AMD Xilinx - XC3S1200E-5FTG256C Datasheet

Welcome to E-XFL.COM

Understanding Embedded - FPGAs (Field Programmable Gate Array)

Embedded - FPGAs, or Field Programmable Gate Arrays, are advanced integrated circuits that offer unparalleled flexibility and performance for digital systems. Unlike traditional fixed-function logic devices, FPGAs can be programmed and reprogrammed to execute a wide array of logical operations, enabling customized functionality tailored to specific applications. This reprogrammability allows developers to iterate designs quickly and implement complex functions without the need for custom hardware.

Applications of Embedded - FPGAs

The versatility of Embedded - FPGAs makes them indispensable in numerous fields. In telecommunications.

Details	
Product Status	Active
Number of LABs/CLBs	2168
Number of Logic Elements/Cells	19512
Total RAM Bits	516096
Number of I/O	190
Number of Gates	1200000
Voltage - Supply	1.14V ~ 1.26V
Mounting Type	Surface Mount
Operating Temperature	0°C ~ 85°C (TJ)
Package / Case	256-LBGA
Supplier Device Package	256-FTBGA (17x17)
Purchase URL	https://www.e-xfl.com/product-detail/xilinx/xc3s1200e-5ftg256c

Storage Element Functions

There are three pairs of storage elements in each IOB, one pair for each of the three paths. It is possible to configure each of these storage elements as an edge-triggered D-type flip-flop (FD) or a level-sensitive latch (LD).

The storage-element pair on either the Output path or the Three-State path can be used together with a special multiplexer to produce Double-Data-Rate (DDR) transmission. This is accomplished by taking data synchronized to the clock signal's rising edge and converting it to bits synchronized on both the rising and the falling edge. The combination of two registers and a multiplexer is referred to as a Double-Data-Rate D-type flip-flop (ODDR2).

Table 4 describes the signal paths associated with the storage element.

Table 4: **Storage Element Signal Description**

Storage Element Signal	Description	Function
D	Data input	Data at this input is stored on the active edge of CK and enabled by CE. For latch operation when the input is enabled, data passes directly to the output Q.
Q	Data output	The data on this output reflects the state of the storage element. For operation as a latch in transparent mode, Q mirrors the data at D.
CK	Clock input	Data is loaded into the storage element on this input's active edge with CE asserted.
CE	Clock Enable input	When asserted, this input enables CK. If not connected, CE defaults to the asserted state.
SR	Set/Reset input	This input forces the storage element into the state specified by the SRHIGH/SRLOW attributes. The SYNC/ASYNC attribute setting determines if the SR input is synchronized to the clock or not. If both SR and REV are active at the same time, the storage element gets a value of 0.
REV	Reverse input	This input is used together with SR. It forces the storage element into the state opposite from what SR does. The SYNC/ASYNC attribute setting determines whether the REV input is synchronized to the clock or not. If both SR and REV are active at the same time, the storage element gets a value of 0.

As shown in Figure 5, the upper registers in both the output and three-state paths share a common clock. The OTCLK1 clock signal drives the CK clock inputs of the upper registers on the output and three-state paths. Similarly, OTCLK2 drives the CK inputs for the lower registers on the output and three-state paths. The upper and lower registers on the input path have independent clock lines: ICLK1 and ICLK2.

The OCE enable line controls the CE inputs of the upper and lower registers on the output path. Similarly, TCE controls the CE inputs for the register pair on the three-state path and ICE does the same for the register pair on the input path.

The Set/Reset (SR) line entering the IOB controls all six registers, as is the Reverse (REV) line.

In addition to the signal polarity controls described in IOB Overview, each storage element additionally supports the controls described in Table 5.

Table 5: **Storage Element Options**

Option Switch	Function	Specificity
FF/Latch	Chooses between an edge-triggered flip-flop or a level-sensitive latch	Independent for each storage element
SYNC/ASYNC	Determines whether the SR set/reset control is synchronous or asynchronous	Independent for each storage element
SRHIGH/SRLOW	Determines whether SR acts as a Set, which forces the storage element to a logic 1 (SRHIGH) or a Reset, which forces a logic 0 (SRLOW)	Independent for each storage element, except when using ODDR2. In the latter case, the selection for the upper element will apply to both elements.
INIT1/INIT0	When Global Set/Reset (GSR) is asserted or after configuration this option specifies the initial state of the storage element, either set (INIT1) or reset (INIT0). By default, choosing SRLOW also selects INIT0; choosing SRHIGH also selects INIT1.	Independent for each storage element, except when using ODDR2, which uses two IOBs. In the ODDR2 case, selecting INIT0 for one IOBs applies to both elements within the IOB, although INIT1 could be selected for the elements in the other IOB.

Digital Clock Managers (DCMs)

For additional information, refer to the “Using Digital Clock Managers (DCMs)” chapter in UG331.

Differences from the Spartan-3 Architecture

- Spartan-3E FPGAs have two, four, or eight DCMs, depending on device size.
- The variable phase shifting feature functions differently on Spartan-3E FPGAs than from Spartan-3 FPGAs.
- The Spartan-3E DLLs support lower input frequencies, down to 5 MHz. Spartan-3 DLLs support down to 18 MHz.

Overview

Spartan-3E FPGA Digital Clock Managers (DCMs) provide flexible, complete control over clock frequency, phase shift and skew. To accomplish this, the DCM employs a Delay-Locked Loop (DLL), a fully digital control system that uses feedback to maintain clock signal characteristics with a high degree of precision despite normal variations in operating temperature and voltage. This section provides a fundamental description of the DCM.

The XC3S100E FPGA has two DCMs, one at the top and one at the bottom of the device. The XC3S250E and XC3S500E FPGAs each include four DCMs, two at the top and two at the bottom. The XC3S1200E and XC3S1600E FPGAs contain eight DCMs with two on each edge (see also Figure 45). The DCM in Spartan-3E FPGAs is surrounded by CLBs within the logic array and is no longer located at the top and bottom of a column of block RAM as in the Spartan-3 architecture. The Digital Clock Manager is instantiated within a design using a “DCM” primitive.

The DCM supports three major functions:

- **Clock-skew Elimination:** Clock skew within a system occurs due to the different arrival times of a clock signal at different points on the die, typically caused by the clock signal distribution network. Clock skew increases setup and hold time requirements and increases clock-to-out times, all of which are undesirable in high frequency applications. The DCM eliminates clock skew by phase-aligning the output clock signal that it generates with the incoming clock signal. This mechanism effectively cancels out the clock distribution delays.
- **Frequency Synthesis:** The DCM can generate a wide range of different output clock frequencies derived from the incoming clock signal. This is accomplished by either multiplying and/or dividing the frequency of the input clock signal by any of several different factors.
- **Phase Shifting:** The DCM provides the ability to shift the phase of all its output clock signals with respect to the input clock signal.

Although a single design primitive, the DCM consists of four interrelated functional units: the Delay-Locked Loop (DLL), the Digital Frequency Synthesizer (DFS), the Phase Shifter (PS), and the Status Logic. Each component has its associated signals, as shown in Figure 40.

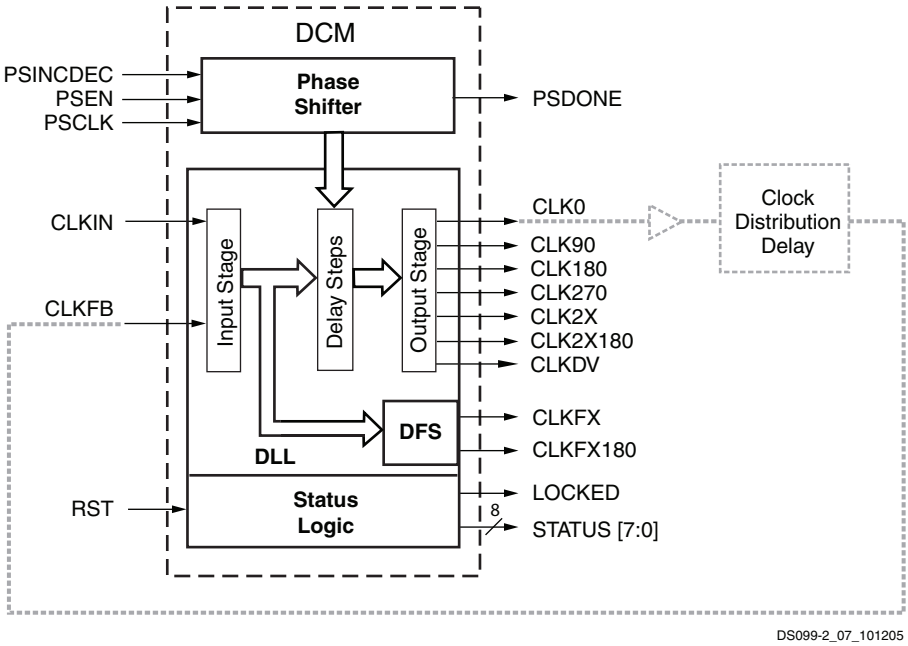


Figure 40: **DCM Functional Blocks and Associated Signals**

Delay-Locked Loop (DLL)

The most basic function of the DLL component is to eliminate clock skew. The main signal path of the DLL consists of an input stage, followed by a series of discrete delay elements or *steps,* which in turn leads to an output stage. This path together with logic for phase detection and control forms a system complete with feedback as shown in Figure 41. In Spartan-3E FPGAs, the DLL is implemented using a counter-based delay line.

