Ti125 Data Sheet

DSTi125-v1.0
April 2026
www.efinixinc.com

Contents

Introduction

The Titanium™ Ti125 FPGA features the high-density, low-power Efinix® Quantum® compute fabric wrapped with an I/O interface in a small footprint package for easy integration. Ti125 FPGAs are designed for highly integrated mobile and edge devices that need low power, a small footprint, and a multitude of I/Os. With ultra-low power Ti125 FPGAs, designers can build products that are always on, providing enhanced capabilities for applications such as industrial cameras, robotics, mobile, edge, AI IoT, and sensor fusion.

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
  - PHY clock network and clock modifiers for high-speed clocking
  - Up to seven PLLs with support for fractional-N division, programmable duty cycle, spread-spectrum clocking, and dynamic reconfiguration
- FPGA interface blocks
  - Three 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 1.5 Gbps
      - MIPI lane (DSI and CSI) in high-speed and low-power modes, up to 1.5 Gbps
    - Enhanced configurable high-speed I/O (HSIO2) support
      - Single-ended and differential I/O
      - LVDS, subLVDS, Mini-LVDS, and RSDS (RX, TX, and bidirectional), up to 1.8 Gbps
      - MIPI lane (DSI and CSI) in high-speed and low-power modes, up to 2.5 Gbps
      - DDR3L up to 1,333 Mbps
  - One oscillator
- 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: Ti125 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]				
122,792	109,440	19,200	Up to 32	5.9	576	288

Table 2: Ti125 Package-Dependent Resources

Resource		F225	M225S4F4
Single-ended GPIO (maximum)	HVIO LVCMOS: 1.8, 2.5, 3.0, 3.3 V LVTTL: 3.0, 3.3 V	23	36
	HSIO LVCMOS, HSTL: 1.2, 1.5, 1.8 V SSTL: 1.2, 1.35, 1.5, 1.8 V	62	26
	HSIO2 LVCMOS, HSTL: 1.2, 1.5, 1.8 V SSTL: 1.2, 1.35, 1.5, 1.8 V	78	84
Differential GPIO (maximum)	HSIO (LVDS, Differential HSTL, and SSTL)	31	13
	HSIO2 (LVDS, Differential HSTL, and SSTL)	39	42
	HSIO (MIPI D-PHY Data Lanes)	25	10
	HSIO (MIPI D-PHY Clock Lanes)	6	3
	HSIO2 (MIPI D-PHY Data Only Lanes)	23	25
	HSIO2 (MIPI D-PHY Data or Clock Lanes)	16	17
	HSIO2 Byte Clock Groups	8	9
	High-Speed DDR PHY Interface	2 x8	2 x8
Global clock or control signals from GPIO pins		24	24
Fractional PLLs		7	7

Available Package Options

Table 3: Available Packages

Package	Dimensions (mm x mm)	Pitch (mm)
225-ball FBGA	10 x 10	0.65
225-ball FBGA (M225S4F4)	8 x 8	0.5

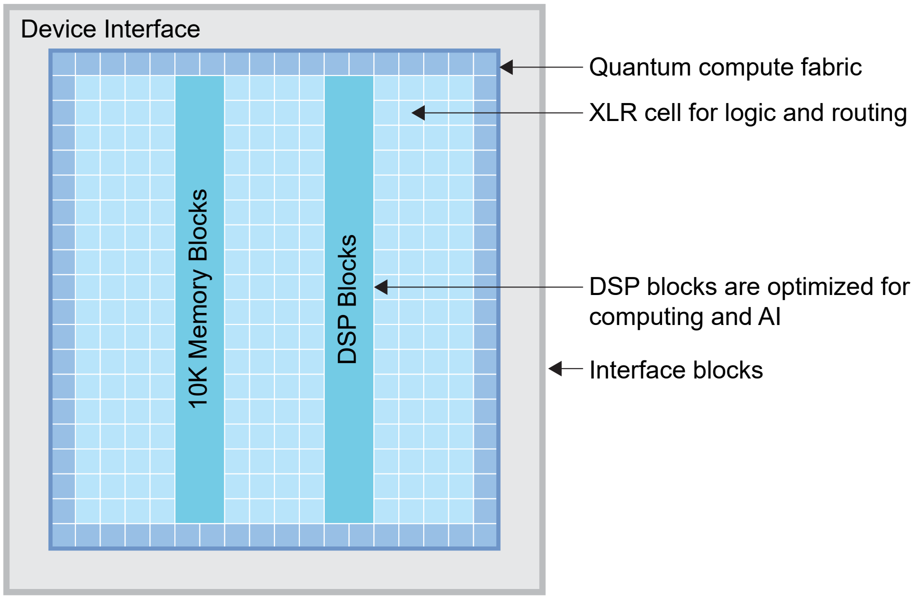
---

(1) Number of XLR cells that can be configured as shift register with 8 maximum taps.

Device Core Functional Description

Ti125 FPGAs feature an eXchangeable Logic and Routing (XLR) cell that Efinix® has optimized for a variety of applications. Titanium™ FPGAs contain LEs that are constructed from XLR cells. Each FPGA in the Titanium™ 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: Ti125 FPGA Block Diagram



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 Titanium™ FPGAs, read the **Why the XLR Cell is a Big Deal White Paper**.

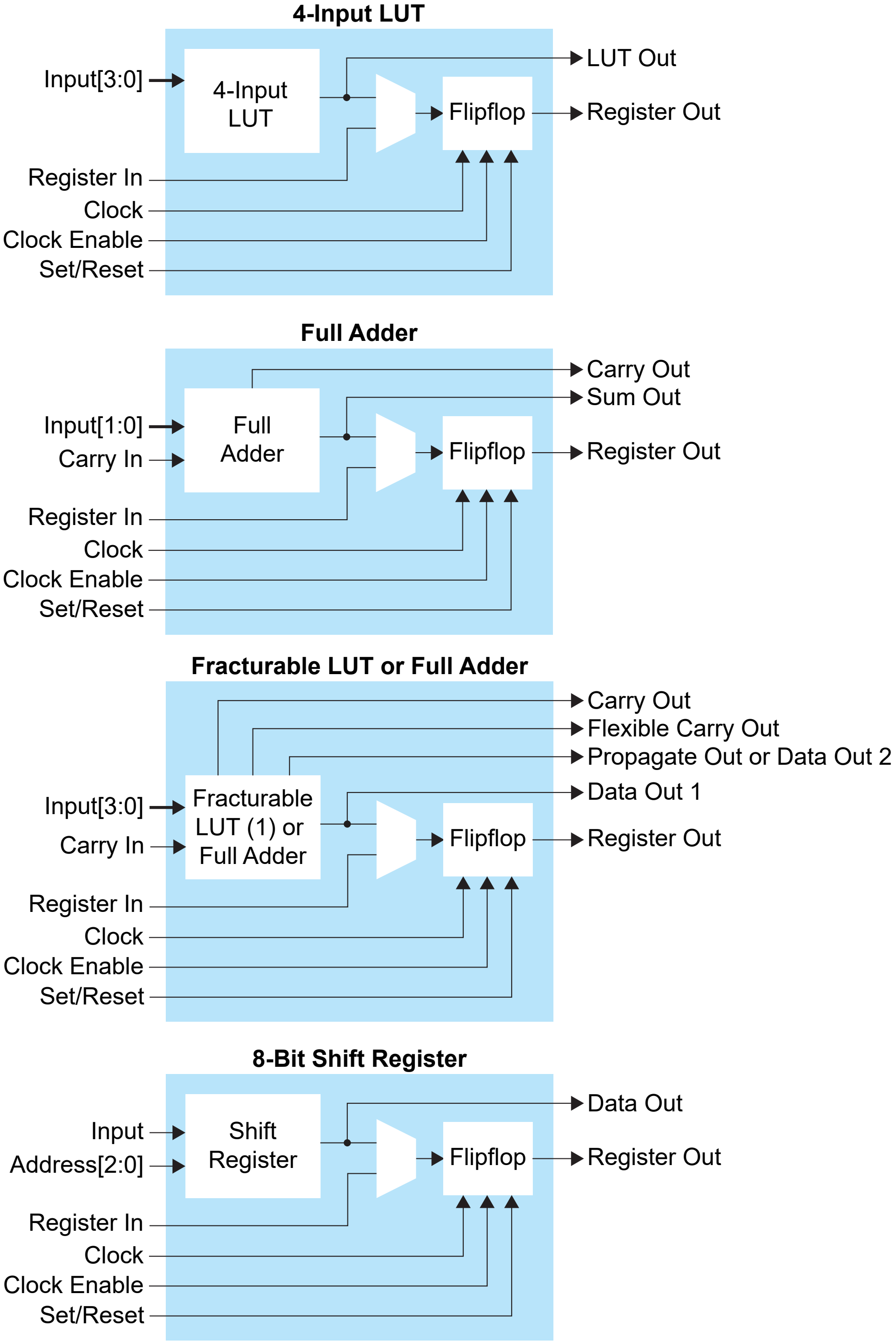
The XLR cell functions as:

- 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.

Ti125 XLR cells have enhanced functionality to improve the quality of results during compilation. For all XLR cell functions you can drive the flipflop without going through the LUT first, which helps to reduce the amount of logic required. This feature helps designs that are logic limited. Additionally, the XLR has an improved routing structure that increases the routability of hard-to-route designs.

Figure 2: Logic Cell Functions



Note:

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® Titanium Primitives User Guide** for details on the Titanium™ logic cell primitives.

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.

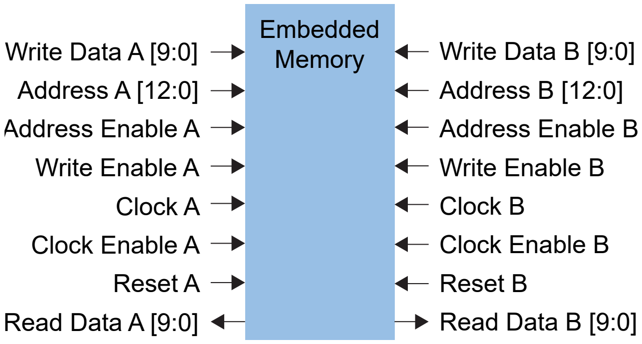
Learn more: Refer to the **Quantum® Titanium Primitives User Guide** for details on the Titanium™ RAM configuration.

True Dual-Port Mode

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

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

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

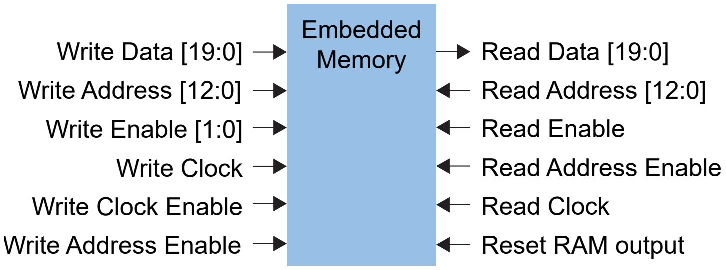


Simple Dual-Port Mode

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

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

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



DSP Block

The Titanium 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:

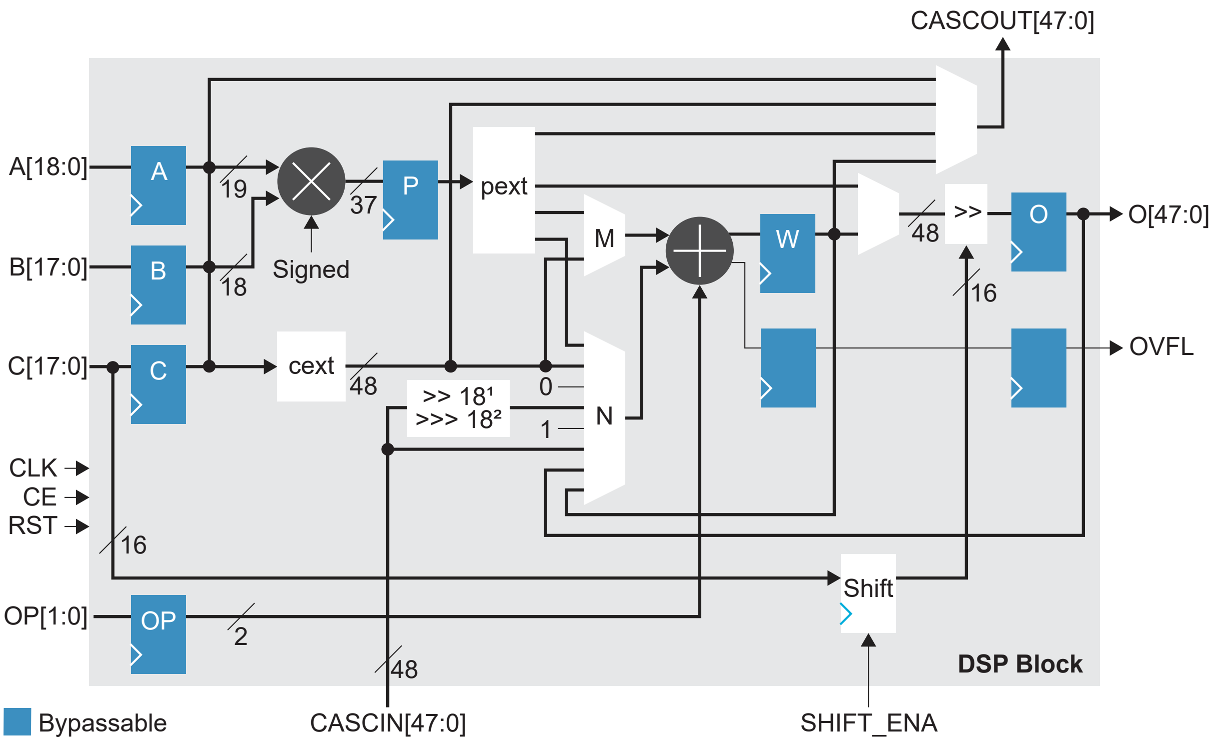
- *Normal*—One 19 × 18 integer multiplication with 48-bit addition/subtraction.
- *Dual*—One 11 × 10 integer multiplication and one 8 × 8 integer multiplication with two 24-bit additions/subtractions.
- *Quad*—One 7 × 6 integer multiplication and three 4 × 4 integer multiplications with four 12-bit additions/subtractions.

Important: The 7 × 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® Titanium Primitives User Guide for details on the Titanium™ DSP block primitives.

Clock and Control Network

The Ti125 FPGA has an enhanced clock network comprising global, regional, and local clocks as well as PHY clocks and byte clocks. 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.

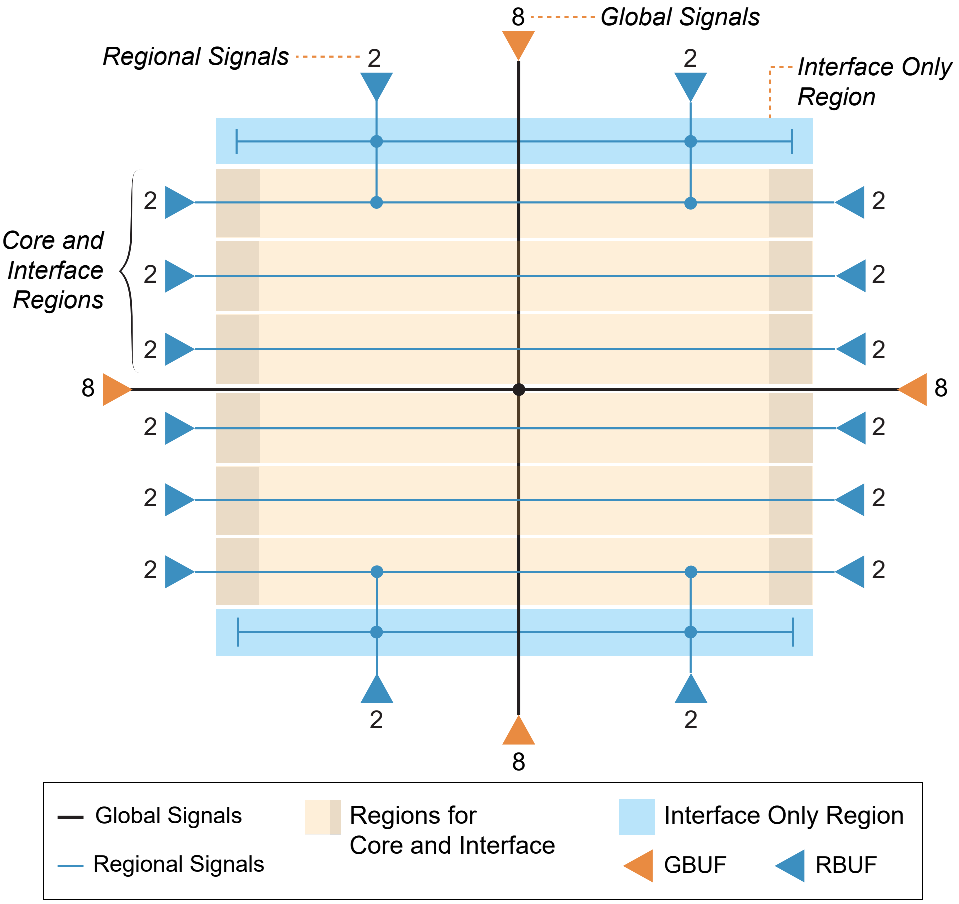
The FPGA has 32 global signals that can be used as either clocks or control signals. The global signals are balanced trees that feed the whole FPGA.

The FPGA also has regional signals that can only reach certain FPGA regions, including the top or bottom edges. The FPGA has 6 regional networks for the core, right interface, and left interface blocks. The top and bottom interface blocks have one 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

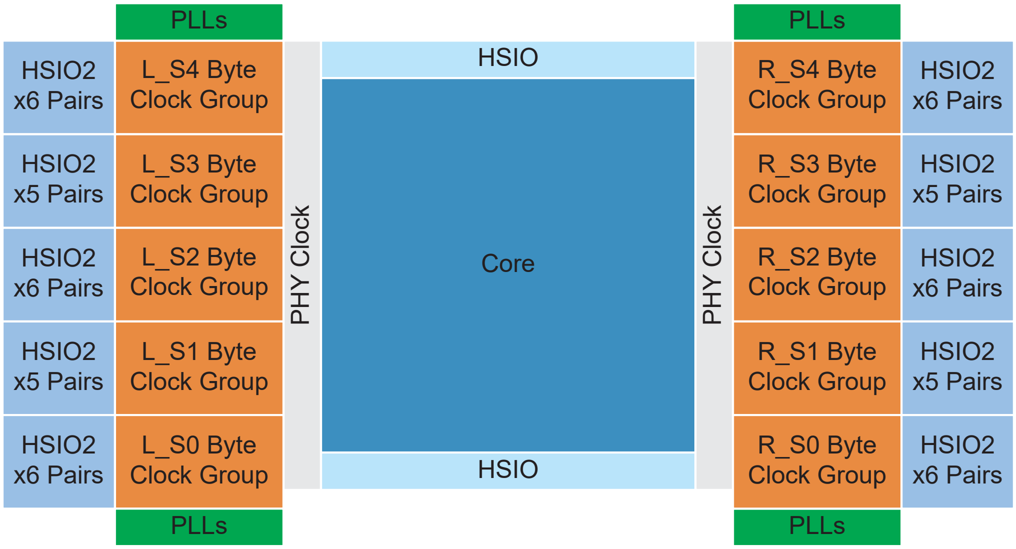


The regional clocks can also drive adjacent regions. See **Driving the Regional Network** on page 20 for details.

This figure shows the regional clock sources driven from the periphery, some devices have additional regional buffers that can only be accessed from the core. The Efinity software shows these regional buffers in report files.

In the Ti125 FPGA, the HSIO on the top and bottom connect to the global and regional clock networks. The HSIO2 on the left and right connect to the global and regional clock networks as well as the PHY clocks and byte clock groups.

Figure 7: PHY and Byte Clock Floorplan Overview



There are five or six differential pairs of HSIO2 blocks in each byte clock group. You can cascade L_S2 with L_S1 and R_S2 with R_S1 to create a x8 MIPI interface. See **Cascaded MIPI Clock Lanes** on page 65.

Clock Sources that Drive the Global and Regional Networks

The Ti125 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 RX Lane (configured as clock lane)	Supports GCLK (default) and RCLK. You can only use resources that are identified as clocks.
Clock modifiers	The byte clock group's clock modifiers have a forward clock (FWDCLK) that can drive the global and regional clock networks.
PLL	All output clocks connect to the global network. The PLL output can also drive the PHY clock network Refer to **Driving the Regional Network** on page 20 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

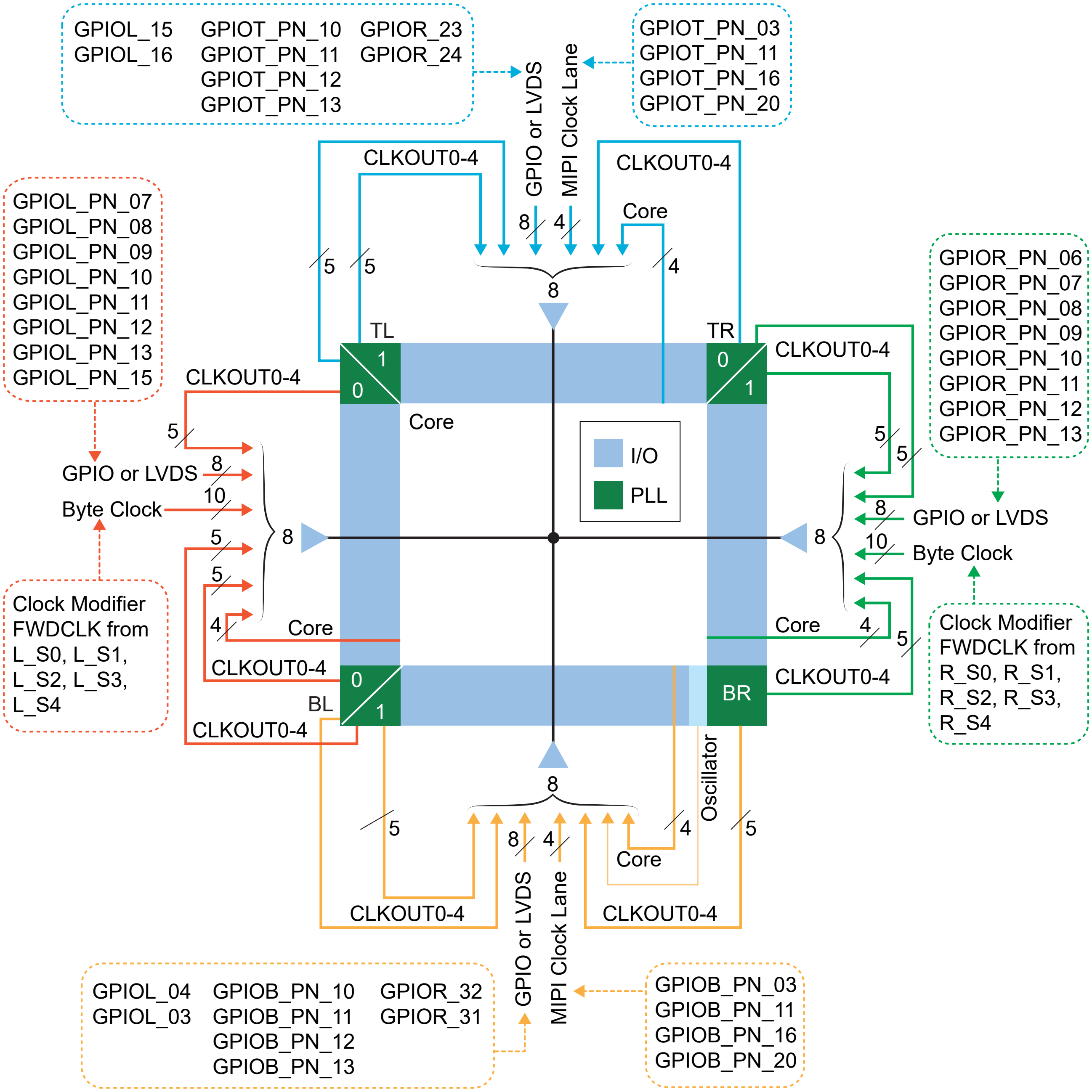
You can access the global clock network using the global clock GPIO pins, PLL outputs, oscillator output, clock modifiers, and core-generated clocks.

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® Titanium 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 8: Clock Sources that Drive the Global Network



Each byte clock group has clock modifiers (CMs) that can drive the global clock network. In the figure, S0L - S4L and S0R - S4R are the byte clock groups on the left and right side.

