuses for the Tz325 FPGA on your board.

2. Power up supplies in group 1 first. You can power up these supplies in any sequence.

   Important: Ensure the power ramp rate is within the values shown in Table 78: Power Supply Ramp Rates on page 93.

3. Power up the group 2 supplies in any sequence at a minimum delay of 10 μs after group 1 supplies have reached 90% of their nominal voltage levels.
4. If you are using transceivers, the external reference clock must be ready before `CRESET_N` is released.
5. Release the `CRESET_N` input to high at a minimum delay of 10 μs after all supplies have reached 90% of their nominal voltage levels.

6. FPGA configuration can begin after there has been:
   - A 4.5 ms minimum delay after all supplies have reached at least 90% of their nominal voltage.
   - A t_{DMIN} minimum delay after `CRESET_N` goes high (see **SPI Passive Mode** on page 108 and **JTAG Mode** on page 106 for the delay specification).

Note: With the configuration bitstream stored in the SPI flash device and the SPI active hardware connection properly established, the SPI active configuration automatically starts after the `CRESET_N` signal transitions from low to high.

Table 71: Power-Up Groups

If you are blowing the security fuses, refer to **Fuse Programming Requirements** on page 85.

Power-Up Sequence	
Group 1	**Group 2**
VCC VCCA VDD_SOC	VCCAUX VCCIO VCCIO33
MIPI D-PHY	
–	VCC18A_MIPI_TX VCC18A_MIPI_RX
DDR DRAM controller	
VDD_PHY VDDPLL_MCB_TOP_PHY	VDDQ_PHY VDDQX_PHY VDDQ_CK_PHY
Transceivers(6)	
VCC_SERDES, VDDA_C_Q, VDDA_D_Q, VDDA_H_Q	

Note: Some DDR DRAM devices have a specific power-up sequence requirement. Ensure this requirement is met when the FPGA and memory share a power supply.

(6) The transceiver supplies can be powered up in any sequence.

Table 72: Connection Requirements for Unused Resources and Features

Unused Resource/Feature	Pin	Note
PLL	VCCA	Connect to VCC.
HSIO Bank	VCCIO	Connect to either 1.2 V, 1.35 V, 1.5 V, or 1.8 V.
HVIO Bank	VCCIO33	Connect to either 1.8 V, 2.5 V, 3.0 V, or 3.3 V.
MIPI	VCC18A_MIPI_TX VCC18A_MIPI_RX	Connect to VCC.
DDR	VDD_PHY	Leave unconnected.
	VDDPLL_MCB_TOP_PHY	Leave unconnected.
	VDDQ_PHY	Leave unconnected.
	VDDQX_PHY	Leave unconnected.
	VDDQ_CK_PHY	Leave unconnected.
Hardened RISC-V block	VDD_SOC	Leave unconnected.
Transceivers	VCC_SERDES	Leave unconnected.
	VDDA_C_Q VDDA_D_Q VDDA_H_Q	Leave unconnected.
Security (Fuse Blowing)	VQPS	Connect to GND.

Power-Down Sequence

There is no specific power-down sequence for Tz325 FPGAs. However, the `VQPS` power supply **must** follow the specifications in Fuse Programming Requirements on page 85.

Power Supply Current Transient

You may observe an inrush current on the dedicated power rail during power-up. You must ensure that the power supplies selected in your board meets the current requirement during power-up and the estimated current during user mode. Use the Power Estimator to calculate the estimated current during user mode.

Table 73: Minimum Power Supply Current Transient

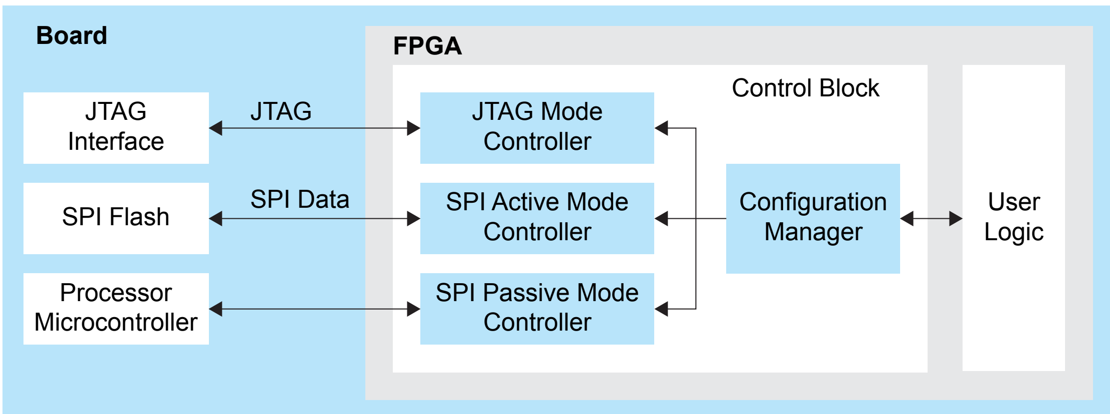
Power Supply	Minimum Power Supply Current Transient	Unit
VCC	1,500[7]	mA

[7] Preliminary

Configuration

The Tz325 FPGA contains volatile Configuration RAM (CRAM). The user must configure the CRAM for the desired logic function upon power-up and before the FPGA enters normal operation. The FPGA's control block manages the configuration process and uses a bitstream to program the CRAM. The Efinity® software generates the bitstream, which is design dependent. You can configure the Tz325 FPGA(s) in SPI active, SPI passive, or JTAG mode.

Figure 57: High-Level Configuration Options



In active mode, the FPGA controls the configuration process. The configuration clock can either be provided by an oscillator circuit within the FPGA or an external clock connected to the `EXT_CONFIG_CLK` pin. The bitstream is typically stored in an external serial flash device, which provides the bitstream when the FPGA requests it.

The control block sends out the instruction and address to read the configuration data. First, it issues a release from power-down instruction to wake up the external SPI flash. Then, it waits for at least 30 μs before issuing a fast read command to read the content of SPI flash from address 24h'000000 for 3-byte addressing mode or 32'h00000000 for 4-byte addressing mode.

In passive mode, the FPGA is the slave and relies on an external master to provide the control, bitstream, and clock for configuration. Typically the master is a microcontroller or another FPGA in active mode. The controller must wait for at least 32 μs after `CRESET` is de-asserted before it can send the bitstream.

In JTAG mode, you configure the FPGA via the JTAG interface.

Note: Tz325 FPGAs have a `JTAG_VCCIO_SEL` pin that selects the voltage to use for JTAG. Refer to Pinout Description on page 110 for more details.

Supported Configuration Modes

Table 74: Tz325 Configuration Modes by Package

Configuration Mode	Width	All Packages
Active	X1	✓
	X2	✓
	X4	✓
	X8	✓
Passive	X1	✓
	X2	✓
	X4	✓
	X8	✓
	X16	✓[8]
	X32	✓[8]
JTAG	X1	✓

[8] Not supported when security mode is enabled.

Characteristics and Timing

The following table shows the specification status for Tz325 packages.

Table 75: Package Status

Package	Status
N484, C529	Preliminary

DC and Switching Characteristics

Important: All specifications are preliminary and pending hardware characterization.

Table 76: Absolute Maximum Ratings[9]

Conditions beyond those listed may cause permanent damage to the device. Device operation at the absolute maximum ratings for extended periods of time has adverse effects on the device.

Symbol	Description	Min	Max	Units
VCC	Core power supply.	-0.5	1.05	V
VCCA	PLL analog power supply.	-0.5	1.05	V
VDD_SOC	Hardened RISC-V block power supply.	-0.5	1.05	V
VCCAUX	1.8 V auxiliary power supply.	-0.5	1.98	V
VQPS	1.8 V security fuse supply.	-0.5	1.98	V
VCCIO	HSIO bank power supply.	-0.5	1.98	V
VCCIO33	HVIO bank power supply.	-0.5	3.63	V
VCC18A_MIPI_TX	1.8 V TX analog power supply for MIPI TX.	-0.5	1.98	V
VCC18A_MIPI_RX	1.8 V TX analog power supply for MIPI RX.	-0.5	1.98	V
VDD_PHY	LPDDR4 digital power supply.	-0.5	1.05	V
VDDPLL_MCB_TOP_PHY	LPDDR4 PLL power supply.	-0.5	1.05	V
VDDQ_PHY	LPDDR4 I/O power supply.	-0.5	1.21	V
VDDQX_PHY	LPDDR4 I/O pre-driver power supply.	-0.5	1.21	V
VDDQ_CK_PHY	LPDDR4 I/O power supply for clock.	-0.5	1.21	V
VCC_SERDES	Transceiver digital PCS and PCIe controller power supplies.	-0.5	1.05	V
VDDA_C_Q	Transceiver analog bias power supply.	-0.5	1.05	V
VDDA_D_Q	Transceiver digital and analog data path power supplies.	-0.5	1.05	V
VDDA_H_Q	Transceiver analog power supply for I/O.	-0.5	1.98	V

(9) Supply voltage specification applied to the voltage taken at the device pins with respect to ground, not at the power supply.