The DLL component has two clock inputs, CLKIN and CLKFB, as well as seven clock outputs, CLK0, CLK90, CLK180, CLK270, CLK2X, CLK2X180, and CLKDV as described in Table 28. The clock outputs drive simultaneously. Signals that initialize and report the state of the DLL are discussed in Status Logic.

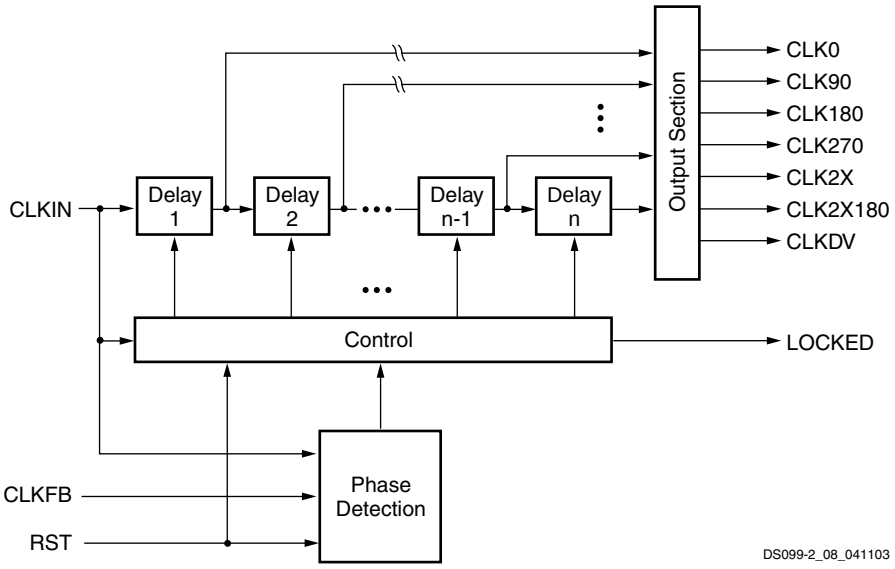


Figure 41: **Simplified Functional Diagram of DLL**

Table 28: **DLL Signals**

Signal	Direction	Description
CLKIN	Input	Receives the incoming clock signal. See Table 30, Table 31, and Table 32 for optimal external inputs to a DCM.
CLKFB	Input	Accepts either CLK0 or CLK2X as the feedback signal. (Set the CLK_FEEDBACK attribute accordingly).
CLK0	Output	Generates a clock signal with the same frequency and phase as CLKIN.
CLK90	Output	Generates a clock signal with the same frequency as CLKIN, phase-shifted by 90°.
CLK180	Output	Generates a clock signal with the same frequency as CLKIN, phase-shifted by 180°.
CLK270	Output	Generates a clock signal with the same frequency as CLKIN, phase-shifted by 270°.
CLK2X	Output	Generates a clock signal with the same phase as CLKIN, and twice the frequency.
CLK2X180	Output	Generates a clock signal with twice the frequency of CLKIN, and phase-shifted 180° with respect to CLK2X.
CLKDV	Output	Divides the CLKIN frequency by CLKDV_DIVIDE value to generate lower frequency clock signal that is phase-aligned to CLKIN.

The clock signal supplied to the CLKIN input serves as a reference waveform. The DLL seeks to align the rising-edge of feedback signal at the CLKFB input with the rising-edge of CLKIN input. When eliminating clock skew, the common approach to using the DLL is as follows: The CLK0 signal is passed through the clock distribution network that feeds all the registers it synchronizes. These registers are either internal or external to the FPGA. After passing through the clock distribution network, the clock signal returns to the DLL via a feedback line called CLKFB. The control block inside the DLL measures the phase error between CLKFB and CLKIN. This phase error is a measure of the clock skew that the clock distribution network introduces. The control block activates the appropriate number of delay steps to

DLL Clock Output and Feedback Connections

As many as four of the nine DCM clock outputs can simultaneously drive four of the BUFGMUX buffers on the same die edge. All DCM clock outputs can simultaneously drive general routing resources, including interconnect leading to OBUF buffers.

The feedback loop is essential for DLL operation. Either the CLK0 or CLK2X outputs feed back to the CLKFB input via a BUFGMUX global buffer to eliminate the clock distribution delay. The specific BUFGMUX buffer used to feed back the CLK0 or CLK2X signal is ideally one of the BUFGMUX buffers associated with a specific DCM, as shown in Table 30, Table 31, and Table 32.

The feedback path also phase-aligns the other seven DLL outputs: CLK0, CLK90, CLK180, CLK270, CLKDV, CLK2X, or CLK2X180. The CLK_FEEDBACK attribute value must agree with the physical feedback connection. Use “1X” for CLK0 feedback and “2X” for CLK2X feedback. If the DFS unit is used stand-alone, without the DLL, then no feedback is required and set the CLK_FEEDBACK attribute to “NONE”.

Two basic cases determine how to connect the DLL clock outputs and feedback connections: on-chip synchronization and off-chip synchronization, which are illustrated in Figure 42a through Figure 42d.

In the on-chip synchronization case in Figure 42a and Figure 42b, it is possible to connect any of the DLL’s seven output clock signals through general routing resources to the FPGA’s internal registers. Either a Global Clock Buffer (BUFG) or a BUFGMUX affords access to the global clock network. As shown in Figure 42a, the feedback loop is created by routing CLK0 (or CLK2X) in Figure 42b to a global clock net, which in turn drives the CLKFB input.

In the off-chip synchronization case in Figure 42c and Figure 42d, CLK0 (or CLK2X) plus any of the DLL’s other output clock signals exit the FPGA using output buffers (OBUF) to drive an external clock network plus registers on the board. As shown in Figure 42c, the feedback loop is formed by feeding CLK0 (or CLK2X) in Figure 42d back into the FPGA, then to the DCM’s CLKFB input via a Global Buffer Input, specified in Table 30.