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

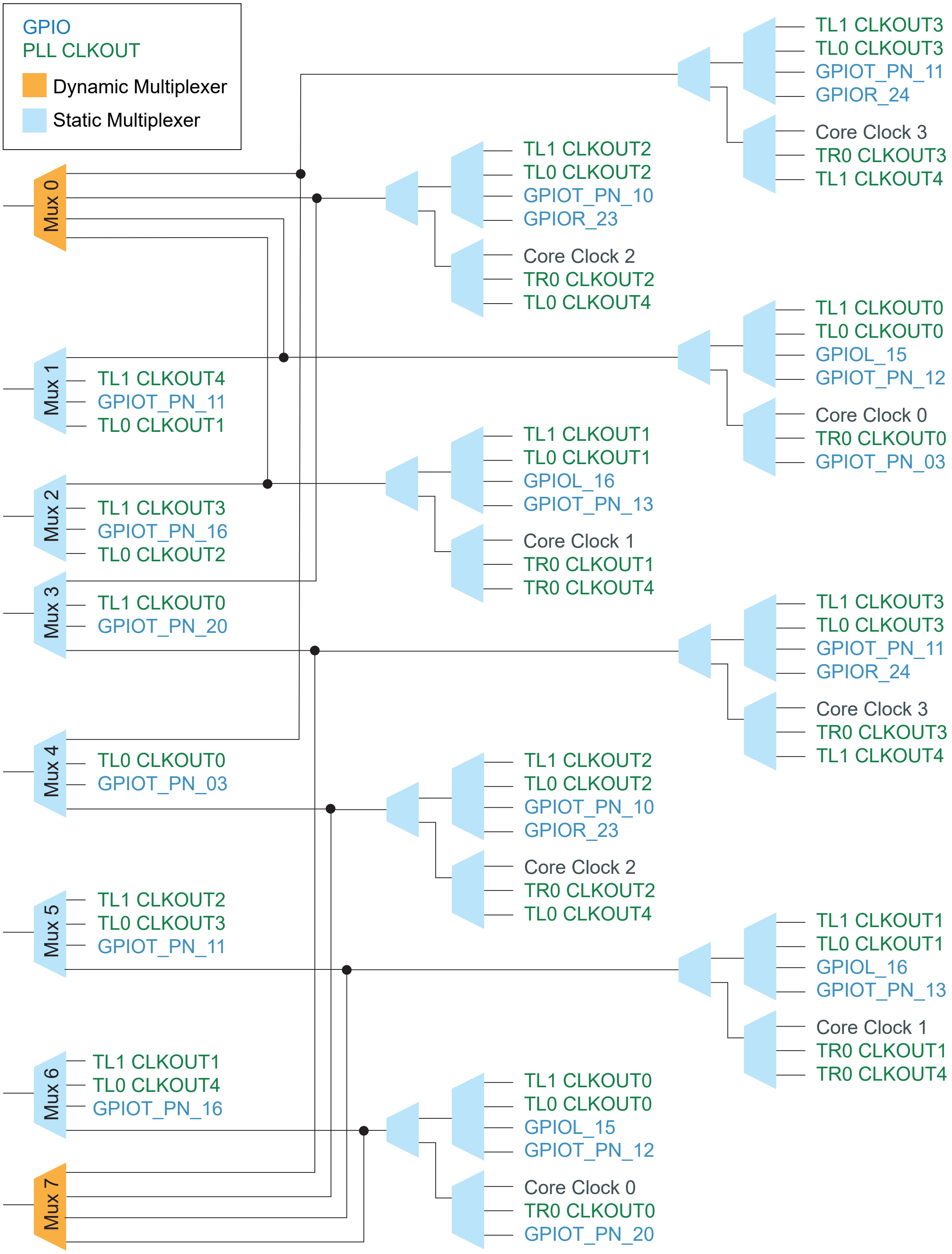


Figure 10: Clock Sources that Drive the Multiplexers: Bottom

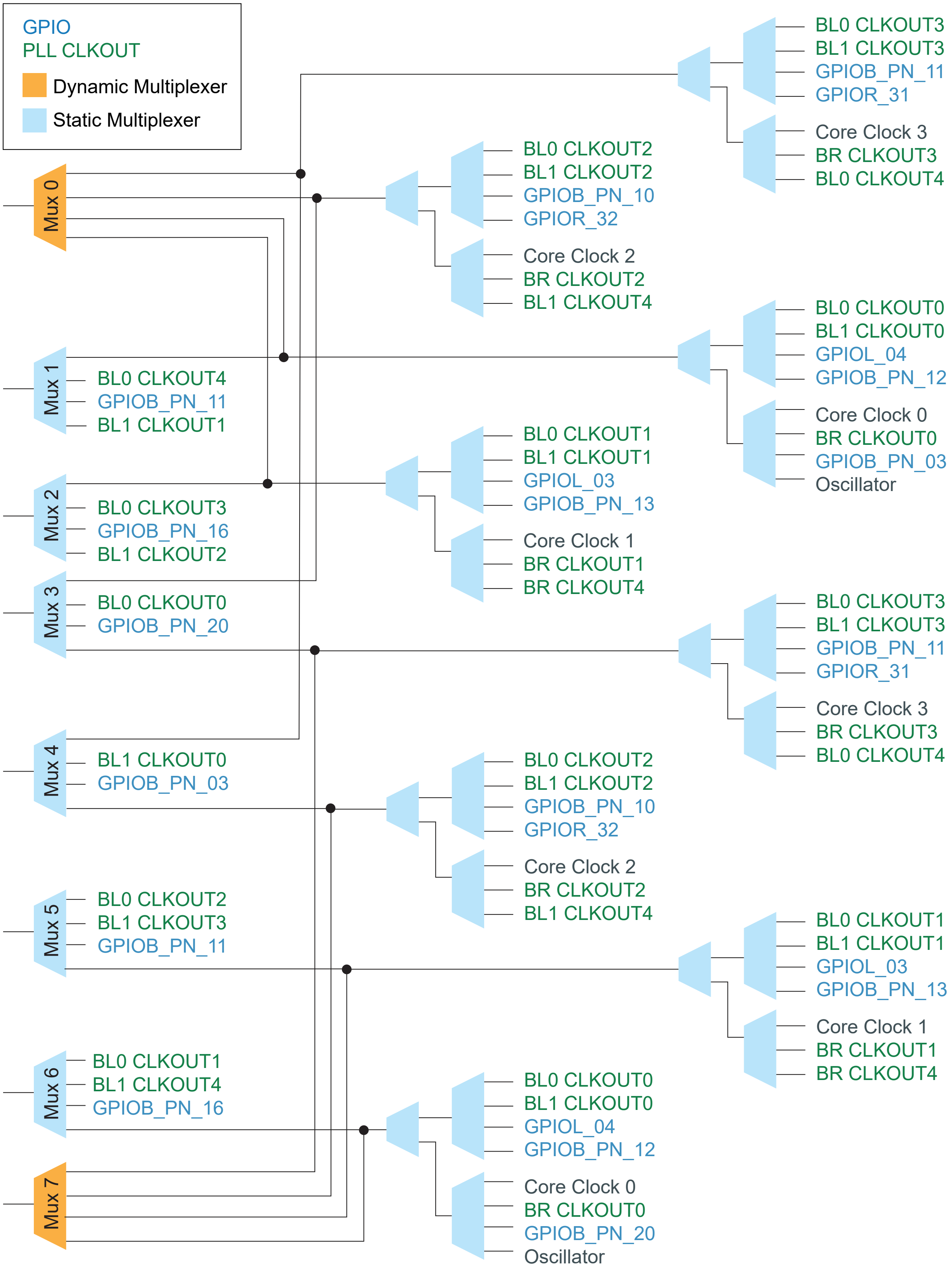


Figure 11: Clock Sources that Drive the Multiplexers: Left

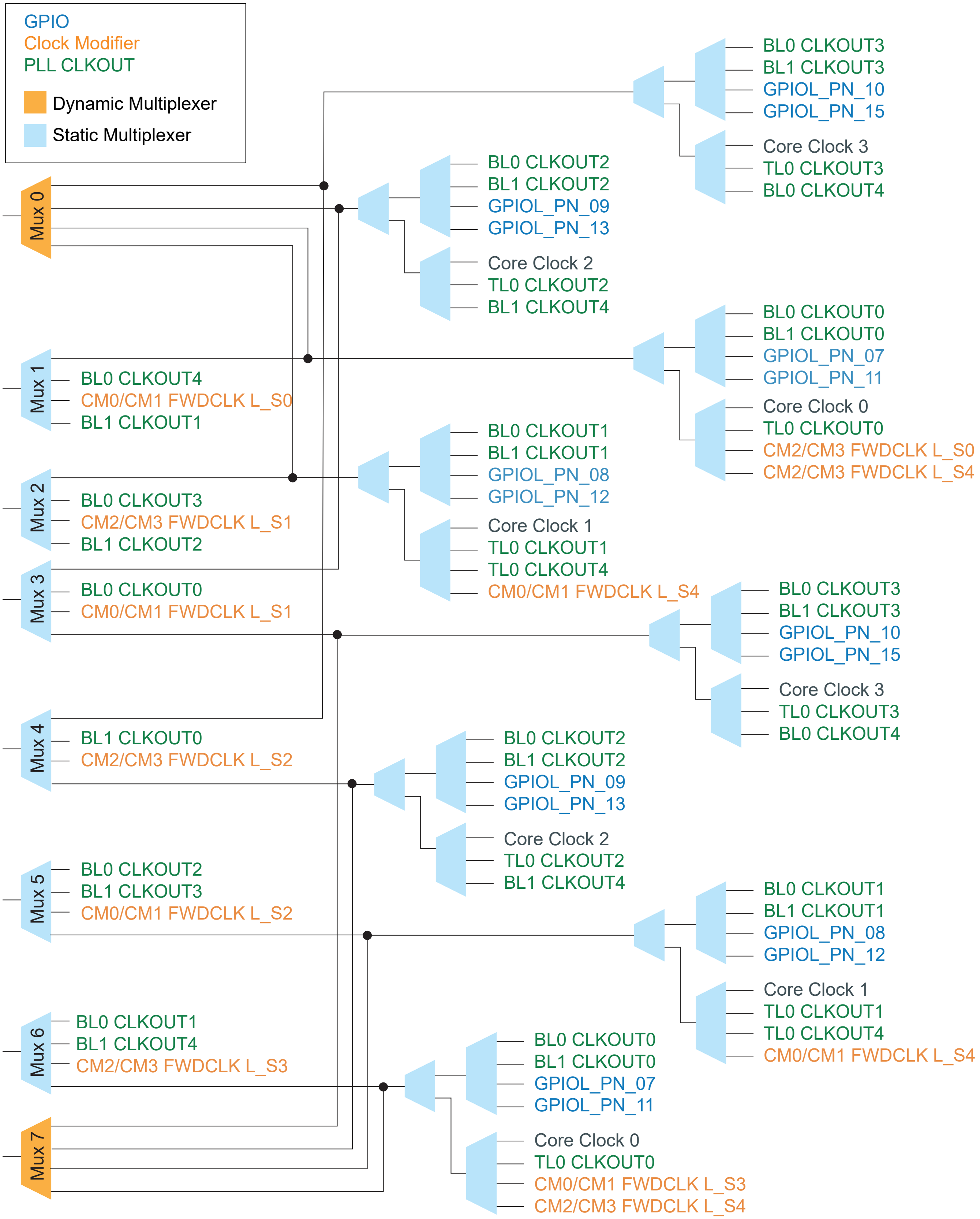
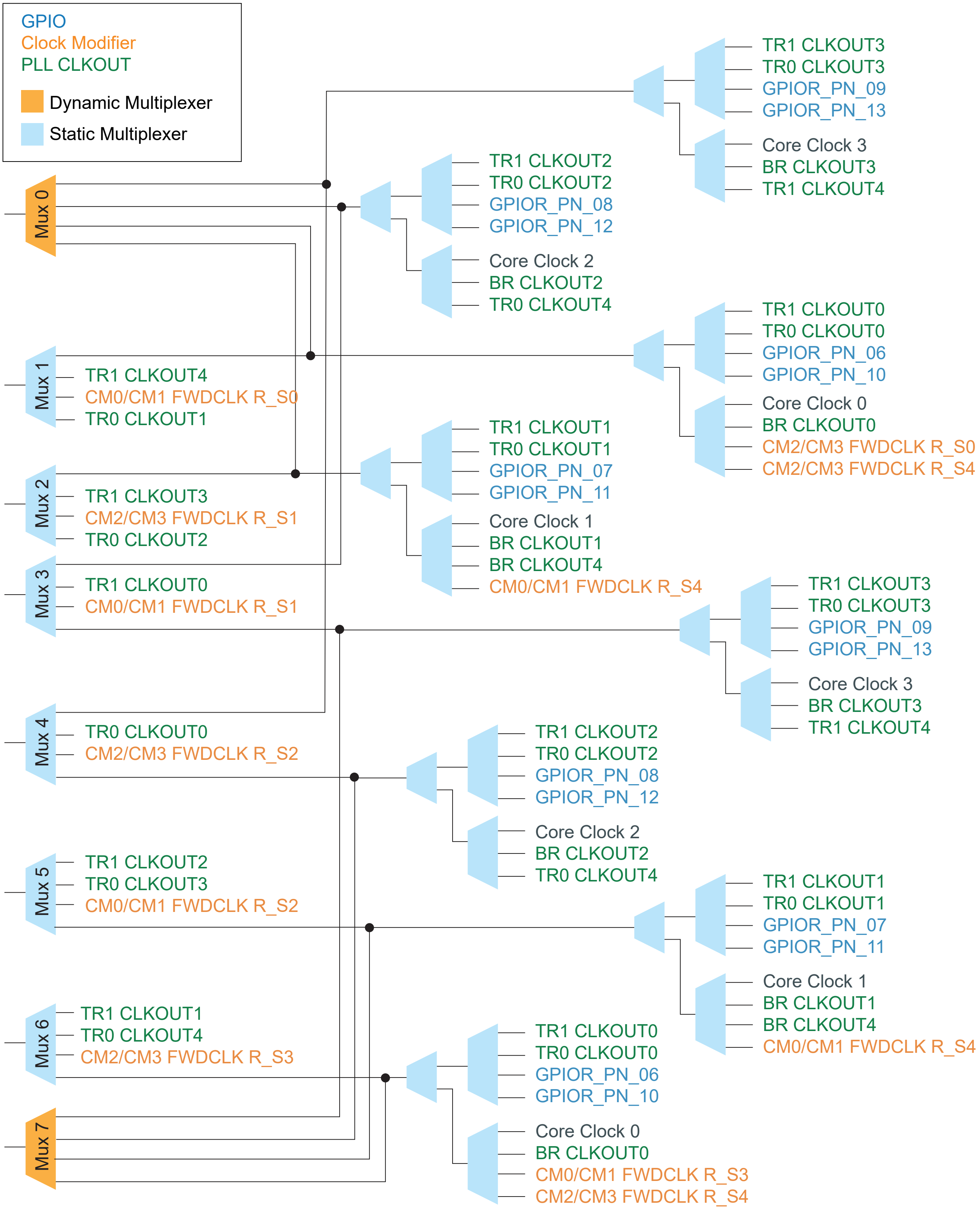


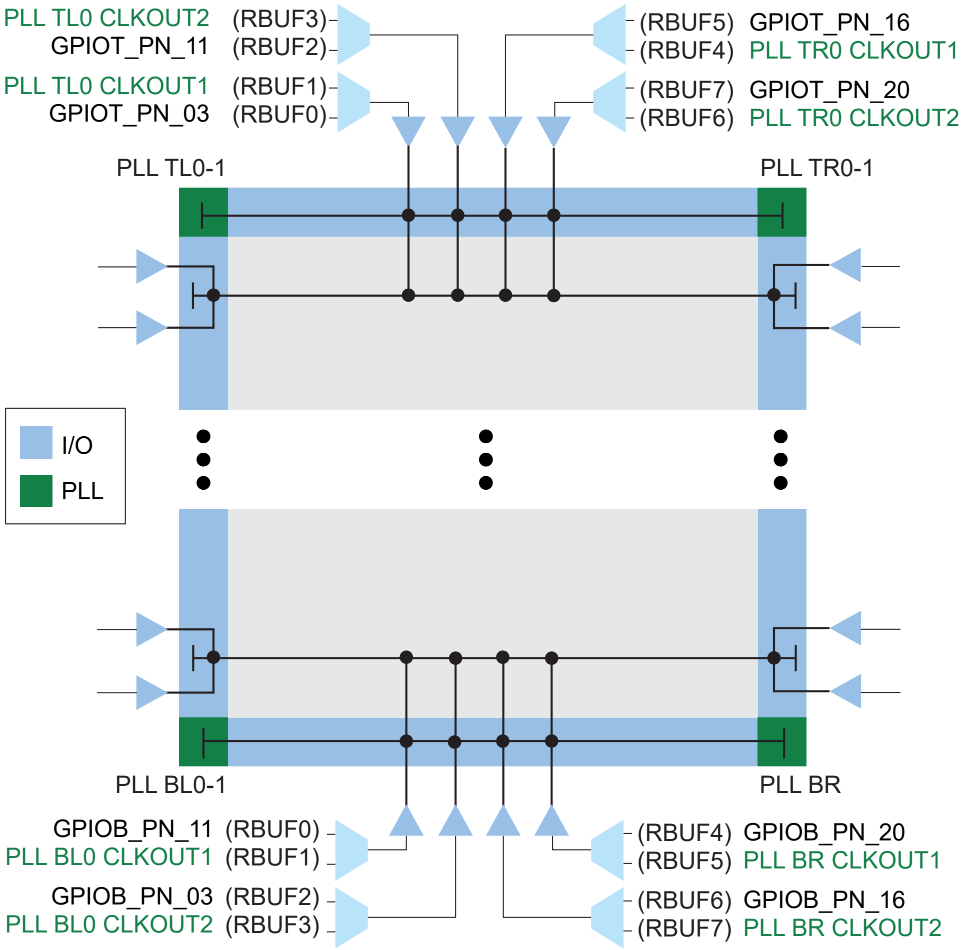
Figure 12: Clock Sources that Drive the Multiplexers: Right



Driving the Regional Network

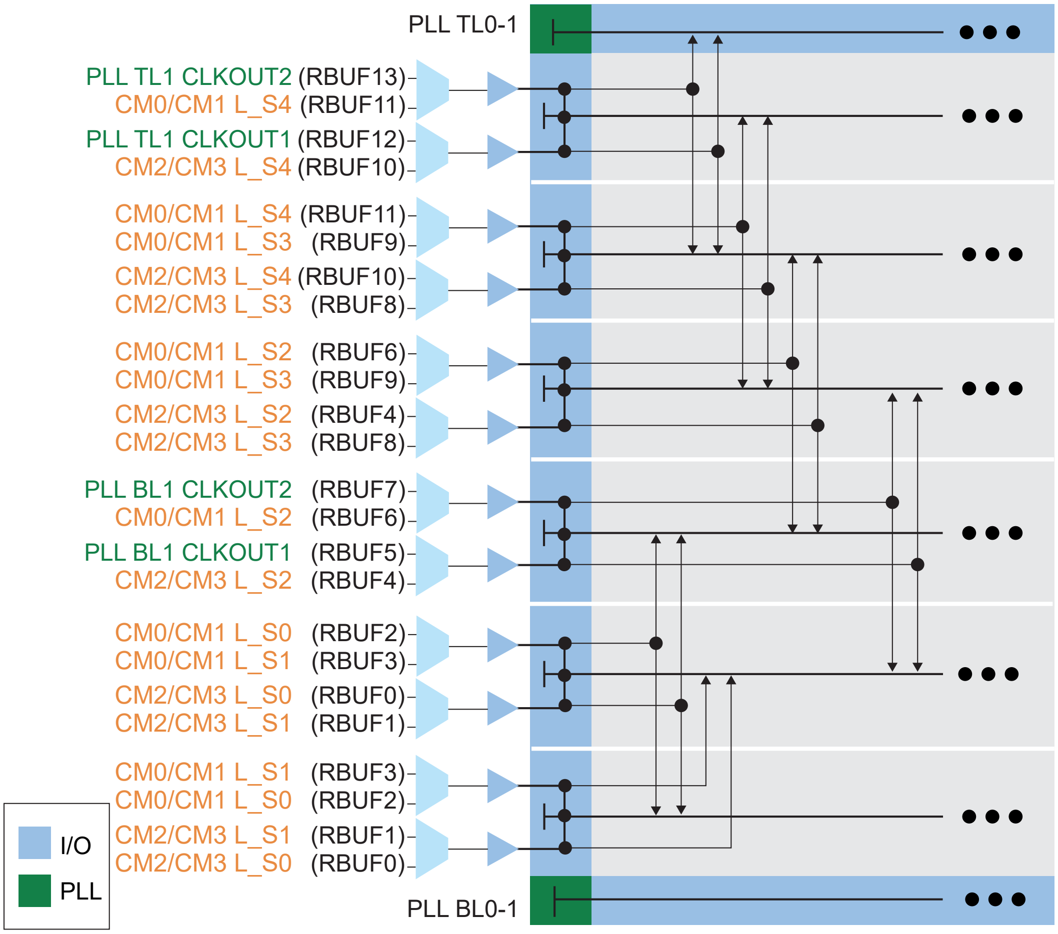
The following figures show the regional network clock sources graphically.

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



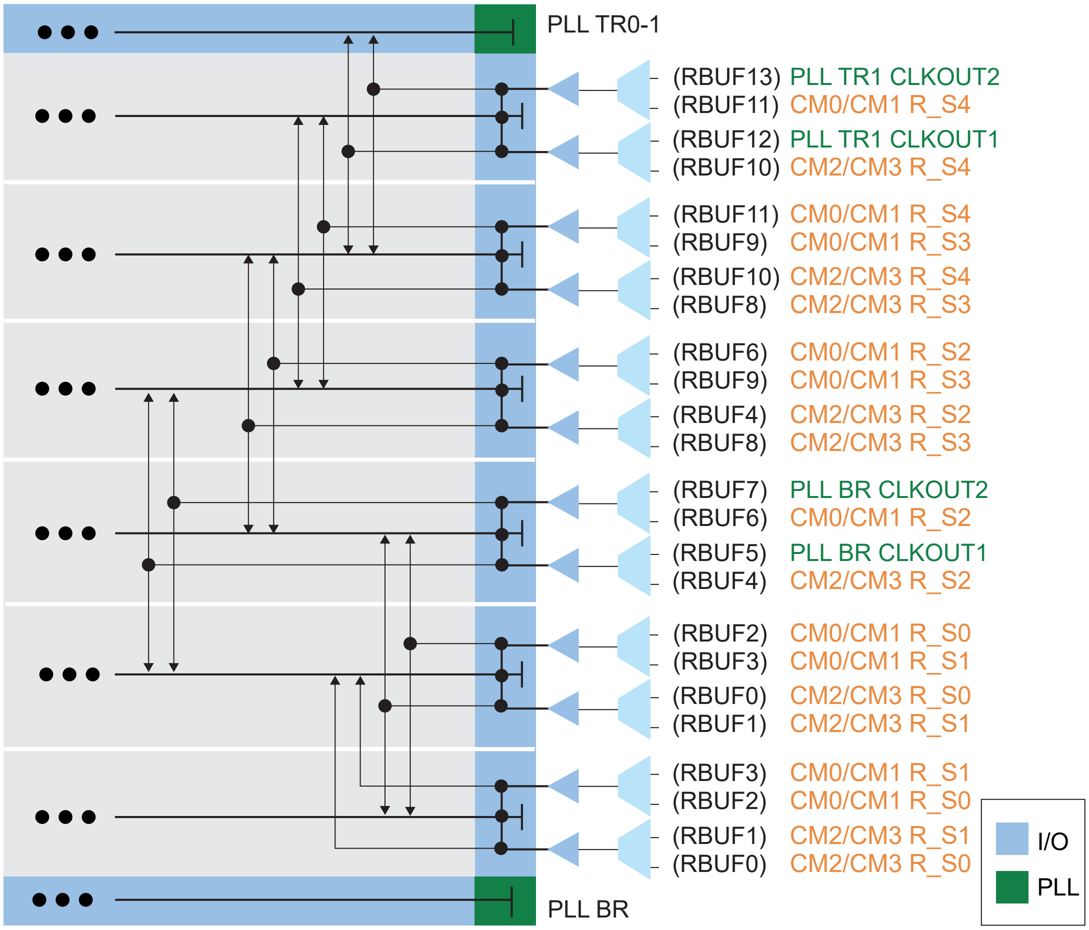
See **Figure 14: Clock Sources that Drive the Regional Network: Left** on page 21 and **Figure 15: Clock Sources that Drive the Regional Network: Right** on page 22 for the left and right connections.

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



The clock modifier (CM) comes from a PLL clock output or an HSIO2 pin.

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



The clock modifier (CM) comes from a PLL clock output or an HSIO2 pin.

