易灵思®

Ti60 Data Sheet

DSTi60-v1.1
October 2021
www.elitestek.com

Contents

Introduction

The 钛金系列 Ti60 FPGA features the high-density, low-power 易灵思® Quantum™compute fabric wrapped with an I/O interface in a small footprint package for easy integration. Ti60 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 Ti60 FPGAs, designers can build products that are always on, providing enhanced capabilities for applications such as 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
  - PLL support
- FPGA interface blocks
  - High-voltage I/O (HVIO) (1.8, 2.5, 3.3 V)
  - High-speed I/O (HSIO), configurable as:
    - LVDS, subLVDS, Mini-LVDS, and RSDS (RX, TX, and bidirectional), up to 1.6 Gbps
    - MIPI lane I/O (DSI and CSI) in high-speed (HS) low-power (LP) modes, up to 1.5 Gbps
    - Single-ended and differential I/O
  - PLL
  - Oscillator
- Flexible device configuration
  - Standard SPI interface (active, passive, and daisy chain[1])
  - JTAG interface
  - Supports internal reconfiguration
- Single-event upset (SEU) detection feature
- Fully supported by the Efinity® software, an RTL-to-bitstream compiler
- Optional security feature
  - Asymmetric bitstream authentication using RSA-4096
  - Optional bitstream encryption/decryption using AES-GCM

Important: All specifications are preliminary and pending hardware characterization.

[1] Daisy-chain is not supported in F100 package.

Table 1: Ti60 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
62,016	60,800	Up to 32	2.6	256	160

Table 2: Ti60 Package-Dependent Resources

Resource		W64	F100	F225
Single-ended GPIO (Max)	HVIO (1.8, 2.5, 3.0, 3.3 V LVCMOS, 3.0, 3.3 V LVTTL)	–	–	23
	HSIO (1.2, 1.5, 1.8 V LVCMOS, HSTL and SSTL)	35	61	140
Differential GPIO (Max)	HSIO (LVDS, Differential HSTL, SSTL, MIPI TX Data and Clock Lanes)	17	30	70
	HSIO (MIPI RX Data Lanes)	8	21	58
	HSIO (MIPI RX Clock Lanes)	2	3	12
Global clock or control signals from GPIO pins		3	8	15
PLLs		2	3	4

Available Package Options

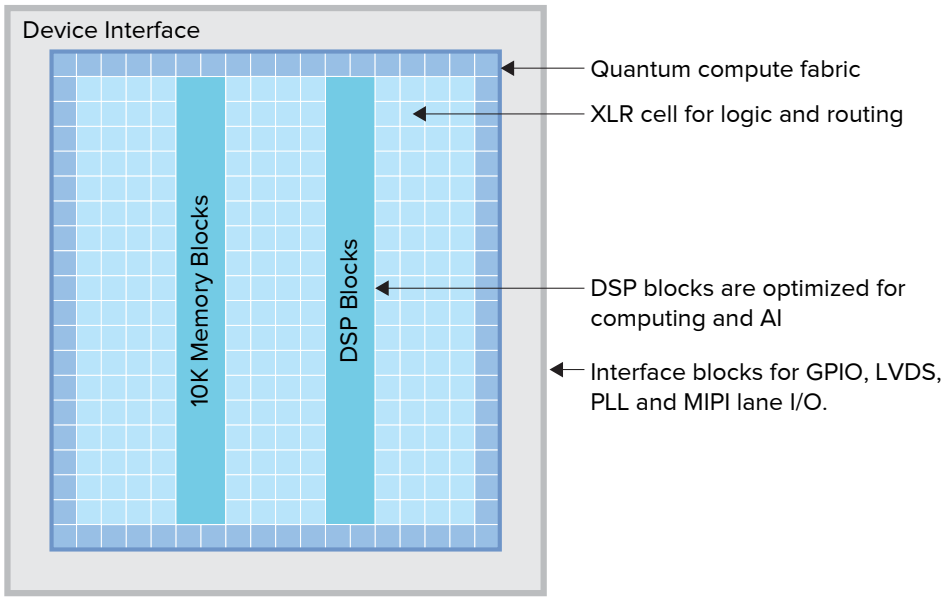
Table 3: Available Packages

Package	Dimensions (mm x mm)	Pitch (mm)
64-ball WLCSP	3.5 x 3.4	0.4
100-ball FBGA	5.5 x 5.5	0.5
225-ball FBGA	10 x 10	0.65

Device Core Functional Description

Ti60 FPGAs feature an eXchangeable Logic and Routing (XLR) cell that 易灵思® has optimized for a variety of applications. 钛金系列 FPGAs contain LEs that are constructed from XLR cells. Each FPGA in the 钛金系列 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: Ti60 FPGA Block Diagram



XLR Cell

The eXchangeable Logic and Routing (XLR) cell is the basic building block of the Quantum™ architecture. The 易灵思® 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.

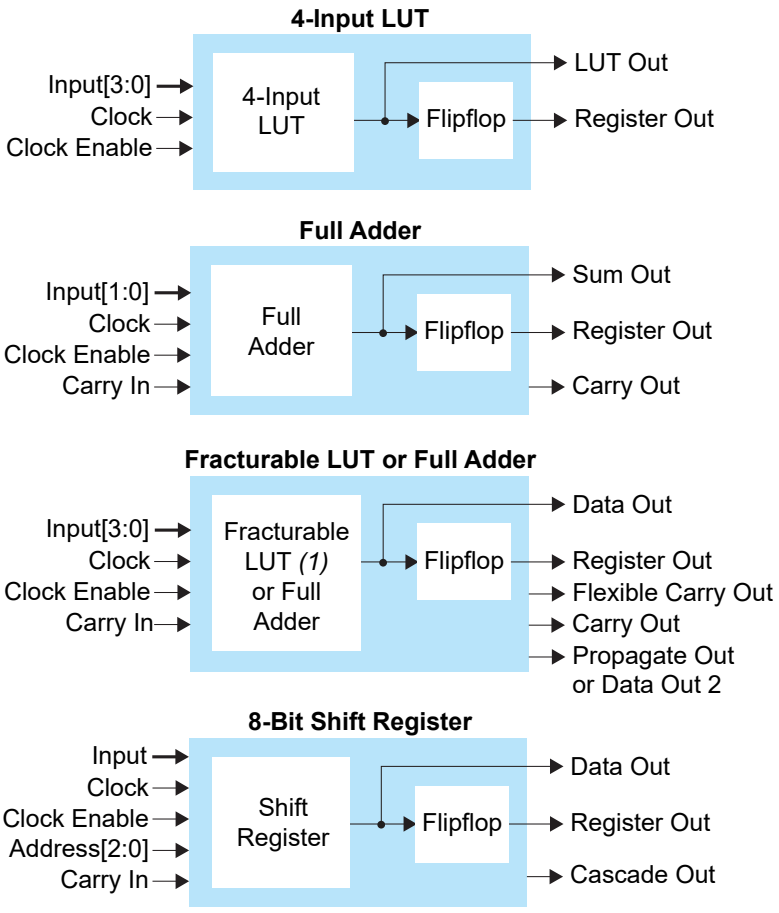
Learn more: For more detailed on the advantages the XLR cell brings to 钛金系列 FPGAs, read the Why the XLR Cell is a Big Deal White Paper.

The XLR cell functions as:

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

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

Figure 2: Logic Cell Functions



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

Learn more: Refer to the Quantum 钛金系列 Primitives User Guide for details on the 钛金系列 logic cell primitves.

Embedded Memory

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

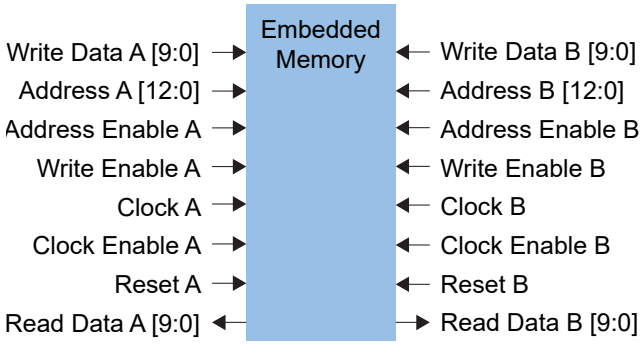
The read and write ports support independently configured data widths, an address enable, and an output register reset. The simple dual-port mode also supports a write byte enable.

True Dual-Port Mode

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

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

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



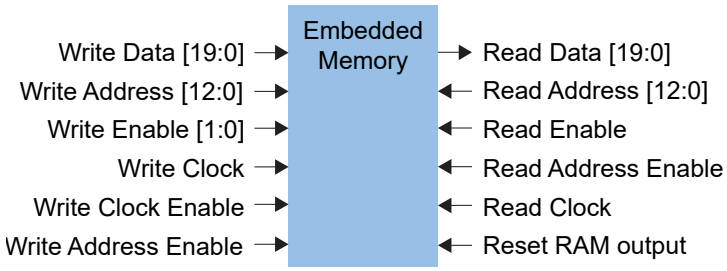
Learn more: Refer to the **Quantum 钛金系列 Primitives User Guide** for details on the RAM configuration.

Simple Dual-Port Mode

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

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

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



Learn more: Refer to the **Quantum 钛金系列 Primitives User Guide** for details on the RAM configuration.

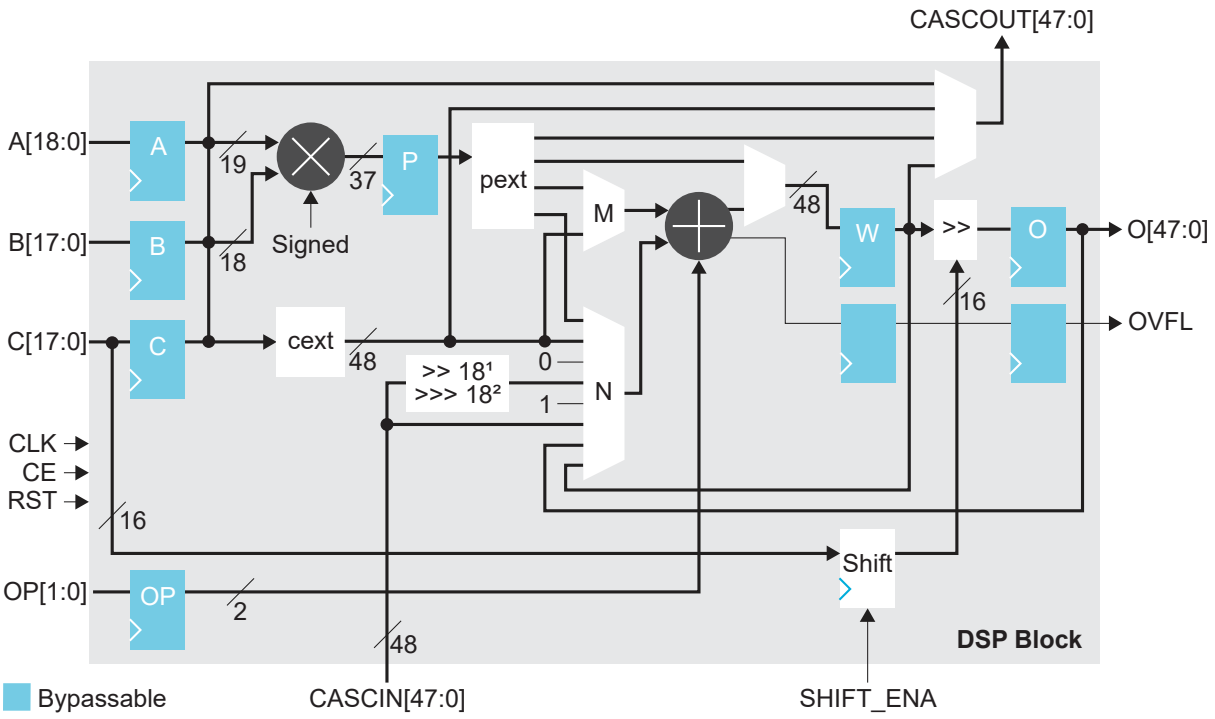
DSP Block

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

- *Normal*—One 19 x 18 integer multiplication with 48-bit addition/subtraction.
- *Dual*—One 11 x 10 integer multiplication and one 8 x 8 integer multiplication with two 24-bit additions/subtractions.
- *Quad*—One 7 x 6 integer multiplication and three 4 x 4 integer multiplications with four 12-bit additions/subtractions.
- *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 钛金系列 Primitives User Guide** for more information about the 钛金系列 DSP block primitives.

Clock and Control Network

The clock and control network is distributed through the FPGA to provide clocking for the core's LEs, memory, DSP blocks, I/O blocks, and control signals. The Ti60 has 32 global signals with 8 per side that can be used as either clocks or control signals. The global signals are balanced trees that feed all FPGA modules.