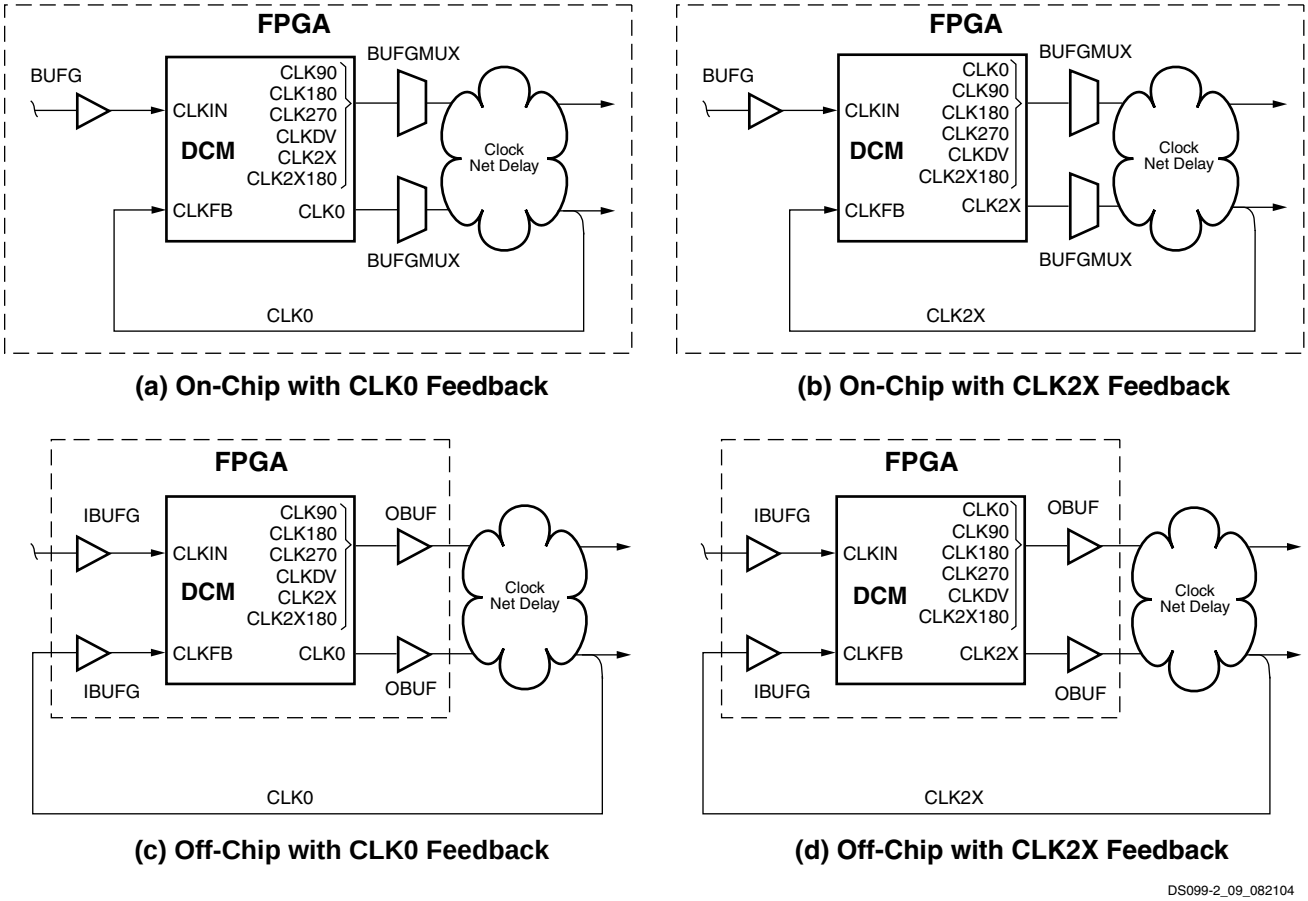


Figure 42: **Input Clock, Output Clock, and Feedback Connections for the DLL**

The CLKFX_DIVIDE is an integer ranging from 1 to 32, inclusive and forms the denominator in Equation 1. For example, if CLKFX_MULTIPLY = 5 and CLKFX_DIVIDE = 3, the frequency of the output clock signal is 5/3 that of the input clock signal. These attributes and their acceptable ranges are described in Table 34.

Table 34: **DFS Attributes**

Attribute	Description	Values
CLKFX_MULTIPLY	Frequency multiplier constant	Integer from 2 to 32, inclusive
CLKFX_DIVIDE	Frequency divisor constant	Integer from 1 to 32, inclusive

Any combination of integer values can be assigned to the CLKFX_MULTIPLY and CLKFX_DIVIDE attributes, provided that two conditions are met:

1. The two values fall within their corresponding ranges, as specified in Table 34.
2. The f_{CLKFX} output frequency calculated in Equation 1 falls within the DCM's operating frequency specifications (see Table 107 in Module 3).

DFS With or Without the DLL

Although the CLKIN input is shared with both units, the DFS unit functions with or separately from the DLL unit. Separate from the DLL, the DFS generates an output frequency from the CLKIN frequency according to the respective CLKFX_MULTIPLY and CLKFX_DIVIDE values. Frequency synthesis does not require a feedback loop. Furthermore, without the DLL, the DFS unit supports a broader operating frequency range.

With the DLL, the DFS unit operates as described above, only with the additional benefit of eliminating the clock distribution delay. In this case, a feedback loop from the CLK0 or CLK2X output to the CLKFB input must be present.

When operating with the DLL unit, the DFS's CLKFX and CLKFX180 outputs are phase-aligned with the CLKIN input every CLKFX_DIVIDE cycles of CLKIN and every CLKFX_MULTIPLY cycles of CLKFX. For example, when CLKFX_MULTIPLY = 5 and CLKFX_DIVIDE = 3, the input and output clock edges coincide every three CLKIN input periods, which is equivalent in time to five CLKFX output periods.

Smaller CLKFX_MULTIPLY and CLKFX_DIVIDE values result in faster lock times. Therefore, CLKFX_MULTIPLY and CLKFX_DIVIDE must be factored to reduce their values wherever possible. For example, given CLKFX_MULTIPLY = 9 and CLKFX_DIVIDE = 6, removing a factor of three yields CLKFX_MULTIPLY = 3 and CLKFX_DIVIDE = 2. While both value-pairs result in the multiplication of clock frequency by 3/2, the latter value-pair enables the DLL to lock more quickly.

Phase Shifter (PS)

The DCM provides two approaches to controlling the phase of a DCM clock output signal relative to the CLKIN signal: First, eight of the nine DCM clock outputs — CLK0, CLK90, CLK180, CLK270, CLK2X, CLK2X180, CLKFX, and CLKFX180 — provide either quadrant or half-period phase shifting of the input clock.

Second, the PS unit provides additional fine phase shift control of all nine DCM outputs. The PS unit accomplishes this by introducing a "fine phase shift" delay (T_{PS}) between the CLKFB and CLKIN signals inside the DLL unit. In FIXED phase shift mode, the fine phase shift is specified at design time with a resolution down to $^{1}/_{256}^{th}$ of a CLKIN cycle or one delay step (DCM_DELAY_STEP), whichever is greater. This fine phase shift value is relative to the coarser quadrant or half-period phase shift of the DCM clock output. When used, the PS unit shifts the phase of all nine DCM clock output signals.

Enabling Phase Shifting and Selecting an Operating Mode

The CLKOUT_PHASE_SHIFT attribute controls the PS unit for the specific DCM instantiation. As described in Table 35, this attribute has three possible values: NONE, FIXED, and VARIABLE. When CLKOUT_PHASE_SHIFT = NONE, the PS unit is disabled and the DCM output clocks are phase-aligned to the CLKIN input via the CLKFB feedback path. Figure 44a shows this case.

The PS unit is enabled when the CLKOUT_PHASE_SHIFT attribute is set to FIXED or VARIABLE modes. These two modes are described in the sections that follow.

Table 35: **PS Attributes**

Attribute	Description	Values
CLKOUT_PHASE_SHIFT	Disables the PS component or chooses between Fixed Phase and Variable Phase modes.	***NONE***, FIXED, VARIABLE
PHASE_SHIFT	Determines size and direction of initial fine phase shift.	Integers from –255 to +255

Status Logic

The Status Logic indicates the present state of the DCM and a means to reset the DCM to its initial known state. The Status Logic signals are described in Table 37.

In general, the Reset (RST) input is only asserted upon configuring the FPGA or when changing the CLKIN frequency. The RST signal must be asserted for three or more CLKIN cycles. A DCM reset does not affect attribute values (for example, CLKFX_MULTIPLY and CLKFX_DIVIDE). If not used, RST is tied to GND.

The eight bits of the STATUS bus are described in Table 38.

Table 37: **Status Logic Signals**

Signal	Direction	Description
RST	Input	A High resets the entire DCM to its initial power-on state. Initializes the DLL taps for a delay of zero. Sets the LOCKED output Low. This input is asynchronous.
STATUS[7:0]	Output	The bit values on the STATUS bus provide information regarding the state of DLL and PS operation
LOCKED	Output	Indicates that the CLKIN and CLKFB signals are in phase by going High. The two signals are out-of-phase when Low.

Table 38: **DCM Status Bus**

Bit	Name	Description
0	Reserved	-
1	CLKIN Stopped	When High, indicates that the CLKIN input signal is not toggling. When Low, indicates CLKIN is toggling. This bit functions only when the CLKFB input is connected.(1)
2	CLKFX Stopped	When High, indicates that the CLKFX output is not toggling. When Low, indicates the CLKFX output is toggling. This bit functions only when the CLKFX or CLKFX180 output are connected.
3-6	Reserved	-

Notes:

1. When only the DFS clock outputs but none of the DLL clock outputs are used, this bit does not go High when the CLKIN signal stops.