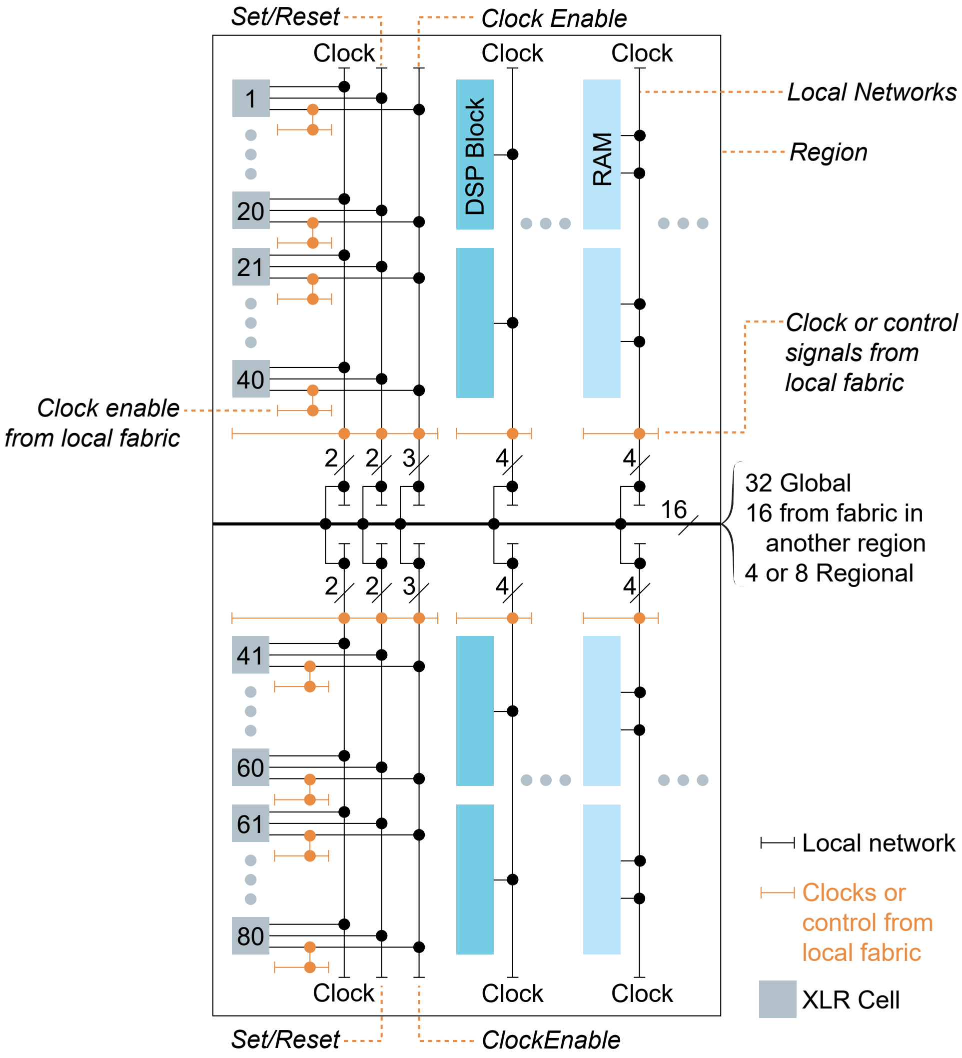
Driving the Local Network

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 its 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)

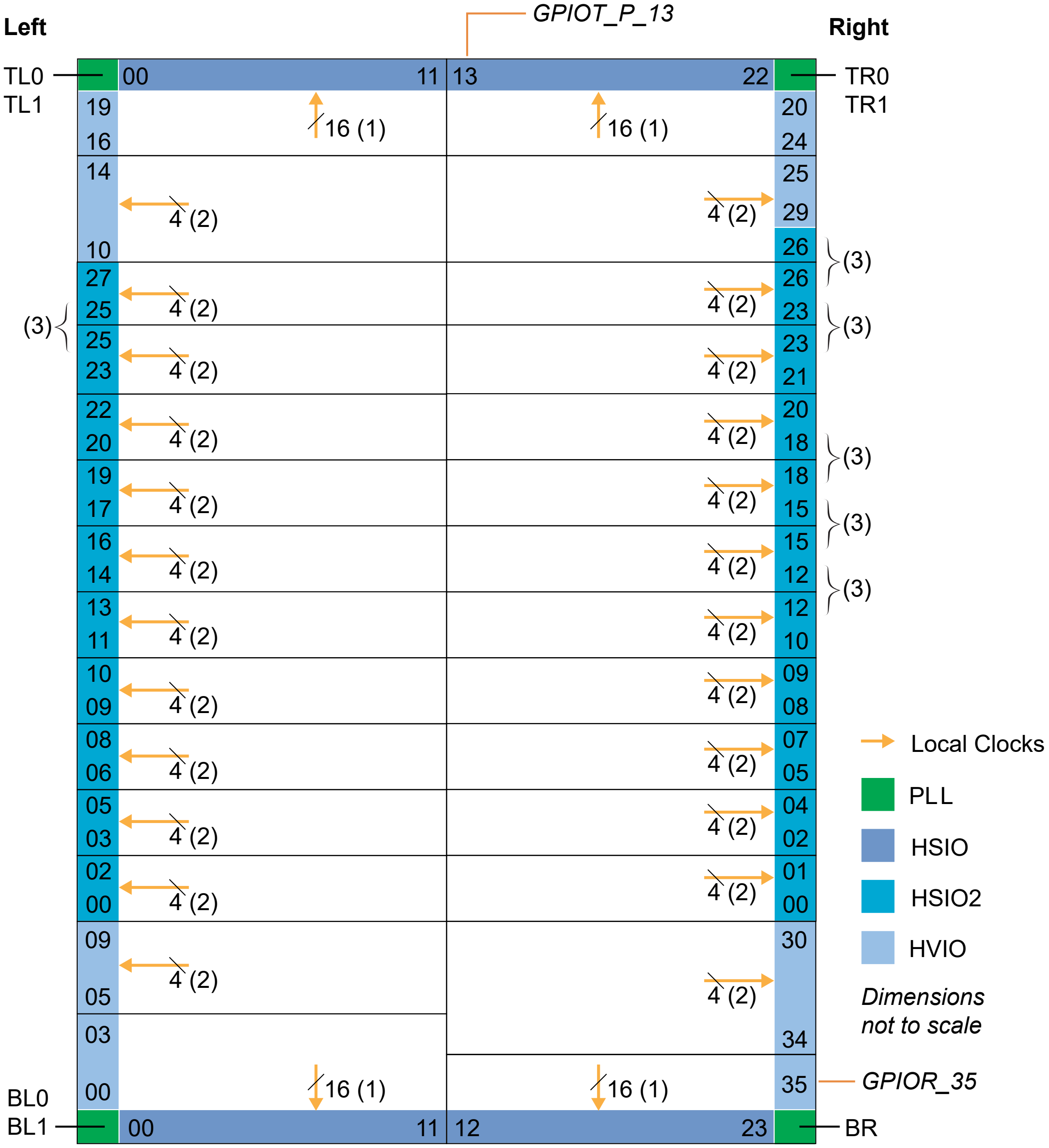
Additionally, the local fabric can generate clock and control signals for the local network. The fabric can also drive the clock enable for the XLR cell directly, allowing each XLR cell to have a unique clock enable.

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 two from the fabric.
- The left and right regions can each support up to four unique clock signals. Up to two 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



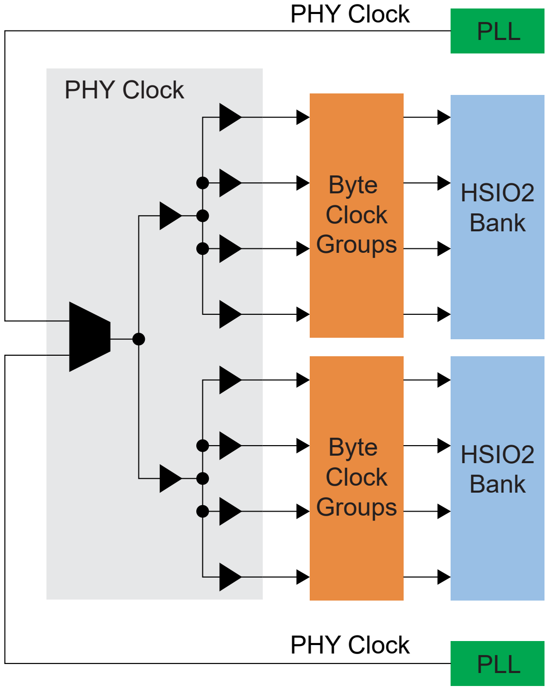
Notes:

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. Some GPIO are in overlapping regions. To make it easier to see to which region the GPIO belongs, the GPIO in overlapping regions are listed twice.

PHY Clock Network

Dedicated low-skew connection from the PLLs to the HSIO2 on the left and right of the FPGA. Because the connection is very low skew, you can run the interfaces at high speed.

Figure 18: PHY Clock Network

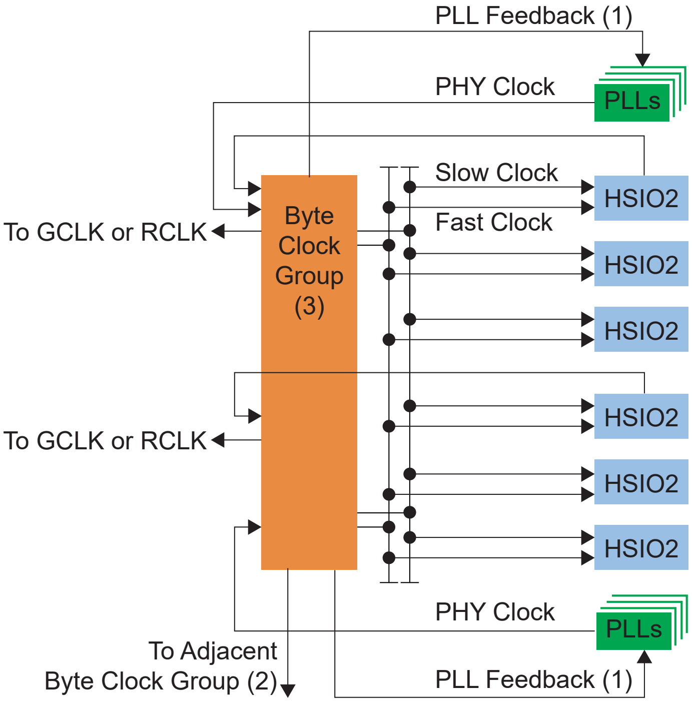


Byte Clock Groups

The PLL clock output can feed the PHY clock network. The PHY clock delivers high-speed clocks to byte clock groups, which performs clock management (division, phase shift, and delay) before the clock signal reaches the HSIO2 blocks. Byte clock groups have either five or six HSIO2 blocks.

One byte clock group on each side of the device can cascade a clock signal to the adjacent group. S2 (x6 group) can cascade to S1 (x5 group) (see Cascaded MIPI Clock Lanes on page 65).

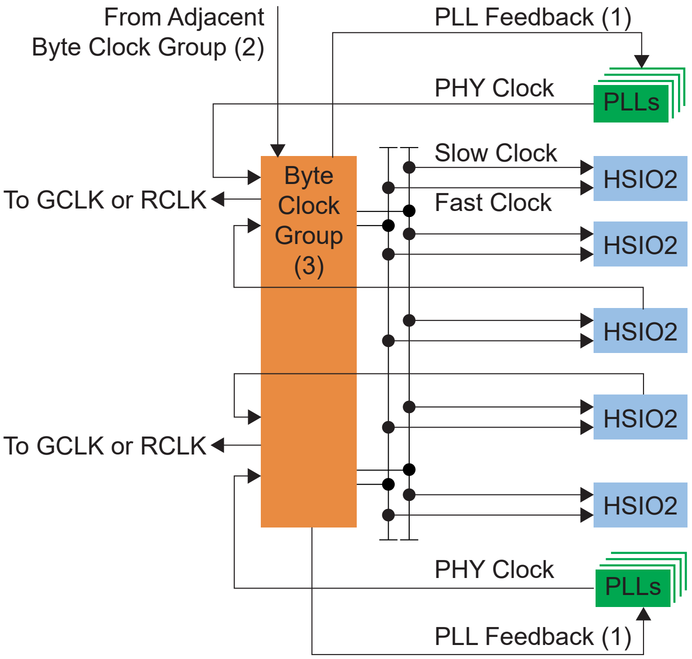
Figure 19: Byte Clock Group with Six HSIO2



Notes:

1. Only the top and bottom byte clock groups on each side can feed back to the PLL.
2. Only one byte clock group on each side can cascade to the adjacent byte clock group.
3. Refer to Figure 21: Byte Clock Group Block Diagram on page 29 for details.

Figure 20: Byte Clock Group with Five HSIO2



Notes:

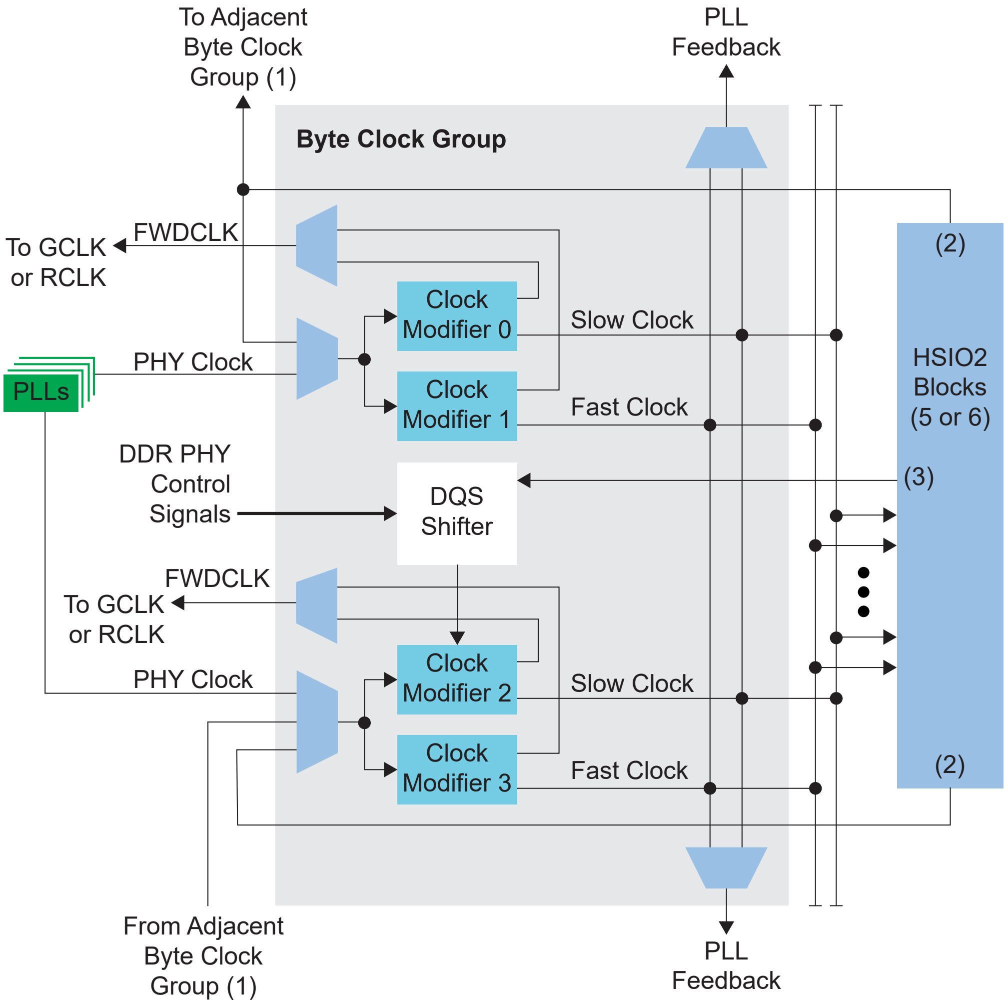
1. Only the top and bottom byte clock groups on each side can feed back to the PLL
2. Only one byte clock group on each side can receive the cascaded signal from the adjacent byte clock group.
3. Refer to Figure 21: Byte Clock Group Block Diagram on page 29 for details.

Each byte clock group has four clock modifiers (CM0 - CM3) that generate fast and slow clocks. These clocks connect to the HSIO2 and can feed back to the global or regional clock networks (`FWDCLK` signal). Each clock modifier has a clock divider and a phase shifter. `FWDCLK` uses the divider; the fast/slow clocks use the divider and phase shifter.

Note: For details on how the clock modifiers are used, refer to:
- HSIO2 Configured as LVDS on page 55
- HSIO2 Configured as MIPI Lane on page 61
- HSIO2 Configured as QDRIO on page 66

In QDRIO mode, the DQS shifter handles the DDR PHY control signals for read operations.

Figure 21: Byte Clock Group Block Diagram



Notes:

1. Only one byte clock group on each side can cascade to the adjacent byte clock group.
2. Only the top and bottom byte clock groups on each side can feed back to the PLL
3. Refer to HSIO2 Configured as QDRIO on page 66 for the DDR PHY read and write clocking.

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 Titanium™ family support a variety of interfaces to meet the needs of different applications.

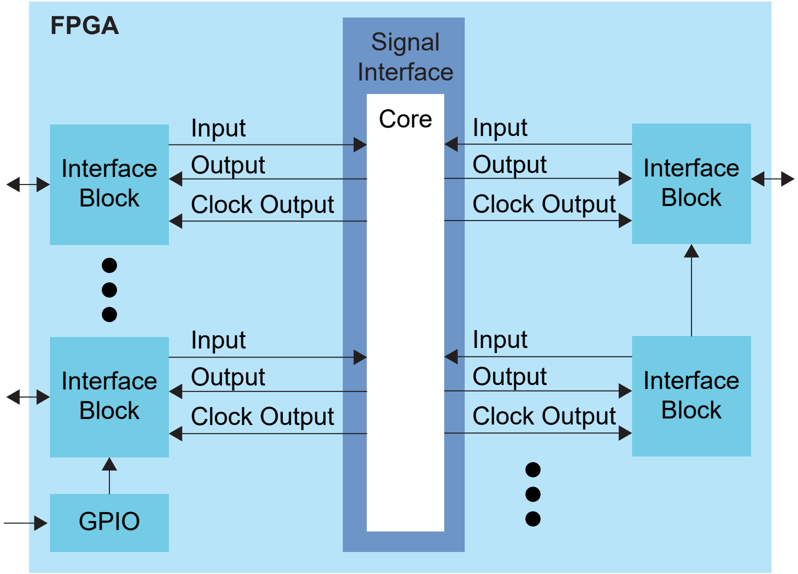
Learn more: The following sections describe the available device interface features in Ti125 FPGAs. Refer to the **Titanium Interfaces User Guide** for details on the Efinity® Interface Designer settings.

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 22: 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 Titanium™ 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 Ti125 FPGA supports the following 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.
- *Enhanced high-speed I/O (HSIO2)*—Complex I/O blocks that can support single-ended and differential I/O functionality at high speeds.

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 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 pins support 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	0.5 * VCCIO	GPIO

Standard	VCCIO (V)		VCCAUX (V)	VREF (V)	When Configured As
	TX	RX			
HSTL/Differential HSTL 1.5 V SSTL/Differential SSTL 1.5 V	1.5	1.5, 1.8[(2)]	1.8	0.5 * VCCIO	GPIO
SSTL/Differential SSTL 1.35 V	1.35	1.35, 1.5, 1.8[(2)]	1.8	0.5 * VCCIO	GPIO
HSTL/Differential HSTL 1.2 V SSTL/Differential SSTL 1.2 V	1.2	1.2, 1.35, 1.5, 1.8[(2)]	1.8	0.5 * VCCIO	GPIO
LVDS/RSDS/mini-LVDS	1.8	1.5, 1.8[(2)]	1.8	–	LVDS
Sub-LVDS	1.8	1.5, 1.8[(2)]	1.8	–	Sub-LVDS
MIPI	1.2	1.2	1.8	–	MIPI Lane
SLVS	1.2	1.2	1.8	–	SLVS

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, HSIO, and HSIO2 Configured as GPIO

The following table describes the features for HVIO, HSIO, and HSIO2 configured as GPIO.

Table 7: Features for HVIO, HSIO, and HSIO2 Configured as GPIO

Feature	HVIO	HSIO and HSIO2 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	–	✓

---

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

Table 8: GPIO Modes

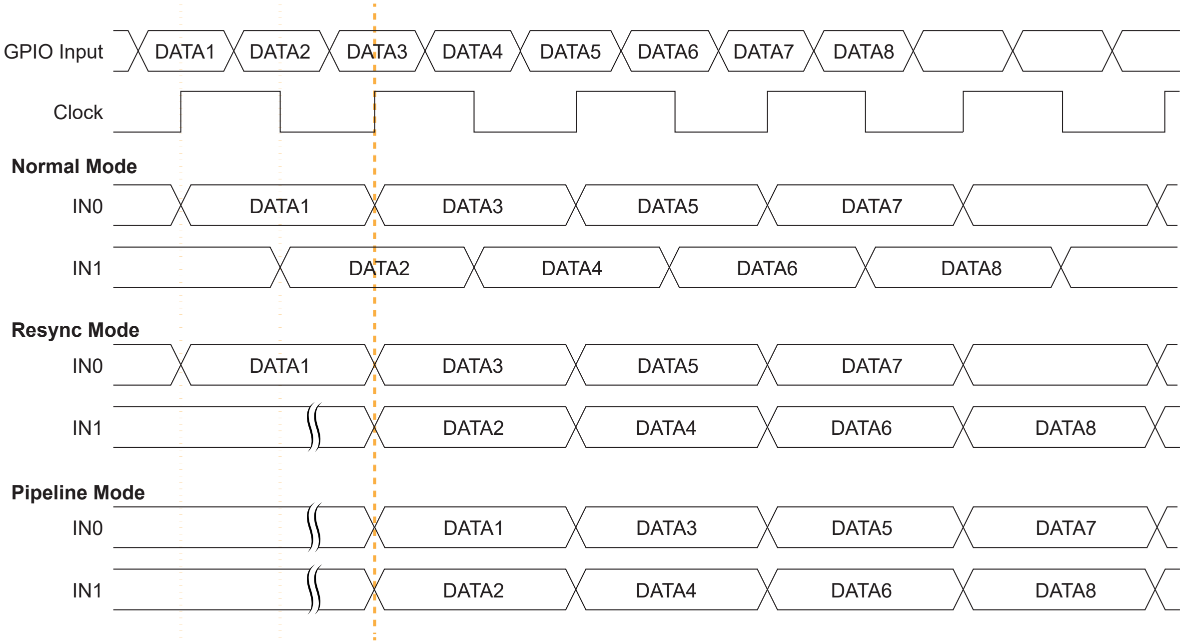
GPIO Mode	Description
Input	Only the input path is enabled; optionally registered. If registered, the input path uses the input clock to control the registers (positively or negatively triggered). Select the alternate input path to drive the alternate function of the GPIO. The alternate path cannot be 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

Ti125 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.

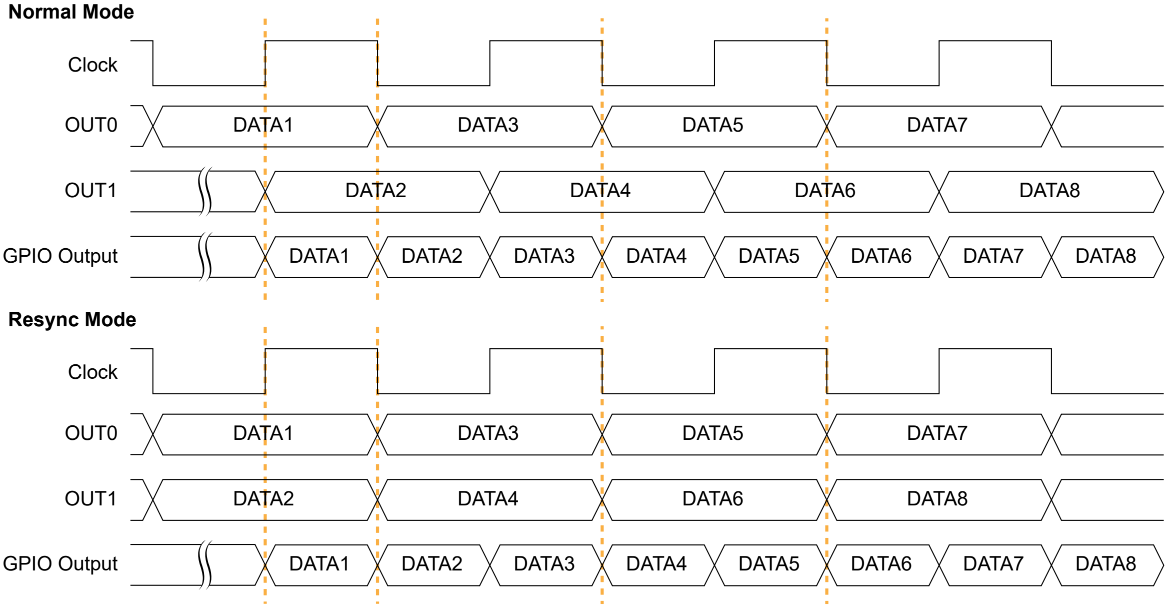
Figure 23: DDIO Input Timing Waveform



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 24: 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, HSIO, and HSIO2 configured as GPIO support programmable delay chain. In some cases you can use static and dynamic delays at the same time.