Symbol	Description	Min	Max	Units
I_{IN}	Maximum current allowed through any I/O pin when the device is not turned on or during power-up/down.(10)	–	10	mA
V_{IN}	HVIO input voltage.	-0.5	3.63	V
	HSIO input voltage.	-0.5	1.98	V
T_J	Operating junction temperature.	-40	125	°C
T_{STG}	Storage temperature, ambient.	-55	150	°C

Table 77: Recommended Operating Conditions (9)

Symbol	Description	Min	Typ	Max	Units
VCC	Core power supply.	0.92	0.95	0.98	V
VCCA	PLL analog power supply.	0.92	0.95	0.98	V
VDD_SOC	Hardened RISC-V block power supply.	0.92	0.95	0.98	V
VCCAUX	1.8 V auxiliary power supply.	1.75	1.8	1.85	V
VQPS	1.8 V security fuse supply.	1.71	1.8	1.89	V
VCCIO	1.2 V HSIO bank power supply.	1.14	1.2	1.26	V
	1.35 V HSIO bank power supply	1.283	1.35	1.417	V
	1.5 V HSIO bank power supply.	1.425	1.5	1.575	V
	1.8 V HSIO bank power supply.	1.71	1.8	1.89	V
VCCIO33	1.8 V HVIO bank power supply.	1.71	1.8	1.89	V
	2.5 V HVIO bank power supply.	2.375	2.5	2.625	V
	3.0 V HVIO bank power supply.	2.85	3.0	3.15	V
	3.3 V HVIO bank power supply.	3.135	3.3	3.465	V
VCC18A_MIPI_TX	1.8 V analog power supply for MIPI TX.	1.71	1.8	1.89	V
VCC18A_MIPI_RX	1.8 V analog power supply for MIPI RX.	1.71	1.8	1.89	V
VDD_PHY	LPDDR4 digital power supply.	0.82	0.85	0.88	V
VDDPLL_MCB_TOP_PHY	LPDDR4 PLL power supply.	0.82	0.85	0.88	V
VDDQ_PHY	LPDDR4 I/O power supply.	1.06	1.1	1.17	V
VDDQX_PHY	LPDDR4 I/O pre-driver power supply.	1.06	1.1	1.17	V
VDDQ_CK_PHY	LPDDR4 I/O power supply for clock.	1.06	1.1	1.17	V
VCC_SERDES	Transceiver digital PCS and PCIe controller power supplies.	0.82	0.85	0.88	V
VDDA_C_Q	Transceiver analog bias power supply.	0.82	0.85	0.88	V
VDDA_D_Q	Transceiver digital and analog data path power supplies.	0.82	0.85	0.88	V
VDDA_H_Q	Transceiver analog power supply (for low power)	(11)	1.5	(11)	V

(10) Should not exceed a total of 100 mA per bank
(11) Pending characterization.

Symbol	Description	Min	Typ	Max	Units
	Transceiver analog power supply (for high performance)	(11)	1.8	(11)	V
T_{JCOM}	Operating junction temperature, commercial.	0	-	85	°C
T_{JIND}	Operating junction temperature, industrial.	-40	-	100	°C

Table 78: Power Supply Ramp Rates

Symbol	Description	Min	Max	Units
t_{RAMP}	Power supply ramp rate for all supplies.	0.1 * V_{supply}	10	V/ms

Table 79: HVIO DC Electrical Characteristics

I/O Standard	V_{IL} (V)		V_{IH} (V)		V_{OL} (V)	V_{OH} (V)
	Min	Max	Min	Max	Max	Min
3.3 V LVCMOS	-0.3	0.8	2.1	3.465	0.2	VCCIO33 - 0.2
3.0 V LVCMOS	-0.3	0.8	2.1	3.15	0.2	VCCIO33 - 0.2
3.3 V LVTTL	-0.3	0.8	2.1	3.465	0.4	2.4
3.0 V LVTTL	-0.3	0.8	2.1	3.15	0.4	2.4
2.5 V LVCMOS	-0.3	0.45	1.7	2.625	0.4	2.0
1.8 V LVCMOS	-0.3	0.58	1.27	1.89	0.45	VCCIO33 - 0.45

Table 80: HVIO DC Electrical Characteristics

Voltage (V)	Typical Hysteresis (mV)(12)	Input Leakage Current (µA)	Tristate Output Leakage Current (µA)
3.3	250	±25	±10
2.5	250	±25	±10
1.8	200	±25	±10

(12) For input pins with Schmitt Trigger enabled

Table 81: HSIO Pins Configured as Single-Ended I/O DC Electrical Characteristics

I/O Standard	V_{IL} (V)		V_{IH} (V)		V_{OL} (V)	V_{OH} (V)
	Min	Max	Min	Max	Max	Min
1.8 V LVCMOS	-0.3	0.58	1.27	1.89	0.45	VCCIO - 0.45
1.5 V LVCMOS	-0.3	0.35 * VCCIO	0.65 * VCCIO	1.575	0.25 * VCCIO	0.75 * VCCIO
1.2 V LVCMOS	-0.3	0.35 * VCCIO	0.65 * VCCIO	1.26	0.25 * VCCIO	0.75 * VCCIO
1.8 V HSTL	–	VREF - 0.1	VREF + 0.1	–	0.4	VCCIO - 0.4
1.5 V HSTL	–	VREF - 0.1	VREF + 0.1	–	0.4	VCCIO - 0.4
1.2 V HSTL	-0.15	VREF - 0.08	VREF + 0.08	VREF + 0.15	0.25 * VCCIO	0.75 * VCCIO
1.8 V SSTL	-0.3	VREF - 0.125	VREF + 0.125	VCCIO + 0.3	VTT - 0.603	VTT + 0.603
1.5 V SSTL	–	VREF - 0.1	VREF + 0.1	–	0.2 * VCCIO	0.8 * VCCIO
1.35 V SSTL	–	VREF - 0.1	VREF + 0.1	–	0.2 * VCCIO	0.8 * VCCIO
1.2 V SSTL	–	VREF - 0.1	VREF + 0.1	–	0.2 * VCCIO	0.8 * VCCIO

Table 82: HSIO Pins Configured as Single-Ended I/O DC Electrical Characteristics

I/O Standard	VREF (V)			Vtt (V)		
	Min	Typ	Max	Min	Typ	Max
1.8 V HSTL	0.85	0.9	0.95	–	0.5 * VCCIO	–
1.5 V HSTL	0.68	0.75	0.9	–	0.5 * VCCIO	–
1.2 V HSTL	0.47 * VCCIO	0.5 * VCCIO	0.53 * VCCIO	–	0.5 * VCCIO	–
1.8 V SSTL	0.833	0.9	0.969	VREF - 0.04	VREF	VREF + 0.04
1.5 V SSTL	0.49 * VCCIO	0.5 * VCCIO	0.51 * VCCIO	0.49 * VCCIO	0.5 * VCCIO	0.51 * VCCIO
1.35 V SSTL	0.49 * VCCIO	0.5 * VCCIO	0.51 * VCCIO	0.49 * VCCIO	0.5 * VCCIO	0.51 * VCCIO
1.2 V SSTL	0.49 * VCCIO	0.5 * VCCIO	0.51 * VCCIO	0.49 * VCCIO	0.5 * VCCIO	0.51 * VCCIO

Table 83: HSIO Pins Configured as Differential SSTL I/O Electrical Characteristics

I/O Standard	$V_{SWING (DC)}$ (V)		$V_{X(AC)}$ (V)			$V_{SWING (AC)}$ (V)	
	Min	Max	Min	Typ	Max	Min	Max
1.8 V SSTL	0.25	VCCIO + 0.6	VCCIO/2 – 0.175	–	VCCIO/2 + 0.175	0.5	VCCIO + 0.6
1.5 V SSTL	0.2	–	VCCIO/2 – 0.15	–	VCCIO/2 + 0.15	0.35	–
1.35 V SSTL	0.2	–	VCCIO/2 – 0.15	–	VCCIO/2 + 0.15	0.35	–
1.2 V SSTL	0.18	–	VREF– 0.15	VCCIO /2	VREF + 0.15	-0.3	0.3

Table 84: HSIO Pins Configured as Differential HSTL I/O Electrical Characteristics

I/O Standard	$V_{DIF\ (DC)}$ (V)		$V_{X\ (AC)}$ (V)			$V_{CM\ (DC)}$ (V)			$V_{DIF\ (AC)}$ (V)	
	Min	Max	Min	Typ	Max	Min	Typ	Max	Min	Max
1.8 V HSTL	0.2	–	0.78	–	1.12	0.78	–	1.12	0.4	–
1.5 V HSTL	0.2	–	0.68	–	0.9	0.68	–	0.9	0.4	–
1.2 V HSTL	0.16	VCCIO + 0.3	–	0.5 * VCCIO	–	0.4 * VCCIO	0.5 * VCCIO	0.6 * VCCIO	0.3	VCCIO + 0.48

Table 85: HSIO Pins Configured as Single-Ended I/O DC Electrical Characteristics

Voltage (V)	Typical Hysteresis (mV)[13]	Input Leakage Current (µA)	Tristate Output Leakage Current (µA)
1.8	200	±25	±10
1.5	160	±25	±10
1.35	–	±25	±10
1.2	140	±25	±10