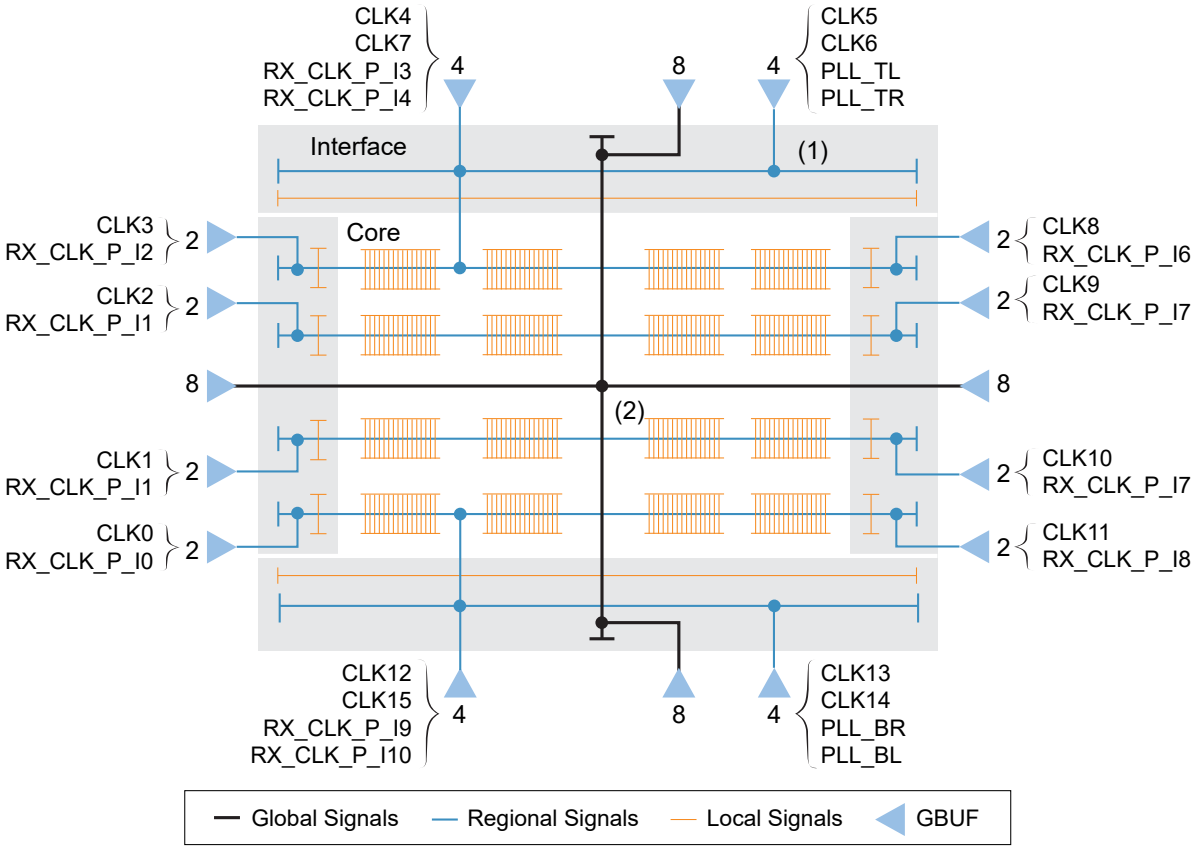
The Ti60 also has regional signals that can only reach certain FPGA regions, including the top or bottom edges. The Ti60 has 4 regional networks for the core, right interface, and left interface blocks. The top and bottom interface blocks have 1 regional clock network each. You can drive the right and left sides of each region independently.

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

The Ti60 local signals are generated locally by logic within each region and can only be used within that region.

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 Clock Network



1. Each blue horizontal line represents a region.
2. The number of local signals shown is for illustration only.

Global and Regional Signal Location

The following table describes the location of the global clock, regional clock, global control and regional control signal in Ti60 FPGAs.

The symbols in the tables represent the clock and control signal access:

- *C*—Access to core fabric
- *I*—Access to interface block

Table 4: Clock or Control Input from GPIO Pins

The clock supports inputs from single-ended and differential I/O. This table lists the function and resource name from single-ended I/O. For differential I/O, use both the P and N resources.

Function Name	Resource Name	Global Clock (GCLK) Global Control (GCTRL)				Regional Clock (RCLK) Regional Control (RCTRL)			
		Left	Top	Right	Bottom	Left	Top	Right	Bottom
CLK0	GPIOL_P_07_CLK0_P	C, I	–	–	–	C, I	–	–	–
CLK1	GPIOL_P_08_CLK1_P	C, I	–	–	–	C, I	–	–	–
CLK2	GPIOL_P_09_CLK2_P	C, I	–	–	–	C, I	–	–	–
CLK3	GPIOL_P_10_CLK3_P	C, I	–	–	–	C, I	–	–	–
CLK4	GPIOT_P_07_CLK4_P	–	C, I	–	–	–	C, I	–	–
CLK5	GPIOT_P_08_CLK5_P	–	C, I	–	–	–	I	–	–
CLK6	GPIOT_P_09_CLK6_P	–	C, I	–	–	–	I	–	–
CLK7	GPIOT_P_10_CLK7_P	–	C, I	–	–	–	C, I	–	–
CLK8	GPIOR_P_11_CLK8_P	–	–	C, I	–	–	–	C, I	–
CLK9	GPIOR_P_10_CLK9_P	–	–	C, I	–	–	–	C, I	–
CLK10	GPIOR_P_09_CLK10_P	–	–	C, I	–	–	–	C, I	–
CLK11	GPIOR_P_08_CLK11_P	–	–	C, I	–	–	–	C, I	–
CLK12	GPIOB_P_10_CLK12_P	–	–	–	C, I	–	–	–	C, I
CLK13	GPIOB_P_09_CLK13_P	–	–	–	C, I	–	–	–	I
CLK14	GPIOB_P_08_CLK14_P	–	–	–	C, I	–	–	–	I
CLK15	GPIOB_P_07_CLK15_P	–	–	–	C, I	–	–	–	C, I

Table 5: Clock from PLL OUTCLK Signal

PLL Reference	GCLK				RCLK			
	Left	Top	Right	Bottom	Left	Top	Right	Bottom
PLL_TL	C, I	C, I	–	–	–	I	–	–
PLL_TR	–	C, I	C, I	–	–	I	–	–
PLL_BR	–	–	C, I	C, I	–	–	–	I
PLL_BL	C, I	–	–	C, I	–	–	–	I

Table 6: Clock from MIPI RX Lane CLKOUT Signal

MIPI RX lane CLKOUT signal is derived from the dedicated clock input of MIPI RX function pin.

MIPI RX Function	Resource Name	GCLK				RCLK			
		Left	Top	Right	Bottom	Left	Top	Right	Bottom
RX_CLK_P_I0 RX_CLK_N_I0	GPIOL_P_02 GPIOL_N_02	C, I	–	–	–	C, I	–	–	–
RX_CLK_P_I1 RX_CLK_N_I1	GPIOL_P_09 GPIOL_N_09	C, I	–	–	–	C, I	–	–	–
RX_CLK_P_I2 RX_CLK_N_I2	GPIOL_P_16 GPIOL_N_16	C, I	–	–	–	C, I	–	–·	–
RX_CLK_P_I3 RX_CLK_N_I3	GPIOT_P_03 GPIOT_N_03	–	C, I	–	–	–	C, I	–	–
RX_CLK_P_I4 RX_CLK_N_I4	GPIOT_P_09 GPIOT_N_09	–	C, I	–	–	–	C, I	–	–
RX_CLK_P_I5 RX_CLK_N_I5	GPIOT_P_14 GPIOT_N_14	–	C, I	–	–	–	–	–	–
RX_CLK_P_I6 RX_CLK_N_I6	GPIOR_P_02 GPIOR_N_02	–	–	C, I	–	–	–	C, I	–
RX_CLK_P_I7 RX_CLK_N_I7	GPIOR_P_10 GPIOR_N_10	–	–	C, I	–	–	–	C, I	–
RX_CLK_P_I8 RX_CLK_N_I8	GPIOR_P_17 GPIOR_N_17	–	–	C, I	–	–	–	C, I	–
RX_CLK_P_I9 RX_CLK_N_I9	GPIOB_P_03 GPIOB_N_03	–	–	–	C, I	–	–	–	C, I
RX_CLK_P_I10 RX_CLK_N_I10	GPIOB_P_09 GPIOB_N_09	–	–	–	C, I	–	–	–	C, I
RX_CLK_P_I11 RX_CLK_N_I11	GPIOB_P_14 GPIOB_N_14	–	–	–	C, I	–	–	–	–

Learn more: Refer to **MIPI RX Lane** on page 29 for more information about the MIPI RX function.

Global Network Multiplexers

You can access the Ti60 global clock network using the global clock GPIO pins, PLL outputs, oscillator output, and core-generated clocks. Similarly, the Ti60 has GPIO pins that you can configure as control inputs to access the high-fanout network connected to the LE's set, reset, and clock enable signals.

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 钛金系列 **Interfaces User Guide** for information on how to configure the global and regional clock networks.

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 钛金系列 family support a variety of interfaces to meet the needs of different applications.

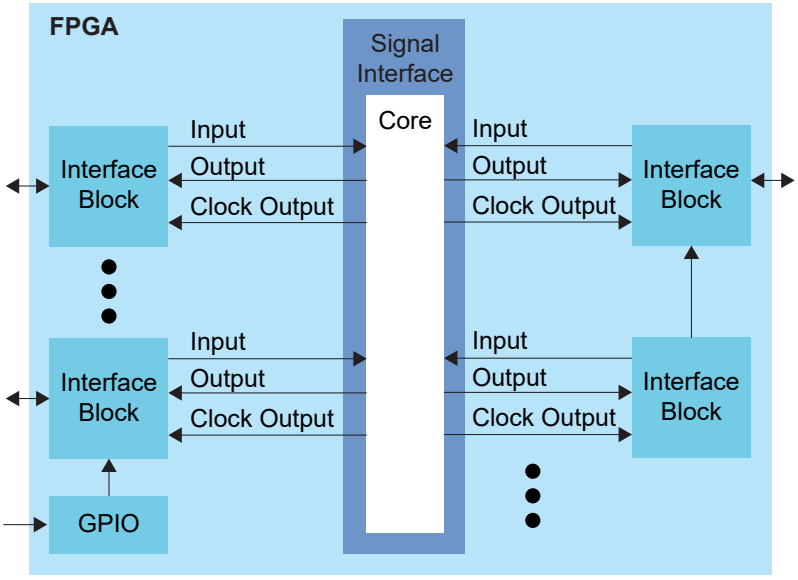
Learn more: The following sections describe the available device interface features in Ti60 FPGAs. Refer to the 钛金系列 **Interfaces User Guide** for details on the Efinity® Interface Designer settings.

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 7: 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 钛金系列 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 in the Ti60. Signals and block diagrams are shown from the perspective of the interface, not the core.

GPIO

The Ti60 FPGA supports two types of GPIO:

- *High-voltage I/O (HVIO)*—Simple I/O blocks that can support single-ended I/O standards.
- *High-speed I/O (HSIO)*—Complex I/O blocks that can support single-ended and differential I/O functionality.

The I/O logic comprises three register types:

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

The HVIO supports the following I/O standards.

Table 7: 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: 易灵思 recommends that you limit the number of 3.0/3.3 V HVIO as I/O or output to 6 per bank
to avoid switching noise. The Efinity® software issues a warning if you exceed the recommended limit.

The HSIO supports the following I/O standards.

Table 8: HSIO Supported I/O Standards

Standard	VCCIO (V)		VCCAUX (V)	VREF (V)	When Configured As
	TX	RX			
LVCMOS 1.8 V	1.8	1.8	1.8	–	GPIO
LVCMOS 1.5 V	1.5	1.5	1.8	–	GPIO
LVCMOS 1.2 V	1.2	1.2	1.8	–	GPIO
HSTL/Differential HSTL 1.8 V SSTL/Differential SSTL 1.8 V	1.8	1.8	1.8	0.9	GPIO
HSTL/Differential HSTL 1.5 V SSTL/Differential SSTL 1.5 V	1.5	1.5, 1.8	1.8	0.75	GPIO
HSTL/Differential HSTL 1.2 V SSTL/Differential SSTL 1.2 V	1.2	1.2, 1.5, 1.8	1.8	0.6	GPIO
LVDS/RSDS/mini-LVDS	1.8	1.8	1.8	–	LVDS
Sub-LVDS	1.8	1.5, 1.8	1.8	–	LVDS
MIPI Lane I/O /SLVS	1.2	1.2	1.8	–	MIPI Lane

The differential receivers are powered by VCCAUX, which gives you the flexibility to choose the VCCIO you want to use.

Features for HVIO and HSIO Configured as GPIO

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

Table 9: Features for HVIO and HSIO Configured as GPIO

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

Table 10: 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.

During configuration, all GPIO pins are configured in weak pull-up mode.