Table 9: HVIO Programmable Delay Support

GPIO Type	Delay Steps	
	Static Delay	Dynamic Delay
Single-Ended		
HVIO input	16	N/A
HVIO output	16	N/A

Table 10: HSIO Programmable Delay Support

GPIO Type	Delay Steps	
	Static Delay	Dynamic Delay
Single-Ended		
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(3)	64(3)

Table 11: HSIO2 Programmable Delay Support

GPIO Type	Delay Steps	
	Static Delay	Dynamic Delay
Single-Ended		
HSIO2 P pin input	16	64
HSIO2 P pin output	16	N/A
HSIO2 N pin input	16	64
HSIO2 N pin output	16	N/A
Differential		
HSIO2 TX	64	N/A
HSIO RX	64(4)	64(4)

Learn more: Refer to the following tables for the delay step size:
Table 66: Single-Ended I/O Programmable Delay Chain Step Size: Static on page 98
Table 67: Single-Ended I/O Programmable Delay Chain Step Size: Dynamic on page 98
Table 68: Differential I/O Programmable Delay Chain Step Size: Static and Dynamic on page 98

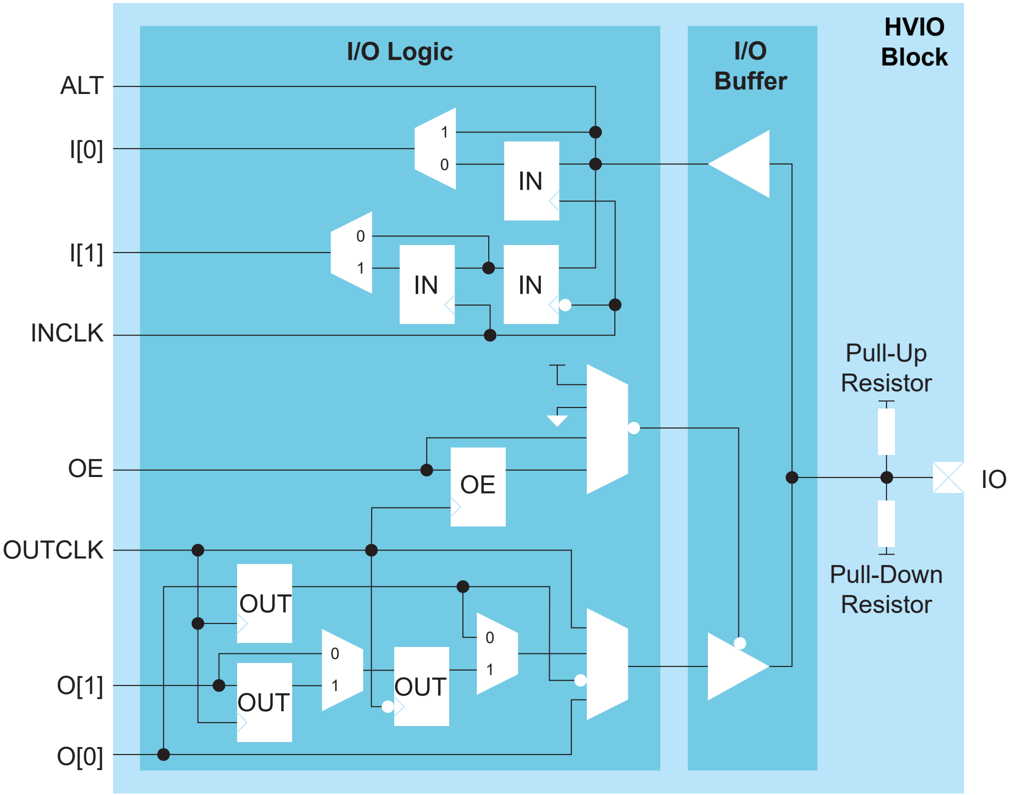
---

(3) 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 25: HVIO Interface Block



---

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

Table 12: 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 13: HVIO Pads

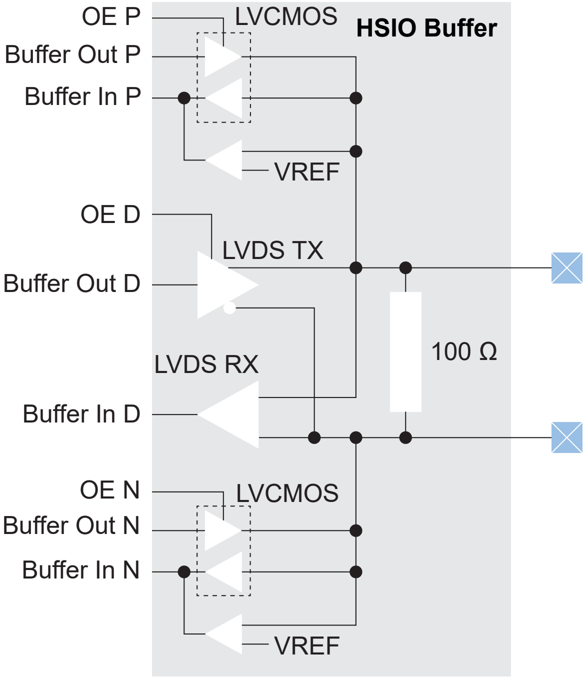
Signal	Direction	Description
IO	Bidirectional	HVIO pad.

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 26: HSIO Buffer Block Diagram



Important: When you are using an HSIO pin as a GPIO, make sure to leave at least one 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 a warning 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 27: I/O Interface Block

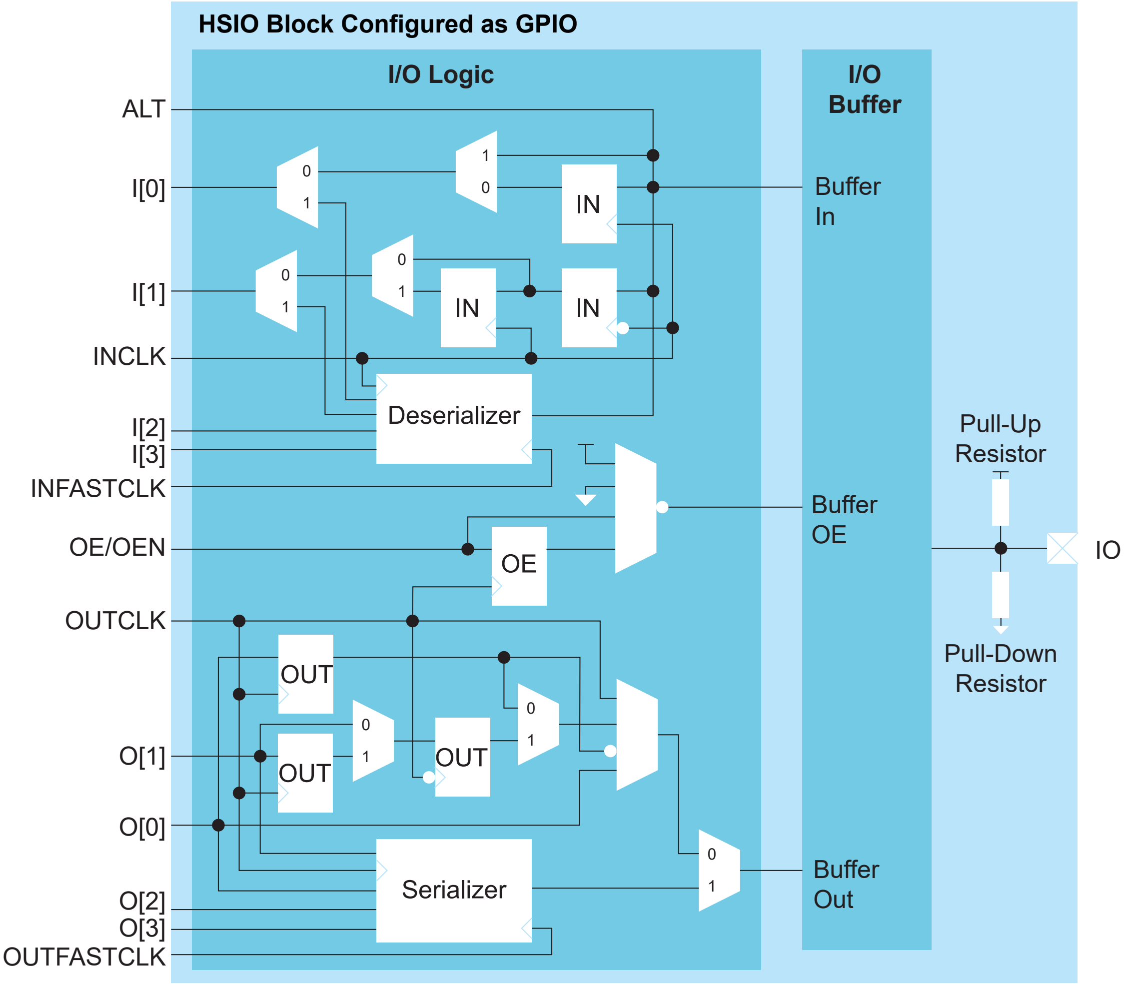


Table 14: HSIO Block Configured as GPIO Signals (Interface to FPGA Fabric)

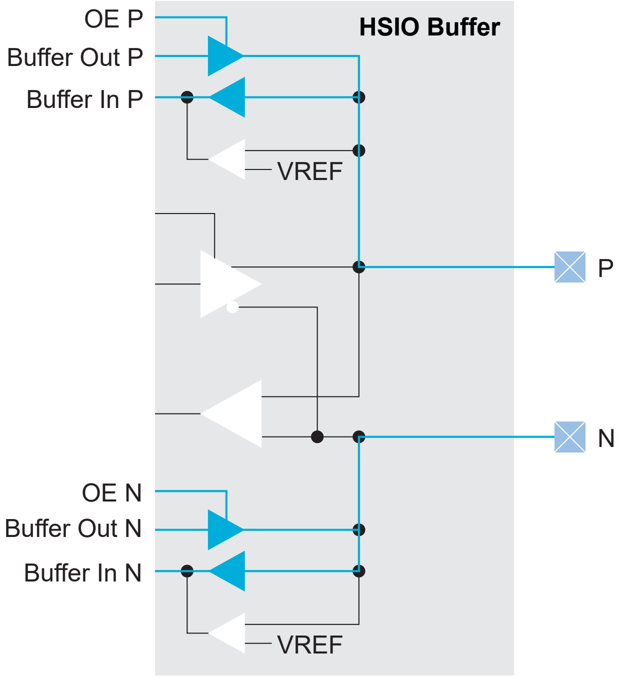
Signal	Direction	Description
I[3:0]	Output	Input data from the pad to the core fabric. I[0] is the normal input to the core. In DDIO mode, I[0] is the data captured on the positive clock edge (HI pin name in the Interface Designer) and I[1] is the data captured on the negative clock edge (LO pin name 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 15: 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.

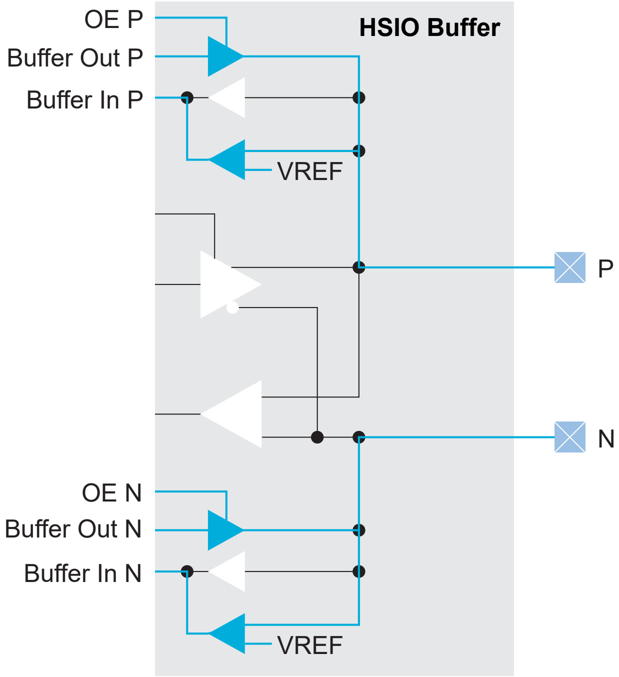
Figure 28: I/O Buffer Path for LVCMOS



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.

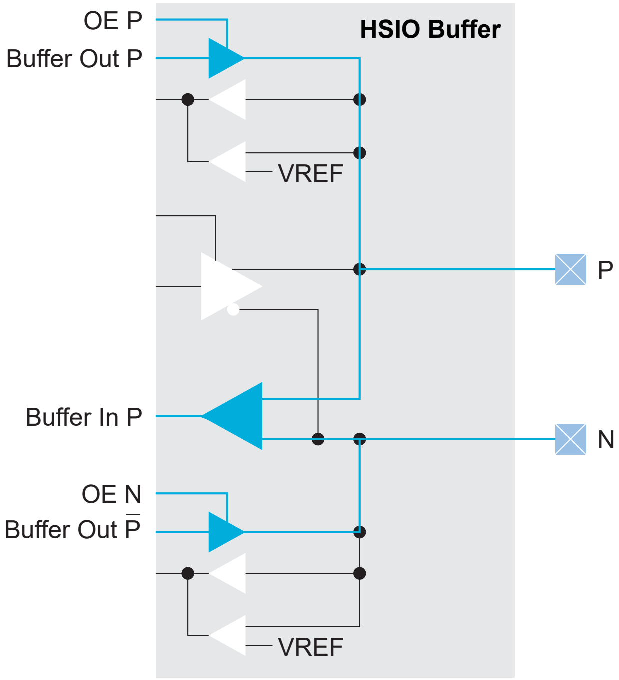
Important: When configuring an I/O pad of the standard's input path as a `VREF` pin, you must use the `VREF` from the same physical I/O bank even when the I/O banks are merged to share a common VCCIO pin.

Figure 29: 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 30: 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 1.5 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 16: 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 31: LVDS RX Interface Block Diagram

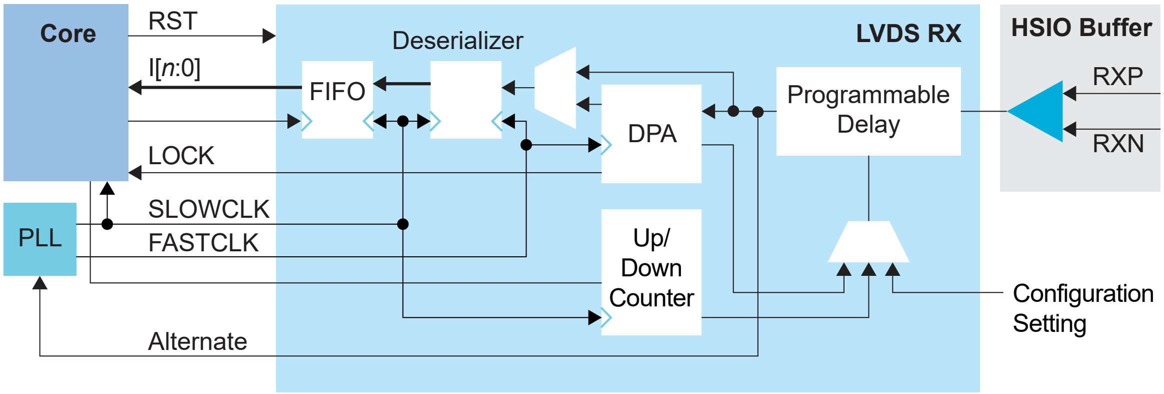
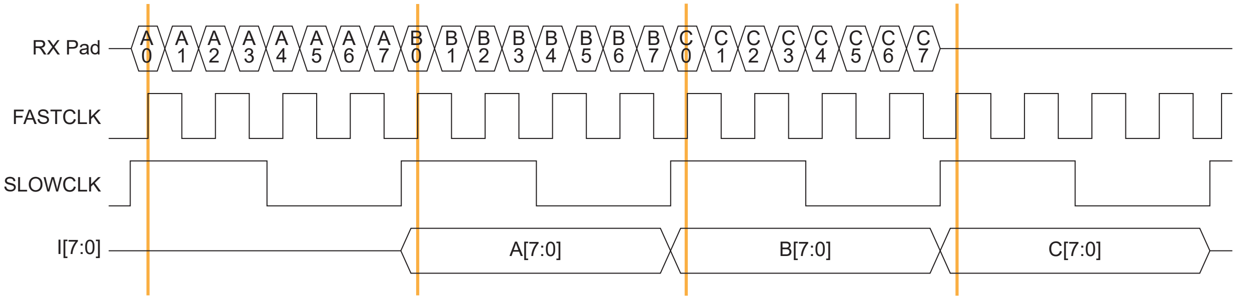


Table 17: LVDS RX Signals (Interface to FPGA Fabric)

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 32: 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 33: LVDS TX Interface Block Diagram

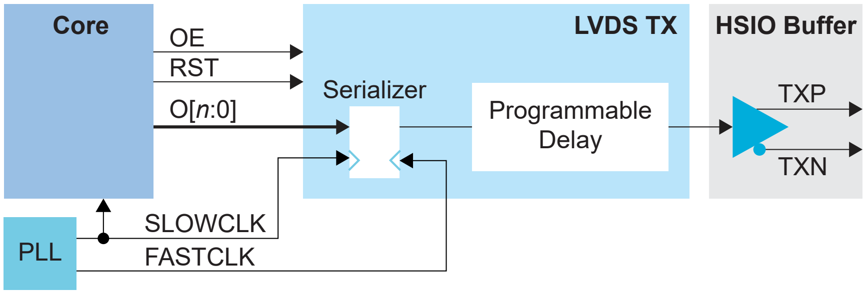
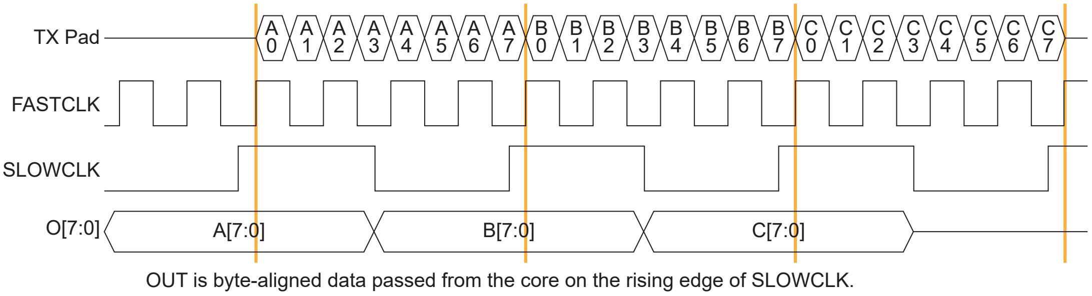


Table 18: LVDS TX Signals (Interface to FPGA Fabric)

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 34: 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 35: LVDS Bidirectional Interface Block Diagram

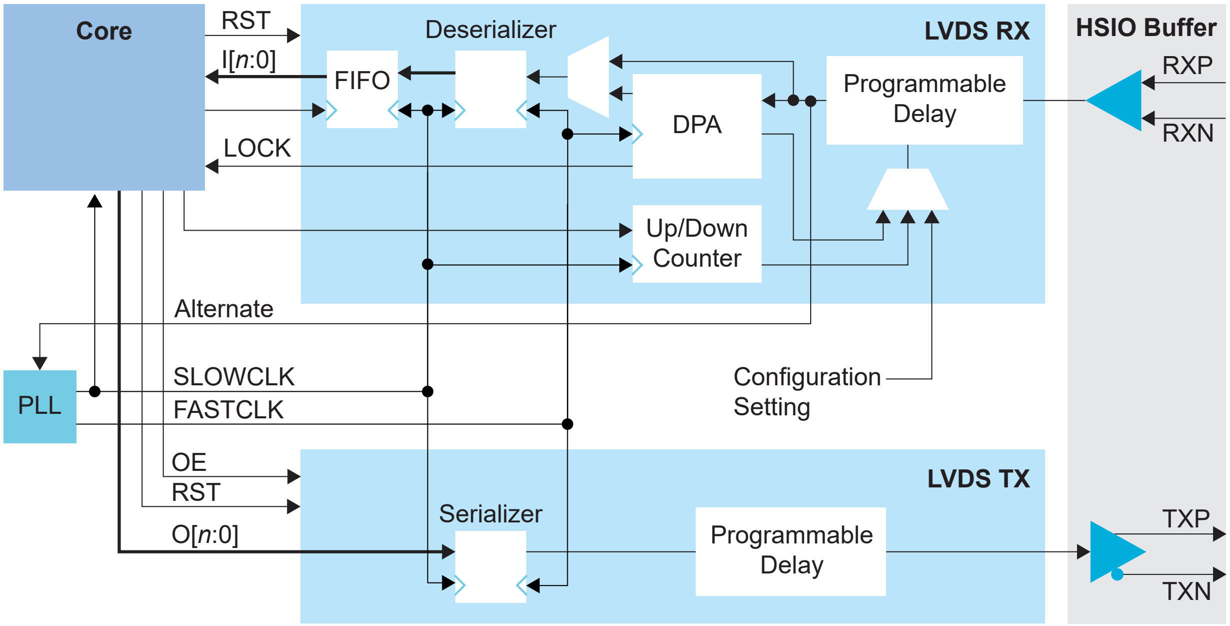


Table 19: LVDS Bidirectional Signals (Interface to FPGA Fabric)

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 20: 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 1.5 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.

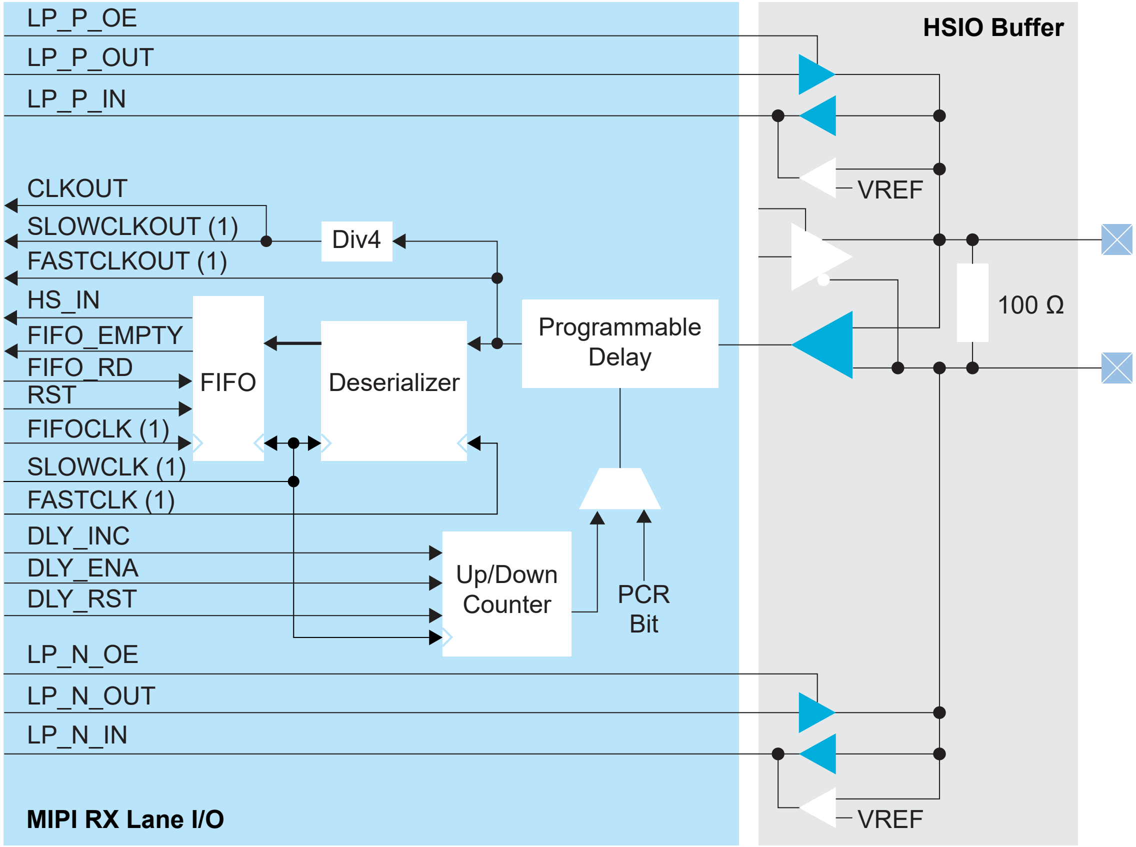
The data lane fast and slow clocks must be driven by a clock lane in the same MIPI group (dedicated buses drive from the clock lane to the neighboring data lanes).