Table 86: Supported HVIO Drive Strength

I/O Standard	Drive Strength	Units
3.3 V LVTTL	4, 8, 12, 16	mA
3.3 V LVCMOS	2, 4, 6, 8	mA
3.0 V LVTTL	4, 8, 12, 16	mA
3.0 V LVCMOS	2, 4, 6, 8	mA
2.5 V LVCMOS	4, 8, 12, 16	mA
1.8 V LVCMOS	4, 8, 12, 16	mA

Table 87: Supported HSIO Drive Strength

I/O Standard	Drive Strength	Units
1.8 V LVCMOS	4, 8, 12, 16	mA
1.5 V LVCMOS	4, 8, 12, 16	mA
1.2 V LVCMOS	2, 4, 8, 12	mA
1.8 V SSTL	4, 8, 10, 12	mA
1.5 V SSTL	4, 8, 10, 12	mA
1.35 V SSTL	4, 8, 10, 12	mA
1.2 V SSTL	4, 8, 10, 12	mA
1.8 V HSTL	4, 8, 10, 12	mA
1.5 V HSTL	4, 8, 10, 12	mA
1.2 V HSTL	4, 8, 10, 12	mA

[13] For LVCMOS input pins with Schmitt Trigger enabled

Table 88: Maximum Toggle Rate

I/O	I/O Standard	Speed Grade	Serialization Mode	Max Toggle Rate (Mbps)[14][15]
HVIO	3.0 V, 3.3 V LVTTL 3.0 V, 3.3 V LVCMOS	All	–	200
HVIO	2.5 V LVCMOS	All	–	100
HVIO	1.8 V LVCMOS	All	–	400
HSIO	1.8 V, 1.5 V, 1.2 V LVCMOS	All	–	400
HSIO	1.8 V, 1.5 V, 1.35 V, 1.2 V SSTL 1.8 V, 1.5 V, 1.2 V HSTL	All	–	800
HSIO	LVDS	C3, I3	Full-rate	850
			Half-rate	1,300
		C2, I2	Full-rate	720
			Half-rate	1,100
HSIO	Sub-LVDS	C3, I3	Full-rate	850
			Half-rate	1,100
		C2, I2	Full-rate	720
			Half-rate	1,100
HSIO	MIPI lane	C3, I3	–	1,300
		C2, I2	–	1,100

Table 89: HVIO Internal Weak Pull-Up and Pull-Down Resistance

I/O Standard	Internal Pull-Up			Internal Pull-Down			Units
	Min	Typ	Max	Min	Typ	Max	
3.3 V LVTTL/LVCMOS	25	42	67	24	29	33	kΩ
3.0 V LVTTL/LVCMOS	25	42	67	24	29	33	kΩ
2.5 V LVCMOS	25	42	67	24	29	33	kΩ
1.8 V LVCMOS	25	35	45	24	29	33	kΩ

Table 90: HSIO Internal Weak Pull-Up and Pull-Down Resistance

CDONE and CRESET_N also have an internal weak pull-up with these values.

I/O Standard	Internal Pull-Up			Internal Pull-Down			Units
	Min	Typ	Max	Min	Typ	Max	
1.8 V LVCMOS, HSTL, SSTL	18	27	47	18	27	47	kΩ
1.5 V LVCMOS, HSTL, SSTL	22	38	65	22	38	65	kΩ
1.35 V SSTL	30	52	100	30	52	100	kΩ
1.2 V LVCMOS, HSTL, SSTL	40	66	135	40	66	135	kΩ

(14) The maximum toggle rate is dependent on the drive strength and external load conditions. Perform IBIS simulation to determine the optimal drive strength setting to achieve the targeted toggle rate.
(15) All I/O standards are characterized with 5 pF load, except for LVTTL and LVCMOS standards which are characterized with 15 pF load.

Table 91: Single-Ended I/O Programmable Delay Chain Step Size: Static

Speed Grade	Delay per Step			Units
	Min	Typ	Max	
All	35	55	75	ps

Table 92: Single-Ended I/O Programmable Delay Chain Step Size: Dynamic

Speed Grade	Delay per Step			Units
	Min	Typ	Max	
All	12	18	24	ps

Table 93: Differential I/O Programmable Delay Chain Step Size: Static and Dynamic

Speed Grade	Delay per Step			Units
	Min	Typ	Max	
All	12	18	24	ps

Table 94: Block RAM, DSP Block, Gobal Clock Buffer, DPA, and RISC-V Performance

Description	Speed Grade		Units
	C3, I3	C2, I2	
Block RAM maximum frequency.	850	720	MHz
DSP block maximum frequency.	850	720	MHz
Global clock buffer block maximum frequency.	850	720	MHz
DPA maximum data rate.	850	720	Mbps
Hardened RISC-V block maximum system clock.	800	800	MHz

Table 95: MIPI D-PHY Interface Performance

Description	Speed Grade		Units
	C3, I3	C2, I2	
MIPI D-PHY block maximum data rate.	2.0	1.8	Gbps

Table 96: LPDDR4 Interface Performance

Description	Speed Grade		Units
	C3, I3	C2, I2	
LPDDR4 DRAM interface maximum data rate.	2.4	1.866	Gbps

Table 97: V_{IH}, V_{IL}, V_{OL}, and V_{OH} Specifications for LPDDR4

V_{IL} (V)		V_{IH} (V)		V_{OL} (V)	V_{OH} (V)
Min	Max	Min	Max	Max	Min
–	$(V_{DDQ\_PHY} / 6) - 0.075$	$(V_{DDQ\_PHY} / 6) + 0.075$	–	$V_{DDQ\_PHY} * 0.1$	$V_{DDQ\_PHY} * 0.5$

HSIO Electrical and Timing Specifications

The HSIO pins comply with the LVDS EIA/TIA-644 electrical specifications.

Important: All specifications are preliminary and pending hardware characterization.

HSIO as LVDS, Sub-LVDS, Bus-LVDS, RSDS, Mini LVDS, and SLVS

Table 98: HSIO Electrical Specifications when Configured as LVDS

Parameter	Description	Test Conditions	Min	Typ	Max	Unit
LVDS TX						
V_{CCIO}	LVDS transmitter voltage supply	–	1.71	1.8	1.89	V
V_{OD}	Output differential voltage	RL = 100 Ω	200	350	450	mV
ΔV_{OD}	Change in V_{OD}	–	–	–	50	mV
V_{OCM}	Output common mode voltage	–	1.125	1.2	1.375	V
ΔV_{OCM}	Change in V_{OCM}	–	–	–	50	mV
LVDS RX						
V_{ID}	Input differential voltage	–	100	–	600	mV
V_{ICM}	Input common mode voltage (fmax <= 1000 Mbps)	–	100	–	1,600	mV
	Input common mode voltage (fmax > 1000 Mbps)	–	700	–	1,400	mV
V_i	Input voltage valid range	–	0	–	1.89	V

Table 99: HSIO Timing Specifications when Configured as LVDS

Parameter	Description	Min	Typ	Max	Unit
t_{LVDS_CPA}	LVDS TX reference clock output phase accuracy	-5	–	+5	%
t_{LVDS_skew}	LVDS TX lane-to-lane skew	–	200	–	ps

Table 100: HSIO Electrical Specifications when Configured as Sub-LVDS

Parameter	Description	Test Conditions	Min	Typ	Max	Unit
Sub-LVDS TX						
VCCIO	Sub-LVDS transmitter voltage supply	–	1.71	1.8	1.89	V
V_{OD}	Output differential voltage	RL = 100 Ω	100	150	200	mV
ΔV_{OD}	Change in V_{OD}	–	–	–	50	mV
V_{OCM}	Output common mode voltage	–	0.8	0.9	1.0	V
ΔV_{OCM}	Change in V_{OCM}	–	–	–	50	mV
Sub-LVDS RX						
V_{ID}	Input differential voltage	–	100	–	600	mV
V_{ICM}	Input common mode voltage	–	100	–	1600	mV
V_i	Input voltage valid range	–	0	–	1.89	V

Table 101: HSIO Electrical Specifications when Configured as Bus-LVDS

Parameter	Description	Test Conditions	Min	Typ	Max	Unit
Bus-LVDS TX						
VCCIO	Voltage supply for LVDS transmitter	–	1.71	1.8	1.89	V
V_{OD}	Differential output voltage	RL = 27 Ω	200	250	300	mV
ΔV_{OD}	Static difference of VOD (between 0 and 1)	–	–	–	50	mV
V_{OC}	Output common mode voltage	–	1.125	1.2	1.375	V
ΔV_{OC}	Output common mode voltage offset	–	–	–	50	mV
Bus-LVDS RX						
V_{ID}	Differential input voltage	–	100	–	600	mV
V_{IC}	Differential input common mode	–	100	–	1600	mV
V_i	Valid input voltage range	–	0	–	1.89	V

Table 102: HSIO Electrical Specifications when Configured as RSDS, Mini LVDS and SLVS

IO standard	V_{ID} (mV)		V_{ICM} (mV)		V_{OD} (mV)			V_{OCM} (mV)		
	Min	Max	Min	Max	Min	Typ	Max	Min	Typ	Max
RSDS	100	-	300	1400	100	200	600	500	1200	1400
Mini LVDS	200	600	400	1325	250	-	600	1000	1200	1400
SLVS	100	400	100	300	150	200	250	140	200	270

HSIO as High-Speed and Low-Power MIPI Lane

The MIPI transmitter and receiver lanes are compliant to the MIPI Alliance Specification for D-PHY Revision 1.1.