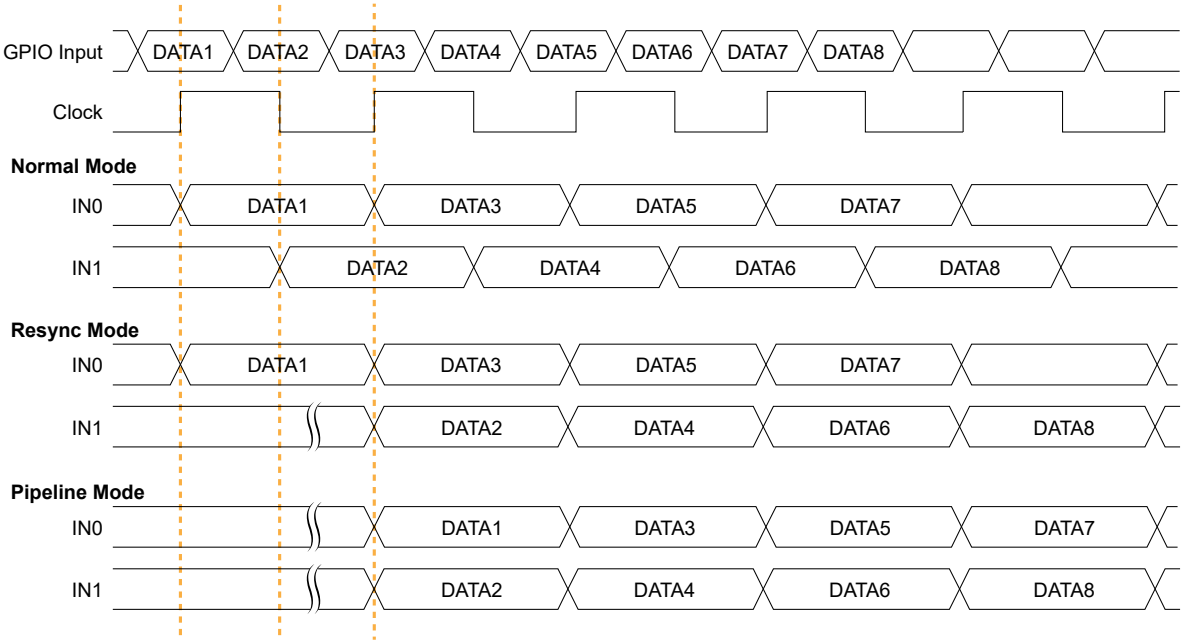
During user mode, unused GPIO pins are tristated and configured in weak pull-up mode. You can change the default mode to weak pull-down in the Interface Designer.

Double-Data I/O

Ti60 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.

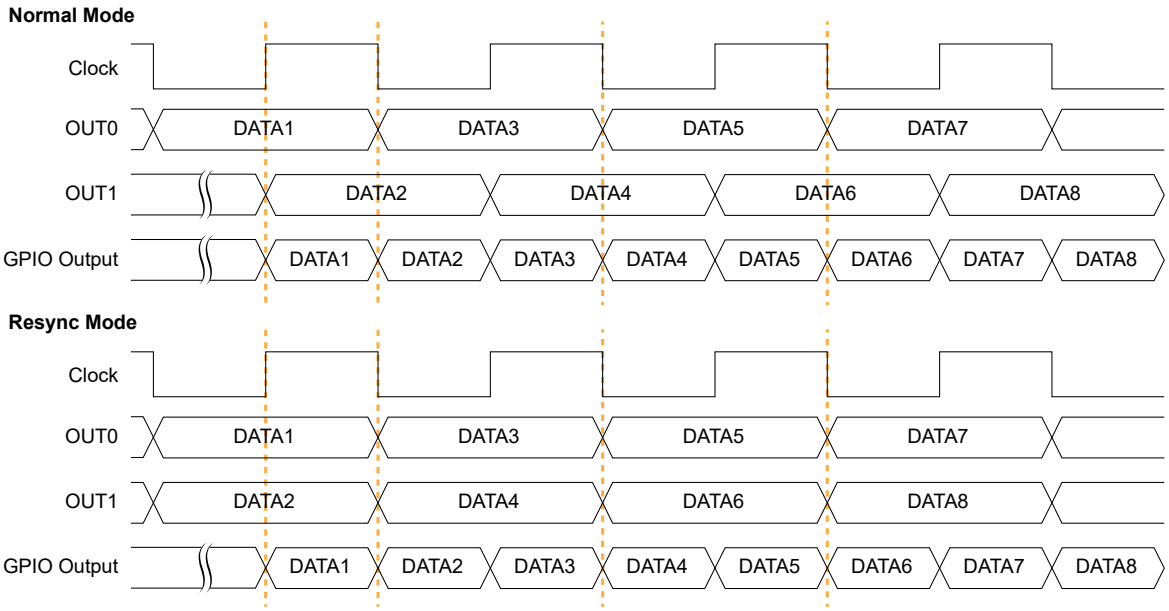
In normal mode, the interface receives or sends data directly to or from the core on the positive and negative clock edges. In resync and pipeline mode, the interface resynchronizes the data to pass both signals on the positive clock edge only.

Figure 8: 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 9: 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 HSIO configured as GPIO supports programmable delay chain. In some cases you can use static and dynamic delays at the same time.

Table 11: Programmable Delay Support

Delay Type	GPIO Type	Description
Static	Single-ended	16 steps with approximately 60 ps of delay per step, both input and output paths.
	Differential RX	64 steps of approximately 25 ps delay per step. You cannot use the static and dynamic delay at the same time. Available only on P input of the HSIO pin pair.
	Differential TX	16 steps with approximately 60 ps of delay per step.
Dynamic	Single ended and differential	64 steps with approximately 25 ps of delay per step, input only. Only available on input (RX) delay; available only on P input of the HSIO pin pair.

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 10: HVIO Interface Block

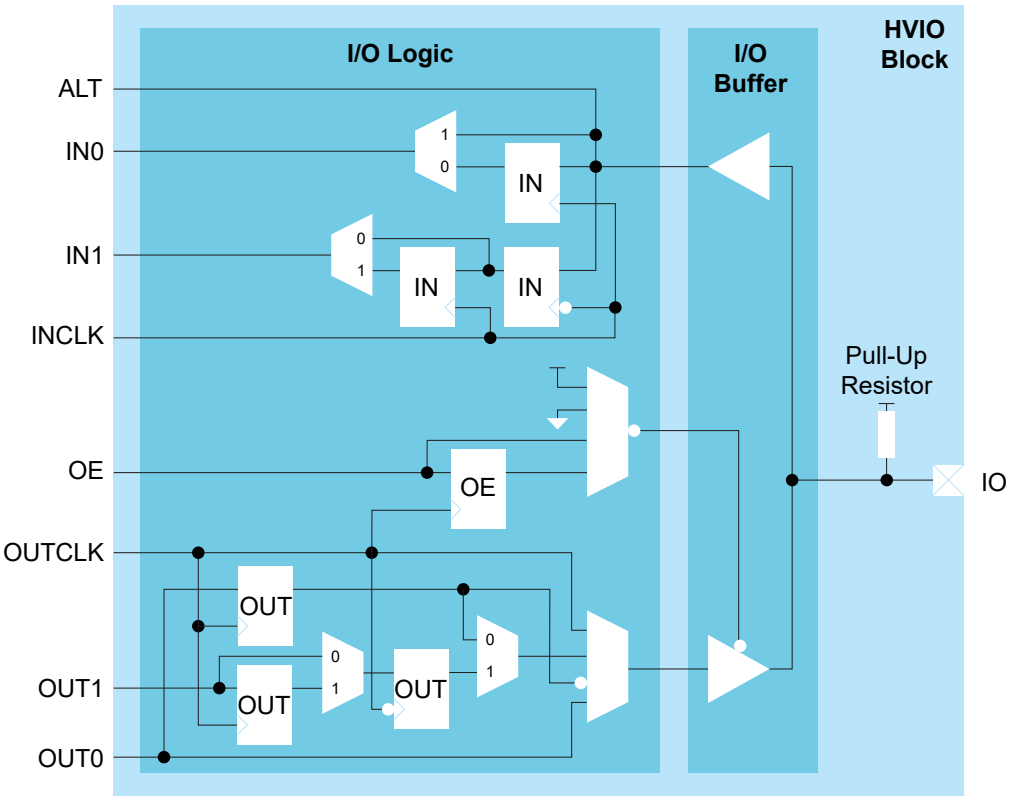


Table 12: HVIO Signals (Interface to FPGA Fabric)

Signal	Direction	Description
IN[1:0]	Output	Input data from the HVIO pad to the core fabric. IN0 is the normal input to the core. In DDIO mode, IN0 is the data captured on the positive clock edge (HI pin name in the Interface Designer) and IN1 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.
OUT[1:0]	Input	Output data to HVIO pad from the core fabric. OUT0 is the normal output from the core. In DDIO mode, OUT0 is the data captured on the positive clock edge (HI pin name in the Interface Designer) and OUT1 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

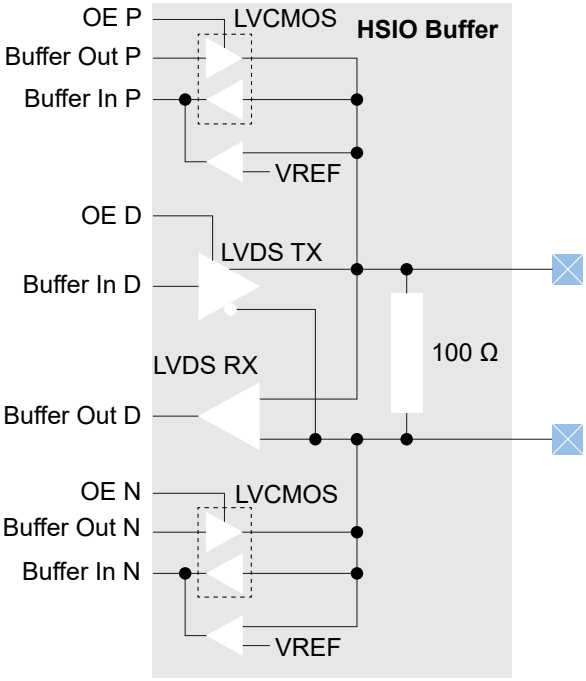
Signal	Direction	Description
IO	Bidirectional	HVIO pad.

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)

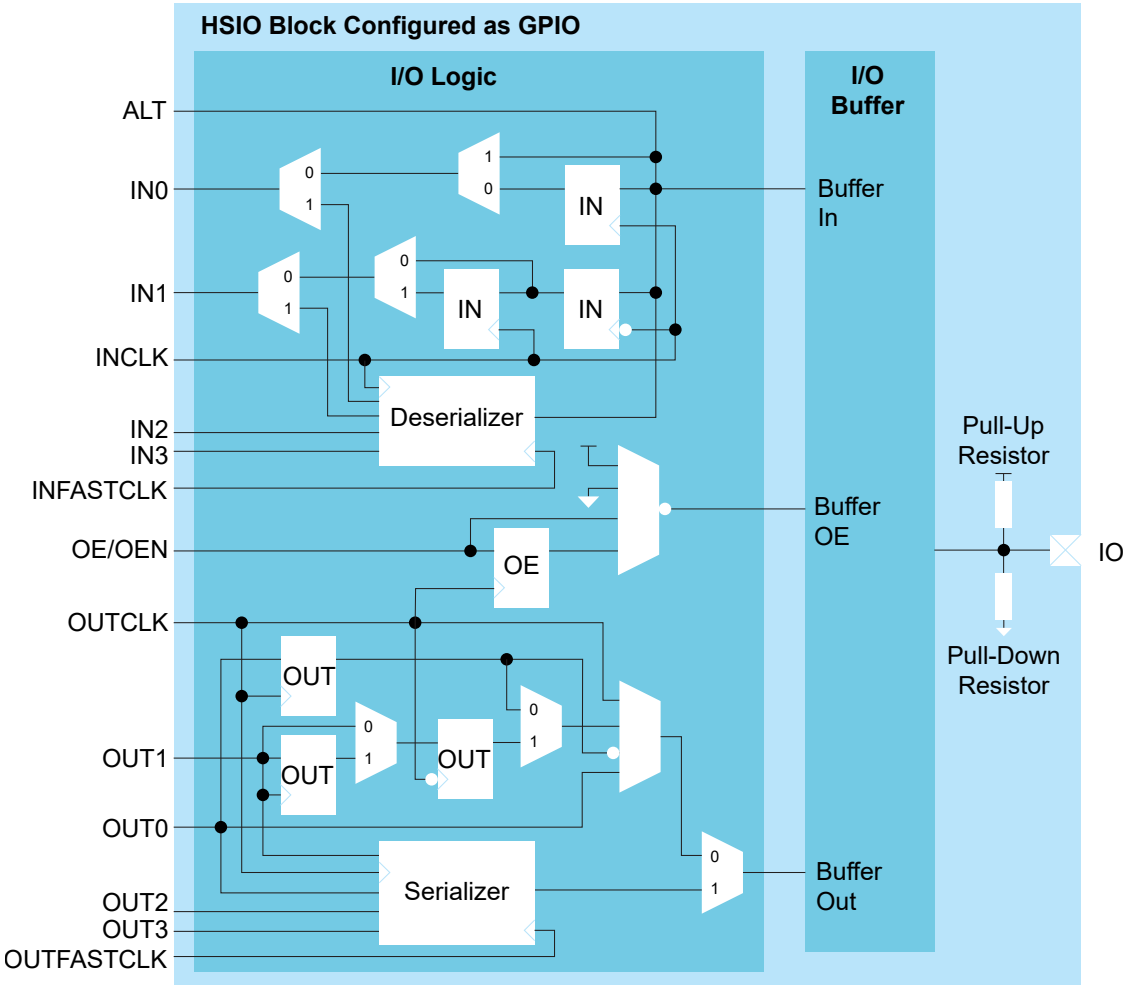
Figure 11: HSIO Buffer Block Diagram



HSIO Configured as GPIO

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

Figure 12: I/O Interface Block



Important: In the same bank, when you are using an HSIO pin as a GPIO, make sure to leave at least 1 pair of unassigned HSIO pins between the HSIO pin as GPIO and HSIO pins. This separation reduces noise. The Efinity software issues an error if you do not leave this separation.