The MIPI RX function is defined as:

Table 21: 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* = I5, I6, I7, I13, I14, I15 MIPI RX channel (up to 12 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* = I5, I6, I7, I13, I14, I15 MIPI RX channel (up to 12 MIPI RX channels)

Figure 36: MIPI RX Lane Block Diagram



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

Table 22: 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.

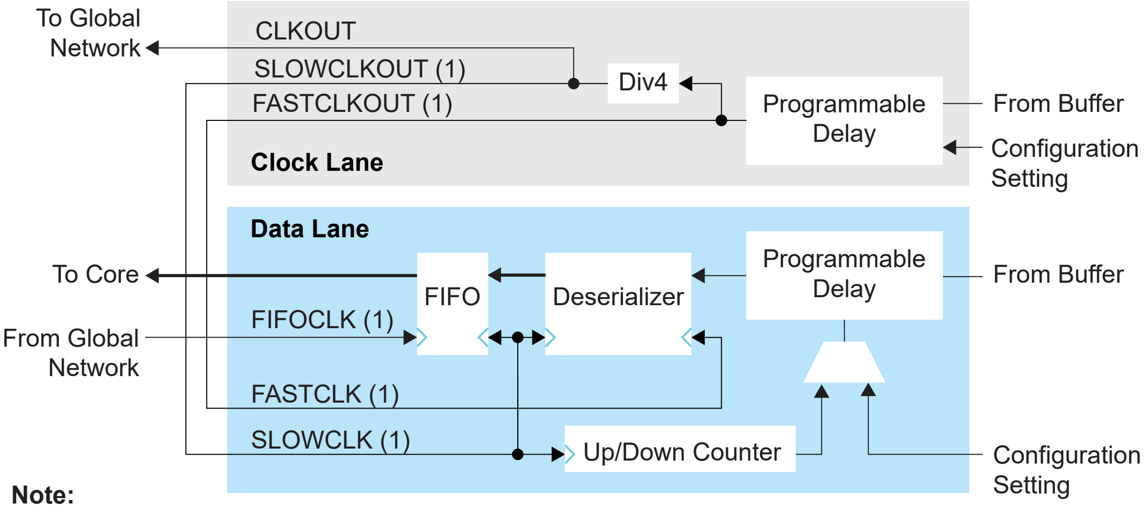
---

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

Signal	Direction	Clock Domain	Description
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.

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



Note:
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 38: MIPI TX Lane Block Diagram

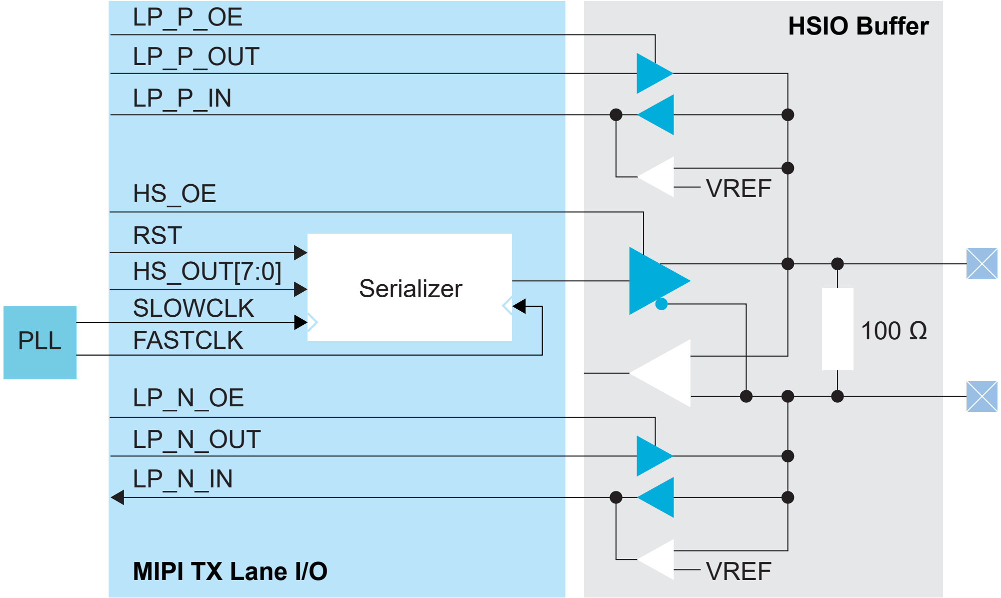


Table 23: MIPI TX Lane Signals

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 24: MIPI Lane Pads

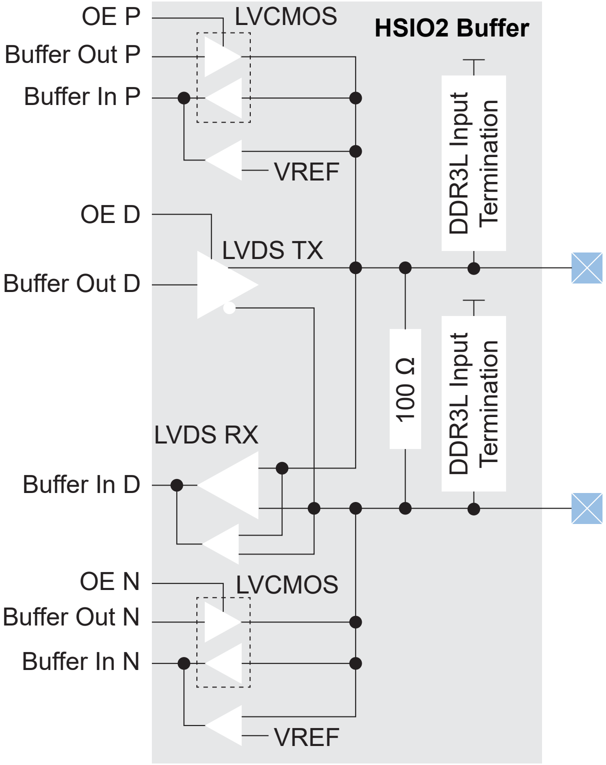
Signal	Direction	Description
P	Bidirectional	Differential pad P.
N	Bidirectional	Differential pad N.

HSIO2

Each HSIO2 block uses a pair of I/O pins as one of the following:

- *Single-ended HSIO2*—Two single-ended I/O pins (LVCMOS, SSTL, HSTL)
- *Differential HSIO2*—One pair of 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 39: HSIO2 Buffer Block Diagram



Important: When you are using an HSIO2 pin as a GPIO, make sure to leave at least one pair of unassigned HSIO2 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 a warning if you do not leave this separation.

HSIO2 Configured as GPIO

You can configure each HSIO2 block as two GPIO (single-ended) or one GPIO (differential). The function is the same as the HSIO pins. See HSIO Configured as GPIO on page 39.

HSIO2 Configured as LVDS

You can configure each HSIO2 block in RX, TX, or bidirectional LVDS mode. As LVDS, the HSIO2 has these features:

- Programmable V_{OD}, depending on the I/O standard used.
- Programmable pre-emphasis.
- Up to 1.8 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.
- Serialization widths:
  - *1 - 8 bits, 10 bits, and 16 bits*—Supports these widths when using a PLL for clocking.
  - *1 - 8 bits and 16 bits*—Supports these widths when using a PLL driving the PHY clock network for clocking.
- 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 25: 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 PLL output clocks feed the PHY clock network, which then goes to the clock modifiers in the byte clock groups. The slow clock runs at the data rate divided by the serialization factor. See Byte Clock Groups on page 27 for related diagrams.

LVDS RX

You can configure an HSIO2 block as one LVDS RX signal.

The LVDS RX fast clock and slow clocks can come from the clock modifier (as shown in the following figure) or from a PLL (as shown in Figure 31: LVDS RX Interface Block Diagram on page 43).

Figure 40: LVDS RX Interface Block Diagram

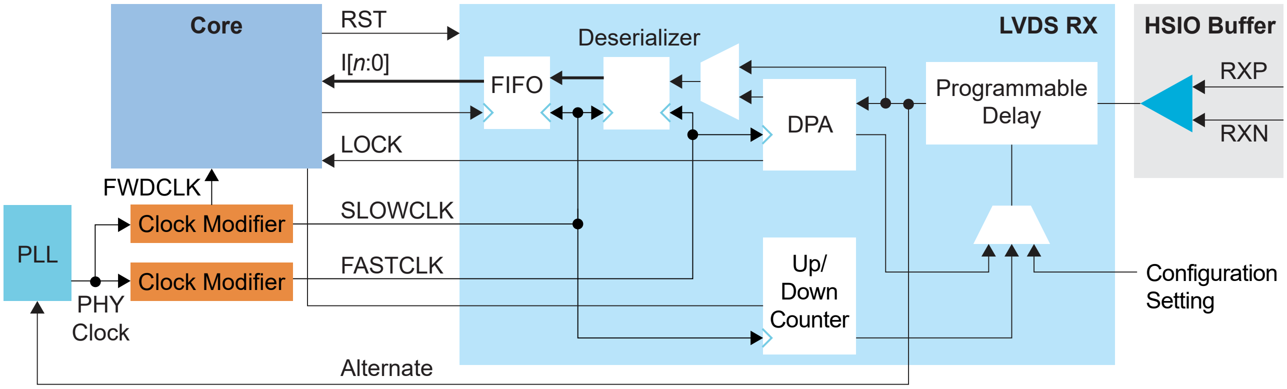


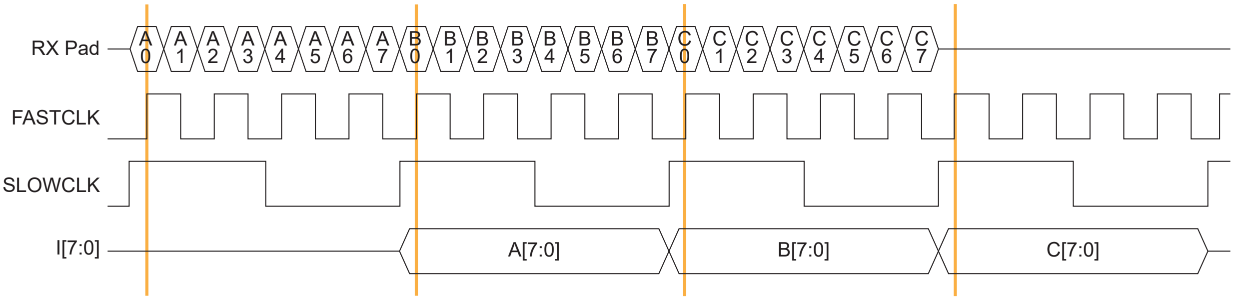
Table 26: LVDS RX Signals (Interface to FPGA Fabric)

Signal	Direction	Clock Domain	Description
I[15:0]	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.

Signal	Direction	Clock Domain	Description
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 41: 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 HSIO2 block as one LVDS TX signal. LVDS TX can be used in the serial data output mode or reference clock output mode.

The LVDS TX fast clock and slow clocks can come from the clock modifier (as shown in the following figure) or from a PLL (as shown in Figure 33: LVDS TX Interface Block Diagram on page 45).

Figure 42: LVDS TX Interface Block Diagram

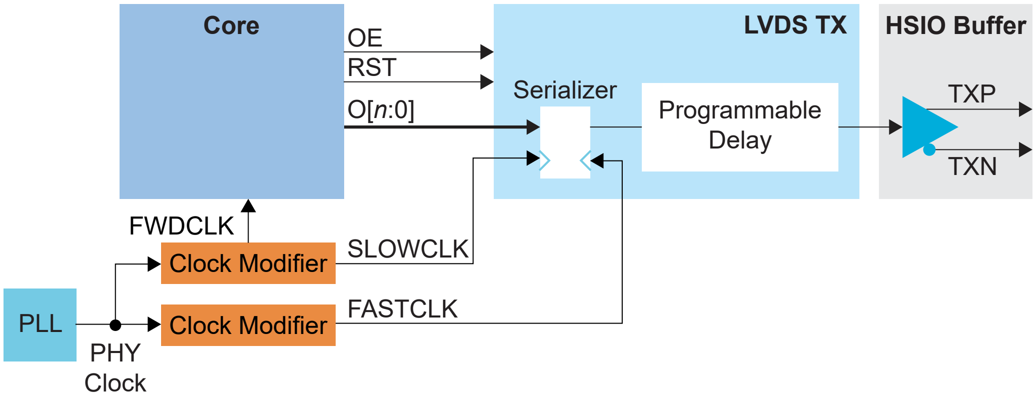
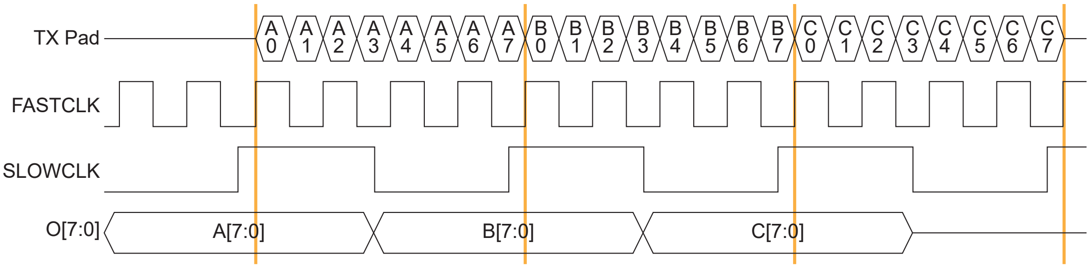


Table 27: LVDS TX Signals (Interface to FPGA Fabric)

Signal	Direction	Clock Domain	Description
O[15:0]	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 43: LVDS Timing Example Serialization Width of 8 (Half Rate)



OUT is byte-aligned data passed from the core on the rising edge of SLOWCLK.

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 HSIO2 block as one LVDS bidirectional signal. You must use the same serialization for the RX and TX.

The LVDS bidirectional fast clock and slow clocks can come from the clock modifier (as shown in the following figure) or from a PLL (as shown in Figure 35: LVDS Bidirectional Interface Block Diagram on page 47).

Figure 44: LVDS Bidirectional Interface Block Diagram

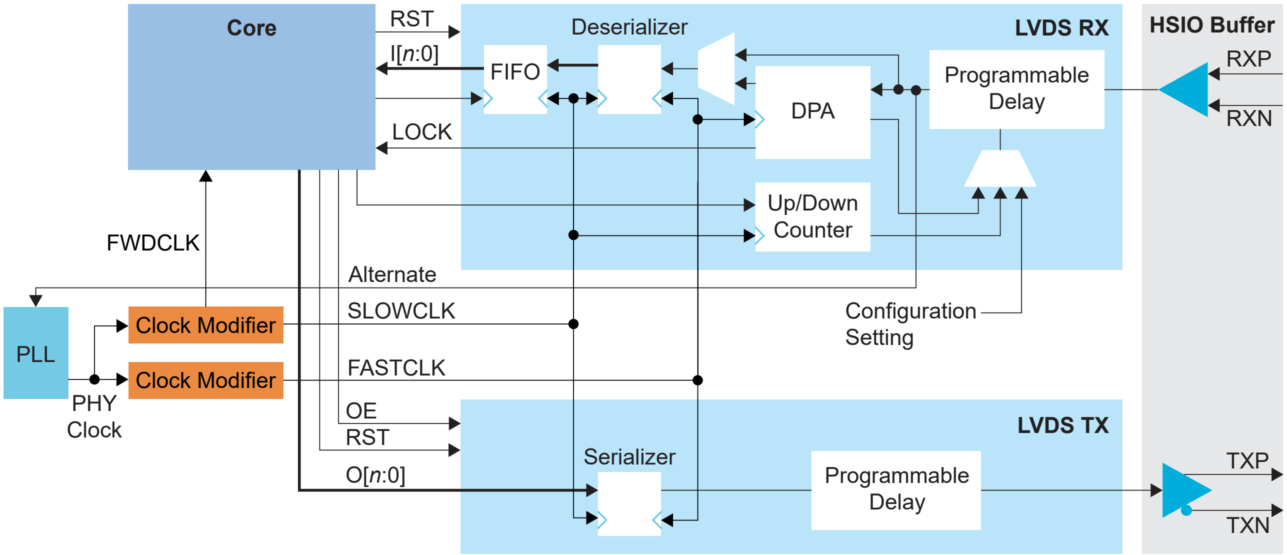


Table 28: LVDS Bidirectional Signals (Interface to FPGA Fabric)

Signal	Direction	Clock Domain	Description
I[15:0]	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

Signal	Direction	Clock Domain	Description
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 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[15:0]	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 29: LVDS Pads

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

HSIO2 Configured as MIPI Lane

You can configure the HSIO2 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 2.5 Gbps. The MIPI lane operates in high-speed (HS) and low-power (LP) modes. In HS mode, the HSIO2 block transmits or receives data with a 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. Dedicated HSIO2 blocks are assigned on the RX interface as clock lanes while the clock lane for TX interface can use any of the HSIO2 blocks 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 HSIO2 blocks to be used for the MIPI data lanes.

Each byte clock group has five or six HSIO2 blocks. Two blocks in each group can be MIPI RX clock lanes. If you do not use a MIPI RX clock lane as a clock, you can use it as a data lane instead. This arrangement supports many combinations, for example, a byte clock group with five HSIO2 blocks supports:

- One clock with four data lanes *or*
- One clock with one data lane and one clock with two data lanes *or*
- Two pairs of one clock with one data lane

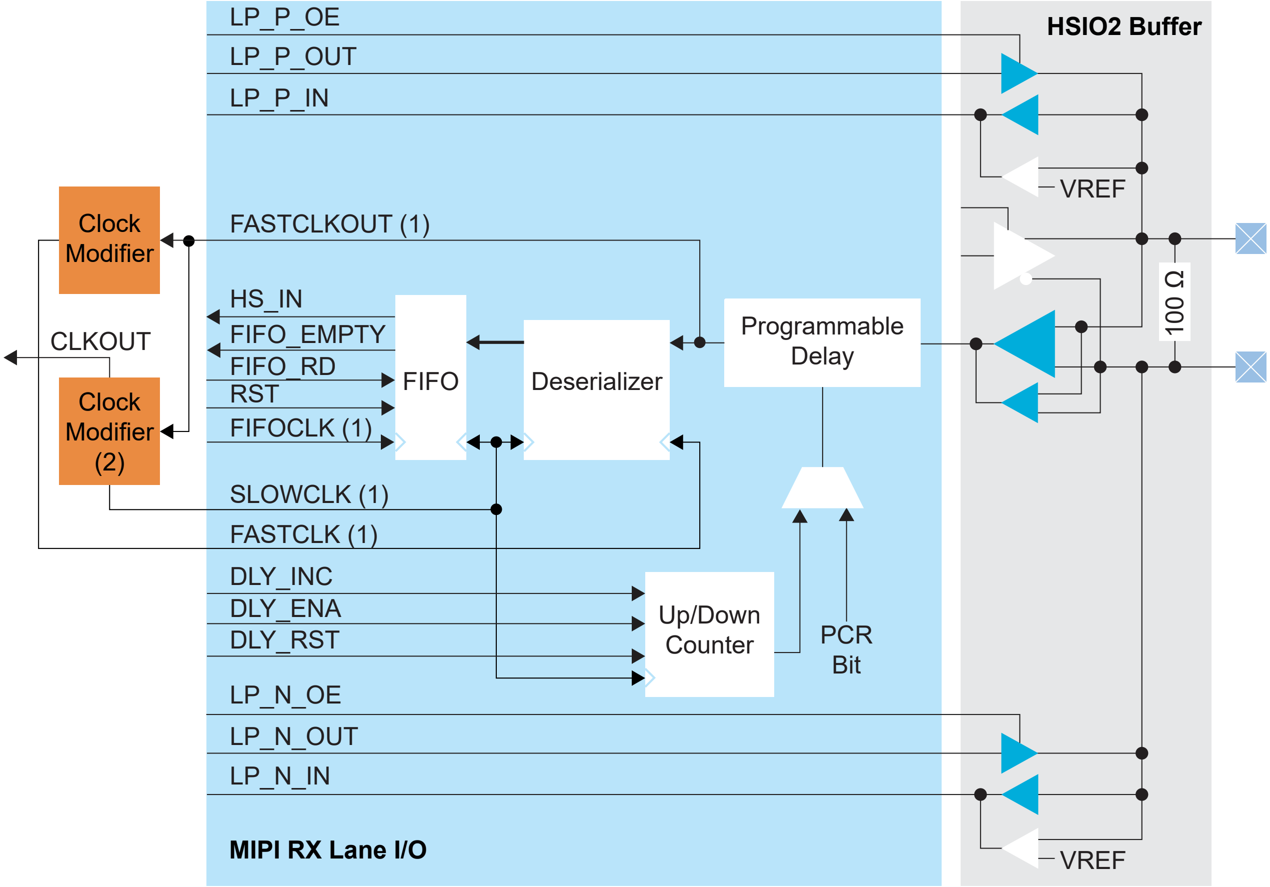
Additionally, one byte clock group on each side of the FPGA can be cascaded with the neighboring byte clock group to build an eight-lane interface. You only need to use one HSIO2 as the MIPI RX clock lane for both cascaded groups. See Cascaded MIPI Clock Lanes on page 65.

The MIPI RX function is defined as:

Table 30: MIPI RX Function

MIPI RX Function	Description
RX_DATA_*xy*_L_S*z* RX_DATA_*xy*_R_S*z*	MIPI RX Data Lane. *x* = P or N *y* = data lane number, 0 - 5 (Up to 4 data lanes in one byte clock group; up to 8 data lanes when cascading, see **Cascaded MIPI Clock Lanes** on page 65) *z* = MIPI byte clock group number, 0 - 4 (five byte clock groups on the left and right sides)
RX_CLK*n*_*x*_L_S*z* RX_CLK*n*_*x*_R_S*z*	MIPI RX Clock Lane. *n* is the clock number *x* = P or N *z* = MIPI byte clock group number, 0 - 4 (five byte clock groups the left and right sides)

Figure 45: MIPI RX Lane Block Diagram



Notes:
1. These signals are in the primitive, but the software automatically connects them for you.
2. The clock modifier performs a divide by 4 (serialization of 8) or by 8 (serialization of 16).

Table 31: MIPI RX Lane Signals

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

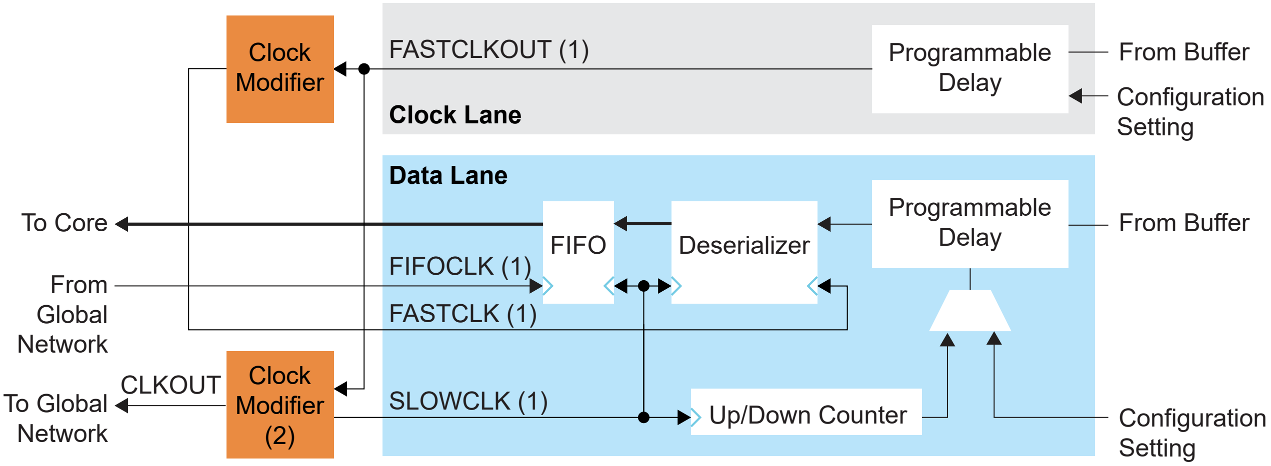
Signal	Direction	Clock Domain	Description
LP_P_OE	Input	-	(Optional) LP output enable signal for P pad.
LP_P_OUT	Input	-	(Optional) LP output data from the core for the P pad. Used if the data lane is reversible.
LP_P_IN	Output	-	LP input data from the P pad.
CLKOUT	Output	-	Divided down parallel (slow) 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(6)	Output	-	Divided down parallel (slow) clock from the pads. Can only drive RX DATA lanes.
FASTCLKOUT(6)	Output	-	Serial (fast) clock from the pads. Can only drive RX DATA lanes.
HS_IN[15:0]	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(6)	Input	-	(Optional) Core clock to read from the FIFO.
SLOWCLK(6)	Input	-	Parallel (slow) clock.
FASTCLK(6)	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.