Table 103: HSIO DC Specifications when Configured as High-Speed MIPI TX Lane

Parameter	Description	Min	Typ	Max	Unit
VCCIO	High-speed transmitter voltage supply	1.14	1.2	1.26	V
V_{CMTX}	High-speed transmit static common-mode voltage	150	200	250	mV
$\lvert\Delta V_{CMTX}\rvert$	V_{CMTX} mismatch when output is Differential-1 or Differential-0	-	-	5	mV
$\lvert V_{OD}\rvert$	High-speed transmit differential voltage	140	200	270	mV
$\lvert\Delta V_{OD}\rvert$	V_{OD} mismatch when output is Differential-1 or Differential-0	-	-	14	mV
V_{OHHS}	High-speed output high voltage	-	-	360	mV
V_{CMRX}	Common mode voltage for high-speed receive mode	70	-	330	mV

Table 104: HSIO DC Specifications when Configured as Low-Power MIPI TX Lane

Parameter	Description	Min	Typ	Max	Unit
V_{OH}	Thevenin output high level	1.1	1.2	1.3	V
V_{OL}	Thevenin output low level	-50	-	50	mV
Z_{OLP}	Output impedance of low-power transmitter	110	-	-	Ω

Table 105: HSIO DC Specifications when Configured as High-Speed MIPI RX Lane

Parameter	Description	Min	Typ	Max	Unit
$V_{CMRX(DC)}$	Common mode voltage high-speed receiver mode	70	-	330	mV
V_{IDTH}	Differential input high threshold	-	-	70	mV
V_{IDTL}	Differential input low threshold	-70	-	-	mV
V_{IHHS}	Single-ended input high voltage		-	460	mV
V_{ILHS}	Single-ended input low voltage	-40	-	-	mV

Table 106: HSIO DC Specifications when Configured as Low-Power MIPI RX Lane

Parameter	Description	Min	Typ	Max	Unit
V_{IH}	Logic 1 input voltage	880	-	-	mV
V_{IL}	Logic 0 input voltage, not in ULP state	-	-	550	mV
$V_{IL-ULPS}$	Logic 0 input voltage, ULPS state	-	-	300	mV
V_{HYST}	Input hysteresis	25	-	-	mV

MIPI Electrical Specifications and Timing

The MIPI D-PHY transmitter and receiver are compliant to the MIPI Alliance Specification for D-PHY Revision 1.1.

Table 107: High-Speed MIPI D-PHY Transmitter (TX) DC Specifications

Parameter	Description	Min	Typ	Max	Unit
V_{CMTX}	High-speed transmit static common-mode voltage	150	200	250	mV
$\lvert\Delta V_{CMTX(1,0)}\rvert$	V_{CMTX} mismatch when output is Differential-1 or Differential-0	-	-	5	mV
$\lvert V_{OD}\rvert$	High-speed transmit differential voltage	140	200	270	mV
$\lvert\Delta V_{OD}\rvert$	VOD mismatch when output is Differential-1 or Differential-0	-	-	14	mV
V_{OHHS}	High-speed output high voltage	-	-	360	mV
Z_{OS}	Single ended output impedance	40	50	60	Ω
ΔZ_{OS}	Single ended output impedance mismatch	-	-	20	%

Table 108: High-Speed MIPI D-PHY Transmitter (TX) AC Specifications

Parameter	Description	Min	Typ	Max	Unit
$\Delta V_{CMTX(HF)}$	Common-level variations above 450 MHz	-	-	15	mV_{RMS}
$\Delta V_{CMTX(LF)}$	Common-level variations between 50 to 450 MHz	-	-	25	mV_{PEAK}
t_R and t_F	Rise and fall time < 1.0 Gbps	-	-	0.3	UI
	Rise and fall time > 1.0 Gbps	-	-	0.35	UI
	Rise and fall time > 1.5 Gbps	-	-	0.4	UI

Table 109: Low-Power MIPI D-PHY Transmitter (TX) DC Specifications

Parameter	Description	Min	Typ	Max	Unit
V_{OH}	Thevenin output high level	0.95	1.2	1.3	V
V_{OL}	Thevenin output low level	-50	-	50	mV
Z_{OLP}	Output impedance of low-power transmitter	110	-	-	Ω

Table 110: Low-Power MIPI D-PHY Transmitter (TX) AC Specifications

Parameter	Description	Min	Typ	Max	Unit
T_{RLP}/T_{FLP}	15%-85% rise time and fall time	-	-	25	ns
T_{REOT}	30%-85% rise time and fall time	-	-	35	ns
$T_{LP\text{-}PULSE\text{-}TX}$	Pulse width of first LP exclusive-OR clock pulse after Stop state or last pulse before Stop state	40	-	-	ns
	Pulse width of all other pulses	-	20	-	ns
$T_{LP\text{-}PER\text{-}TX}$	Period of the LP exclusive-OR clock	90	-	-	ns
$\delta V/\delta t_{SR}$	Slew rate at C_{LOAD} = 50 pF < 1.5 Gbps	30	-	150	mV/ns
	Slew rate at C_{LOAD} = 50 pF > 1.5 Gbps	25	-	150	mV/ns

Table 111: High-Speed MIPI D-PHY Receiver (RX) DC Specifications

Parameter	Description	Min	Typ	Max	Unit
$V_{CMRX(DC)}$	Common mode voltage high-speed receive mode	70	-	330	mV
Z_{ID}	Differential input impedance	80	100	120	Ω

Table 112: High-Speed MIPI D-PHY Receiver (RX) AC Specifications

Parameter	Description	Min	Typ	Max	Unit
$\Delta V_{CMRX(HF)}$	Common-point interference above 450 MHz	-	-	50	mV
$\Delta_{VCMRX(LF)}$	Common-point interference between 50 MHz to 450 MHz	-	-	25	mV
V_{IDTH}	Differential input high threshold	-	-	40	mV
V_{IDTL}	Differential input low threshold	-40	-	-	mV
V_{IHHS}	Single-ended input high voltage	-	-	460	mV
V_{ILHS}	Single-ended input low voltage	-40	-	-	mV
$V_{TERM-EN}$	Single-ended threshold for high-speed termination enable	-	-	450	mV
CCP	Common-point termination	-	-	60	pF

Table 113: Low-Power MIPI D-PHY Receiver (RX) DC Specifications

Parameter	Description	Min	Typ	Max	Unit
V_{IH}	Logic 1 input voltage	740	-	-	mV
V_{IL}	Logic 0 input voltage, not in ULP state	-	-	550	mV
$V_{IL-ULPS}$	Logic 0 input voltage, ULP state	-	-	300	mV
V_{HYST}	Input hysteresis	25	-	-	mV

Table 114: Low-Power MIPI D-PHY Receiver (RX) AC Specifications

Parameter	Description	Min	Typ	Max	Unit
T_{MIN-RX}	Minimum pulse width response	20	-	-	ns
V_{INT}	Peak interference amplitude	-	-	200	mV
f_{INT}	Interference frequency	450	-	-	MHz

MIPI Reset Timing

The MIPI RX and TX interfaces have two reset signals (`RESET` and `RST0_N`) to reset the D-PHY controller logic. These signals are active low, and you should use them together to reset the MIPI interface.

The following waveform illustrates the minimum time required to reset the MIPI interface.

Figure 58: RESET and RST0_N Timing Diagram

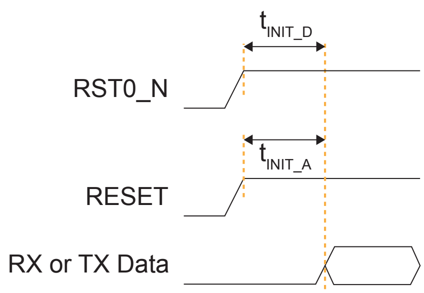


Table 115: MIPI Timing

Symbol	Parameter	Min	Typ	Max	Units
t_{INIT_A}	Minimum time between the rising edge of RESET and the start of MIPI RX or TX data.	350	-	-	µs
t_{INIT_D}	Minimum time between the rising edge of RST0_N and the start of MIPI RX or TX data.	1	-	-	clk

PLL Timing and AC Characteristics

The following tables describe the PLL timing and AC characteristics.