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

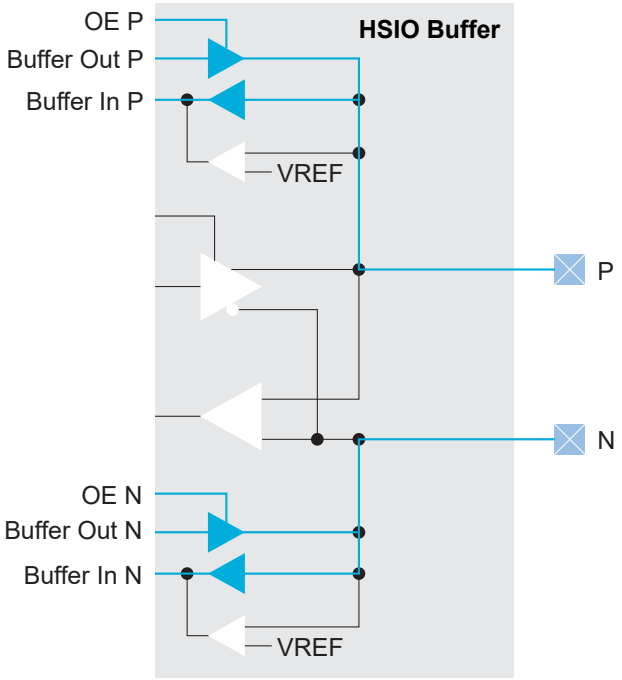
Signal	Direction	Description
IN[3:0]	Output	Input data from the pad to the core fabric. IN0 is the normal input to the core. In DDIO mode, IN0 is the data captured on the positive clock edge (HI pin name in the Interface Designer) and IN1 is the data captured on the negative clock edge (LO pin name in the Interface Designer). When using the deserializer, the first bit is on IN0 and the last bit is on IN3.
ALT	Output	Alternative input connection for GCLK, GCTRL, PLL_CLKIN, RCLK, RCTRL, PLL_EXTFB, and VREF. (In the Interface Designer, **Register Option** is **none**).
OUT[3:0]	Input	Output data to GPIO pad from the core fabric. OUT0 is the normal output from the core. In DDIO mode, OUT0 is the data captured on the positive clock edge (HI pin name in the Interface Designer) and OUT1 is the data captured on the negative clock edge (LO pin name in the Interface Designer). When using the serializer, the first bit is on OUT0 and the last bit is on OUT3.
OE/OEN	Input	Output enable from core fabric to the I/O block. Can be registered. OEN is used in diffferential mode. Drive it with the same signal as OE.
DLY_ENA	Input	(Optional) Enable the dynamic delay control.
DLY_INC	Input	(Optional) Dynamic delay control. When DLY_ENA = 1, 1: Increments 0: Decrements
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.

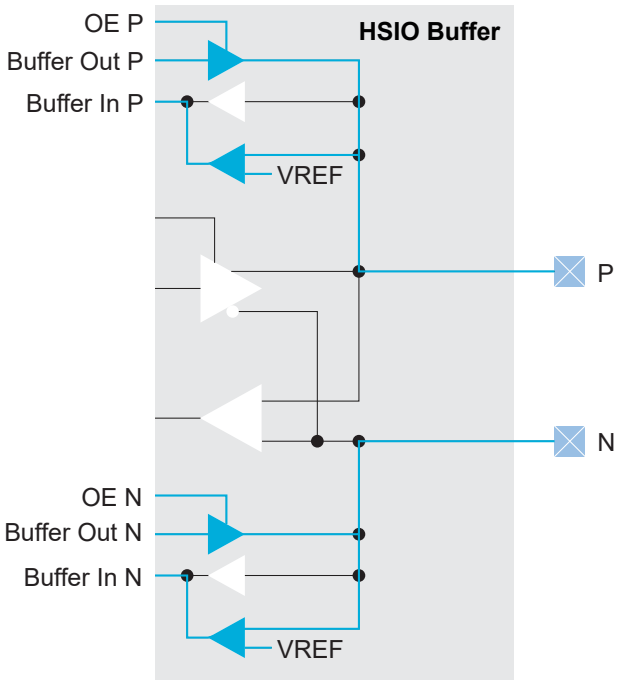
Figure 13: I/O Buffer Path for LVCMOS



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.

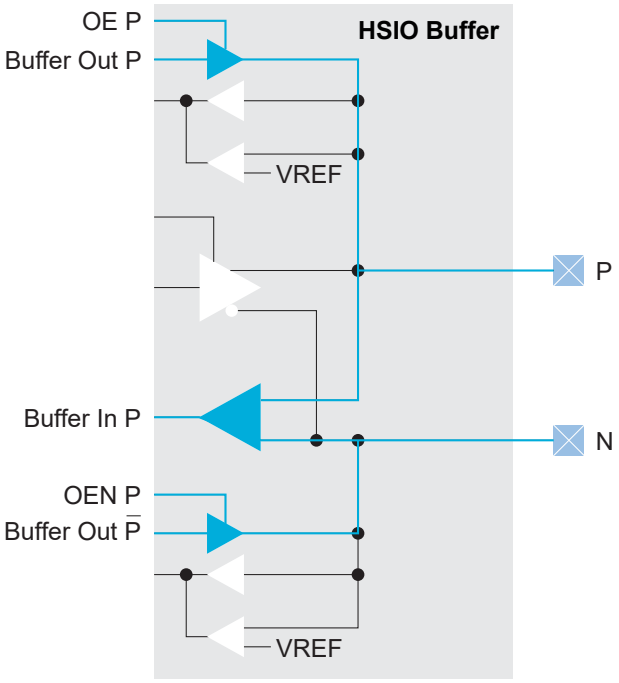
Important: When configuring an I/O pad of the standard's input path as a `VREF` pin, you must use the `VREF` from the same physical I/O bank even when the I/O banks are merged to share a common VCCIO pin.

Figure 14: 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 15: 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 1.6 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.

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.

Ti60 FPGAs do not have a dedicated LVDS clock tree; therefore, the fast and slow clocks must use global or regional clocks to feed the LVDS primitives.

LVDS RX

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

Figure 16: LVDS RX Interface Block Diagram

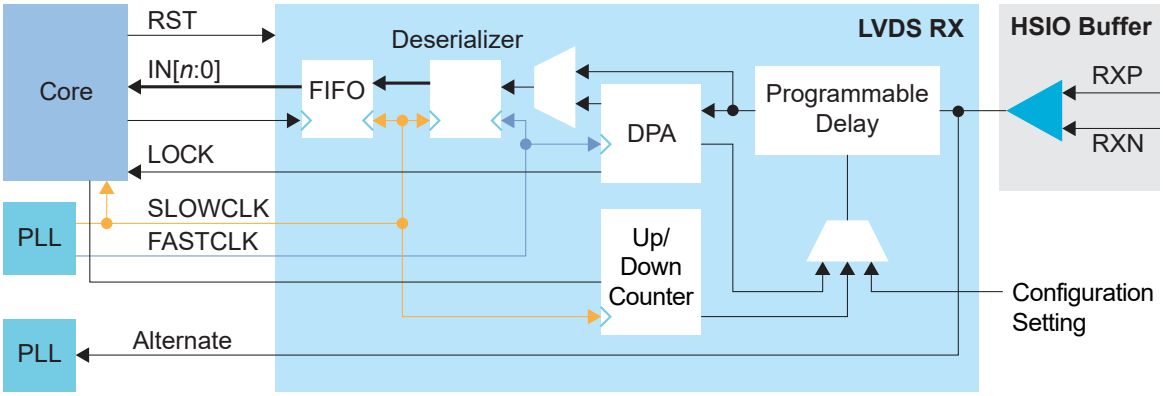
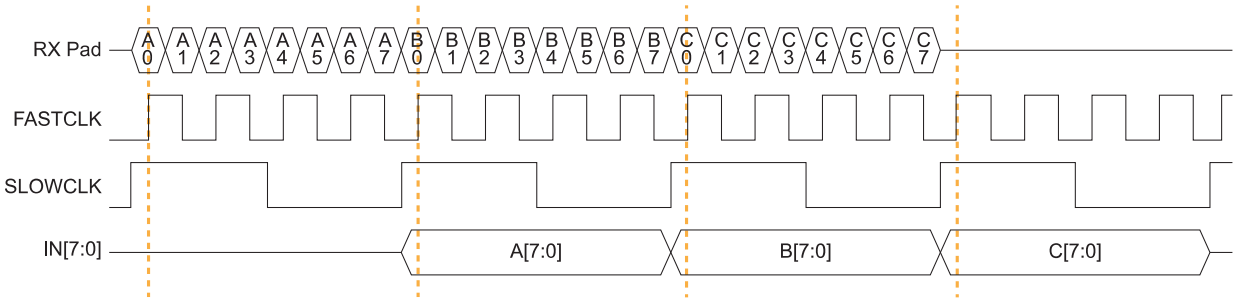


Table 17: LVDS RX Signals (Interface to FPGA Fabric)

Signal	Direction	Clock Domain	Description
IN[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.
LOCK	Output		(Optional) When DPA is enabled, this signal indicates that the DPA has achieved training lock and data can be passed.
FIFO_EMPTY	Output	FIFOCLK	(Optional) When FIFO is enabled, this signal indicates that the FIFO is empty.
SLOWCLK	Input	–	Parallel (slow) clock.
FASTCLK	Input	–	Serial (fast) clock.
FIFOCLK	Input	–	(Optional) Core clock to read from the FIFO.
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.
ENA	Input	–	Dynamically enable or disable the LVDS input buffer. Can save power when disabled. 1: Enabled 0: Disabled
TERM	Input	–	Enables or disables termination in dynamic termination mode. 1: Enabled 0: Disabled
DLY_ENA	Input	SLOWCLK	(Optional) Enable the dynamic delay control or the DPA circuit, depending on which one is enabled.
DLY_INC	Input	SLOWCLK	(Optional) Dynamic delay control. Cannot be used with DPA enabled. When DLY_ENA is 1: 1: Increments 0: Decrements
DLY_RST	Input	SLOWCLK	(Optional) Reset the delay counter or the DPA circuit, depending on which one is enabled.

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



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

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 18: LVDS TX Interface Block Diagram

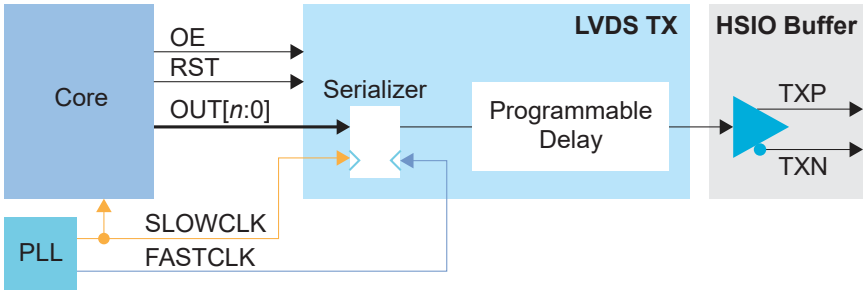
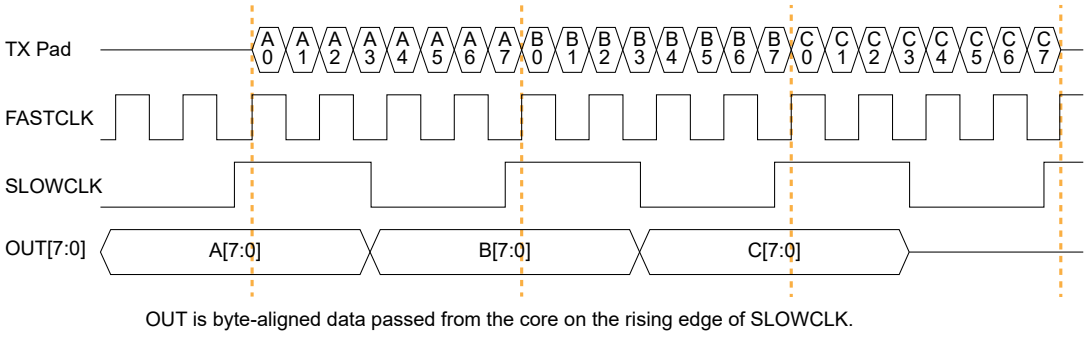


Table 18: LVDS TX Signals (Interface to FPGA Fabric)

Signal	Direction	Clock Domain	Description
OUT[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) 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 19: LVDS Timing Example Serialization Width of 8 (Half Rate)



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 20: LVDS Bidirectional Interface Block Diagram

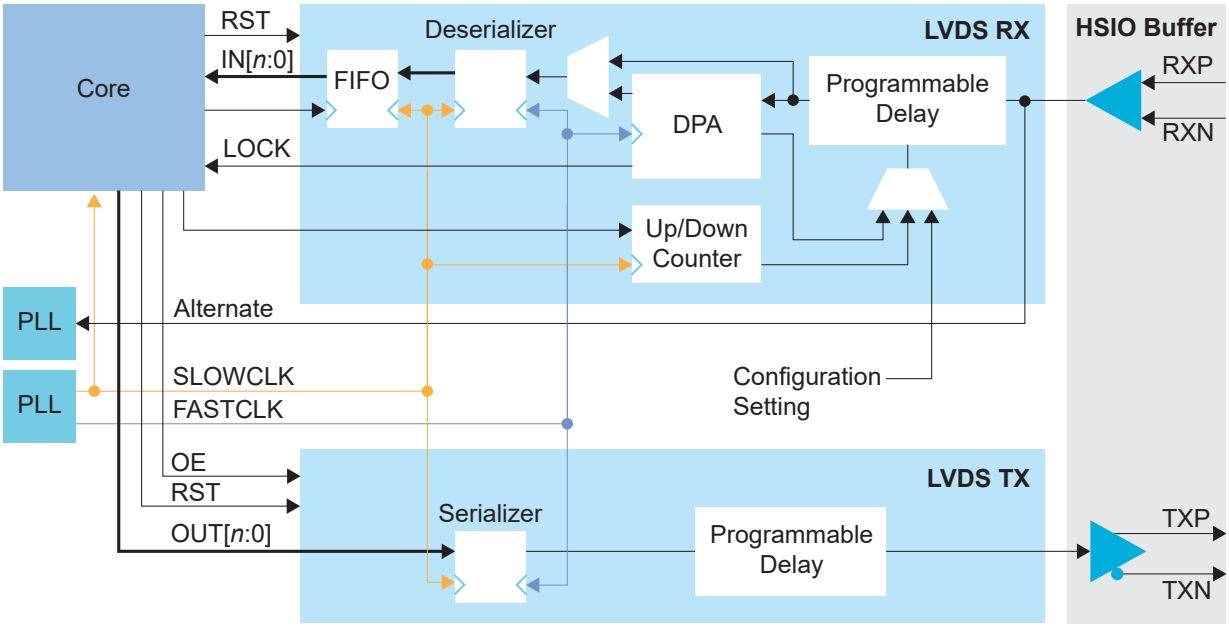


Table 19: LVDS Bidirectional Signals (Interface to FPGA Fabric)