Stabilizing DCM Clocks Before User Mode

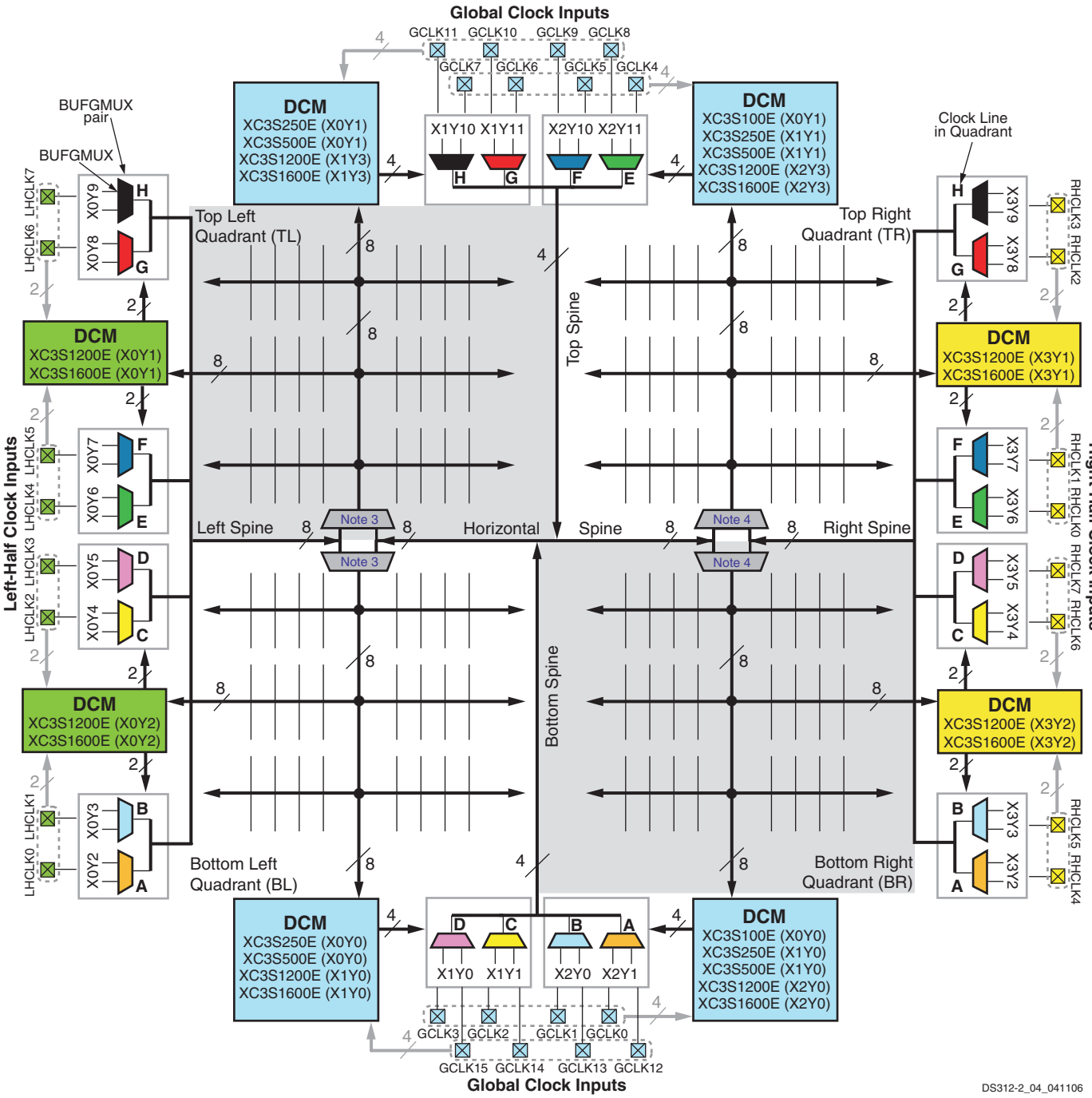
The STARTUP_WAIT attribute shown in Table 39 optionally delays the end of the FPGA's configuration process until after the DCM locks to its incoming clock frequency. This option ensures that the FPGA remains in the Startup phase of configuration until all clock outputs generated by the DCM are stable. When all DCMs that have their STARTUP_WAIT attribute set to TRUE assert the LOCKED signal, then the FPGA completes its configuration process and proceeds to user mode. The associated bitstream generator (BitGen) option ***LCK_cycle*** specifies one of the six cycles in the Startup phase. The selected cycle defines the point at which configuration stalls until all the LOCKED outputs go High. See Start-Up, page 105 for more information.

Table 39: **STARTUP_WAIT Attribute**

Attribute	Description	Values
STARTUP_WAIT	When TRUE, delays transition from configuration to user mode until DCM locks to the input clock.	TRUE, ***FALSE***

Spread Spectrum

DCMs accept typical spread spectrum clocks as long as they meet the input requirements. The DLL will track the frequency changes created by the spread spectrum clock to drive the global clocks to the FPGA logic. See XAPP469, *Spread-Spectrum Clocking Reception for Displays* for details.



Notes:

1. The diagram presents electrical connectivity. The diagram locations do not necessarily match the physical location on the device, although the coordinate locations shown are correct.
2. Number of DCMs and locations of these DCM varies for different device densities. The left and right DCMs are only in the XC3S1200E and XC3S1600E. The XC3S100E has only two DCMs, one on the top right and one on the bottom right of the die.
3. See Figure 47a, which shows how the eight clock lines are multiplexed on the left-hand side of the device.
4. See Figure 47b, which shows how the eight clock lines are multiplexed on the right-hand side of the device.
5. For best direct clock inputs to a particular clock buffer, not a DCM, see Table 41.
6. For best direct clock inputs to a particular DCM, not a BUFGMUX, see Table 30, Table 31, and Table 32. Direct pin inputs to a DCM are shown in gray.

Figure 45: **Spartan-3E Internal Quadrant-Based Clock Network (Electrical Connectivity View)**

Stepping 0 Limitations when Reprogramming via JTAG if FPGA Set for BPI Configuration

The FPGA can always be reprogrammed via the JTAG port, regardless of the mode pin (M[2:0]) settings. However, Stepping 0 devices have a minor limitation. If a Stepping 0 FPGA is set to configure in BPI mode and the FPGA is attached to a parallel memory containing a valid FPGA configuration file, then subsequent reconfigurations using the JTAG port will fail. Potential workarounds include setting the mode pins for JTAG configuration (M[2:0] = <1:0:1>) or offsetting the initial memory location in Flash by 0x2000.

Stepping 1 devices fully support JTAG configuration even when the FPGA mode pins are set for BPI mode.