Table 116: PLL Timing

Symbol	Parameter	Min	Typ	Max	Units
F_{IN}	Input clock frequency.	16	–	800	MHz
F_{OUT}	Output clock frequency.	0.1342	–	1,000	MHz
F_{VCO}	PLL VCO frequency.	2,200	–	5,500	MHz
F_{PLL}	Post-divider PLL VCO frequency.	–	–	4,000	MHz
F_{PFD}	Phase frequency detector input frequency.	16	–	800	MHz

Table 117: PLL AC Characteristics[16]

Symbol	Parameter	Min	Typ	Max	Units
t_{DT}	Output clock duty cycle.	45	50	55	%
t_{OPJIT} (PK - PK) [17]	Output clock period jitter (PK-PK).	–	–	200	ps
t_{OPJITN} (PK - PK)[18][19]	Output clock period jitter (PK-PK) with noisy input.	–	–	400	ps
t_{PLL_HLW}	PLL input clock HIGH/LOW pulse width	0.56	–	–	ns
t_{LOCK}	PLL lock-in time.	–	300	500	PFD[20]

(16) Test conditions at nominal voltage and room temperature.
(17) The output jitter specification applies to the PLL jitter when an input jitter of 20 ps is applied.
(18) The output jitter specification applies to the PLL jitter with maximum allowed input jitter of 800 ps.
(19) The period jitter is measured over 10,000 sample size with minimal core and I/O activity.
(20) PFD cycle equals to reference clock division divided by reference clock frequency.

Configuration Timing

Important: All specifications are preliminary and pending hardware characterization.

The Tz325 FPGA has the following configuration timing specifications.

Timing Parameters Applicable to All Modes

Table 118: All Modes

Symbol	Parameter	Min	Typ	Max	Units
t_{CRESET_N}	Minimum CRESET_N low pulse width required to trigger re-configuration.	0.32	-	-	µs
t_{USER}	Minimum configuration duration after CDONE goes high before entering user mode. Test condition at 10 kΩ pull-up resistance and 10 pF output loading on CDONE pin.	25	-	-	µs

Note: The FPGA may go into user mode before t_{USER} has elapsed. However, Efinix recommends that you keep the system interface to the FPGA in reset until t_{USER} has elapsed.

For JTAG programming, the min t_{USER} configuration time is required after `CDONE` goes high and the FPGA receives the ENTERUSER instruction from the JTAG host (TAP controller in UPDATE_IR state).

JTAG Mode

Figure 59: JTAG Timing Waveform

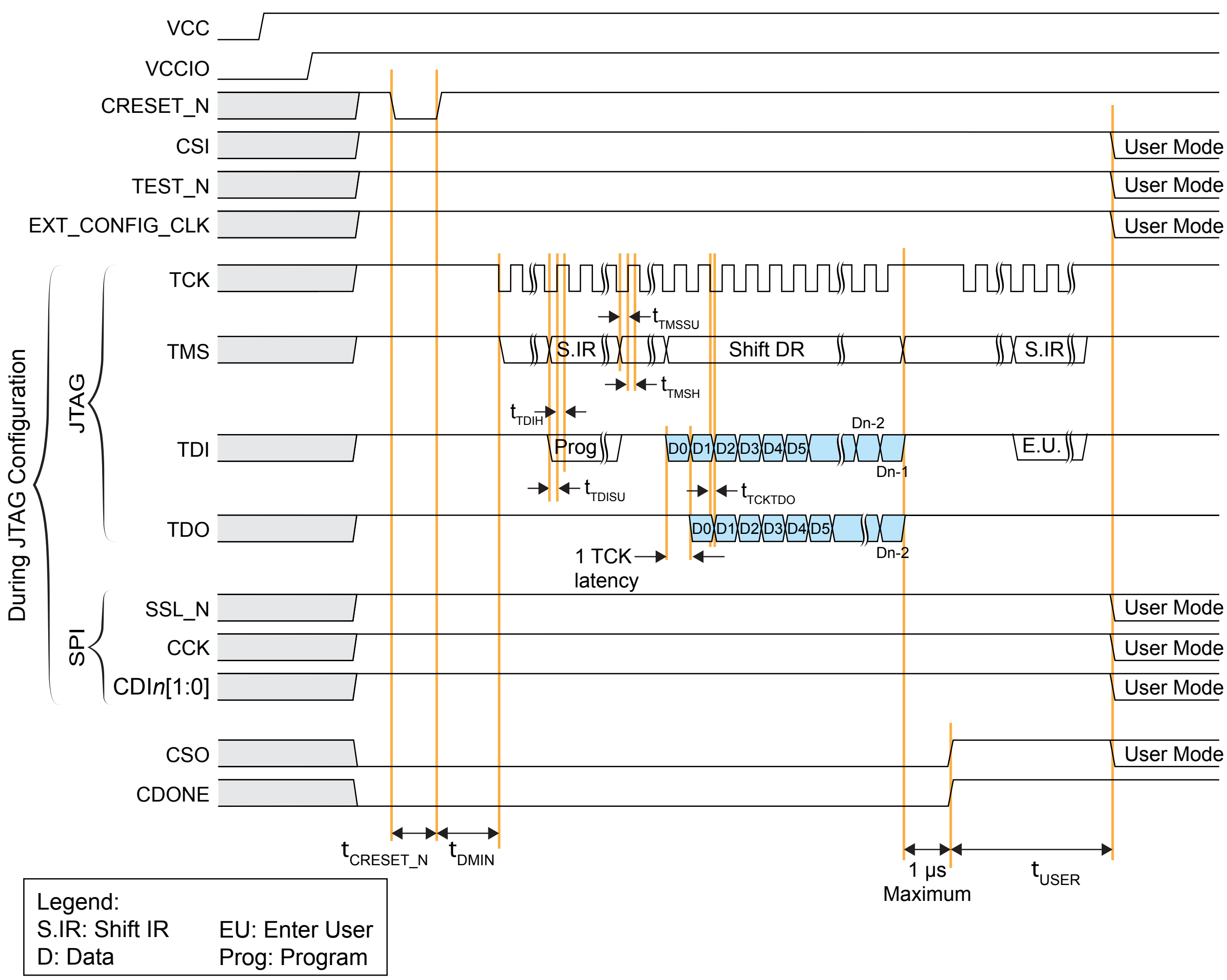


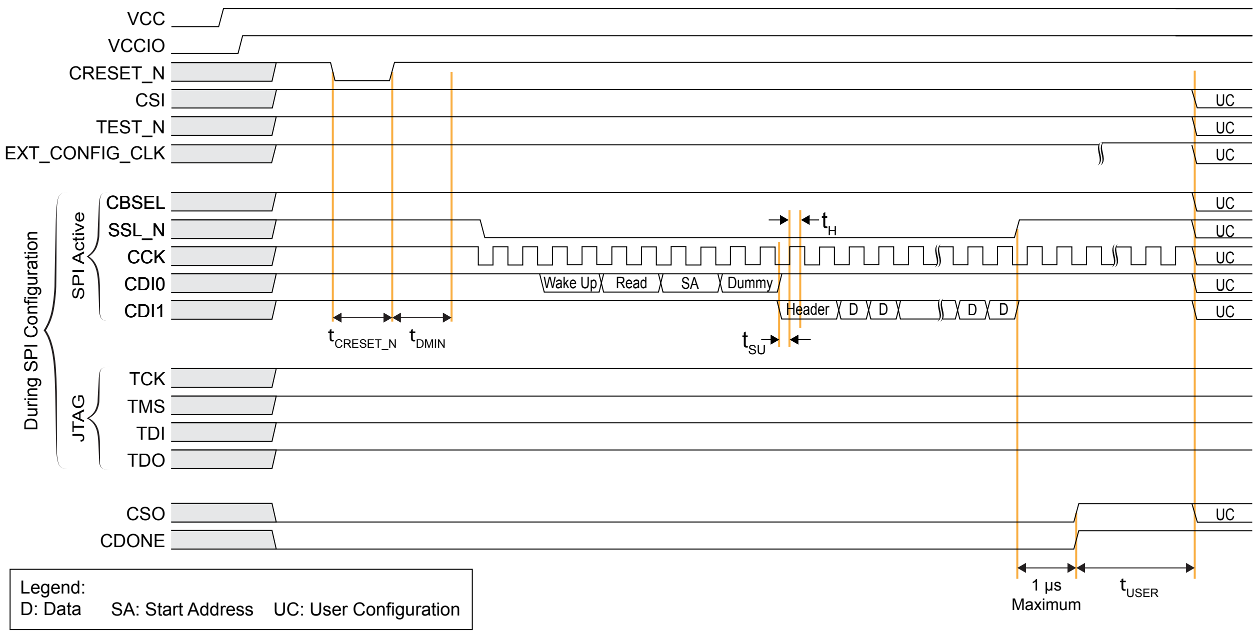
Table 119: JTAG Mode Timing

Symbol	Parameter	Min	Typ	Max	Units
f_{TCK}	TCK frequency.	-	-	10	MHz
t_{TDISU}	TDI setup time.	15	-	-	ns
t_{TDIH}	TDI hold time.	3	-	-	ns
t_{TMSSU}	TMS setup time.	15	-	-	ns
t_{TMSH}	TMS hold time.	3	-	-	ns
t_{TCKTDO}	TCK falling edge to TDO output.	-	-	30	ns
t_{DMIN}	Minimum time between deassertion of CRESET_N to the start of JTAG configuration.	32	-	-	µs

Important: The SPI bus must be inactive during JTAG configuration.
The `EXT_CONFIG_CLK` pin must be inactive during JTAG configuration.

SPI Active Mode

Figure 60: SPI Active (x1) Timing Sequence



The waveform shows the perspective from the control block without any optional external pull-up or pull-down resistors connected.