---

(6) 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 or 8 that feeds the global network. The following figure shows the clock connections between the clock and data lanes.

Figure 46: Connections for Clock and RX Data Lane in the Same MIPI RX Channel



Note:

1. The software automatically connects this signal for you.
2. The clock modifier performs a divide by 4 (serialization of 8) or by 8 (serialization of 16).

Cascaded MIPI Clock Lanes

The HSIO2 block supports clock cascading. You can cascade the clock lane from one byte clock group to another byte clock group, which allows you to have eight MIPI RX data lanes (x8) and one (cascaded) MIPI RX clock lane. The HSIO2 supports cascading for the following groups:

Table 32: MIPI Clock Cascading

MIPI RX Clock Lane	Group	Cascaded Group
RX_CLK0_P_R_S2, RX_CLK0_N_R_S2	R_S2	R_S1
RX_CLK0_P_L_S2, RX_CLK0_N_L_S2	L_S2	L_S1

Download: Refer to the Ti125 Pinout for information on the MIPI byte clock groups.

MIPI TX Lane

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

Figure 47: MIPI TX Lane Block Diagram

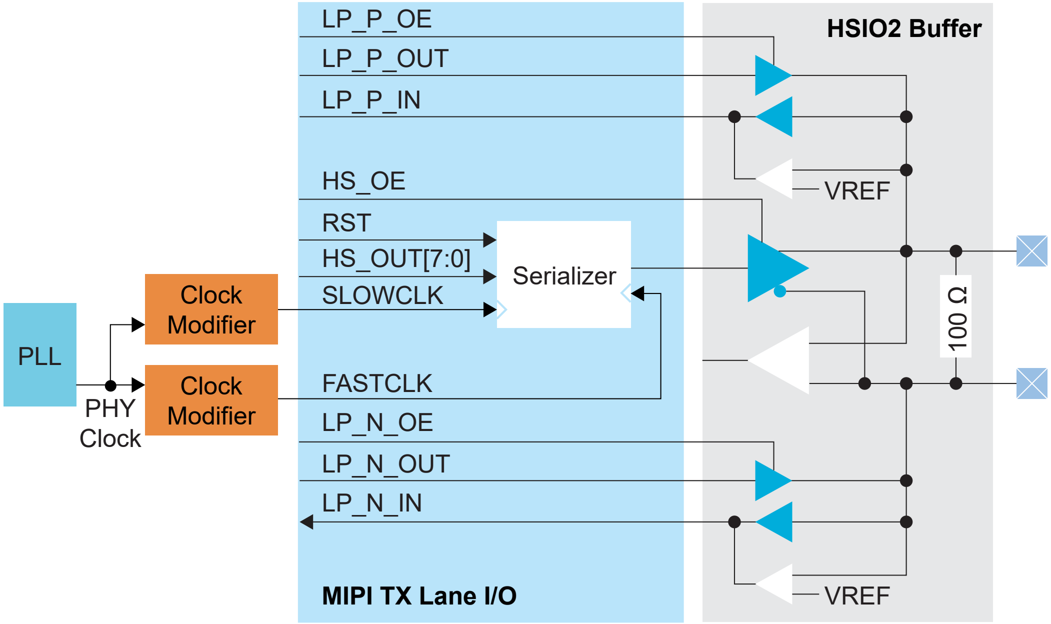


Table 33: MIPI TX Lane Signals

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.

Signal	Direction	Clock Domain	Description
RST	Input	SLOWCLK	(Optional) Resets the serializer.
HS_OUT[15:0]	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 34: MIPI Lane Pads

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

HSIO2 Configured as QDRIO

To create a DDR PHY interface, you use the HSIO2 blocks in quad data rate I/O (QDRIO) mode with a soft-core DDR controller implemented in your RTL design. In this mode, the HSIO2 blocks support quad data rate I/O with a dynamic programmable delay chain and on-die 60-Ω input termination for DDR3L. In the Interface Designer, you add a DDR PHY block that instantiates the clock modifiers, DQS shifter, and the QDRIO blocks for you. It supports x8 or x16 data widths and a PHY rate up to 1,333 Mbps.

The soft-core DDR controller sends control signals to the DQS shifter in the byte clock group, which sends the gated `DQS` signal to the HSIO2 blocks. The `FWDCLK` from the clock modifier feeds a clock in the global clock network, which in turn drives the slow clock to the HSIO2 (PHY Div2).

Note: A x8 DDR interface has eight single-ended `DQ` pins and one pair of differential `DQS` pins; all pins must be in the same byte clock group. Some packages may not have enough pins bonded out in a given byte clock group to implement the interface.

Refer to Table 2: Ti125 Package-Dependent Resources on page 5 for the number of interfaces and data widths supported by package.

Refer to the Ti125 Pinout for the byte clock group(s) that can fit all eight single-ended DQ pins and one pair of differential DQS pin in the same byte clock group.

Figure 48: DDR Read Clocking

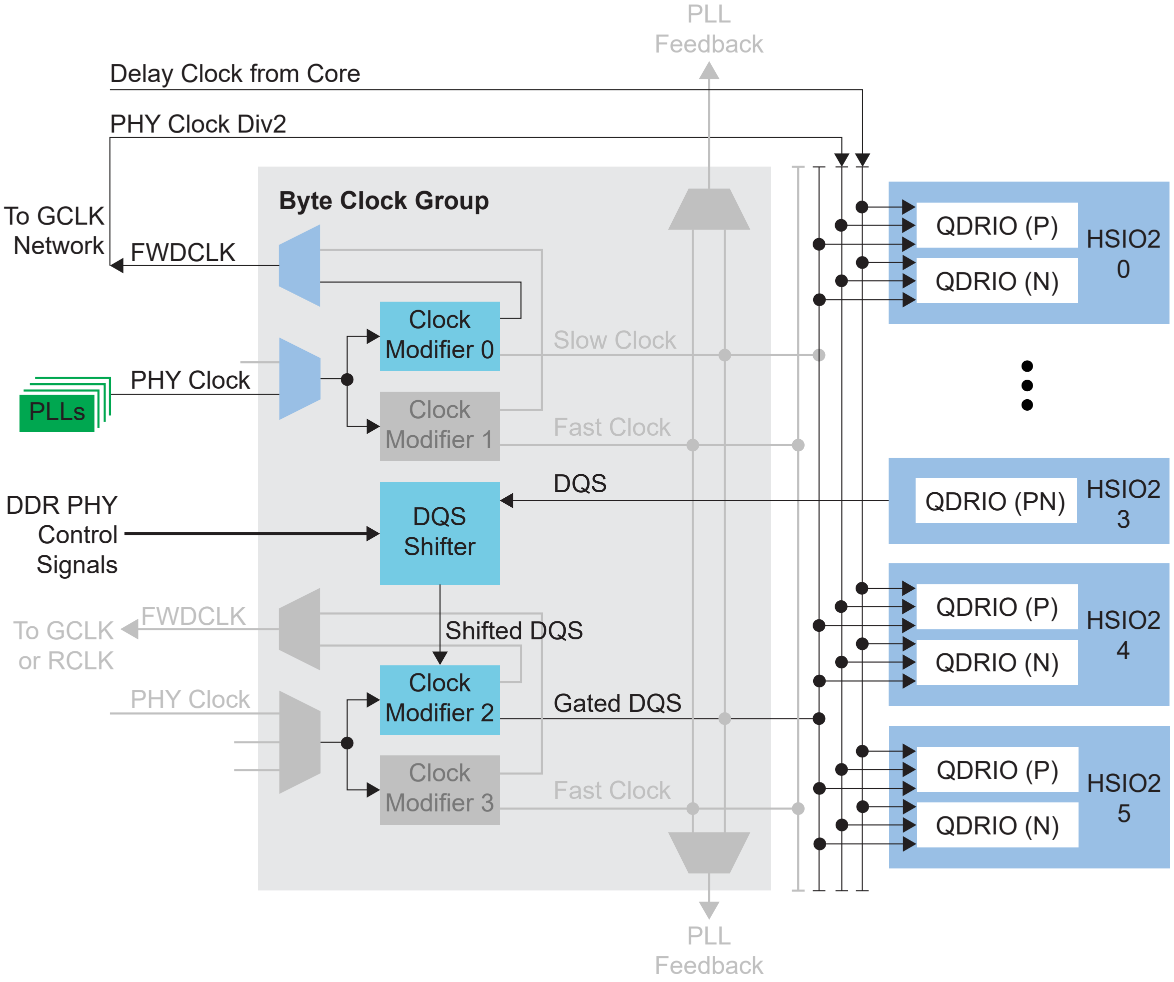


Figure 49: DDR Write Clocking

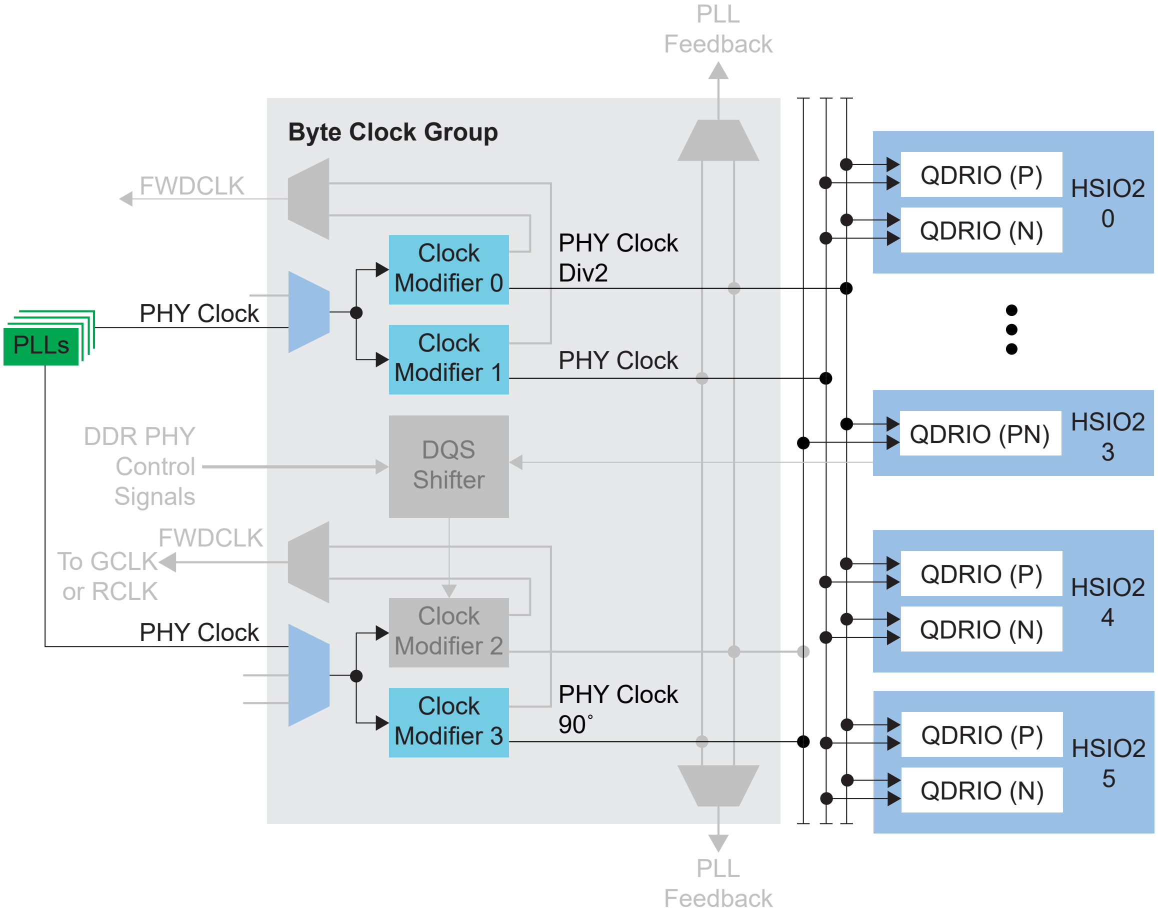


Table 35: QDRIO Modes

GPIO Mode	Description
Output	Only the output path is enabled; always registered. Four registers capture the data on the positive edge of the slow output clock, multiplexing them into one data stream clocked out both edges of the fast output clock.
Bidirectional	The input, output, and OE paths are enabled; always registered. Four registers capture the data on the positive edge of the slow output clock, multiplexing them into one data stream clocked out both edges of the fast output clock. On the input side, a FIFO captures data on both edges of the input fast clock and reads four bits of data on the positive edge of the input slow clock.

Table 36: QDRIO Signals (Interface to FPGA Fabric)

Port	Direction	Clock Domain	Notes
OUTFASTCLK	Input	N/A	Byte clock signal. This signal is the TX PHY clock running at half the data rate, and it comes from the clock modifier in the byte clock group. If this signal is a DQ or DM bit, the phase shift is 0. If it is the DQS bit, the phase shift is 90°.

Port	Direction	Clock Domain	Notes
OUTSLOWCLK	Input	N/A	Byte clock signal. This signal is the TX PHY clock Div2 running at one quarter the data rate. It comes from the clock modifier in the byte clock group.
INFASTCLK	Input	N/A	Byte clock signal. This signal is the RX PHY clock running at half the data rate. It comes from the DQS shifter. Only available in bidirectional mode.
INSLOWCLK	Input	N/A	This signal is the RX PHY clock Div2 running at one quarter the data rate. It comes from the core, and is the forwarded clock (FWDCLK) from the clock modifier. Only available in bidirectional mode.
DLYCLK	Input	N/A	Delay clock signal from the global or regional clock network. Only available in bidirectional mode.
O[3:0]	Input	OUTSLOWCLK	DQ data output synchronous to to the PHY clock Div2.
OE[1:0]	Input	OUTSLOWCLK	DQ data output enable synchronous to PHY clock Div2. Only available in bidirectional mode.
I[3:0]	Output	INSLOWCLK	DQ data input synchronous to the PHY clock Div2 forwarded to the core. Only available in bidirectional mode.
FIFO_RSTN	Input	N/A	Reset for the read FIFO. Only available in bidirectional mode.
FIFO_EMPTY	Output	INSLOWCLK	Read FIFO empty. Only available in bidirectional mode.
FIFO_REN	Input	INSLOWCLK	FIFO read enable. The DQS shifter drives this signal. Only available in bidirectional mode.
FIFO_WEN	Input	INFASTCLK	FIFO write enable. The DQS shifter drives this signal. Only available in bidirectional mode.
TERM	Input	N/A	Enable dynamic input termination. The DQS shifter drives this signal. Only available in bidirectional mode.
DLY_INC	Input	DLYCLK	Dynamic delay control. 1: Increment 0: Decrement when DLY_ENA = 1. Only available in bidirectional mode.
DLY_ENA	Input	DLYCLK	Enable the dynamic delay control. Only available in bidirectional mode.
DLY_RST	Input	DLYCLK	Reset the delay counter. Each HSIO2 block has one reset. Only available in bidirectional mode.
DBG[5:0]	Output	DLYCLK	Debug pin.
ALT	Output	DLYCLK	DQS_IN clock signal that drives the DQS shifter's DQS_IN pin.

Table 37: QDRIO Pads

Signal	Direction	Description
IO	Bidirectional	DDR pad. TX traffic outputs on OUTFASTCLK; RX traffic inputs on INFASTCLK.

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 38: I/O Banks by Package

Package	I/O Banks	Voltage (V)	Dynamic Voltage Support	DDIO Support	Merged Banks
F225	1A, 1B, 2A, 2B, 3A, 3B, 4A, 4B	1.2, 1.35, 1.5, 1.8	-	All	-
	BL0, BR0, TL0, TR0	1.8, 2.5, 3.0, 3.3	✓	All	-
M225S4F4	1A, 1B, 3A, 3B, 4A, 4B	1.2, 1.35, 1.5, 1.8	-	All	-
	BL0, BR0, TL0, TR0	1.8, 2.5, 3.0, 3.3	✓	All	-

Learn more: Refer to the Ti125 Pinout for information on the I/O bank assignments.

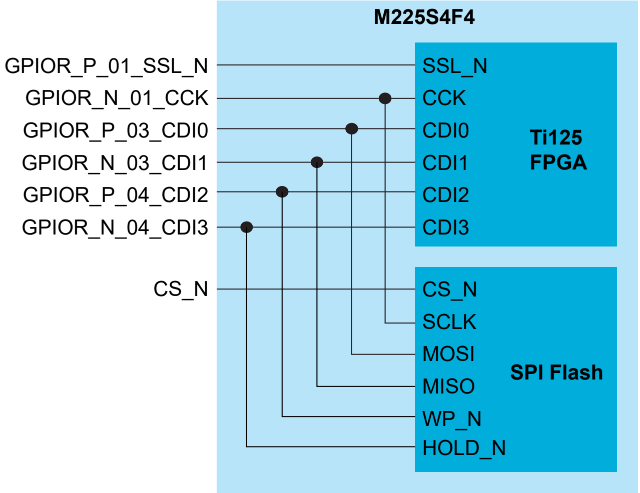
SPI Flash Memory

Ti125 FPGAs in the M225S4F4 package include a SPI flash memory. The SPI flash memory has a density of 256 Mbits and a clock rate of up to 166 MHz in user mode. You can fit four compressed bitstream images into the SPI flash.

Important: You cannot enable the Ti125 FPGA security features when using compressed bitstreams.

In SPI active configuration mode, the FPGA is configured using a bitstream stored in the SPI flash device. During configuration, the maximum clock frequency for the flash device is specified in SPI Active Mode on page 106. When the FPGA is in user mode, you can access the flash at the flash device's maximum clock frequency (although different SPI flash commands may have different maximum clock frequencies).

Figure 50: Connections between FPGA and SPI Flash inside the Package



Important: The SPI flash memory's VCC is connected to `VCCIO_3B`. If you are using the SPI flash memory, drive the `VCCIO_3B` with a 1.8 V supply.

Table 39: SPI Flash Memory Signals (Interface to FPGA Fabric)

SPI Name	Signal	Direction	Description
SCLK	SCLK_OUT	Input	Clock output from FPGA CCK pin to SPI flash memory.
	SCLK_OE	Input	Output enable. Required for multiple controller.
MOSI	MOSI_IN	Output	Required for ×2 or ×4 data width.
	MOSI_OUT	Input	Data output from FPGA CDI0 to SPI flash memory.
	MOSI_OE	Input	Output enable. Required for ×2 data width, ×4 data width, or multiple controller.
MISO	MISO_IN	Output	Data input to FPGA CDI1 from SPI flash memory.
	MISO_OUT	Input	Required for ×2 or ×4 data width.
	MISO_OE	Input	Output enable. Required for ×2 or ×4 data width.
WP_N	WP_N_IN	Output	Required for ×4 data width.
	WP_N_OUT	Input	Data output from FPGA CDI2 pin to SPI flash memory.

SPI Name	Signal	Direction	Description
	WP_N_OE	Input	Output enable. Required for ×4 data width or multiple controller.
HOLD_N	HOLD_N_IN	Output	Required for ×4 data width.
	HOLD_N_OUT	Input	Data output from FPGA CDI3 pin to SPI flash memory
	HOLD_N_OE	Input	Output enable. Required for ×4 data width or multiple controller.
CS_N	CS_N_OUT	Input	Chip select output from FPGA SSL_N pin to SPI flash memory.
	CS_N_OE	Input	Output enable. Required for multiple controller.
CLK	CLK	Input	Required for register interface.

To program the SPI flash memory, use one of these modes:

- SPI Active using JTAG Bridge mode
- SPI Active mode

Learn more: Refer to the AN 033: Configuring Titanium FPGAs for information on programming the SPI flash memory.

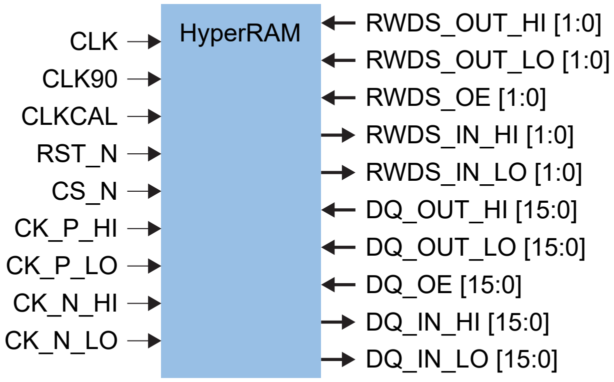
The GPIOR_P_01 (`SSL_N`), GPIOR_N_01 (`CCK`), GPIOR_P_03 (`CDI0`), and GPIOR_N_03 (`CDI1`) resources are for the SPI active interface. You can use these signals to read/write user data to/from the SPI flash memory while the Ti125 M225S4F4 FPGA is in user mode. You enable this feature by adding the SPI flash block to your interface design. These resources are not available as user I/O pins if you use the SPI flash block.

You can also write a new bitstream to the SPI flash memory by controlling the SPI signals with an external controller. In this case, the `CRESET_N` signal should stay low and the FPGA remains in reset mode, even though you stored a new bitstream in the SPI flash memory. To enable this mode, turn on **Configuration > External Flash Control > Enable external controller acccess to flash memory** in the Interface Designer.

HyperRAM Interface

The Ti125 FPGA in M225S4F4 package includes two HyperRAM devices. Each HyperRAM device has a density of 256 Mbits and a clock rate of up to 250 MHz. The HyperRAM supports double-data rates of up to 500 Mbps and supports a 16-bit data bus.

Figure 51: HyperRAM Block Diagram



Important: HyperRAM 0 (HYPER_RAM0 in the Interface Designer): Drive `VCCIO_2A` with a 1.8-V supply. HyperRAM 1 (HYPER_RAM1 in the Interface Designer): Drive `VCCIO_2B` with a 1.8-V supply.

Table 40: HyperRAM Signals (Interface to FPGA Fabric)

Signal	Direction	Description
CLK	Input	HyperRAM controller clock.
CLK90	Input	90 degree phase-shifted version of CLK.
CLKCAL	Input	Calibration clock for input data.
RST_N	Input	Active-low HyperRAM reset.
CS_N	Input	Active-low HyperRAM chip select signal.
CK_P_HI	Input	The clock provided to the HyperRAM. The clock is not required to be free-running. Registered in normal mode of DDIO.
CK_P_LO	Input	
CK_N_HI	Input	
CK_N_LO	Input	
RWDS_OUT_HI [1:0]	Input	Read/write data strobe input ports for data mask during write operation. Registered in normal mode/resync mode of DDIO.
RWDS_OUT_LO [1:0]	Input	
RWDS_OE [1:0]	Input	Read/write data strobe output enable port.
RWDS_IN_HI [1:0]	Output	Read/write data strobe output ports for latency indication, also center-aligned reference strobe for read data. Registered in normal mode/resync mode of DDIO.
RWDS_IN_LO [1:0]	Output	
DQ_OUT_HI [15:0]	Input	DQ input ports for command, address and data. Registered in normal mode of DDIO.
DQ_OUT_LO [15:0]	Input	
DQ_OE [15:0]	Input	DQ output enable port.
DQ_IN_HI [15:0]	Output	DQ output ports for data.
DQ_IN_LO [15:0]	Output	

Oscillator

The Ti125 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. Its 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

Ti125 FPGAs have seven 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.)

Ti125 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.

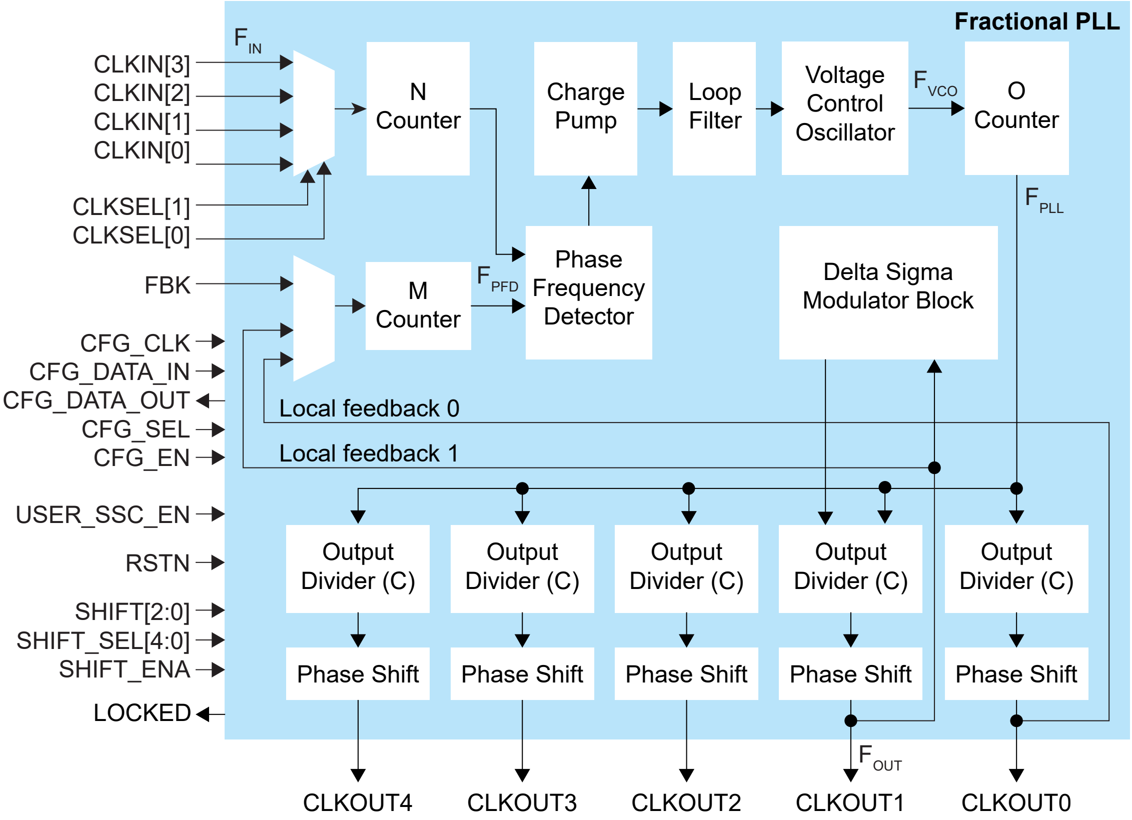
At startup, Efinix recommends that you hold the PLL in reset until the PLL's reference clock source is stable.