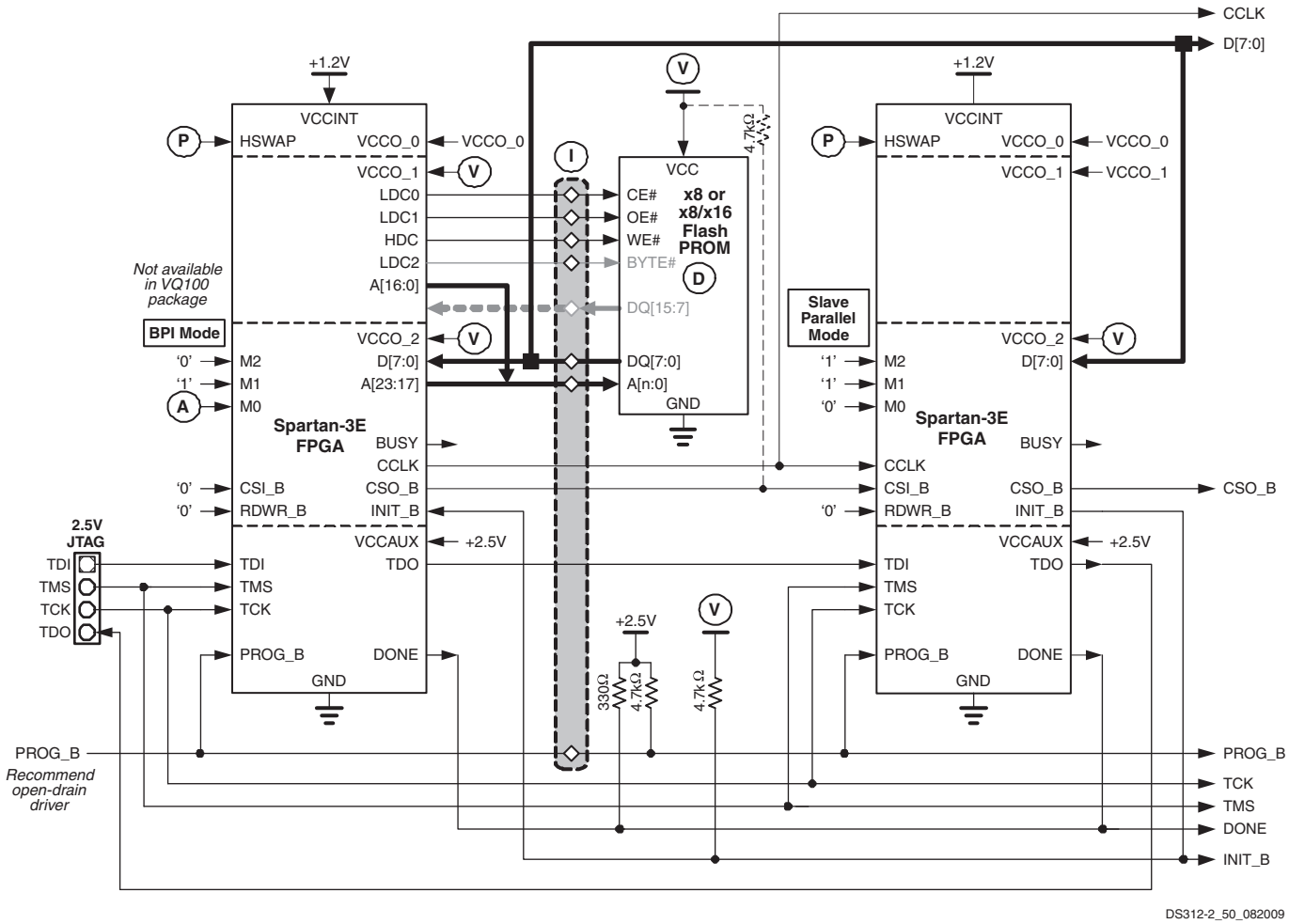


Figure 59: **Daisy-Chaining from BPI Flash Mode**

In-System Programming Support

(I) In a production application, the parallel Flash PROM is usually preprogrammed before it is mounted on the printed circuit board. In-system programming support is available from third-party boundary-scan tool vendors and from some third-party PROM programmers using a socket adapter with attached wires. To gain access to the parallel Flash signals, drive the FPGA's PROG_B input Low with an open-drain driver. This action places all FPGA I/O pins, including those attached to the parallel Flash, in high-impedance (Hi-Z). If the HSWAP input is Low, the I/Os have pull-up resistors to the V_{CCO} input on their respective I/O bank. The external programming hardware then has direct access to the parallel Flash pins. The programming access points are highlighted in the gray boxes in Figure 58 and Figure 59.

The FPGA itself can also be used as a parallel Flash PROM programmer during development and test phases. Initially, an FPGA-based programmer is downloaded into the FPGA via JTAG. Then the FPGA performs the Flash PROM programming algorithms and receives programming data from the host via the FPGA's JTAG interface. See the Embedded System Tools Reference Manual.

Dynamically Loading Multiple Configuration Images Using MultiBoot Option

For additional information, refer to the "Reconfiguration and MultiBoot" chapter in UG332.

Start-Up

At the end of configuration, the FPGA automatically pulses the Global Set/Reset (GSR) signal, placing all flip-flops in a known state. After configuration completes, the FPGA switches over to the user application loaded into the FPGA. The sequence and timing of how the FPGA switches over is programmable as is the clock source controlling the sequence.

The default start-up sequence appears in Figure 68, where the Global Three-State signal (GTS) is released one clock cycle after DONE goes High. This sequence allows the DONE signal to enable or disable any external logic used during configuration before the user application in the FPGA starts driving output signals. One clock cycle later, the Global Write Enable (GWE) signal is released. This allows signals to propagate within the FPGA before any clocked storage elements such as flip-flops and block ROM are enabled.

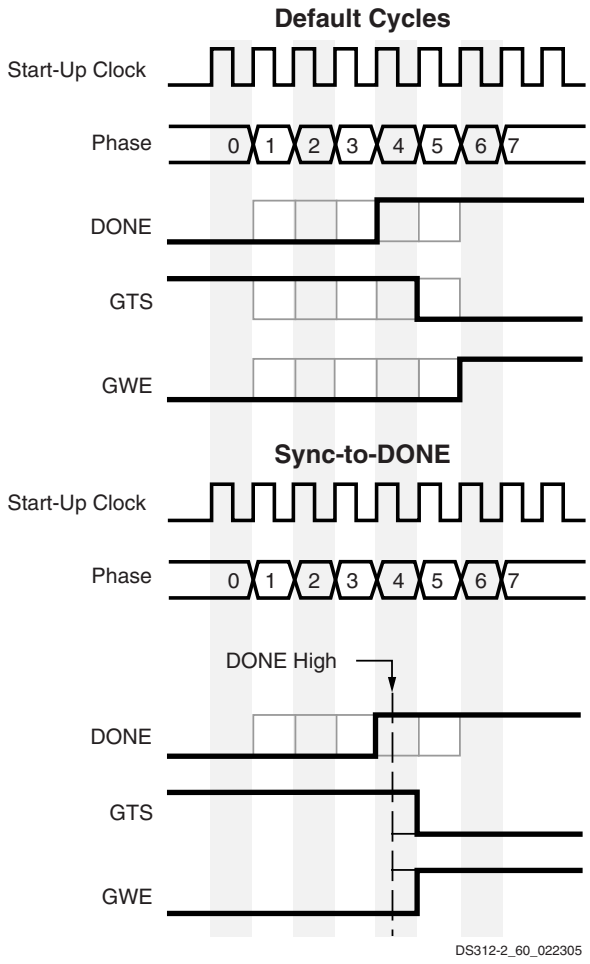


Figure 68: **Default Start-Up Sequence**

The function of the dual-purpose I/O pins, such as M[2:0], VS[2:0], HSWAP, and A[23:0], also changes when the DONE pin goes High. When DONE is High, these pins become user I/Os. Like all user-I/O pins, GTS controls when the dual-purpose pins can drive out.

The relative timing of configuration events is programmed via the Bitstream Generator (BitGen) options in the Xilinx development software. For example, the GTS and GWE events can be programmed to wait for all the DONE pins to High on all the devices in a multiple-FPGA daisy-chain, forcing the FPGAs to start synchronously. Similarly, the start-up sequence can be paused at any stage, waiting for selected DCMs to lock to their respective input clock signals. See also Stabilizing DCM Clocks Before User Mode.

By default, the start-up sequence is synchronized to CCLK. Alternatively, the start-up sequence can be synchronized to a user-specified clock from within the FPGA application using the STARTUP_SPARTAN3E library primitive and by setting the ***StartupClk*** bitstream generator option. The FPGA application can optionally assert the GSR and GTS signals via the STARTUP_SPARTAN3E primitive. For JTAG configuration, the start-up sequence can be synchronized to the TCK clock input.

Differential I/O Standards

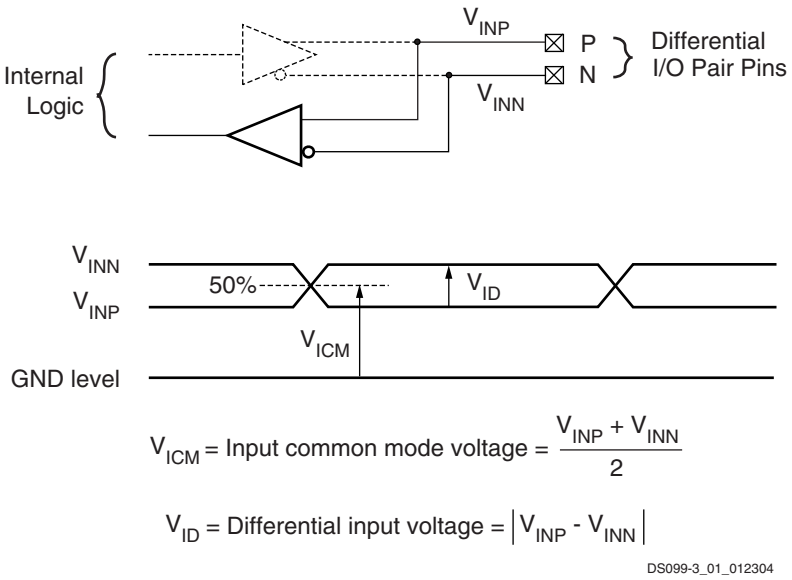


Figure 69: **Differential Input Voltages**

Table 82: **Recommended Operating Conditions for User I/Os Using Differential Signal Standards**