Table 120: Active Mode Timing

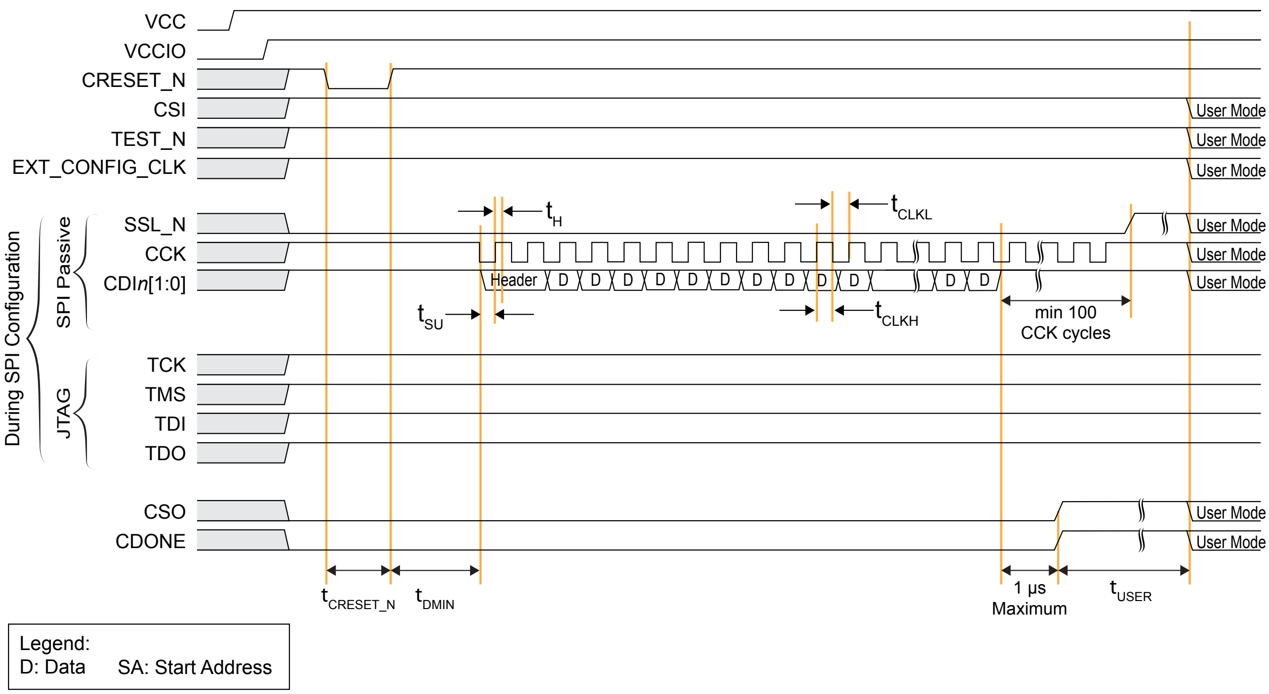
Symbol	Parameter	Frequency	Min	Typ	Max	Units
f_{MAX_M}	Active mode internal configuration clock frequency.	DIV1	52	80	100	MHz
		DIV2	26	40	52	MHz
		DIV4	13	20	26	MHz
		DIV8	6.5	10	13	MHz
$f_{MAX\_M\_EXTCLK}$	Active mode external configuration clock frequency.	-	-	-	100	MHz
t_{SU}	Setup time. Test condition at 1.8 V I/O standard and 0 pF output loading.	-	5	-	-	ns
t_H	Hold time. Test condition at 1.8 V I/O standard and 0 pF output loading.	-	0	-	-	ns

Important: The JTAG pins must be inactive during SPI active configuration.

The `EXT_CONFIG_CLK` pin must be inactive during SPI active configuration if the internal oscillator is selected as the configuration clock source (default).

SPI Passive Mode

Figure 61: SPI Passive Mode (x1, Mode 3) Timing Sequence



Note:

- The waveform shows the perspective from the control block without any optional external pull-up or pull-down resistors connected.
- CDI input data is clocked by `CCK`. To prevent configuration failure, `CCK` must stop toggling if the bitstream data becomes invalid. You must resume with the next bitstream data before stopping to continue the configuration.
- `CSI` must stay high during configuration.
- `SSL_N` must stay low during configuration.
- Efinix does not recommend connecting multiple slaves on the same SPI bus.

Important: To ensure successful configuration, the microprocessor must continue to supply the configuration clock to the Topaz FPGA for at least 100 cycles after sending the last configuration data.

Table 121: Passive Mode Timing

Symbol	Parameter	Min	Typ	Max	Units
f_{MAX_S}	Passive mode configuration clock frequency.	-	-	100	MHz
t_{CLKH}	Configuration clock pulse width high.	4.8	-	-	ns
t_{CLKL}	Configuration clock pulse width low.	4.8	-	-	ns
t_{SU}	Setup time (x1, x2, x4, x8).	2	-	-	ns
	Setup time (x16, x32).	6	-	-	ns
t_H	Hold time.	2	-	-	ns
t_{DMIN}	Minimum time between deassertion of CRESET_N to first valid configuration data.	32	-	-	µs

Important: The JTAG pins must be inactive during SPI passive configuration.
The EXT_CONFIG_CLK pin must be inactive during SPI passive configuration.

Pinout Description

The following tables describe the pinouts for power, ground, configuration, and interfaces.

Table 122: Power and Ground Pinouts

xx indicates the bank location.

Function	Description
VCC	Core power supply.
VCCA_*xx*	PLL analog power supply.
VDD_SOC	Hardened RISC-V block power supply.
VCCAUX	1.8 V auxiliary power supply.
VCCIO33_*xx*	HVIO bank power supply.
VCCIO33_*xx_yy_zz*	Power for HVIO banks that are shorted together. *xx*, *yy*, and *zz* are the bank locations. For example: VCCIO33_BR1_BR2 shorts banks BR1 and BR2.
VCCIO*xx*	HSIO bank power supply.
VCCIO*xx_yy_zz*	Power for HSIO banks that are shorted together. *xx*, *yy*, and *zz* are the bank locations. For example: VCCIO1B_1C shorts banks 1B and 1C
VQPS	1.8 V supply for security fuse. During configuration and normal operation, keep this pin at 0 V. When you want to blow the security fuses, power this pin up to 1.8 V.
GND	Ground.

Table 123: GPIO Pinouts

x indicates the location (T, B, L, or R); *xx* indicates the bank location; *n* indicates the number; *yyyy* indicates the function.

Function	Direction	Description
GPIO*x_n*	I/O	HVIO for user function. User I/O pins are single-ended.
GPIO*x_n_yyyy*	I/O	HVIO or multi-function pin.
GPIO*x*_N_*n* GPIO*x*_P_*n*	I/O	HSIO transmitter, receiver, or both.
GPIO*x*_N_*n_yyyy* GPIO*x*_P_*n_yyyy*	I/O	HSIO transmitter, receiver, both, or multi-function.
REF_RES_*xx*	-	REF_RES is a reference resistor to generate constant current for the related circuits. Connect all REF_RES pins to ground through a 10 kΩ resistor with a tolerance of ±1%.

Table 124: Alternate Function Pinouts

n is the number.

Function	Direction	Description
CLK*n*	Input	Single ended input for global clock and control network resource. The number of inputs is package dependent.
CLK*n*_P/N	Input	Differential input pair for global clock and control network resource. P pins can access to global clock and control network resource if it is in single-ended configuration.
EXTFB	Input	PLL external feedback CLKIN.
PLLIN*n*	Input	PLL reference clock resource. The number of reference clock resources is package dependent.

Configuration Pins

Table 125: Dedicated Configuration Pins

These pins cannot be used as general-purpose I/O after configuration.
All the pins are in internal weak pull-up during configuration mode except for `TCK` and `TDO`.

Pins	Direction	Description	External Weak Pull Up/ Pull Down Requirement
CDONE	I/O	Configuration done status pin. CDONE is an open drain output; connect it to an external pull-up resistor to VCCIO. When CDONE = 1, the configuration is complete and the FPGA enters user mode. You can hold CDONE low and release it to synchronize the FPGAs entering user mode.	Pull up
CRESET_N	Input	Active-low FPGA reset and re-configuration trigger. Pulse CRESET_N low for a duration of t_{creset_N} before releasing CRESET_N from low to high to initiate FPGA re-configuration. This pin does not perform a system reset.	Pull up
TCK	Input	JTAG test clock input (TCK). The rising edge loads signals applied at the TAP input pins (TMS and TDI). The falling edge clocks out signals through the TAP TDO pin.	Pull up
TMS	Input	JTAG test mode select input (TMS). The I/O sequence on this input controls the test logic operation . The signal value typically changes on the falling edge of TCK. TMS is typically a weak pull-up; when it is not driven by an external source, the test logic perceives a logic 1.	Pull up
TDI	Input	JTAG test data input (TDI). Data applied at this serial input is fed into the instruction register or into a test data register depending on the sequence previously applied at TMS. Typically, the signal applied at TDI changes state following the falling edge of TCK while the registers shift in the value received on the rising edge. Like TMS, TDI is typically a weak pull-up; when it is not driven from an external source, the test logic perceives a logic 1.	Pull up
TDO	Output	JTAG test data output (TDO). This serial output from the test logic is fed from the instruction register or a test data register depending on the sequence previously applied at TMS. The shift out content is based on the issued instruction. The signal driven through TDO changes state following the falling edge of TCK. When data is not being shifted through the device, TDO is set to an inactive drive state (e.g., high-impedance).	Pull up
JTAG_VCCIO_SEL	Input	JTAG voltage select pin. This pin affects the BR4 bank voltage, or any banks merged with the BR4 bank. Supply VCCIO33_BR4 with 1.8 V and connect a 1 kΩ external resistor between this pin and ground to use JTAG at 1.8 V. Leave this pin floating to use the default JTAG at 3.3 V or 2.5 V.	Floating or pull down

[21] CDONE has a drive strength of 12 mA at 1.8 V.