Note: You can cascade the PLLs in Ti125 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 52: Fractional 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 102 for F_{VCO}, F_{OUT}. F_{IN}, F_{PLL}, and F_{PFD} values.

Figure 53: PLL Interface Block Diagram

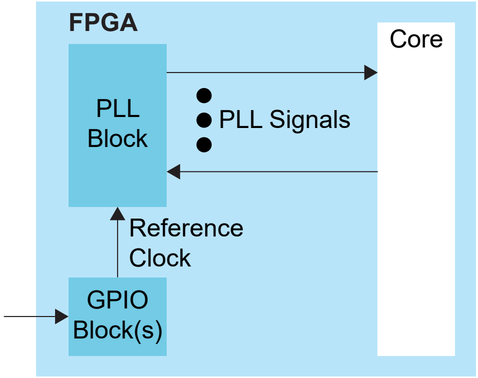


Table 41: Fractional PLL Signals (Interface to FPGA Fabric)

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.
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.
CFG_EN	Input	Enable for dynamic configuration.
USER_SSC_EN	Input	User spread-spectrum clocking enable pin name.

Reference Clock Resource Assignments

Table 42: PLL Reference Clock Resource Assignments (F225)

PLL	REFCLK0	REFCLK1	REFCLK2	External Feedback I/O
BL0	Differential: GPIOL_P_00_PLLIN0, GPIOL_N_00_CDI22 Single-ended: GPIOL_P_00_PLLIN0	Differential: GPIOB_P_00_PLLIN1, GPIOB_N_00 Single-ended: GPIOB_P_00_PLLIN1	Single-ended: GPIOL_00_PLLIN2	Differential: GPIOL_P_01_EXTFB, GPIOL_N_01_CDI23 Single-ended: GPIOL_P_01_EXTFB
BL1	Differential: GPIOL_P_03_CDI24_PLLIN0, GPIOL_N_03_CDI25 Single-ended: GPIOL_P_03_CDI24_PLLIN0	–	Single-ended: GPIOL_01_PLLIN2	Differential: GPIOL_P_02_CDI26_EXTFB, GPIOL_N_02_CDI27 Single-ended: GPIOL_P_02_CDI26_EXTFB
BR0	Differential: GPIOR_P_00_PLLIN0, GPIOR_N_00 Single-ended: GPIOR_P_00_PLLIN0	Differential: GPIOB_P_23_PLLIN1, GPIOB_N_23 Single-ended: GPIOB_P_23_PLLIN1	–	Differential: GPIOB_P_22_EXTFB, GPIOB_N_22 Single-ended: GPIOB_P_22_EXTFB
TL0	Differential: GPIOL_P_27_PLLIN0, GPIOL_N_27 Single-ended: GPIOL_P_27_PLLIN0	Differential: GPIOT_P_01_PLLIN1, GPIOT_N_01 Single-ended: GPIOT_P_01_PLLIN1	–	Differential: GPIOT_P_02_EXTFB, GPIOT_N_02 Single-ended: GPIOT_P_02_EXTFB
TL1	Differential: GPIOL_P_25_PLLIN0, GPIOL_N_25 Single-ended: GPIOL_P_25_PLLIN0	–	Single-ended: GPIOL_18_PLLIN2	Differential: GPIOT_P_03_EXTFB, GPIOT_N_03 Single-ended: GPIOT_P_03_EXTFB
TR0	Differential: GPIOR_P_26_PLLIN0, GPIOR_N_26 Single-ended: GPIOR_P_26_PLLIN0	Differential: GPIOT_P_22_PLLIN1, GPIOT_N_22 Single-ended: GPIOT_P_22_PLLIN1	Single-ended: GPIOR_20_PLLIN2	Differential: GPIOR_P_25_EXTFB, GPIOR_N_25 Single-ended: GPIOR_P_25_EXTFB
TR1	Differential: GPIOR_P_24_PLLIN0, GPIOR_N_24 Single-ended: GPIOR_P_24_PLLIN0	Differential: GPIOT_P_21_PLLIN1, GPIOT_N_21 Single-ended: GPIOT_P_21_PLLIN1	Single-ended: GPIOR_21_PLLIN2	–

Table 43: PLL Reference Clock Resource Assignments (M225S4F4)

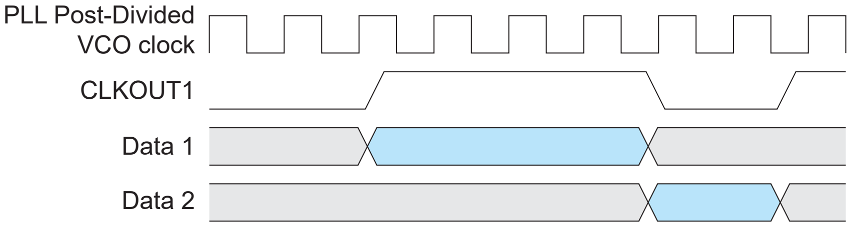
PLL	REFCLK0	REFCLK1	REFCLK2	External Feedback I/O
BL0	Differential: GPIOL_P_00_PLLIN0, GPIOL_N_00_CDI22 Single-ended: GPIOL_P_00_PLLIN0	Differential: GPIOB_P_00_PLLIN1, GPIOB_N_00 Single-ended: GPIOB_P_00_PLLIN1	Single-ended: GPIOL_00_PLLIN2	Differential: GPIOL_P_01_EXTFB, GPIOL_N_01_CDI23 Single-ended: GPIOL_P_01_EXTFB
BL1	Differential: GPIOL_P_03_CDI24_PLLIN0, GPIOL_N_03_CDI25 Single-ended: GPIOL_P_03_CDI24_PLLIN0	–	Single-ended: GPIOL_01_PLLIN2	Differential: GPIOL_P_02_CDI26_EXTFB, GPIOL_N_02_CDI27 Single-ended: GPIOL_P_02_CDI26_EXTFB
BR0	Differential: GPIOR_P_00_PLLIN0, GPIOR_N_00 Single-ended: GPIOR_P_00_PLLIN0	Differential: GPIOB_P_23_PLLIN1, GPIOB_N_23 Single-ended: GPIOB_P_23_PLLIN1	–	Differential: GPIOB_P_22_EXTFB, GPIOB_N_22 Single-ended: GPIOB_P_22_EXTFB
TL0	Differential: GPIOL_P_27_PLLIN0, GPIOL_N_27 Single-ended: GPIOL_P_27_PLLIN0	–	–	–
TL1	Differential: GPIOL_P_25_PLLIN0, GPIOL_N_25 Single-ended: GPIOL_P_25_PLLIN0	–	Single-ended: GPIOL_18_PLLIN2	–
TR0	Differential: GPIOR_P_26_PLLIN0, GPIOR_N_26 Single-ended: GPIOR_P_26_PLLIN0	–	Single-ended: GPIOR_20_PLLIN2	Differential: GPIOR_P_25_EXTFB, GPIOR_N_25 Single-ended: GPIOR_P_25_EXTFB
TR1	Differential: GPIOR_P_24_PLLIN0, GPIOR_N_24 Single-ended: GPIOR_P_24_PLLIN0	–	Single-ended: GPIOR_21_PLLIN2	–

Programmable Duty Cycle

Ti125 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 54: Programmable Duty Cycle Example on page 80).

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

Figure 54: Programmable Duty Cycle Example



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

Fractional Output Divider

Ti125 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.

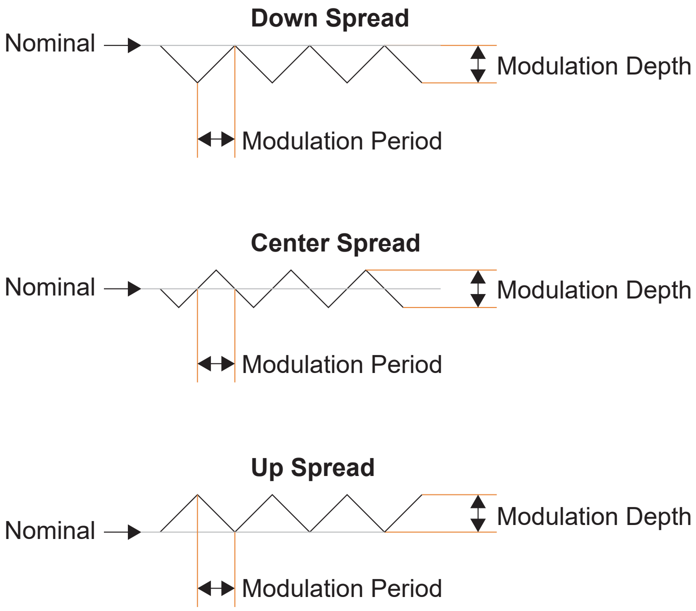
To use this feature you choose local feedback mode in the Interface Designer and specify the fractional options. The PLL feeds `CLKOUT1` back into the M counter, which causes the fractional part to propagate to all of the clock output signals.

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

Spread-Spectrum Clocking

Ti125 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 55: Supported Modulation Types



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

Dynamic PLL Reconfiguration

Ti125 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

Ti125 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 one 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 four clock cycles of the targeted output clock.
4. De-assert `SHIFT_ENA` and `SHIFT_SEL[n]`. Wait for at least four 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 56: Single Dynamic Phase Shift Waveform Example for CLKOUT1

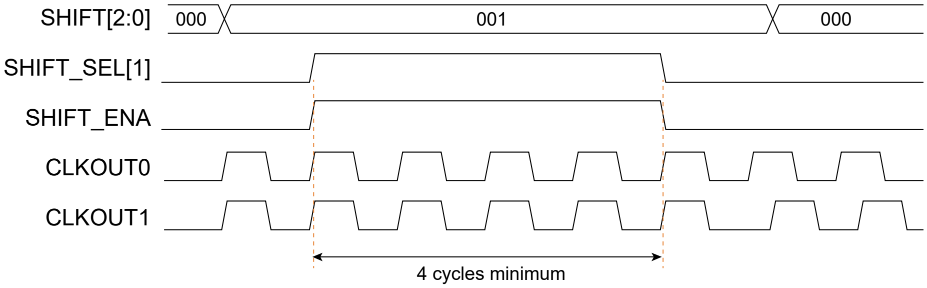
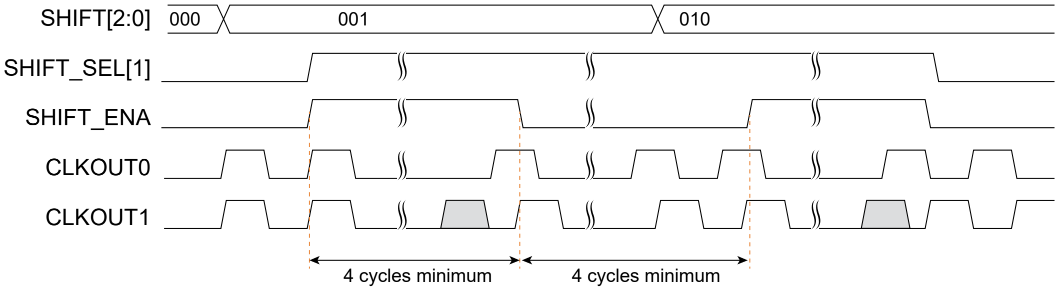


Figure 57: Consecutive Dynamic Phase Shift Waveform Example for CLKOUT1



Single-Event Upset Detection

The Ti125 FPGA has a hard block for detecting single-event upset (SEU). The SEU detection feature has two modes:

- *Auto mode*—The Ti125 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 Ti125 when an error is detected.

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

Important: For applications that require the SEU feature, you must implement an external system-level primary functional detection mechanism. You should only use the Titanium™ FPGA SEU detection feature as a secondary indicator and it should not take precedence over the primary detection mechanism.

Note: Contact your local Efinix representative for more system-level design details if your application needs both Over-the-Air and SEU detection.

Internal Reconfiguration Block

The Ti125 FPGAs have built-in hardware that supports an internal reconfiguration feature. The Ti125 can reconfigure itself from a bitstream image stored in flash memory.

Note: Refer to **AN 010: Using the Internal Reconfiguration Feature to Update Efinix FPGAs Remotely** for details regarding reconfiguration.

Security Feature

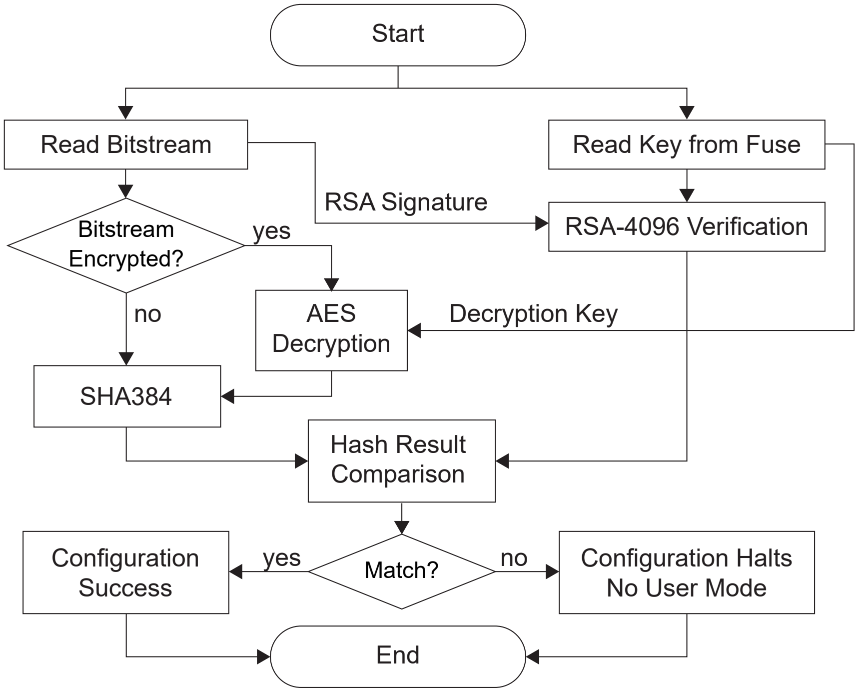
The FPGA security feature includes:

- 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 58: Security Flow



Download: Refer to the "Securing Titanium Bitstreams" section of the "Configuring an FPGA" chapter in the **Efinity Software User Guide** for instructions on how to enable these features.

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

Ti125 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.

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.

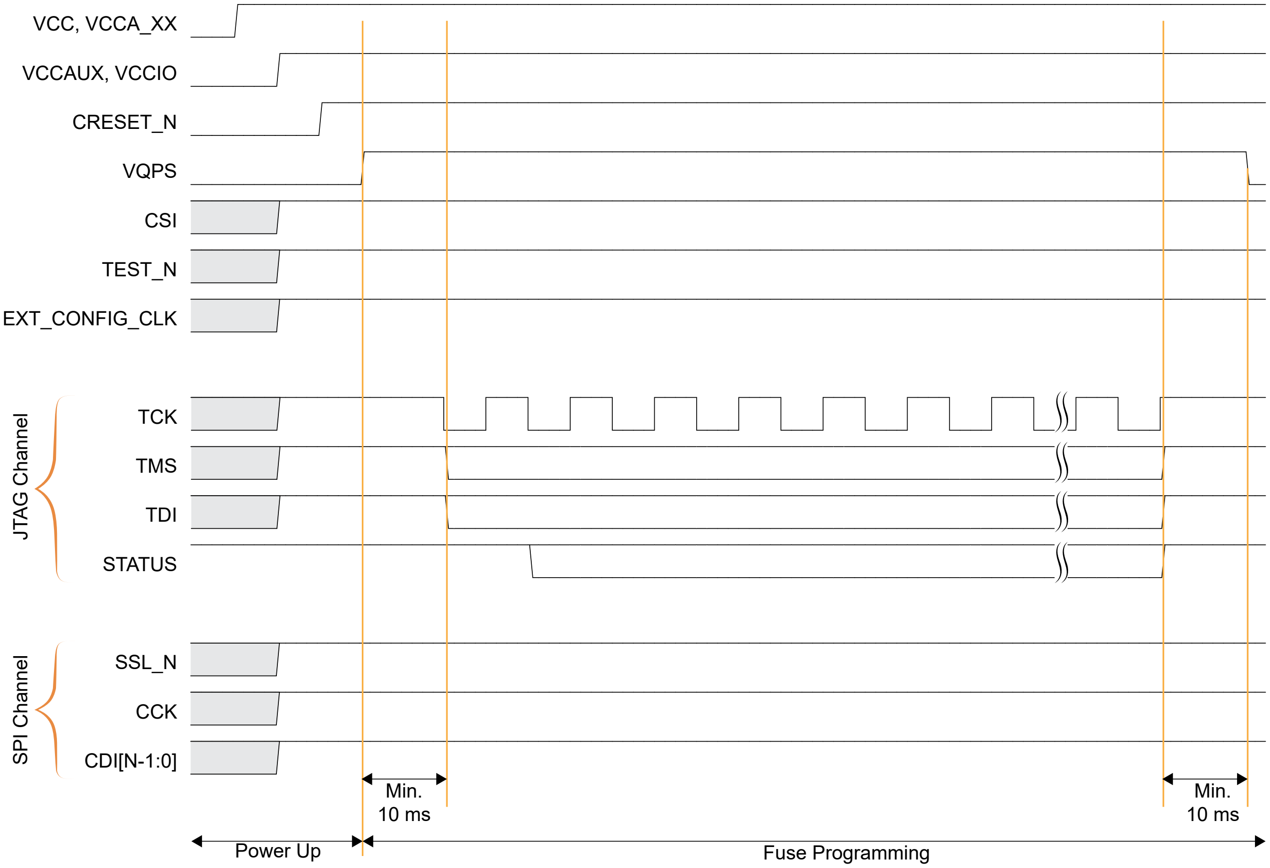
Fuse Programming Requirements

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 51: 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 59: Fuse Programming Waveform



This waveform assumes you are using an SVF file generated with the Efinity Bitstream Security Key Generator.

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

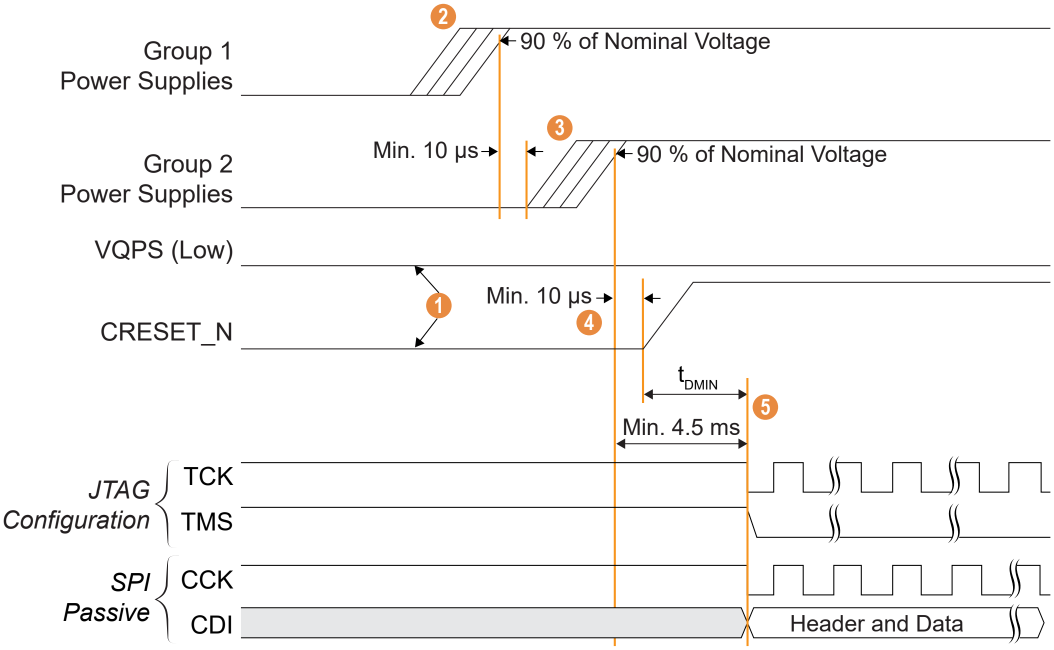
Learn more: Refer to the "Securing Bitstreams" section in the Efinity Programmer User Guide.
Refer to AN 057: Controlling VQPS with the Efinity SVF Player for more details about fuse programming.

Power Sequence

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

Power-Up Sequence

Figure 60: 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 Ti125 security fuses.

   Note: Refer to Fuse Programming Requirements on page 86 if you need to blow the security fuses for the Ti125 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 51: 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. Release the `CRESET_N` input to high at a minimum delay of 10 μs after all FPGA supplies have reached 90% of their nominal voltage levels.
5. 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 107 and **JTAG Mode** on page 105 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 44: Power-Up Groups

If you are blowing the security fuses, refer to **Fuse Programming Requirements** on page 86.

Power-Up Sequence	
Group 1	**Group 2**
VCC	VCCAUX
VCCA	VCCIO
	VCCIO33

Power-Down Sequence

There is no specific power-down sequence for Ti125 FPGAs. However, the `VQPS` power supply **must** follow the specifications in **Fuse Programming Requirements** on page 86.

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 45: Minimum Power Supply Current Transient

Power Supply	Minimum Power Supply Current Transient	Unit
VCC	1,500(7)	mA

Unused Resources and Features

Table 46: 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.
HSIO2 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.
Security (Fuse Blowing)	VQPS	Connect to GND.
Internal HyperRAM (M225S4F4 only)	VCCIO2A (HyperRAM 0), VCCIO2B (HyperRAM 1)	Connect to 1.8 V.

Learn more: For more information, refer to the Titanium Hardware Design Checklist and Guidelines page of the Support Center.

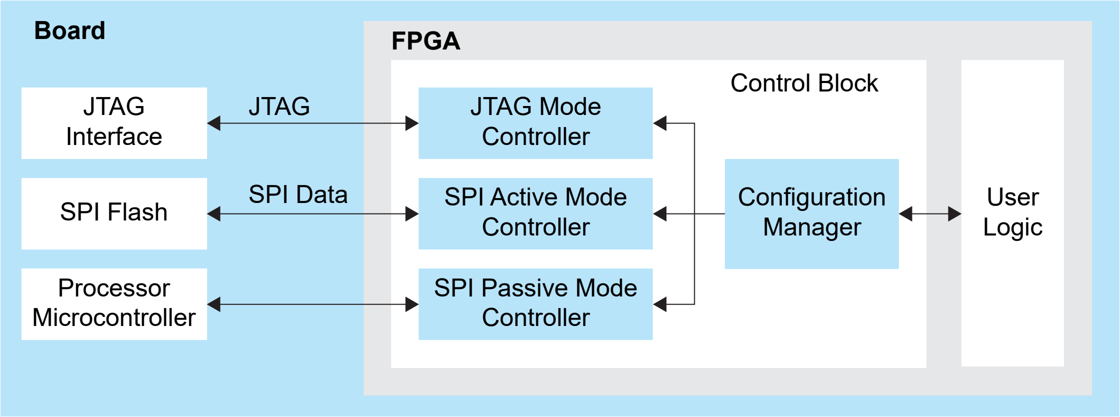
---

(7) Preliminary

Configuration

The Ti125 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 Ti125 FPGA(s) in SPI active, SPI passive, or JTAG mode.

Figure 61: 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.

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.