IOSTANDARD Attribute	V_{CCO} for Drivers[1]			V_{ID}			V_{ICM}		
	Min (V)	Nom (V)	Max (V)	Min (mV)	Nom (mV)	Max (mV)	Min (V)	Nom (V)	Max (V)
LVDS_25	2.375	2.50	2.625	100	350	600	0.30	1.25	2.20
BLVDS_25	2.375	2.50	2.625	100	350	600	0.30	1.25	2.20
MINI_LVDS_25	2.375	2.50	2.625	200	-	600	0.30	-	2.2
LVPECL_25[2]	Inputs Only			100	800	1000	0.5	1.2	2.0
RSDS_25	2.375	2.50	2.625	100	200	-	0.3	1.20	1.4
DIFF_HSTL_I_18	1.7	1.8	1.9	100	-	-	0.8	-	1.1
DIFF_HSTL_III_18	1.7	1.8	1.9	100	-	-	0.8	-	1.1
DIFF_SSTL18_I	1.7	1.8	1.9	100	-	-	0.7	-	1.1
DIFF_SSTL2_I	2.3	2.5	2.7	100	-	-	1.0	-	1.5

Notes:

1. The V_{CCO} rails supply only differential output drivers, not input circuits.
2. V_{REF} inputs are not used for any of the differential I/O standards.

Table 90: **Propagation Times for the IOB Input Path**

Symbol	Description	Conditions	IFD_DELAY_VALUE=	Device	Speed Grade		Units
					-5	-4	
					Min	Min	
Propagation Times							
T_{IOPLI}	The time it takes for data to travel from the Input pin through the IFF latch to the I output with no input delay programmed	LVCMOS25(2), IFD_DELAY_VALUE = 0	0	All	1.96	2.25	ns
T_{IOPLID}	The time it takes for data to travel from the Input pin through the IFF latch to the I output with the input delay programmed	LVCMOS25(2), IFD_DELAY_VALUE = default software setting	2	XC3S100E	5.40	5.97	ns
			3	All Others	6.30	7.20	

Notes:

1. The numbers in this table are tested using the methodology presented in Table 95 and are based on the operating conditions set forth in Table 77 and Table 80.
2. This propagation time requires adjustment whenever a signal standard other than LVCMOS25 is assigned to the data Input. When this is true, *add* the appropriate Input adjustment from Table 91.

Table 91: **Input Timing Adjustments by IOSTANDARD**

Convert Input Time from LVCMOS25 to the Following Signal Standard (IOSTANDARD)	Add the Adjustment Below		Units
	Speed Grade		
	-5	-4	
Single-Ended Standards			
LVTTL	0.42	0.43	ns
LVCMOS33	0.42	0.43	ns
LVCMOS25	0	0	ns
LVCMOS18	0.96	0.98	ns
LVCMOS15	0.62	0.63	ns
LVCMOS12	0.26	0.27	ns
PCI33_3	0.41	0.42	ns
PCI66_3	0.41	0.42	ns
HSTL_I_18	0.12	0.12	ns
HSTL_III_18	0.17	0.17	ns
SSTL18_I	0.30	0.30	ns
SSTL2_I	0.15	0.15	ns

Table 91: **Input Timing Adjustments by IOSTANDARD *(Cont'd)***

Convert Input Time from LVCMOS25 to the Following Signal Standard (IOSTANDARD)	Add the Adjustment Below		Units
	Speed Grade		
	-5	-4	
Differential Standards			
LVDS_25	0.48	0.49	ns
BLVDS_25	0.39	0.39	ns
MINI_LVDS_25	0.48	0.49	ns
LVPECL_25	0.27	0.27	ns
RSDS_25	0.48	0.49	ns
DIFF_HSTL_I_18	0.48	0.49	ns
DIFF_HSTL_III_18	0.48	0.49	ns
DIFF_SSTL18_I	0.30	0.30	ns
DIFF_SSTL2_I	0.32	0.32	ns

Notes:

1. The numbers in this table are tested using the methodology presented in Table 95 and are based on the operating conditions set forth in Table 77, Table 80, and Table 82.
2. These adjustments are used to convert input path times originally specified for the LVCMOS25 standard to times that correspond to other signal standards.

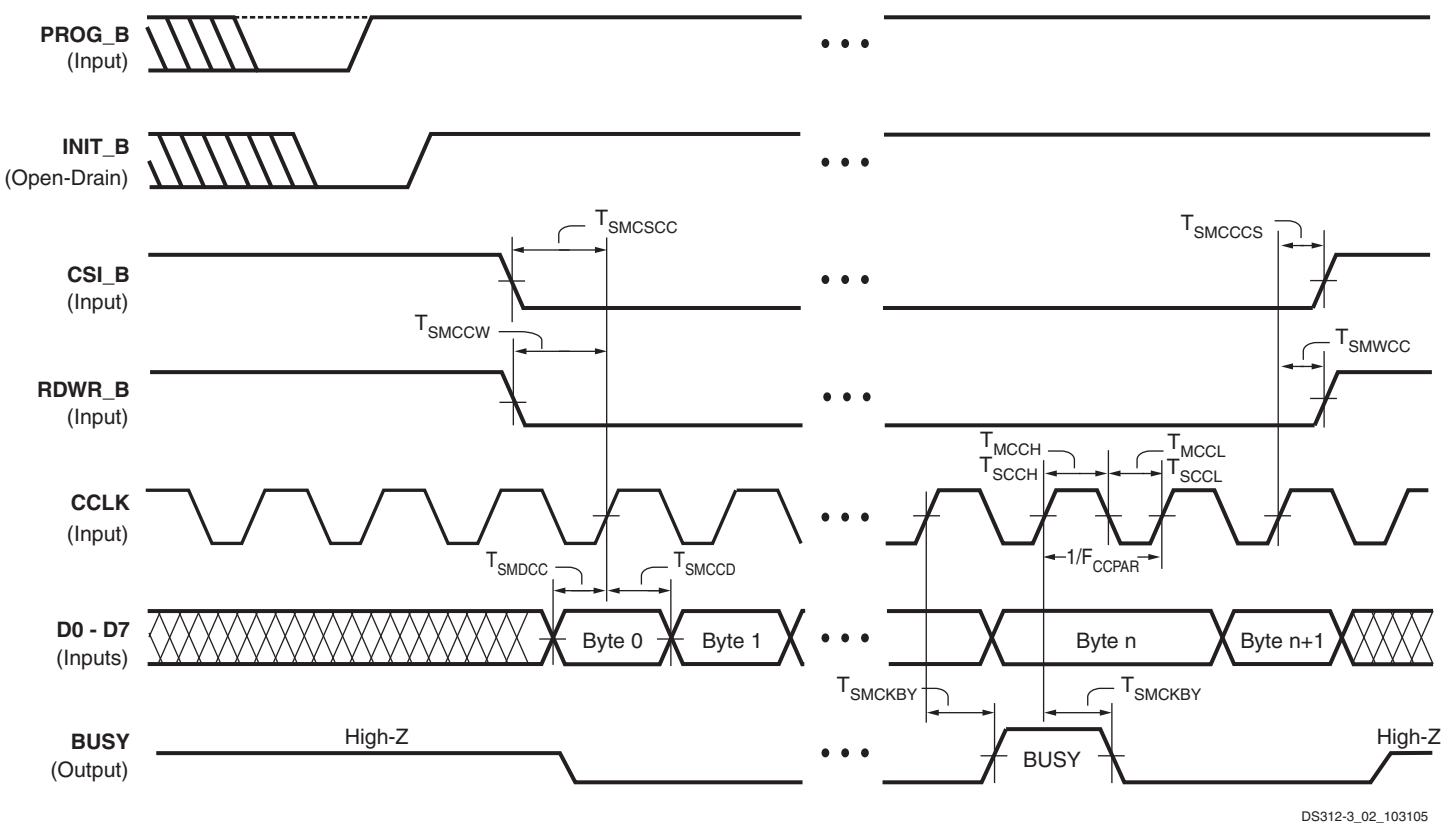
Table 93: **Timing for the IOB Three-State Path**