Table 126: Dual-Purpose Configuration Pins

In user mode (after configuration), you can use these dual-purpose pins as general I/O.

Configuration Functions	Direction	Description	External Weak Pull Up/ Pull Down Requirement
CBSEL[1:0]	Input	Multi-image configuration selection pin. This function is not applicable to single-image bitstream configuration or internal reconfiguration (remote update). Connect CBSEL[1:0] to the external resistors for the image you want to use: 00 for image 1 01 for image 2 10 for image 3 11 for image 4 0: Connect to an external weak pull down. 1: Connect to an external weak pull up.	Pull up or pull down
CCK	I/O	Passive SPI input configuration clock or active SPI output configuration clock.	Optional pull up if required by external load
CDI*n*	I/O	Data input for SPI configuration. *n* is a number from 0 to 31 depending on the SPI configuration data width. CDI0 is an output in x1 active configuration mode and is a bidirectional pin in all other active configuration modes. CDI4 is a bidirectional pin in x8 active configuration mode. In a multi-bit daisy chain connection, CDI[31:0] connects to the data bus in parallel.	Optional pull up if required by external load
CSI	Input	Chip select. 0: The FPGA is not selected or enabled and will not be configured. 1: Select the FPGA for configuration modes. This pin is not bonded out in some of the smaller packages, such as the F100. CSI must remain high throughout configuration.	Pull up
CSO	Output	Chip select output. Asserted after configuration is complete. Connect this pin to the chip select pin of the next FPGA for daisy chain configuration.	-
NSTATUS	Output	Indicates a configuration error. When the FPGA drives this pin low, it indicates an ID mismatch, the bitstream CRC check has failed, or remote update has failed.	-
SSL_N	I/O	SPI configuration mode select. The FPGA senses the value of SSL_N when it comes out of reset (i.e., CRESET_N transitions from low to high). 0: Passive mode; connect to external weak pull down. 1: Active mode; connect to external weak pull up. In active configuration mode, SSL_N is an active-low chip select to the flash device (CDI0 - CDI3).	Pull up or pull down
SSU_N	Output	Active-low chip select to the upper flash device (CDI4 - CDI17) in active x8 configuration mode (dual quad mode). Not available on F100 packages.	Optional pull up if required by external load

Configuration Functions	Direction	Description	External Weak Pull Up/ Pull Down Requirement
EXT_CONFIG_CLK	Input	Alternative clock in active configuration mode.	Optional pull up if required by external load
TEST_N	Input	Active-low test mode enable signal. Set to 1 to disable test mode. During all configuration modes, rely on the external weak pull-up or drive this pin high.	Pull up

Note: Refer to the column Configuration Functions in the pinout file.

Dedicated DDR Pinout

Table 127: Dedicated DDR Pinout

n indicates the number.

Function	Direction	Description
DDR_A[*n*]	Output	Address signals to the memories.
DDR_CKE	Output	Active-high clock enable signals to the memories.
DDR_CK DDR_CK_N	Output	Differential clock output pins to the memories.
DDR_CS_N	Output	Active-low chip select signals to the memories.
DDR_DQ[*n*]	I/O	Data bus to/from the memories.
DDR_DM[*n*]	I/O	Active-high data-mask signals to the memories.
DDR_DQS[*n*] DDR_DQS_N[*n*]	I/O	Differential data strobes to/from the memories.
DDR_RST_N	Output	Active-low reset signals to the memories.
DDR_CAL	Input	240 Ω to ground reference resistor port.
VDD_PHY_DDR*n*	-	DDR digital power supply.
VDDQ_PHY_DDR*n*	-	DDR I/O power supply.
VDDQX_PHY_DDR*n*	-	DDR I/O pre-driver power supply.
VDDPLL_MCB_TOP_PHY_DDR*n*	-	DDR PLL power supply.
VDDQ_CK_PHY_DDR*n*	-	DDR I/O power supply for clock.

Dedicated MIPI D-PHY Pinout

Table 128: Dedicated MIPI D-PHY Pinouts

n indicates the number. *L* indicates the lane

Function	Direction	Description
VCC18A_MIPI*n*_TX	-	Power supply for the MIPI D-PHY TX block.
VCC18A_MIPI*n*_RX	-	Power supply for the MIPI D-PHY RX block.
MIPI*n*_TXDP*L* MIPI*n*_TXDN*L*	Output	MIPI differential transmit data lane.
MIPI*n*_RXDP*L* MIPI*n*_RXDN*L*	Input	MIPI differential receive data lane.

Dedicated Transceiver Pinout

Table 129: Dedicated Transceiver Pinout

n indicates the bank number; *L* indicates the lane; *nn* indicates the merged transceiver bank numbers.

Function	Direction	Description
VCC_SERDES*nn*	-	Transceiver digital PCS and PCIe controller power supplies.
VDDA_C_Q*nn*	-	Transceiver analog bias power supply.
VDDA_D_Q*nn*	-	Transceiver digital and analog data path power supplies.
VDDA_H_Q*nn*	-	Transceiver analog power supply for I/O.
PERST_Q*n*_N	Input	Active-low PCIe reset pin.
Q*n*_CAL_RES	Input	Transceiver calibration resistor. Connect an external resistor between this pin and ground. The resistor value is 3.01 kΩ with 1% tolerance.
Q*n*_REFCLK*n*_N Q*n*_REFCLK*n*_P	Input	Transceiver differential external reference clock input. REFCLK0: Main clock. REFCLK1: Alternative clock.
Q*n*_RXDP*L* Q*n*_RXDN*L*	Input	Transceiver differential receive serial data inputs. Q0 and Q2 include the PCIe controller.
Q*n*_TXDP*L* Q*n*_TXDN*L*	Output	Transceiver differential transmitter serial data inputs Q0 and Q2 include the PCIe controller.

Pin States

HVIO pins have an internal pullup (see Figure 21 on page 32); HSIO configured as GPIO have as internal pull up/down (see Figure 23 on page 35). The following table shows the pin state during reset, configuration, and when unused in user mode.

Note: For the DDR pin states, refer to the **Titanium DDR DRAM Block User Guide**.

Table 130: I/O Pin States

Pin Type	During Reset (CRESET_N Low)	During Configuration (CRESET_N High, CDONE Low)	When Unused in User Mode (Default)
User Pins			
HSIO	Input tri-state with weak pull up.	Input tri-state with weak pull up.	Input tri-state with weak pull up.[22]
HVIO	Input tri-state with weak pull up.	Input tri-state with weak pull up.	Input tri-state with weak pull up.[22]
Dual-Purpose Configuration Pins			
CSO	0	0[23]	Input tri-state with weak pull up.
NSTATUS	1	1[24]	Input tri-state with weak pull up.
CCK	Input tri-state with weak pull up.	SPI active output clock. SPI passive input with weak pull up.	Input tri-state with weak pull up.
CDI0	Input tri-state with weak pull up.	SPI active output. SPI passive input with weak pull up.	Input tri-state with weak pull up.

As shown in Power-Up Sequence on page 86, `CRESET_N` must be kept low during power up.

Note: Refer to the following tables for details:
Table 89: HVIO Internal Weak Pull-Up and Pull-Down Resistance on page 96
Table 90: HSIO Internal Weak Pull-Up and Pull-Down Resistance on page 96

The following table shows the states for the MIPI D-PHY pins.

Table 131: MIPI D-PHY Pin States

Pin Type	During Reset (CRESET_N Low)	During Configuration (CRESET_N High, CDONE Low)	When Unused in User Mode (Default)
MIPI 2.5G D-PHY TX	Input tri-state, no pull up or pull down.	Input tri-state, no pull up or pull down.	Input tri-state, no pull up or pull down.
MIPI 2.5G D-PHY RX	Input.	Input.	Input.

[22] You can change the default mode to weak pull-down in the Interface Designer.
[23] CSO is driven to 1 when the bitstream is done transmitting.
[24] NSTATUS is driven to 0 if the FPGA detects an incorrect JTAG ID or detects a CRC error.

Tz325 Interface Floorplan

Note: The numbers in the floorplan figures indicate the HVIO and HSIO number ranges. Some packages may not have all HVIO or HSIO pins in the range bonded out.