Signal	Direction	Clock Domain	Description
IN[9:0]	Output	SLOWCLK	Parallel input data to the core. The width is programmable.
LOCK	Output	–	(Optional) When DPA is enabled, this signal indicates that the DPA has achieved training lock and data can be passed.
FIFO_EMPTY	Output	FIFOCLK	(Optional) When the FIFO is enabled, this signal indicates that the FIFO is empty.
INSLOWCLK	Input	–	Parallel (slow) clock for RX.
INFASTCLK	Input	–	Serial (fast) clock for RX.
FIFOCLK	Input	–	(Optional) Core clock to read from the FIFO.
FIFO_RD	Input	FIFOCLK	(Optional) Enables FIFO to read.
INRST	Input	FIFOCLK SLOWCLK	(Optional) Asynchronous. Resets the FIFO and RX serializer. 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	–	Enables or disables termination in dynamic termination mode. 1: Enabled 0: Disabled

Signal	Direction	Clock Domain	Description
DLY_ENA	Input	SLOWCLK	(Optional) Enable the dynamic delay control or the DPA circuit, depending on which one is enabled.
DLY_INC	Input	SLOWCLK	(Optional) Dynamic delay control. Cannot be used with DPA enabled. When DLY_ENA is 1, 1: Increments 0: Decrements
DLY_RST	Input	SLOWCLK	(Optional) Reset the delay counter or the DPA circuit, depending on which one is enabled.
OUT[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	(Optional) 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.

Important: When using HSIO pins as GPIO, make sure to leave at least 1 pair of unassigned HSIO pins between any GPIO and HSIO used as MIPI lanes in the same bank. This separation reduces noise. The Efinity software issues an error if you do not leave this separation.

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).

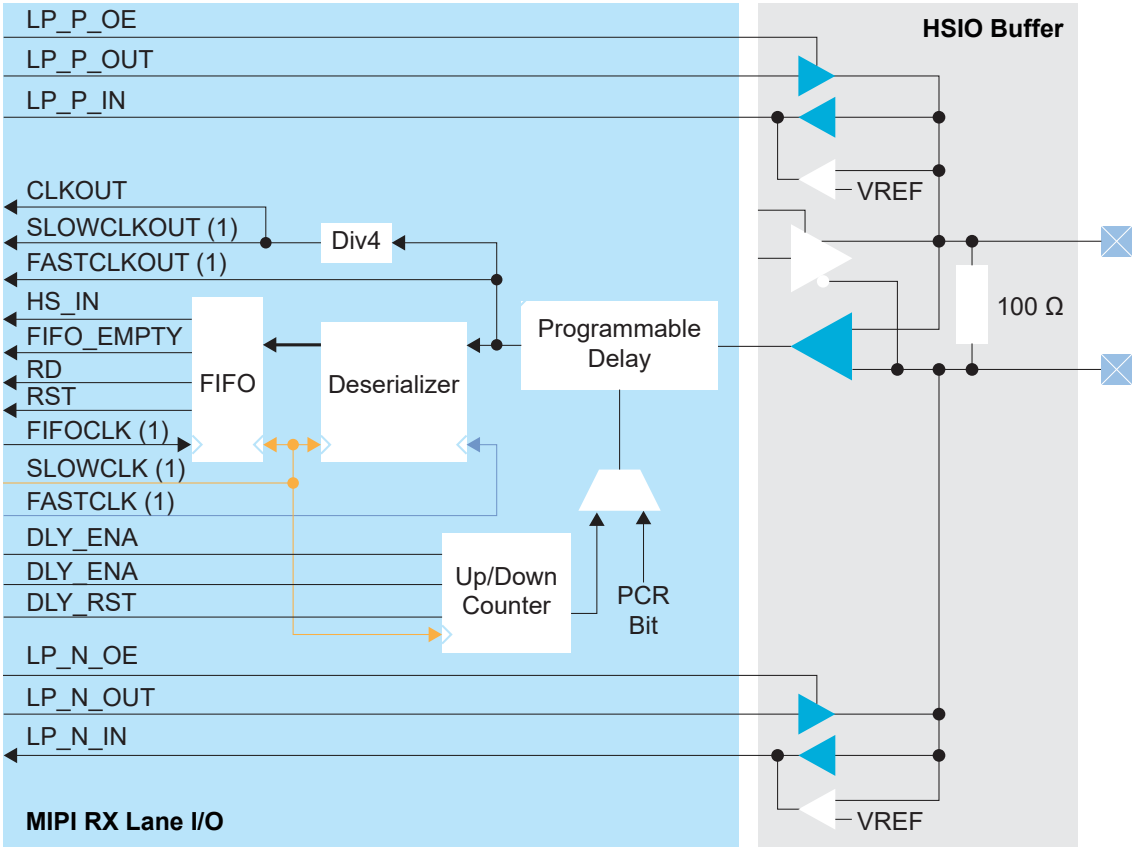
The MIPI RX function is defined as:

Table 21: MIPI RX Function

MIPI RX Function	Description
RX_DATA_*xy_zz*	MIPI RX Lane. You can use any data lanes within the same group to form multiple lanes of MIPI RX group. *x* = P or N *y* = 0 to 7 data lanes (Up to 8 data lanes) *zz* = I0 to I11 MIPI RX group (Up to 12 MIPI RX groups)
RX_CLK_*x_zz*	MIPI RX Clock Lane. *x* = P or N *zz* = I0 to I11 MIPI group

Learn more: Refer to the Ti60 Pinout Pinout for more information about the MIPI RX function for each HSIO and for which pins are in the same MIPI group.

Figure 21: MIPI RX Lane Block Diagram



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

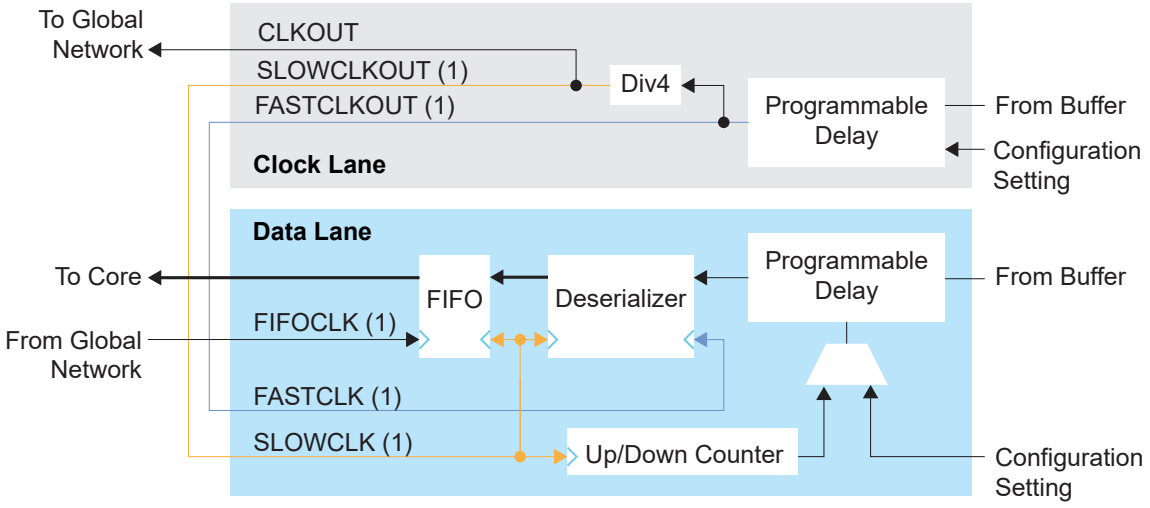
Signal	Direction	Clock Domain	Description
HS_IN[7:0]	Output	SLOWCLK	High-speed parallel data input.
LP_P_IN	Output	–	Low-power input data from the P pad.
LP_N_IN	Output	–	Low-power input data from the N pad.
LP_P_OUT	Input	–	(Optional) Low-power output data from the core for the P pad. Used if the data lane is reversible.
LP_N_OUT	Input	–	(Optional) Low-power output data from the core for the N pad. Used if the data lane is reversible.
FIFO_EMPTY	Output	FIFOCLK	(Optional) When the FIFO is enabled, this signal indicates that the FIFO is empty.
SLOWCLKOUT[2]	Output	–	Divided down parallel (slow) clock from the pads. Can only drive RX DATA lanes.
FASTCLKOUT[2]	Output	–	Serial (fast) clock from the pads. Can only drive RX DATA lanes.

[2] These signals are in the primitive, but the software automatically connects them for you.

Signal	Direction	Clock Domain	Description
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.
SLOWCLK(2)	Input	–	Parallel (slow) clock.
FASTCLK(2)	Input	–	Serial (fast) clock.
FIFOCLK(2)	Input	–	(Optional) Core clock to read from the FIFO.
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.
HS_ENA	Input	–	Dynamically enable the differential input buffer when in high-speed mode.
HS_TERM	Input	–	Dynamically enables input termination high-speed mode.
DLY_ENA	Input	SLOWCLK	(Optional) Enable the dynamic delay control.
DLY_INC	Input	SLOWCLK	(Optional) Dynamic delay control. When DLY_ENA is 1, 1: Increments 0: Decrements
DLY_RST	Input	SLOWCLK	(Optional) Reset the delay counter.

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

Figure 22: Connections for Clock and RX Data Lane



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 23: MIPI TX Lane Block Diagram

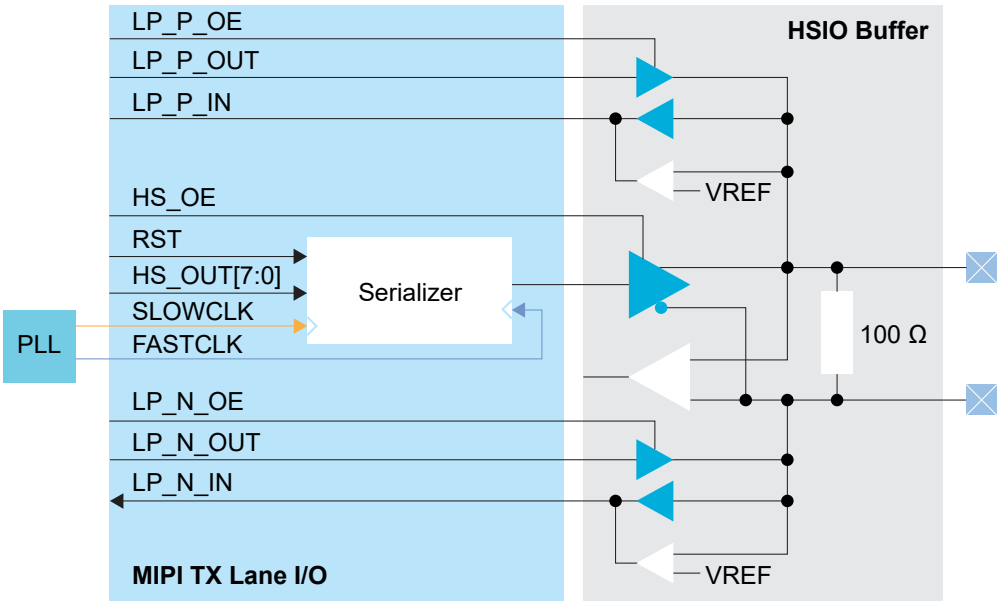


Table 23: MIPI TX Lane Signals

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

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

MIPI Lane Pads

Table 24: MIPI Lane Pads

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

I/O Banks

易灵思 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. Merged banks have underscores (_) between banks in the name (e.g., 1B_1C means 1B and 1C are connected).