Supported Configuration Modes

Table 47: Ti125 Configuration Modes by Package

Configuration Mode	Width	All Packages
Active	×1	✓
	×2	✓
	×4	✓
	×8	✓
Passive	×1	✓
	×2	✓
	×4	✓
	×8	✓
	×16	✓(8)
	×32	✓(8)
JTAG	×1	✓

Learn more: Refer to AN 033: Configuring Titanium FPGAs for more information.

---

(8) Not supported when security mode is enabled.

Characteristics and Timing

The following table shows the specification status for Ti125 packages.

Table 48: Package Status

Package	Status
F225, M225S4F4	Preliminary

DC and Switching Characteristics

Table 49: 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
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
I_{IN}	Maximum current allowed through any I/O pin when the device is not turned on or during power-up/down with forward biasing of the clamp diode.(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 50: Recommended Operating Conditions (9)

Symbol	Description	Min	Typ	Max	Units
VCC	C3L, C4L, I3L, I4L speed grade core power supply.	0.82	0.85	0.88	V
	C3, C4, I3, I4 speed grade core power supply.	0.92	0.95	0.98	V
VCCA	C3L, C4L, I3L, I4L speed grade PLL analog power supply.	0.82	0.85	0.88	V

---

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

(10) Should not exceed a total of 100 mA per bank

Symbol	Description	Min	Typ	Max	Units
	C3, C4, I3, I4 speed grade PLL analog 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
T_{JCOM}	Operating junction temperature, commercial.	0	-	85	°C
T_{JIND}	Operating junction temperature, industrial.	-40	-	100	°C

Table 51: 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 52: 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 53: HVIO DC Electrical Characteristics

Voltage (V)	Typical Hysteresis (mV)(11)	Input Leakage Current (µA)	Tristate Output Leakage Current (µA)
3.3	250	±25	±10
2.5	250	±25	±10
1.8	200	±25	±10

---

(11) For input pins with Schmitt Trigger enabled

Table 54: 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 55: 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 56: 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 57: 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 58: HSIO Pins Configured as Single-Ended I/O DC Electrical Characteristics

Voltage (V)	Typical Hysteresis (mV)[12]	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 59: 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 60: 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

---

(12) For LVCMOS input pins with Schmitt Trigger enabled

Table 61: HVIO Maximum Toggle Rate

I/O Standard	Speed Grade	Serialization Mode	Max Toggle Rate (Mbps)[13][14]
3.0 V, 3.3 V LVTTL 3.0 V, 3.3 V LVCMOS	All	–	200
2.5 V LVCMOS	All	–	100
1.8 V LVCMOS	All	–	400

Table 62: HSIO Maximum Toggle Rate

I/O Standard	Speed Grade	Serialization Mode	Max Toggle Rate (Mbps)[13][14]
1.8 V, 1.5 V, 1.2 V LVCMOS	All	–	400
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
LVDS	C4, I3, I4	Full-rate	1,000
		Half-rate	1,500
	C3	Full-rate	1,000
		Half-rate	1,300
	C4L, I3L, I4L	Full-rate	800
		Half-rate	1,250
	C3L	Full-rate	800
		Half-rate	1,100
Sub-LVDS	C3, C4, I3, I4	Full-rate	1,000
		Half-rate	1,250
	C3L, C4L, I3L, I4L	Full-rate	800
		Half-rate	1,250
MIPI lane	C4, I3, I4	–	1,500
	C3	–	1,300
	I3L, C4L, I4L	–	1,250
	C3L	–	1,100

Table 63: HSIO2 Maximum Toggle Rate

I/O Standard	Speed Grade	Serialization Mode	Max Toggle Rate (Mbps)[13][14]
1.8 V, 1.5 V, 1.2 V LVCMOS	All	–	400
1.8 V, 1.5 V, 1.2 V SSTL 1.8 V, 1.5 V, 1.2 V HSTL	All	–	800
1.35 V SSTL	C4, I3, I4	–	1,333

---

(13) 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.

(14) All I/O standards are characterized with 5 pF load, except for LVTTL and LVCMOS standards which are characterized with 15 pF load.

I/O Standard	Speed Grade	Serialization Mode	Max Toggle Rate (Mbps)(13)(14)
	C3, C4L, I3L, I4L, C3L	–	(15)
LVDS	C4, I3, I4	Full-rate	1,000
		Half-rate	1,800
	C3	Full-rate	1,000
		Half-rate	1,300
	C4L, I3L, I4L	Full-rate	800
		Half-rate	1,250
	C3L	Full-rate	800
		Half-rate	1,100
Sub-LVDS	C3, C4, I3, I4	Full-rate	1,000
		Half-rate	1,250
	C3L, C4L, I3L, I4L	Full-rate	800
		Half-rate	1,250
MIPI lane	C4, I3, I4	–	2,500
	C3	–	(15)
	I3L, C4L, I4L	–	(15)
	C3L	–	(15)

Table 64: 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 65: 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

---

(15) Pending characterization.

Table 66: Single-Ended I/O Programmable Delay Chain Step Size: Static

Speed Grade	Delay per Step			Units
	Min	Typ	Max	
C3, C4, I3, I4	35	55	75	ps
C3L, C4L, I3L, I4L	50	68	89	ps

Table 67: Single-Ended I/O Programmable Delay Chain Step Size: Dynamic

Speed Grade	Delay per Step			Units
	Min	Typ	Max	
C3, C4, I3, I4	12	18	24	ps
C3L, C4L, I3L, I4L	15	22	28	ps

Table 68: Differential I/O Programmable Delay Chain Step Size: Static and Dynamic

Speed Grade	Delay per Step			Units
	Min	Typ	Max	
C3, C4, I3, I4	12	23	30	ps
C3L, C4L, I3L, I4L	20	26	34	ps

Table 69: Block RAM, DSP Block, Gobal Clock Buffer, and DPA Performance

Description	Speed Grade				Units
	C4	C3, I3, I4	C4L	C3L, I3L, I4L	
Block RAM maximum frequency.	1,000	1,000	800	800	MHz
DSP block maximum frequency.	1,000	1,000	800	800	MHz
Global clock buffer block maximum frequency.	1,000	1,000	800	800	MHz
DPA maximum data rate.	1,000	1,000	800	800	Mbps

HSIO Electrical and Timing Specifications

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

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

Table 70: 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 71: 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 72: 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 73: 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 74: 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 75: 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
$\|\Delta V_{CMTX}\|$	V_{CMTX} mismatch when output is Differential-1 or Differential-0.	–	–	5	mV
$\|V_{OD}\|$	High-speed transmit differential voltage.	140	200	270	mV
$\|\Delta V_{OD}\|$	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 76: 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 77: 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 78: 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
$V_{IL\text{-}ULPS}$	Logic 0 input voltage, ULPS state.	–	–	300	mV
V_{HYST}	Input hysteresis.	25	–	–	mV

PLL Timing and AC Characteristics

The following tables describe the PLL timing and AC characteristics.

Table 79: 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 80: PLL AC Characteristics

Test conditions at nominal voltage and room temperature.

Symbol	Parameter	Min	Typ	Max	Units
t_{DT}	Output clock duty cycle.	45	50	55	%
t_{OPJIT}	Output clock period jitter (PK-PK). This specification applies when an input jitter of 20 ps is applied.	-	-	200	ps
t_{OPJITN}	Output clock period jitter (PK-PK) with noisy input. This specification applies for a maximum allowed input jitter of 800 ps. The period jitter is measured over 10,000 sample size with minimal core and I/O activity.	-	-	400	ps
$t_{OPJITFRAC}$	Output clock period jitter for fractional mode (PK-PK).	-	-	650	ps
$t_{OPJITNFRAC}$	Output clock period jitter for fractional mode with noisy input (PK-PL).	-	-	850	ps
t_{ILJIT}	Input clock long-term jitter (PK-PK).	-	-	800	ps
t_{PLL_HLW}	PLL input clock high/low pulse width.	0.56	-	-	ns
t_{LOCK}	PLL lock-in time.	-	300	600	PFD(16)

---

(16) The PFD cycle is equal to the reference clock divider divided by the reference clock frequency.

HyperRAM Characteristics

The Ti125 FPGA in the M225S4F4 package includes a HyperRAM device. This topic describes the distributed refresh interval.

The HyperRAM device has a volatile memory array that requires a periodic refresh of all bits in the array. The refresh operation is performed by internal self-refresh logic that evenly refreshes the memory array automatically.

This automatic refresh operation occurs in standby mode or hybrid sleep mode when it is not being actively read or written by the host system. The refresh logic waits for the end of any active read or write before doing a refresh, if a refresh is needed. If a new read or write begins before the refresh completesc, the memory drives `RWDS` high during the CA period to indicate that additional initial latency time is required at the start of the new access to allow the refresh operation to complete before starting the new access. The automatic refresh operation continues to run for data retention for as long as the HyperRAM remains in standby mode or hybrid sleep mode.

The evenly distributed refresh operations requires a maximum refresh interval between two adjacent refresh operations. The maximum distributed refresh interval varies based on the temperature as shown in the following table.

Table 81: Distributed Refresh Interval by Temperature

Device Temperature (Tj °C)	Maximum Distributed Refresh Interval (µs)	CR1[1:0]
$T_J < 85$	4	01b
$85 < T_J < 125$	1	10b

The host should not perform burst transactions longer than the distributed refresh interval because longer bursts prevent distributed refresh operations happening when needed. Therefore, there is an upper limit to the length of read and write transactions so that the automatic distributed refresh operation can be done between transactions. This limit is called the CS# low maximum time (t_{CSM}); it is equal to the maximum distributed refresh interval.

The host system must respect the t_{CSM} value by ending each transaction before violating t_{CSM}, which would result in a data retention fault. The host memory controller can split long transactions when it reaches the t_{CSM} limit, or the host system hardware or software should not perform a single read or write transaction longer than the t_{CSM} time. As noted in Table 81: Distributed Refresh Interval by Temperature on page 103, the maximum refresh interval is longer at lower temperatures, and the increase in the t_{CSM} allows for longer transactions. The host system uses the `CR1[1:0]` value shown in the table to determine the maximum operating temperature. Alternatively, use a temperature sensor to determine the current system operating temperature and determine the distributed refresh interval based upon this sensor data.

Note: Refer to the HyperRAM Controller Core User Guide for managing t_{CSM} if you are using the HyperRAM Controller IP core.

Configuration Timing

The Ti125 FPGA has the following configuration timing specifications.

Note: Refer to **AN 033: Configuring Titanium FPGAs** for detailed configuration information.

Timing Parameters Applicable to All Modes

Table 82: All Modes

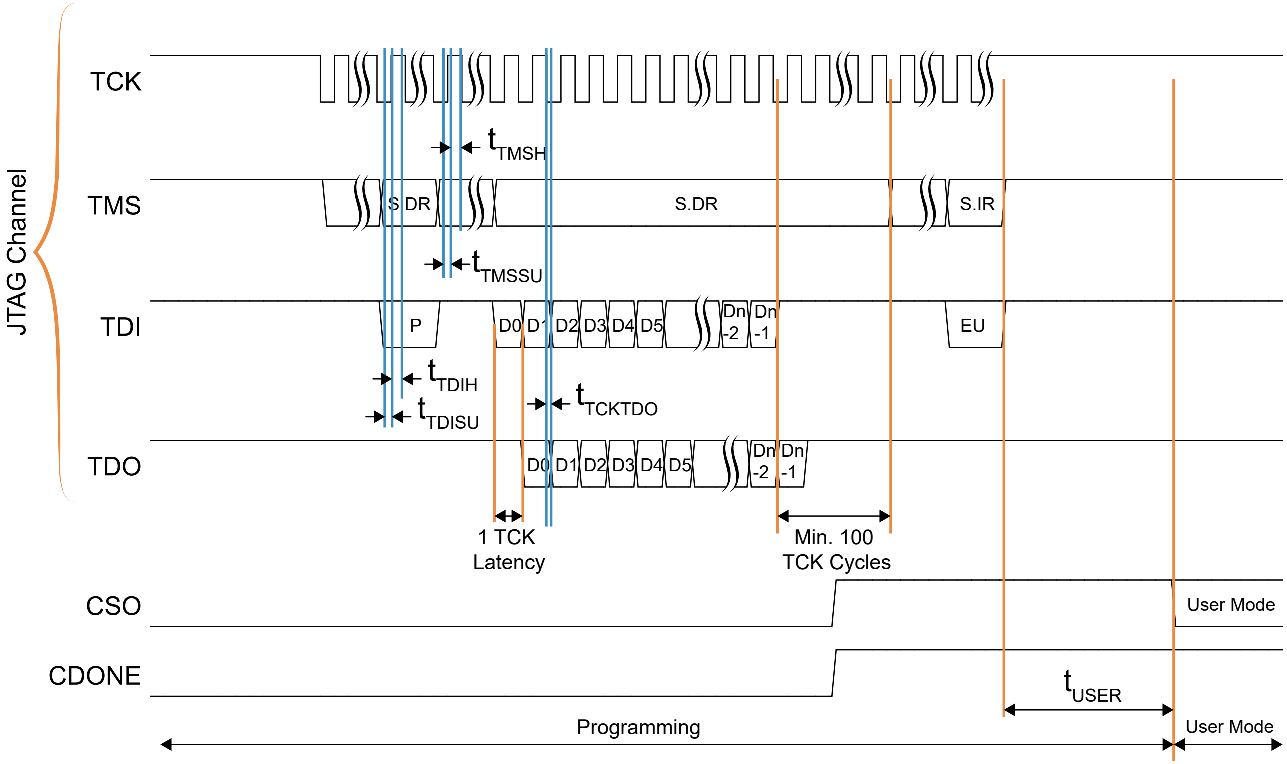
Symbol	Parameter	Min	Typ	Max	Units
t_{CRESET_N}	Minimum CRESET_N low pulse width required to trigger re-configuration.	0.32	-	-	µs
t_{USER}	Minimum configuration duration after CDONE goes high before entering user mode. Test condition at 10 kΩ pull-up resistance and 10 pF output loading on CDONE pin.	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 62: JTAG Timing Waveform



Legend:

S. DR: Shift DR S. IR: Shift IR P: PROGRAM EU: ENTERUSER

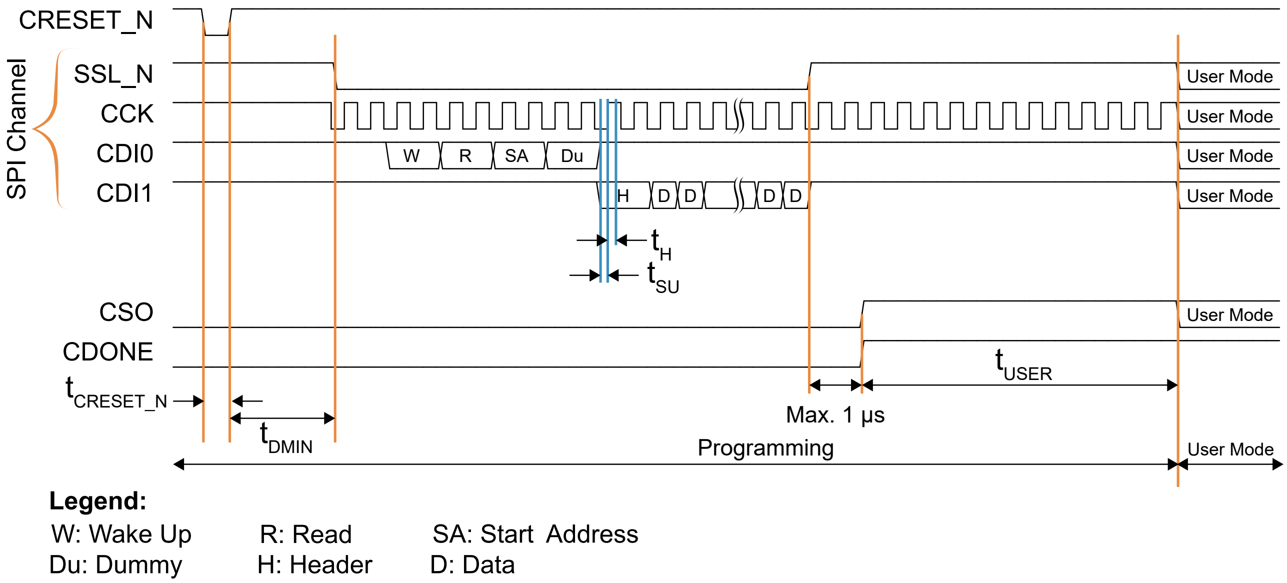
Table 83: JTAG Mode Timing

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 63: 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 84: Active Mode Timing

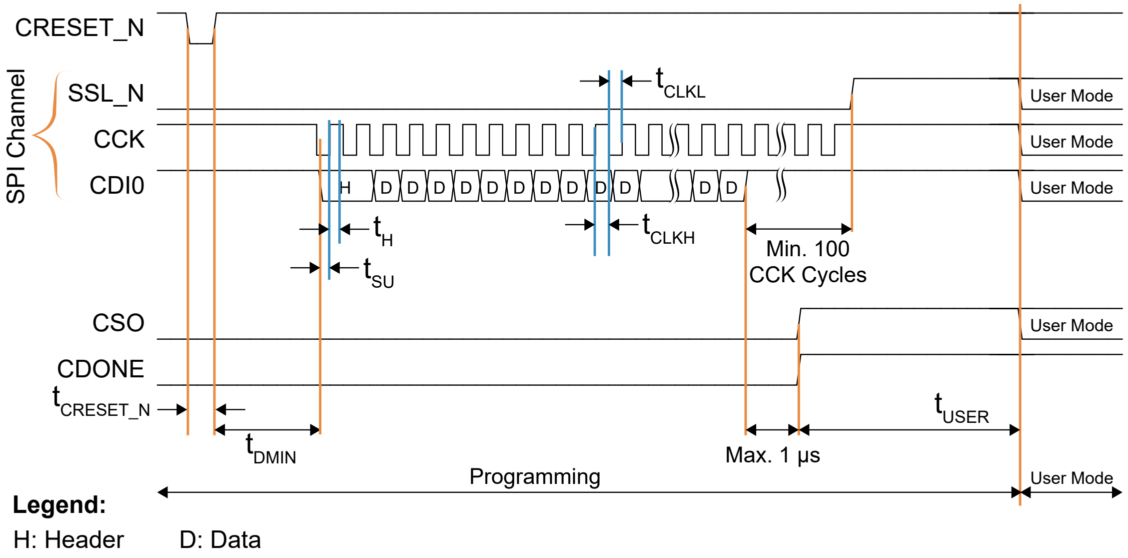
Symbol	Parameter	Frequency	Min	Typ	Max	Units
f_{MAX_M}	Active mode internal configuration clock frequency.	DIV1	52	80	100	MHz
		DIV2	26	40	52	MHz
		DIV4	13	20	26	MHz
		DIV8	6.5	10	13	MHz
$f_{MAX\_M\_EXTCLK}$	Active mode external configuration clock frequency.	-	-	-	100	MHz
t_{SU}	Setup time. Test condition 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
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 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 64: 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 Titanium™ FPGA for at least 100 cycles after sending the last configuration data.

Table 85: 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 86: Power and Ground Pinouts

xx indicates the bank location.

Function	Description
VCC	Core power supply.
VCCA_*xx*	PLL analog power supply.
VCCAUX	1.8 V auxiliary power supply.
VCCIO33_*xx*	HVIO bank power supply.
VCCIO*xx*	HSIO bank power supply.
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 87: 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 88: 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.
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 89: 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`.
Calculate the resistor value as described in "Resistors in Configuration Circuitry" in **AN 033: Configuring Titanium FPGAs**.

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 voltage for the bank in which it is located (BR0), or any banks merged with it. Supply VCCIO33_BR0 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

---

(17) CDONE has a drive strength of 12 mA at 1.8 V.

Table 90: Dual-Purpose Configuration Pins

In user mode (after configuration), you can use these dual-purpose pins as general I/O.
Calculate the resistor value as described in "Resistors in Configuration Circuitry" in **AN 033: Configuring Titanium FPGAs**.

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. 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 either a device mismatch or a failed bitstream CRC check. Refer to **Table 1**.	-
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).	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.

Pin States

HVIO pins have an internal pullup/down (see Figure 25: HVIO Interface Block on page 36); HSIO and HSIO2 configured as GPIO have an internal pull up/down (see Figure 27: I/O Interface Block on page 39). 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 91: 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.(18)
HSIO2	Input tri-state with weak pull up.	Input tri-state with weak pull up.	Input tri-state with weak pull up.(18)
HVIO	Input tri-state with weak pull up.	Input tri-state with weak pull up.	Input tri-state with weak pull up.(18)
Dual-Purpose Configuration Pins			
CSO	0	0(19)	Input tri-state with weak pull up.
NSTATUS	0	1(20)	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 , `CRESET_N` must be kept low during power up.

Note: Refer to the following tables for details:
Table 64: HVIO Internal Weak Pull-Up and Pull-Down Resistance on page 97

---

(18) You can change the default mode to weak pull-down in the Interface Designer.
(19) CSO is driven to 1 when the bitstream is done transmitting.
(20) NSTATUS set to 1 if valid bit stream detected. Remains at 0 if bit stream is authenticated or encrypted.

Ti125 Interface Floorplan

Note: The numbers in the floorplan figures indicate the HVIO, HSIO and HSIO2 number ranges. Some packages may not have all pins in the range bonded out.

Refer to the Ti125 Pinout for information on which pins are available in each package.

Figure 65: Floorplan Diagram for F225 Packages

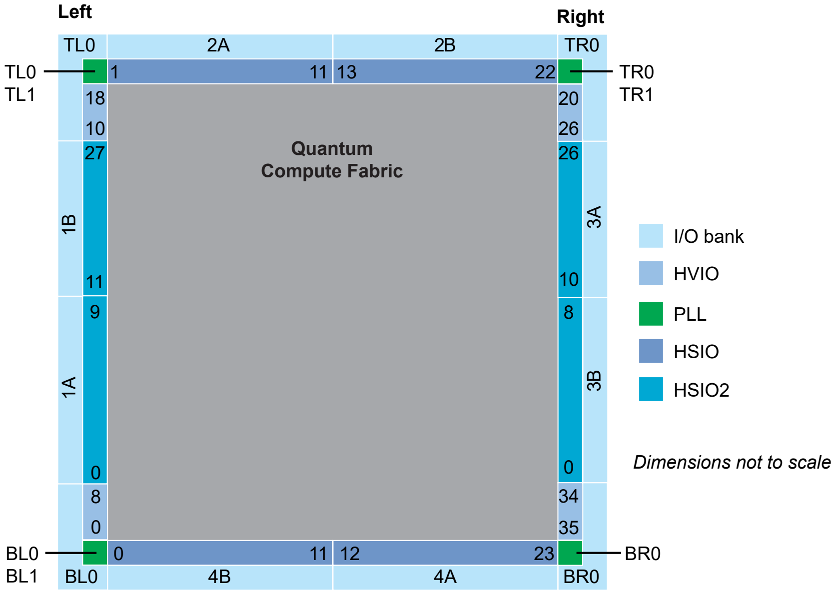
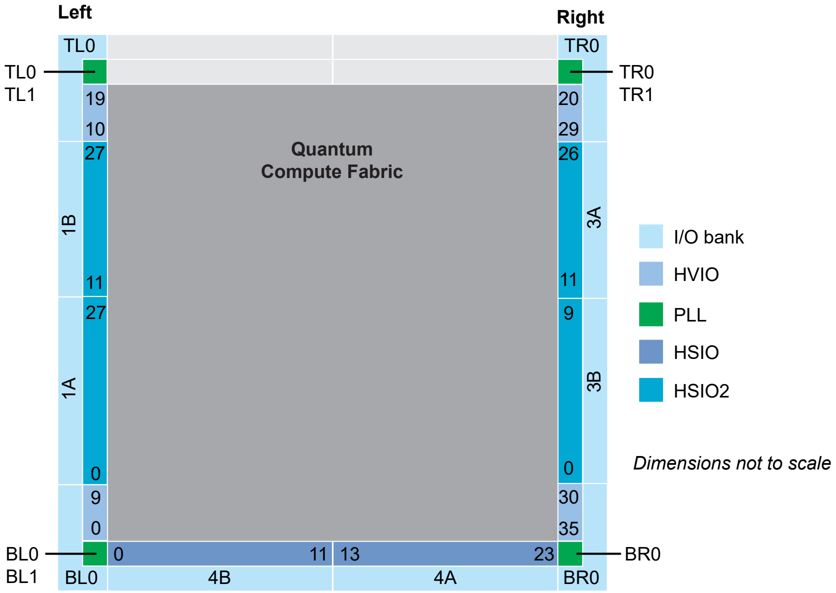


Figure 66: Floorplan Diagram for M225S4F4 Packages



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 Ti125 FPGA. The software supports the Verilog HDL, SystemVerilog, and VHDL languages.

Ordering Codes

Refer to the **Titanium Selector Guide** for the full listing of Ti125 ordering codes.

Revision History

Table 92: Revision History

Date	Version	Description
April 2026	1.0	Initial release.