Figure 62: Floorplan Diagram for N484 Packages

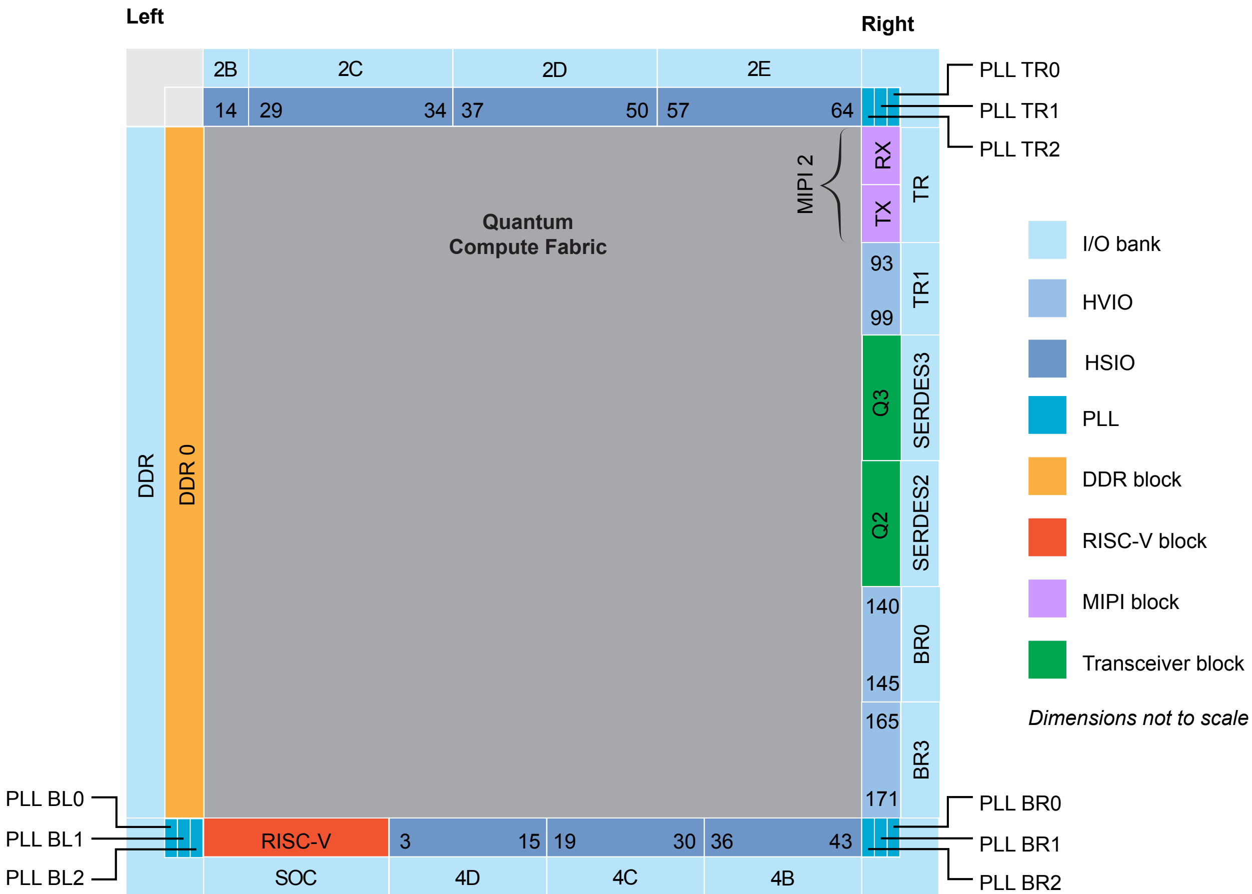
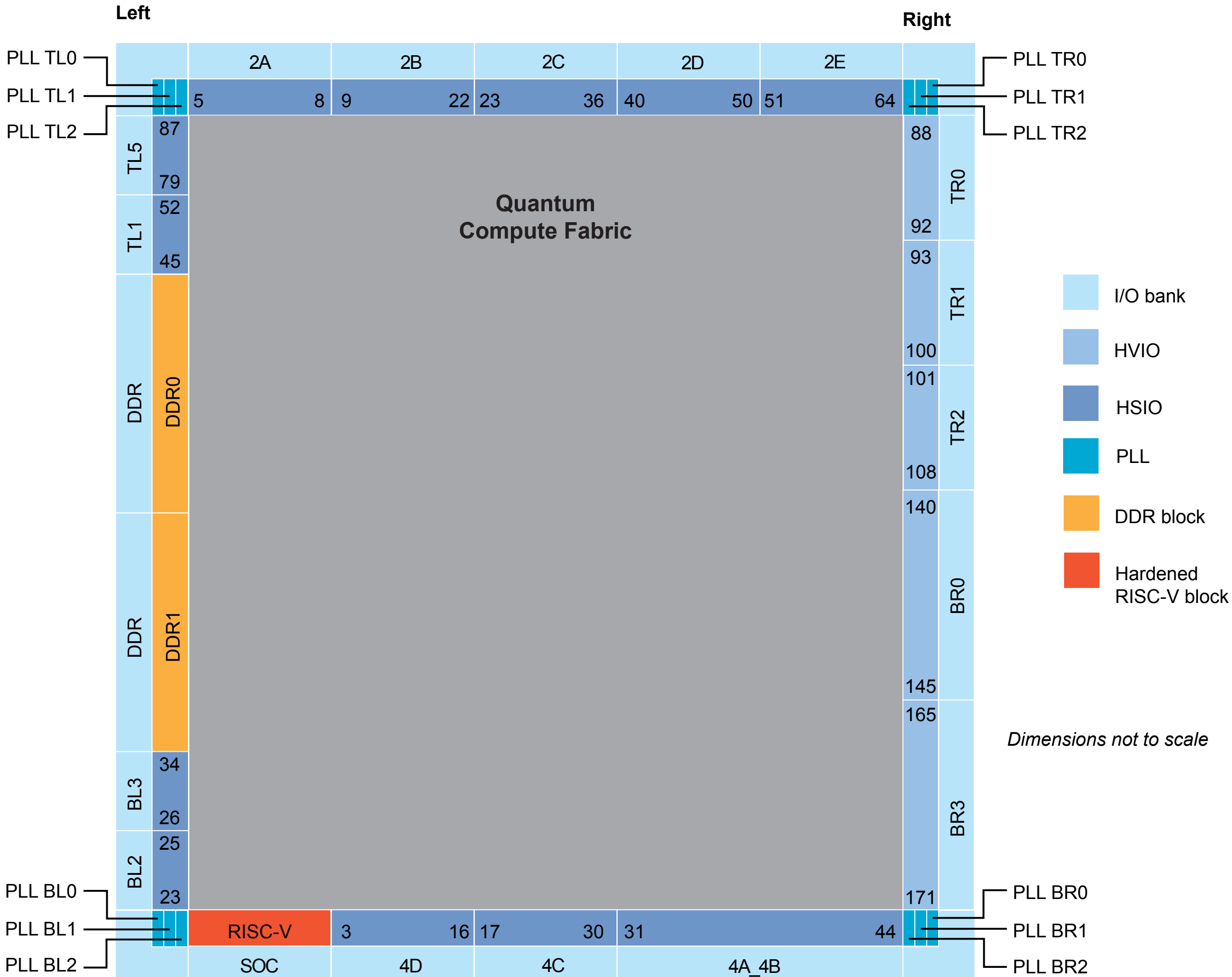


Figure 63: Floorplan Diagram for C529 Packages



Efinity Software Support

The Efinity® software provides a complete tool flow from RTL design to bitstream generation, including synthesis, place-and-route, and timing analysis. The software has a graphical user interface (GUI) that provides a visual way to set up projects, run the tool flow, and view results. The software also has a command-line flow and Tcl command console. The Efinity® software supports simulation flows using the ModelSim, NCSim, or free iVerilog simulators. An integrated hardware Debugger with Logic Analyzer and Virtual I/O debug cores helps you probe signals in your design. The software-generated bitstream file configures the Tz325 FPGA. The software supports the Verilog HDL and VHDL languages.

Ordering Codes

Refer to the Topaz Selector Guide for the full listing of Tz325 ordering codes.

Revision History

Table 132: Revision History

Date	Version	Description
November 2024	1.1	Added DLYCLK GPIO signal. (DOC-2159) Updated GPIO and LVDS interface pin names (IN to I and OUT to O) to align with primitives. (DOC-2086) Removed PLL IOFBK interface pin. The SAMPLE/PRELOAD instruction is available after JTAG fuses have been blown. (DOC-2225) Fixed typo in **Table 126: Dual-Purpose Configuration Pins** on page 113. (DOC-2038) Changed column name from Pins to Configuration Functions in **Table 126: Dual-Purpose Configuration Pins** on page 113. (DOC-2038) Added note after **Table 126: Dual-Purpose Configuration Pins** on page 113 directing the reader to the device pinout file. (DOC-2038) Updated **Fuse Programming Requirements** on page 85 with details of VQPS current. (DOC-1999) Clarified HVIO and HSIO pin states during configuration and when unused in user mode. (DOC-2041) Added notes to the configuration timing and security feature topics about not using SPI and JTAG at the same time. (DOC-2047) The transceiver supplies can be powered up in any sequence. (DOC-2131) Updated configuration timing and fuse programming waveforns. (DOC-2156)
October 2024	1.0	Initial release.