Table 25: I/O Banks by Package

Package	I/O Banks	Voltage (V)	Banks with DDIO Support	Merged Banks
W64	1A, 1B, 2A, 3B	1.2, 1.5, 1.8	All	1A_4B, 1B_2A, 2A_3A_3B_4A
F100	1A, 2A, 2B, 4B	1.8	All	1A_4B, 2A_2B
	1B, 3A, 3B, 4A	1.2, 1.5, 1.8	All	3B_4A
	BL	1.8, 2.5, 3.0, 3.3	All	-
F225	BL, TL, TR, BR,	1.8, 2.5, 3.0, 3.3	All	-
	1A, 1B, 2A, 2B, 3A, 3B, 4A, 4B	1.2, 1.5, 1.8	All	-

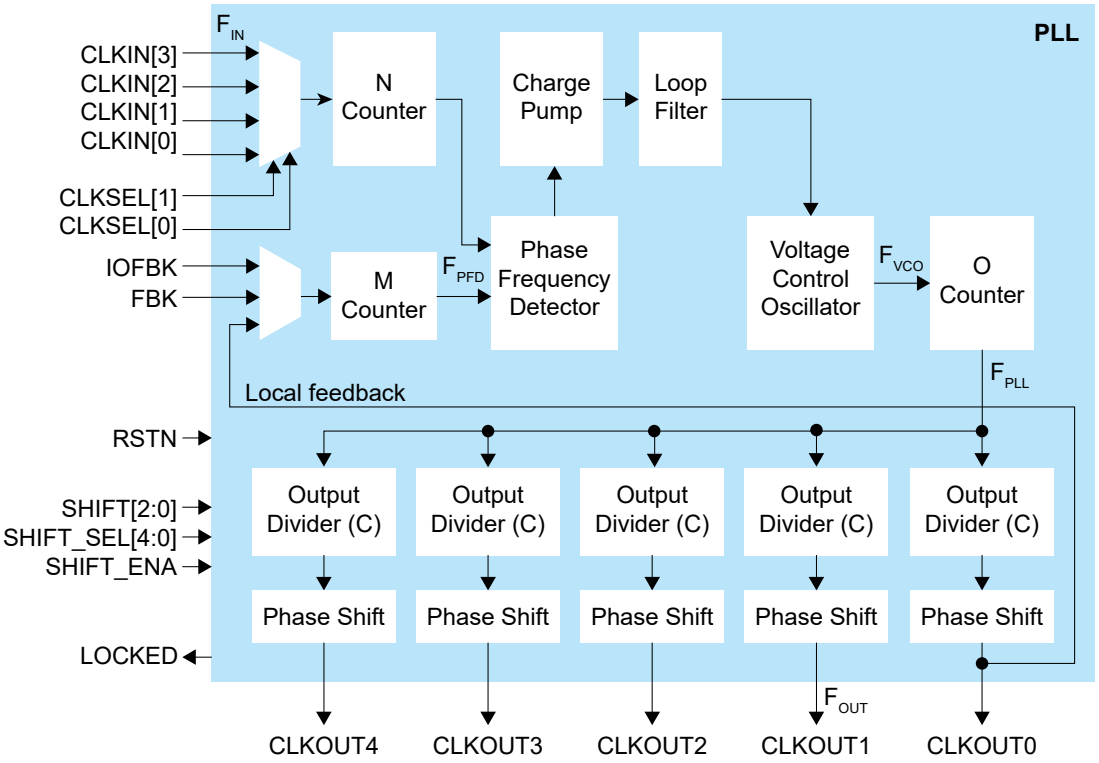
Oscillator

The Ti60 has one low-frequency oscillator tailored for low-power operation. The oscillator runs at a nominal frequency of 10, 20, 40, or 80 MHz. You can use the oscillator to perform always-on functions with the lowest power possible. It's output clock is available to the core. You can enable or disable the oscillator to allow power savings when not in use.

PLL

钛金系列 FPGAs have 4 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 application. 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.) 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).

Figure 24: PLL Block Diagram



The counter settings define the PLL output frequency:

Local and Core Feedback Mode	Where:
$F_{PFD} = F_{IN} / N$ $F_{VCO} = (F_{PFD} \times M \times O \times C_{FBK})$ [3] $F_{OUT} = (F_{IN} \times M \times C_{FBK}) / (N \times C)$	F_{VCO} is the voltage control oscillator frequency F_{OUT} is the output clock frequency F_{IN} is the reference clock frequency F_{PFD} is the phase frequency detector input frequency C is the output divider

Note: Refer to the PLL Timing and AC Characteristics on page 52 for F_{VCO}, F_{OUT}. F_{IN}, F_{PLL}, and F_{PFD} values.

[3] $(M \times O \times C_{FBK})$ must be ≤ 255.

Figure 25: PLL Interface Block Diagram

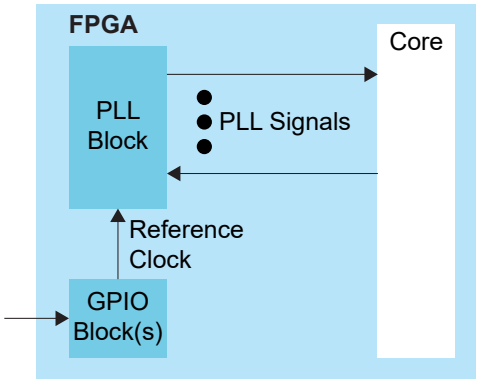


Table 26: PLL Signals (Interface to FPGA Fabric)

Signal	Direction	Description
CLKIN[3:0]	Input	Reference clocks driven by I/O pads or core clock tree.
CLKSEL[1:0]	Input	You can dynamically select the reference clock from one of the clock in pins.
RSTN	Input	Active-low PLL reset signal. When asserted, this signal resets the PLL; when de-asserted, it enables the PLL. 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 in core feedback mode.
IOFBK	Input	Connect to a clock out interface pin when the PLL is in external I/O feedback mode.
CLKOUT0 CLKOUT1 CLKOUT2 CLKOUT3 CLKOUT4	Output	PLL output. You can route these signals as input clocks to the core's GCLK network. CLKOUT4 can only feed the top or bottom regional clocks. All PLL outputs lock on the negative clock edge. The Interface Designer inverts the clock polarity on the leaf cells by default (**Invert Output Clock** option unchecked). You can use CLKOUT0 only for clocks with 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 clock for CLKOUT0.
LOCKED	Output	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.
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 VCO cycle delay). Each increment adds 0.5 VCO cycle delay.
SHIFT_SEL[4:0]	Input	(Optional) Choose the output(s) affected by the dynamic phase shift.
SHIFT_ENA	Input	(Optional) When high, changes the phase shift of the selected PLL(s) to the new value.

Table 27: PLL Reference Clock Resource Assignments (W64)

PLL	REFCLK0	REFCLK1	REFCLK2	External Feedback I/O
PLL_TL	Single-ended: GPIOL_P_18_PLLIN0 Differential: GPIOL_P_18_PLLIN0, GPIOL_N_18	Unbonded(4)	Unbonded(4)	Single-ended: GPIOL_P_17_EXTFB Differential: GPIOL_P_17_EXTFB, GPIOL_N_17
PLL_BR	Single-ended: GPIOR_P_00_PLLIN0 Differential: GPIOR_P_00_PLLIN0, GPIOR_N_00_CDI22	Unbonded(4)	Unbonded(4)	Single-ended: GPIOR_P_01_EXTFB Differential: GPIOR_P_01_EXTFB, GPIOR_N_01_CDI23

Table 28: PLL Reference Clock Resource Assignments (F100)

PLL	REFCLK0	REFCLK1	REFCLK2	External Feedback I/O
PLL_TL	Single-ended: GPIOL_P_18_PLLIN0 Differential: GPIOL_P_18_PLLIN0, GPIOL_N_18	Unbonded(4)	Unbonded(4)	Single-ended: GPIOL_P_17_EXTFB Differential: GPIOL_P_17_EXTFB, GPIOL_N_17
PLL_TR	Single-ended: GPIOR_P_19_PLLIN0 Differential: GPIOR_P_19_PLLIN0, GPIOR_N_19	Unbonded(4)	Unbonded(4)	Unbonded(4)
PLL_BR	Single-ended: GPIOR_P_00_PLLIN0 Differential: GPIOR_P_00_PLLIN0, GPIOR_N_00_CDI22	Unbonded(4)	Unbonded(4)	Single-ended: GPIOR_P_01_EXTFB Differential: GPIOR_P_01_EXTFB, GPIOR_N_01_CDI23

Table 29: PLL Reference Clock Resource Assignments (F225)

PLL	REFCLK0	REFCLK1	REFCLK2	External Feedback I/O
PLL_BL	Single-ended: GPIOL_P_00_PLLIN0 Differential: GPIOL_P_00_PLLIN0, GPIOL_N_00	Single-ended: GPIOB_P_00_PLLIN1 Differential: GPIOB_P_00_PLLIN1, GPIOB_N_00	Unbonded(4)	Single-ended: GPIOB_P_01_EXTFB Differential: GPIOB_P_01_EXTFB, GPIOB_N_01
PLL_TL	Single-ended: GPIOL_P_18_PLLIN0 Differential: GPIOL_P_18_PLLIN0, GPIOL_N_18	Single-ended: GPIOT_P_00_PLLIN1 Differential: GPIOT_P_00_PLLIN1 GPIOT_N_00	GPIOL_11_PLLIN2	Single-ended: GPIOL_P_17_EXTFB Differential: GPIOL_P_17_EXTFB, GPIOL_N_17
PLL_TR	Single-ended: GPIOR_P_19_PLLIN0 Differential: GPIOR_P_19_PLLIN0, GPIOR_N_19	Single-ended: GPIOT_P_17_PLLIN1 Differential: GPIOT_P_17_PLLIN1, GPIOT_N_17	Unbonded(4)	Single-ended: GPIOT_P_16_EXTFB Differential: GPIOT_P_16_EXTFB, GPIOT_N_16
PLL_BR	Single-ended: GPIOR_P_00_PLLIN0 Differential: GPIOR_P_00_PLLIN0, GPIOR_N_00_CDI22	Single-ended: GPIOB_P_17_PLLIN1 Differential: GPIOB_P_17_PLLIN1, GPIOB_N_17	GPIOR_29_PLLIN2	Single-ended: GPIOR_P_01_EXTFB Differential: GPIOR_P_01_EXTFB, GPIOR_N_01_CDI23

(4) There is no dedicated pin assigned to this reference clock.

Dynamic Phase Shift

Ti60 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 VCO cycles. For example, to phase shift a 400 MHz clock by 90-degree, configure the PLL to have a VCO frequency of 800 MHz, set the output counter division to 2, and set `SHIFT[2:0]` to `001`.

Implementing Dynamic Phase Shift

Use these steps to implement the dynamic phase shift:

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

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

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

Figure 26: Single Dynamic Phase Shift Waveform Example for CLKOUT1

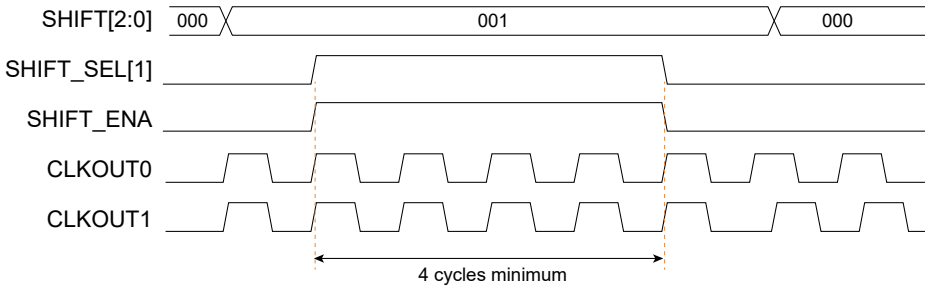
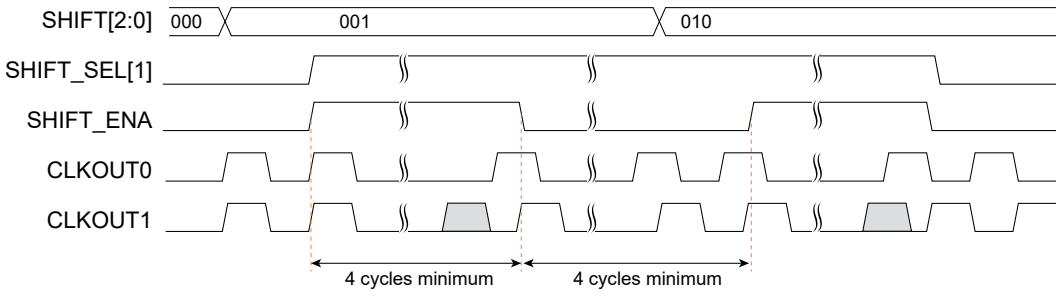


Figure 27: Consecutive Dynamic Phase Shift Waveform Example for CLKOUT1



SPI Flash Memory

Ti60 FPGAs in the F100 package include a SPI flash memory. The SPI flash memory has a density of 16 Mbits and a clock rate of up to 85 MHz. In active configuration mode, the FPGA is configured using the configuration bitstream in the SPI flash memory. Typically you can fit two compressed bitstream images into the F100 SPI flash.

Important: You cannot enable the Ti60 FPGA security features when using compressed bitstreams.

To use active programming mode for the Ti60 F100 with the SPI flash memory, you must use the SPI Active using JTAG Bridge configuration mode to configure the SPI flash memory.

Learn more: Refer to the **AN 033: Configuring 钛金系列 FPGAs** for information on programming the SPI flash memory.

Figure 28: SPI Flash Memory Block Diagram

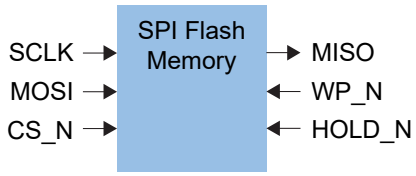


Table 30: SPI Flash Memory Signals (Interface to FPGA Fabric)

Signal	Direction	Description
SCLK	Input	Clock output to SPI flash memory.
MOSI	Input	Data output to SPI flash memory.
CS_N	Input	Active-low SPI flash memory chip select.
WP_N	Input	Active-low write protect signal.
HOLD_N	Input	Active-low hold signal.
MISO	Output	Data input from SPI flash memory.