Symbol	Description	Conditions	Device	Speed Grade		Units
				-5	-4	
				Max	Max	
Synchronous Output Enable/Disable Times						
T_{IOCKHZ}	Time from the active transition at the OTCLK input of the Three-state Flip-Flop (TFF) to when the Output pin enters the high-impedance state	LVCMOS25, 12 mA output drive, Fast slew rate	All	1.49	1.71	ns
T_{IOCKON}(2)	Time from the active transition at TFF's OTCLK input to when the Output pin drives valid data		All	2.70	3.10	ns
Asynchronous Output Enable/Disable Times						
T_{GTS}	Time from asserting the Global Three State (GTS) input on the STARTUP_SPARTAN3E primitive to when the Output pin enters the high-impedance state	LVCMOS25, 12 mA output drive, Fast slew rate	All	8.52	9.79	ns
Set/Reset Times						
T_{IOSRHZ}	Time from asserting TFF's SR input to when the Output pin enters a high-impedance state	LVCMOS25, 12 mA output drive, Fast slew rate	All	2.11	2.43	ns
T_{IOSRON}(2)	Time from asserting TFF's SR input at TFF to when the Output pin drives valid data		All	3.32	3.82	ns

Notes:

1. The numbers in this table are tested using the methodology presented in Table 95 and are based on the operating conditions set forth in Table 77 and Table 80.
2. This time requires adjustment whenever a signal standard other than LVCMOS25 with 12 mA drive and Fast slew rate is assigned to the data Output. When this is true, *add* the appropriate Output adjustment from Table 94.
3. For minimum delays use the values reported by the Timing Analyzer.

Slave Parallel Mode Timing



Notes:

1. It is possible to abort configuration by pulling CSI_B Low in a given CCLK cycle, then switching RDWR_B Low or High in any subsequent cycle for which CSI_B remains Low. The RDWR_B pin asynchronously controls the driver impedance of the D0 - D7 bus. When RDWR_B switches High, be careful to avoid contention on the D0 - D7 bus.

Figure 75: **Waveforms for Slave Parallel Configuration**

Table 117: **Timing for the Slave Parallel Configuration Mode**

Symbol	Description	All Speed Grades		Units
		Min	Max	
Clock-to-Output Times				
T_{SMCKBY}	The time from the rising transition on the CCLK pin to a signal transition at the BUSY pin	-	12.0	ns
Setup Times				
T_{SMDCC}	The time from the setup of data at the D0-D7 pins to the active edge the CCLK pin	11.0	-	ns
T_{SMCSCC}	Setup time on the CSI_B pin before the active edge of the CCLK pin	10.0	-	ns
T_{SMCCW}(2)	Setup time on the RDWR_B pin before active edge of the CCLK pin	23.0	-	ns
Hold Times				
T_{SMCCD}	The time from the active edge of the CCLK pin to the point when data is last held at the D0-D7 pins	1.0	-	ns
T_{SMCCCS}	The time from the active edge of the CCLK pin to the point when a logic level is last held at the CSO_B pin	0	-	ns
T_{SMWCC}	The time from the active edge of the CCLK pin to the point when a logic level is last held at the RDWR_B pin	0	-	ns

Table 119: **Configuration Timing Requirements for Attached SPI Serial Flash**

Symbol	Description	Requirement	Units
T_{CCS}	SPI serial Flash PROM chip-select time	$T_{CCS} \leq T_{MCCL1} - T_{CCO}$	ns
T_{DSU}	SPI serial Flash PROM data input setup time	$T_{DSU} \leq T_{MCCL1} - T_{CCO}$	ns
T_{DH}	SPI serial Flash PROM data input hold time	$T_{DH} \leq T_{MCCH1}$	ns
T_V	SPI serial Flash PROM data clock-to-output time	$T_V \leq T_{MCCLn} - T_{DCC}$	ns
f_C or f_R	Maximum SPI serial Flash PROM clock frequency (also depends on specific read command used)	$f_C \geq \frac{1}{T_{CCLKn(min)}}$	MHz

Notes:

1. These requirements are for successful FPGA configuration in SPI mode, where the FPGA provides the CCLK frequency. The post configuration timing can be different to support the specific needs of the application loaded into the FPGA and the resulting clock source.
2. Subtract additional printed circuit board routing delay as required by the application.

Byte Peripheral Interface (BPI) Configuration Timing

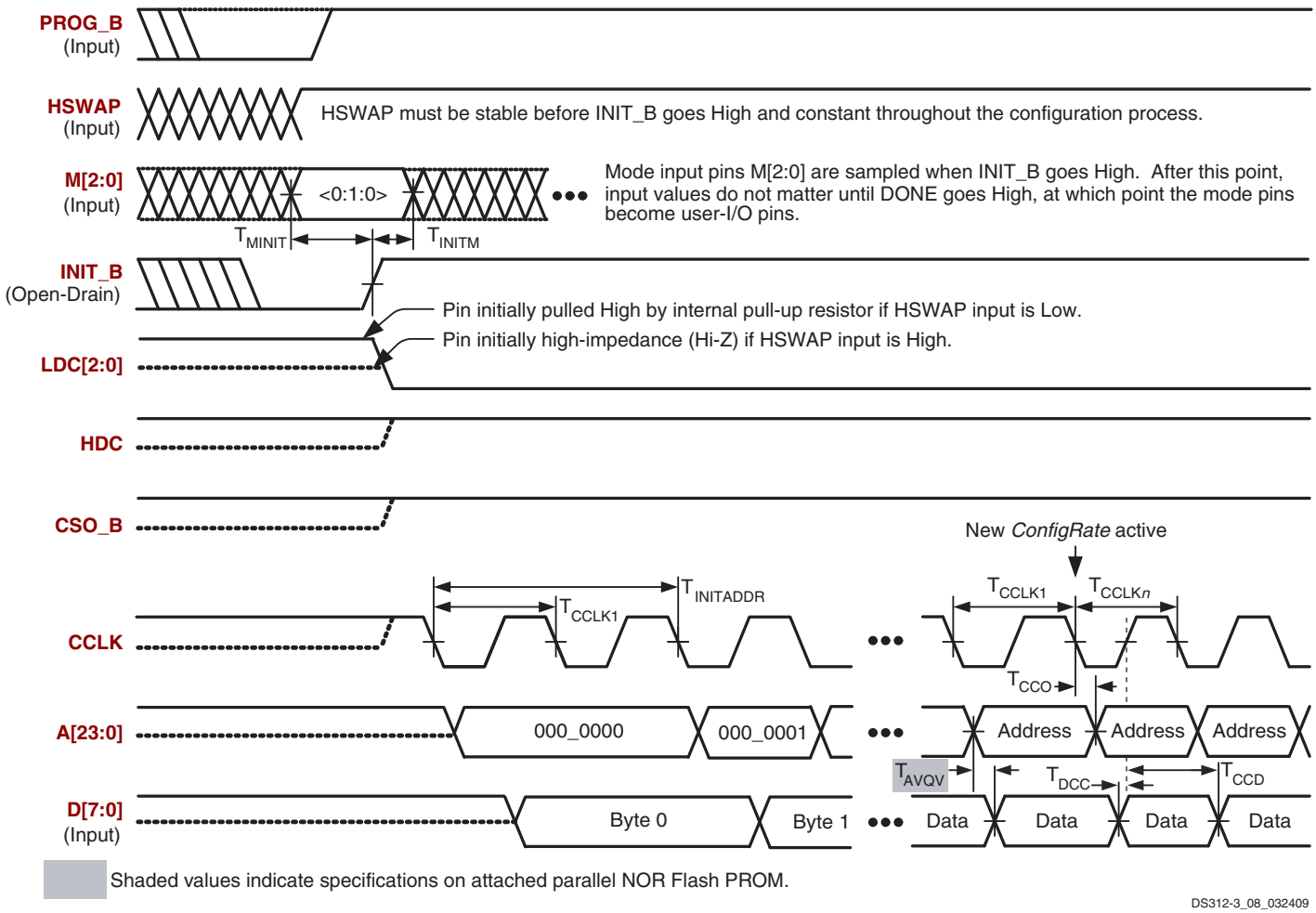


Figure 77: **Waveforms for Byte-wide Peripheral Interface (BPI) Configuration (BPI-DN mode shown)**

Table 120: **Timing for Byte-wide Peripheral Interface (BPI) Configuration Mode**