HyperRAM

The Ti60 FPGA in F100 package includes a HyperRAM. The HyperRAM 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 29: HyperRAM Block Diagram

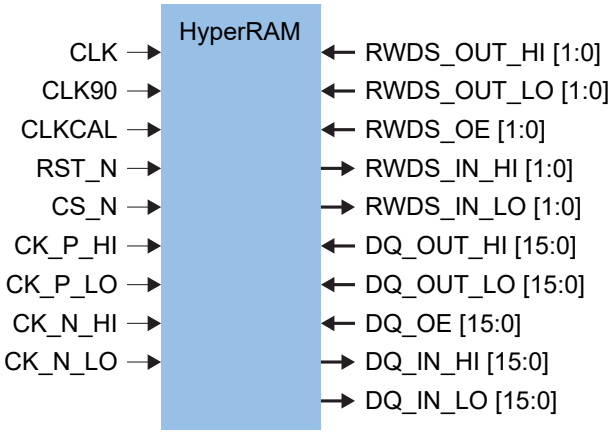


Table 31: HyperRAM Signals (Interface to FPGA Fabric)

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	

Single-Event Upset Detection

The Ti60 FPGA has a hard block for detecting single-event upset (SEU). The SEU detection feature has two modes:

- *Auto mode*—The Ti60 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 Ti60 when an error is detected.

Internal Reconfiguration Block

The Ti60 FPGAs have built-in hardware that supports an internal reconfiguration feature. The Ti60 can reconfigure itself from a bitstream image stored in flash memory.

Security Feature

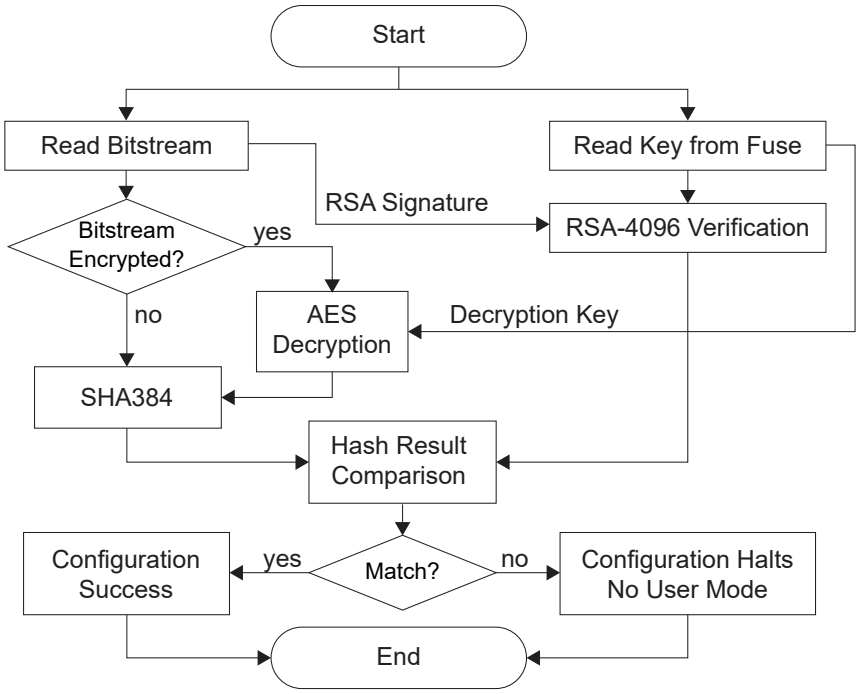
The Ti60 FPGA security feature includes:

- 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 Ti60 FPGA security features when using compressed bitstreams.

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

Figure 30: Security Flow



Bitstream Encryption

Symmetric bitstream encryption uses a 256-bit key and the AES-GCM-256 algorithm. You create the key and then use it to encrypt the bitstream. You also need to store the key into the FPGA's fuses. During configuration, the Ti60 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.

Advanced Security Setting

Most user applications should be sufficiently secure with bitstream encryption and authentication. However, some applications may have advanced higher-level security needs. To support these situations, Ti60 FPGAs support JTAG blocking. With this method, you can disable 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` and enabling the JTAG `BYPASS` mode.

By default, JTAG programming already has security built in. For example, the JTAG bitstream (**.bit**) automatically requires a valid signature when RSA authentication is enabled. So typical users should not need to disable JTAG access. Once you blow the fuse, you cannot use JTAG ever again in that FPGA (except for `IDCODE` and `BYPASS`). So blowing this fuse should be the very last step in your manufacturing process.

Power Up Sequence

You **must** use the following power up sequence when powering 钛金系列 FPGAs:

1. Power up `VCC` and `VCCA_xx` first.
2. When `VCC` and `VCCA_xx` are stable, power up all VCCIO pins. There is no specific timing delay between the VCCIO pins.
3. After all power supplies are stable, hold `CRESET_N` low for a duration of t_{CRESET_N} before asserting `CRESET_N` from low to high to trigger active SPI programming (the FPGA loads the configuration data from an external flash device).

Note: Refer to Configuration Timing on page 53 for timing information.

Figure 31: Power Up Sequence

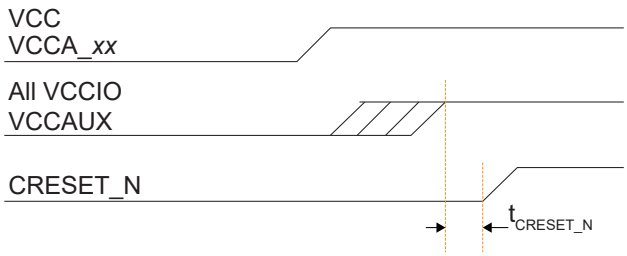


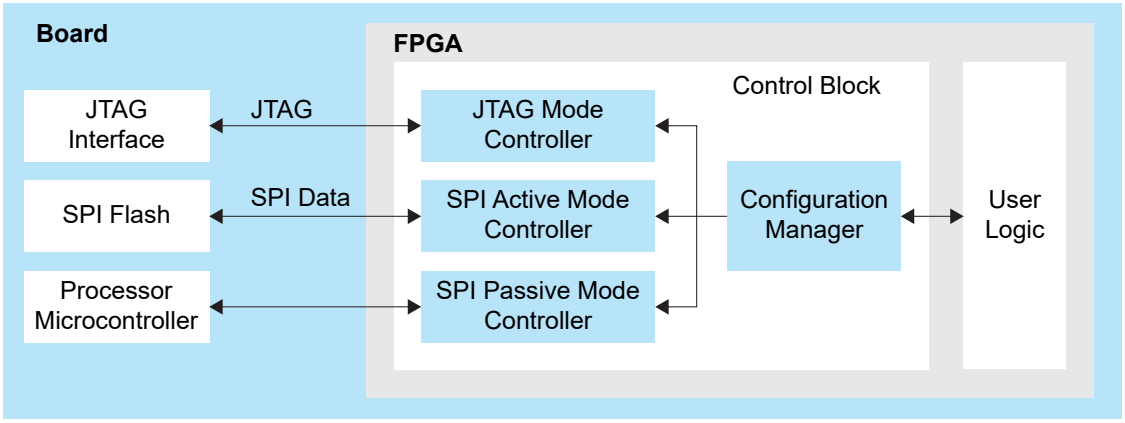
Table 32: Connection Requirements for Unused Resources

Unused Resource	Pin	Note
HSIO Bank	VCCIOxx	Connect to either 1.2 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.
PLL	VCCA_PLL	Connect to VCC.

Configuration

The Ti60 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 Ti60 FPGA(s) in active, passive, or JTAG mode.

Figure 32: High-Level Configuration Options



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

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

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 FPGA Configuration Modes

Table 33: Ti60 Configuration Modes by Package

Configuration Mode	Width	W64	F100	F225
Active	X1	✓	✓	✓
	X2	✓	✓	✓
	X4	✓	-	✓
	X8	-	-	✓
Passive	X1	✓	✓	✓
	X2	✓	✓	✓
	X4	✓	-	✓
	X8	-	-	✓
	X16	-	-	✓[5]
	X32	-	-	✓[5]
JTAG	X1	✓	✓	✓

[5] Not supported when security mode is enabled.

Characteristics and Timing

DC and Switching Characteristics

Important: All specifications are preliminary and pending hardware characterization.

Table 34: Absolute Maximum Ratings

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
VCCIO	HSIO bank power supply	-0.5	1.98	V
VCCA_PLL	PLL analog power supply	-0.5	1.05	V
VCCAUX	1.8 V auxiliary power supply	-0.5	1.98	V
VCCIO33	HVIO bank power supply	-0.5	3.6	V
T_J	Operating junction temperature	-40	125	°C

Table 35: Recommended Operating Conditions (C3, C4, and I4 Speed Grades)[6]

Symbol	Description	Min	Typ	Max	Units
VCC	Core power supply	0.92	0.95	0.98	V
VCCIO	1.2 V I/O bank power supply	1.14	1.2	1.26	V
	1.5 V I/O bank power supply	1.425	1.5	1.575	V
	1.8 V I/O bank power supply	1.71	1.8	1.89	V
VCCIO33	1.8 V I/O bank power supply	1.71	1.8	1.89	V
	2.5 V I/O bank power supply	2.375	2.5	2.625	V
	3.0 V I/O bank power supply	2.85	3.0	3.15	V
	3.3 V I/O bank power supply	3.135	3.3	3.465	V
VCCA_PLL	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
T_{JCOM}	Operating junction temperature, commercial	0	–	85	°C
T_{JIND}	Operating junction temperature, industrial	-40	–	100	°C

[6] Supply voltage specification applied to the voltage taken at the device pins with respect to ground, not at the power supply.

Table 36: Recommended Operating Conditions (C3L, I4L, and C4L Speed Grades)[6]

Symbol	Description	Min	Typ	Max	Units
VCC	Core power supply	0.82	0.85	0.88	V
VCCIO	1.2 V I/O bank power supply	1.14	1.2	1.26	V
	1.5 V I/O bank power supply	1.425	1.5	1.575	V
	1.8 V I/O bank power supply	1.71	1.8	1.89	V
VCCIO33	1.8 V I/O bank power supply	1.71	1.8	1.89	V
	2.5 V I/O bank power supply	2.375	2.5	2.625	V
	3.0 V I/O bank power supply	2.85	3.0	3.15	V
	3.3 V I/O bank power supply	3.135	3.3	3.465	V
VCCA_PLL	PLL analog power supply	0.82	0.85	0.88	V
VCCAUX	1.8 V auxiliary power supply	1.75	1.8	1.85	V
T_{JCOM}	Operating junction temperature, commercial	0	–	85	°C

Table 37: Power Supply Ramp Rates

Symbol	Description	Min	Max	Units
t_{RAMP}	Power supply ramp rate for all supplies.	0.01	10	V/ms

Table 38: 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	0.7	(7)	(7)	0.4	2.0
1.8 V LVCMOS	-0.3	0.58	1.27	1.89	0.45	VCCIO33 - 0.45

Table 39: HVIO DC Electrical Characteristics

Voltage (V)	Typical Hysteresis (mV)	Input Leakage Current (µA)	Tristate Output Leakage Current (µA)
3.3	250	±10	±10
2.5	(8)	±10	±10
1.8	200	±10	±10

(7) Pending hardware characterization.
(8) Pending hardware characterization.

Table 40: 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
1.2 V SSTL	–	VREF - 0.1	VREF + 0.1	–	0.2 * VCCIO	0.8 * VCCIO

Table 41: 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	0.8333	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
1.2 V SSTL	0.49 * VCCIO	0.5 * VCCIO	0.51 * VCCIO	0.49 * VCCIO	0.5 * VCCIO	0.51 * VCCIO

Table 42: 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	Max	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	–
1.2 V SSTL	0.18	–	VREF– 0.15	VCCIO /2	VREF + 0.15	-0.3	0.3

Table 43: 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 44: HSIO Pins Configured as Single-Ended I/O DC Electrical Characteristics

Voltage (V)	Typical Hysteresis (mV)	Input Leakage Current (µA)	Tristate Output Leakage Current (µA)
1.8	200	±10	±10
1.5	160	±10	±10
1.2	140	±10	±10

Table 45: Maximum Toggle Rate

I/O	I/O Standard	Test Condition Load (pF)	Max Toggle Rate (Mbps)
HVIO	3.0 V / 3.3 V LVTTL, 3.0 V / 3.3 V LVCMOS	40	100
HVIO	2.5 V LVCMOS	40	100
HVIO	1.8 V LVCMOS	20	400
HSIO	1.8 V / 1.5 V / 1.2 V LVCMOS	20	400
HSIO	1.8 V / 1.5 V / 1.2 V SSTL, 1.8 V / 1.5 V / 1.2 V HSTL	5	800
HSIO	LVDS	5	1600
HSIO	Sub-LVDS	5	1250
HSIO	MIPI lane	5	1500

Table 46: Block RAM Characteristics

Symbol	Description	C4, I4 Speed Grade	C3 Speed Grade	C4L, I4L Speed Grade	Units
f_{MAX}	Block RAM maximum frequency.	1000	(9)	(9)	MHz

Table 47: DSP Block Characteristics

Symbol	Description	C4, I4 Speed Grade	C3 Speed Grade	C4L, I4L Speed Grade	Units
f_{MAX}	DSP block maximum frequency.	1000	(9)	(9)	MHz

(9) Pending hardware characterization.

HSIO Electrical and Timing Specifications

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

Important: All specifications are preliminary and pending hardware characterization.

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

Table 48: 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 49: HSIO Timing Specifications when Configured as LVDS

Parameter	Description	Min	Typ	Max	Unit
t_{LVDS_DT}	LVDS TX reference clock output duty cycle	45	50	55	%
t_{LVDS_skew}	LVDS TX lane-to-lane skew	–	200	–	ps

Table 50: 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 51: 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 52: 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 53: 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 54: 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 55: 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 56: HSIO DC Specifications when Configured as Low-Power MIPI RX Lane

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

PLL Timing and AC Characteristics

The following tables describe the PLL timing and AC characteristics.

Important: All specifications are preliminary and pending hardware characterization.

Table 57: 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 58: PLL AC Characteristics[10]

Symbol	Parameter	Min	Typ	Max	Units
t_{DT}	Output clock duty cycle.	45	50	55	%
t_{OPJIT} (PK - PK) (11)	Output clock period jitter (PK-PK).	–	–	200	ps
t_{LOCK}	PLL lock-in time.	–	300	500	PFD[12]

(10) Test conditions at 3.3 V and room temperature.
(11) The output jitter specification applies to the PLL jitter when an input jitter of 20 ps is applied.
(12) PFD cycle equals to reference clock division divided by reference clock frequency.

Configuration Timing

Important: All specifications are preliminary and pending hardware characterization.

The Ti60 FPGA has the following configuration timing specifications.

Note: Refer to **AN 033: Configuring 钛金系列 FPGAs** for detailed configuration information.

Timing Waveforms

Figure 33: SPI Active Mode (x1) Timing Sequence

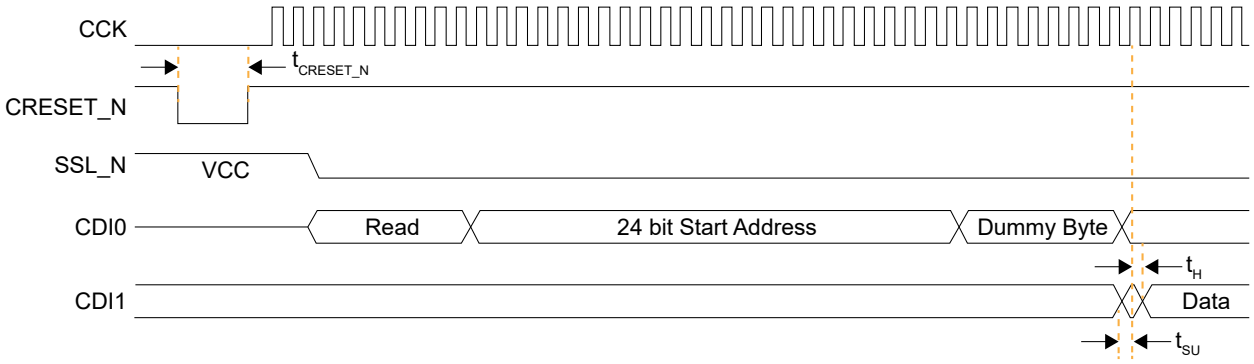


Figure 34: SPI Passive Mode (x1) Timing Sequence

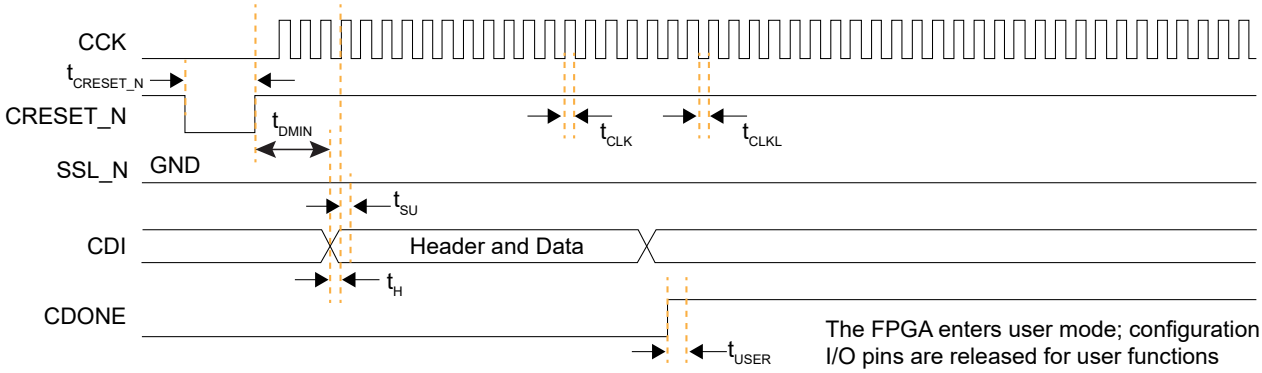
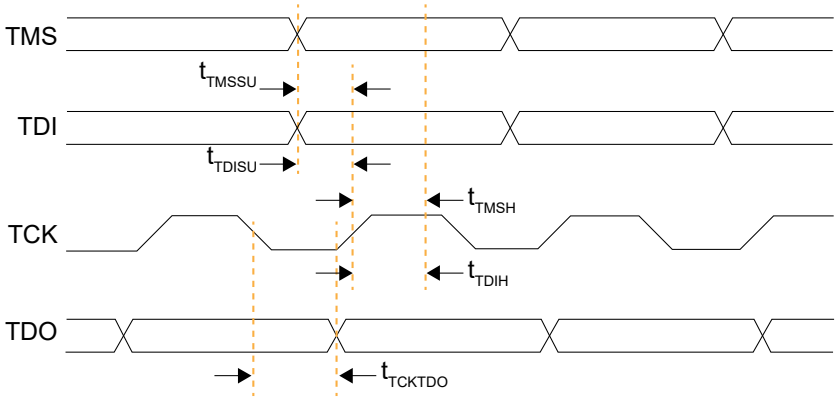


Figure 35: Boundary-Scan Timing Waveform



Timing Parameters

Table 59: 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.[13][14] Test condition at 10 kΩ pull-up resistance and 10 pF output loading on CDONE pin.	25	–	–	µs

Table 60: Active Mode

Symbol	Parameter	Frequency	Min	Typ	Max	Units
f_{MAX_M}	Active mode internal configuration clock frequency.	DIV1	52	80	104	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.	–	[15]	[15]	[15]	MHz
t_{SU}	Setup time. Test condition at 1.8 V I/O standard and 0 pF output loading.	–	3.5	–	–	ns
t_H	Hold time. Test condition at 1.8 V I/O standard and 0 pF output loading.	–	2	–	–	ns

Table 61: Passive Mode

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.	5	–	–	ns
t_H	Hold time.	1	–	–	ns
t_{DMIN}	Minimum time between deassertion of CRESET_N to first valid configuration data.	32	–	–	µs

[13] The FPGA may go into user mode before t_{USER} has elapsed. However, 易灵思 recommends that you keep the system interface to the FPGA in reset until t_{USER} has elapsed.

[14] 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).

[15] Pending hardware characterization.

Table 62: JTAG Mode

Symbol	Parameter	Min	Typ	Max	Units
f_{TCK}	TCK frequency.	–	–	30	MHz
t_{TDISU}	TDI setup time.	3.5	–	–	ns
t_{TDIH}	TDI hold time.	2.5	–	–	ns
t_{TMSSU}	TMS setup time.	3	–	–	ns
t_{TMSH}	TMS hold time.	2.5	–	–	ns
t_{TCKTDO}	TCK falling edge to TDO output.	–	–	10	ns

Pinout Description

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

Table 63: Power and Ground Pinouts

xx indicates the bank location.

Function	Description
VCC	Core power supply.
VCCA_*xx*	PLL analog power supply.
VCCAUX_*xx*	1.8 V auxiliary power supply.
VCCIO33_*xx*	HVIO bank power supply.
VCCIO*xx*	HSIO bank power supply.
VCCIO*xx*_*yy*_*zz*	Power for HSIO banks that are shorted together. *xx*, *yy*, and *zz* are the bank locations. For example: VCCIO1B_1C shorts banks 1B and 1C
GND	Ground.

Table 64: 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*	–	Connect a 10 kΩ resistor with a tolerance of ±0.1% to the REF_RES pin with respect to ground.

Table 65: Alternate Function Pinouts

n is the number.

Function	Direction	Description
CLK*n*	Input	Global clock and control network resource. The number of inputs is package dependent.
PLLIN*n*	Input	PLL reference clock resource. The number of reference clock resources is package dependent.

Table 66: Dedicated Configuration Pins

These pins cannot be used as general-purpose I/O after configuration.

Pins	Direction	Description	Use External Weak Pull-Up During Configuration
CDONE	Output	Configuration done status pin. CDONE is an open drain output; connect it to an external pull-up resistor to VCCIO. When CDONE = 1, configuration is complete. If you hold CDONE low, the device will not enter user mode.	✓
CRESET_N	Input	Initiates FPGA re-configuration (active low). Pulse CRESET_N low for a duration of t_{creset_N} before asserting CRESET_N from low to high to initiate FPGA re-configuration. This pin does not perform a system reset.	✓
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.	✓
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.	✓
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.	✓
TDO	Output	JTAG test data output (TDO). This serial output from the test logic is fed from the instruction register or from a test data register depending on the sequence previously applied at TMS. During shifting, data applied at TDI appears at TDO after a number of cycles of TCK determined by the length of the register included in the serial path. 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).	✓

Table 67: Dual-Purpose Configuration Pins

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

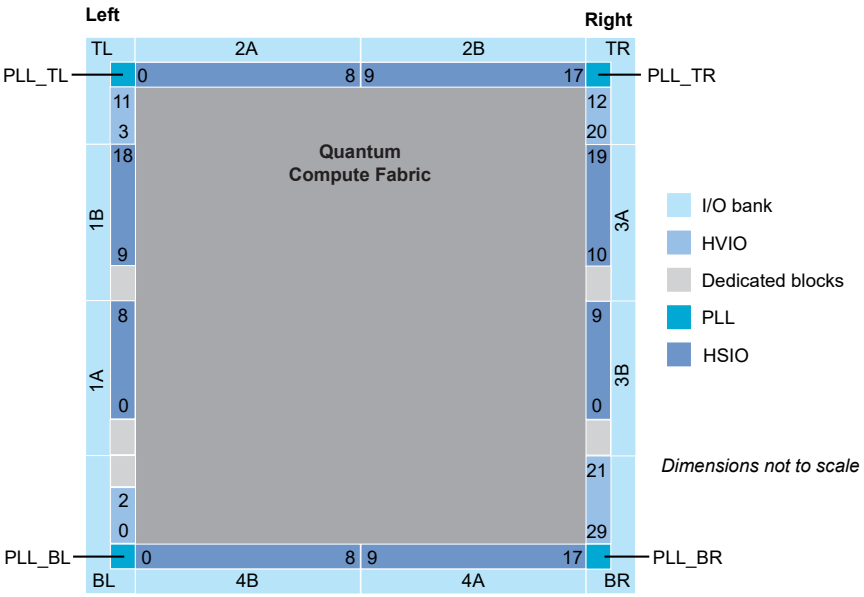
Pins	Direction	Description	Use External Weak Pull-Up During Configuration
CBSEL[1:0]	Input	Optional multi-image selection input (if external multi-image configuration mode is enabled).	N/A
CCK	I/O	Passive SPI input configuration clock or active SPI output configuration clock.	N/A
CDI*n*	I/O	*n* is a number from 0 to 31 depending on the SPI configuration. 0: Passive serial data input or active serial output. 1: Passive serial data output or active serial input. *n*: Parallel I/O. In multi-bit daisy chain connection, the CDI*n* (31:0) connects to the data bus in parallel.	N/A
CSI	Input	Chip select. 0: The FPGA is not selected or enabled and will not be configured. 1: Selects the FPGA for configuration.	✓
CSO	Output	Chip select output. Selects the next device for cascading configuration. [16]	N/A
NSTATUS	Output	Status (active low). Indicates a configuration error. When the FPGA drives this pin low, it indicates an ID mismatch, the bitstream CRC check has failed, or remote update has failed.	N/A
SSL_N	Input	Active-low configuration mode select. The FPGA senses the value of SSL_N when it comes out of reset (pulse CRESET_N low to high). 0: Passive mode 1: Active mode In active configuration mode, SSL_N serves as a chip select to the flash device 1 (CDI0 - CDI3).	✓
SSU_N	Input	In active configuration mode (dual quad mode), SSU_N serves as a chip select to the flash device 2 (CDI4 - CDI7).	✓
EXT_CONFIG_CLK	I/O	In active mode, EXT_CONFIG_CLK pin is connected from an external clock, to be used as a configuration clock.	N/A
TEST_N	Input	Active-low test mode enable signal. Set to 1 to disable test mode. During configuration, rely on the external weak pull-up or drive this pin high.	✓

[16] Cascaded configuration is not supported in the F100 package.

Ti60 Interface Floorplan

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

Figure 36: Floorplan Diagram for Ti60



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 Ti60 FPGA. The software supports the Verilog HDL and VHDL languages.

Revision History

Table 68: Revision History

Date	Version	Description
October 2021	1.1	Updated the Security Feature topic. Updated DIV 2 active mode f_{MAX_M} typical and maximum values. (DOC-509) Added F_{PLL} specs and updated the PLL block diagram to include F_{PLL}. (DOC-512) Added note stating that all PLL outputs lock on the negative clock edge. (DOC-511) Updated sub-LVDS maximum toggle rate. (DOC-541) Added description about CLKOUT0 limitation. (DOC-533) Updated HyperRAM double-data rates to up to 500 Mbps. (DOC-554) Added connection requirements for unused resources in Power Up Sequence topic.
June 2021	1.0	Initial release.