Symbol	Description		Minimum	Maximum	Units
T_{CCLK1}	Initial CCLK clock period		See Table 112		
T_{CCLKn}	CCLK clock period after FPGA loads ConfigRate setting		See Table 112		
T_{MINIT}	Setup time on CSI_B, RDWR_B, and M[2:0] mode pins before the rising edge of INIT_B		50	-	ns
T_{INITM}	Hold time on CSI_B, RDWR_B, and M[2:0] mode pins after the rising edge of INIT_B		0	-	ns
$T_{INITADDR}$	Minimum period of initial A[23:0] address cycle; LDC[2:0] and HDC are asserted and valid	BPI-UP: (M[2:0] = <0:1:0>)	5	5	T_{CCLK1} cycles
		BPI-DN: (M[2:0] = <0:1:1>)	2	2	
T_{CCO}	Address A[23:0] outputs valid after CCLK falling edge		See Table 116		
T_{DCC}	Setup time on D[7:0] data inputs before CCLK rising edge		See Table 116		
T_{CCD}	Hold time on D[7:0] data inputs after CCLK rising edge		See Table 116		

Table 133: **CP132 Package Pinout *(Cont'd)***

Bank	XC3S100E Pin Name	XC3S250E XC3S500E Pin Name	CP132 Ball	Type
0	VCCO_0	VCCO_0	B10	VCCO
1	IO/A0	IO/A0	F12	DUAL
1	IO/VREF_1	IO/VREF_1	K13	VREF
1	IO_L01N_1/A15	IO_L01N_1/A15	N14	DUAL
1	IO_L01P_1/A16	IO_L01P_1/A16	N13	DUAL
1	IO_L02N_1/A13	IO_L02N_1/A13	M13	DUAL
1	IO_L02P_1/A14	IO_L02P_1/A14	M12	DUAL
1	IO_L03N_1/A11	IO_L03N_1/A11	L14	DUAL
1	IO_L03P_1/A12	IO_L03P_1/A12	L13	DUAL
1	IO_L04N_1/A9/RHCLK1	IO_L04N_1/A9/RHCLK1	J12	RHCLK/DUAL
1	IO_L04P_1/A10/RHCLK0	IO_L04P_1/A10/RHCLK0	K14	RHCLK/DUAL
1	IO_L05N_1/A7/RHCLK3/TRDY1	IO_L05N_1/A7/RHCLK3/TRDY1	J14	RHCLK/DUAL
1	IO_L05P_1/A8/RHCLK2	IO_L05P_1/A8/RHCLK2	J13	RHCLK/DUAL
1	IO_L06N_1/A5/RHCLK5	IO_L06N_1/A5/RHCLK5	H12	RHCLK/DUAL
1	IO_L06P_1/A6/RHCLK4/IRDY1	IO_L06P_1/A6/RHCLK4/IRDY1	H13	RHCLK/DUAL
1	IO_L07N_1/A3/RHCLK7	IO_L07N_1/A3/RHCLK7	G13	RHCLK/DUAL
1	IO_L07P_1/A4/RHCLK6	IO_L07P_1/A4/RHCLK6	G14	RHCLK/DUAL
1	IO_L08N_1/A1	IO_L08N_1/A1	F13	DUAL
1	IO_L08P_1/A2	IO_L08P_1/A2	F14	DUAL
1	IO_L09N_1/LDC0	IO_L09N_1/LDC0	D12	DUAL
1	IO_L09P_1/HDC	IO_L09P_1/HDC	D13	DUAL
1	IO_L10N_1/LDC2	IO_L10N_1/LDC2	C13	DUAL
1	IO_L10P_1/LDC1	IO_L10P_1/LDC1	C14	DUAL
1	IP/VREF_1	IP/VREF_1	G12	VREF
1	VCCO_1	VCCO_1	E13	VCCO
1	VCCO_1	VCCO_1	M14	VCCO
2	IO/D5	IO/D5	P4	DUAL
2	IO/M1	IO/M1	N7	DUAL
2	IP/VREF_2	IO/VREF_2	P11	***100E:*** VREF(INPUT) ***Others:*** VREF(I/O)
2	IO_L01N_2/INIT_B	IO_L01N_2/INIT_B	N1	DUAL
2	IO_L01P_2/CSO_B	IO_L01P_2/CSO_B	M2	DUAL
2	IO_L02N_2/MOSI/CSI_B	IO_L02N_2/MOSI/CSI_B	N2	DUAL
2	IO_L02P_2/DOUT/BUSY	IO_L02P_2/DOUT/BUSY	P1	DUAL
2	IO_L03N_2/D6/GCLK13	IO_L03N_2/D6/GCLK13	N4	DUAL/GCLK
2	IO_L03P_2/D7/GCLK12	IO_L03P_2/D7/GCLK12	M4	DUAL/GCLK
2	IO_L04N_2/D3/GCLK15	IO_L04N_2/D3/GCLK15	N5	DUAL/GCLK
2	IO_L04P_2/D4/GCLK14	IO_L04P_2/D4/GCLK14	M5	DUAL/GCLK
2	IO_L06N_2/D1/GCLK3	IO_L06N_2/D1/GCLK3	P7	DUAL/GCLK
2	IO_L06P_2/D2/GCLK2	IO_L06P_2/D2/GCLK2	P6	DUAL/GCLK
2	IO_L07N_2/DIN/D0	IO_L07N_2/DIN/D0	N8	DUAL
2	IO_L07P_2/M0	IO_L07P_2/M0	P8	DUAL
2	N.C. (◆)	IO_L08N_2/A22	M9	***100E:*** N.C. ***Others:*** DUAL

Table 143: **FT256 Package Pinout *(Cont'd)***

Bank	XC3S250E Pin Name	XC3S500E Pin Name	XC3S1200E Pin Name	FT256 Ball	Type
0	IO_L12P_0	IO_L12P_0	IO_L12P_0	E8	I/O
0	N.C. (◆)	IO_L13N_0	IO_L13N_0	C7	***250E:*** N.C. ***500E:*** I/O ***1200E:*** I/O
0	N.C. (◆)	IO_L13P_0	IO_L13P_0	B7	***250E:*** N.C. ***500E:*** I/O ***1200E:*** I/O
0	IO_L14N_0/VREF_0	IO_L14N_0/VREF_0	IO_L14N_0/VREF_0	D7	VREF
0	IO_L14P_0	IO_L14P_0	IO_L14P_0	E7	I/O
0	IO_L15N_0	IO_L15N_0	IO_L15N_0	D6	I/O
0	IO_L15P_0	IO_L15P_0	IO_L15P_0	C6	I/O
0	IO_L17N_0/VREF_0	IO_L17N_0/VREF_0	IO_L17N_0/VREF_0	A4	VREF
0	IO_L17P_0	IO_L17P_0	IO_L17P_0	A5	I/O
0	IO_L18N_0	IO_L18N_0	IO_L18N_0	C4	I/O
0	IO_L18P_0	IO_L18P_0	IO_L18P_0	C5	I/O
0	IO_L19N_0/HSWAP	IO_L19N_0/HSWAP	IO_L19N_0/HSWAP	B3	DUAL
0	IO_L19P_0	IO_L19P_0	IO_L19P_0	C3	I/O
0	IP	IP	IP	A3	INPUT
0	IP	IP	IP	C13	INPUT
0	IP_L02N_0	IP_L02N_0	IP_L02N_0	C12	INPUT
0	IP_L02P_0	IP_L02P_0	IP_L02P_0	D12	INPUT
0	IP_L07N_0	IP_L07N_0	IP_L07N_0	C9	INPUT
0	IP_L07P_0	IP_L07P_0	IP_L07P_0	C10	INPUT
0	IP_L10N_0/GCLK9	IP_L10N_0/GCLK9	IP_L10N_0/GCLK9	B8	GCLK
0	IP_L10P_0/GCLK8	IP_L10P_0/GCLK8	IP_L10P_0/GCLK8	A8	GCLK
0	IP_L16N_0	IP_L16N_0	IP_L16N_0	E6	INPUT
0	IP_L16P_0	IP_L16P_0	IP_L16P_0	D5	INPUT
0	VCCO_0	VCCO_0	VCCO_0	B5	VCCO
0	VCCO_0	VCCO_0	VCCO_0	B12	VCCO
0	VCCO_0	VCCO_0	VCCO_0	F7	VCCO
0	VCCO_0	VCCO_0	VCCO_0	F10	VCCO
1	IO_L01N_1/A15	IO_L01N_1/A15	IO_L01N_1/A15	R15	DUAL
1	IO_L01P_1/A16	IO_L01P_1/A16	IO_L01P_1/A16	R16	DUAL
1	IO_L02N_1/A13	IO_L02N_1/A13	IO_L02N_1/A13	P15	DUAL
1	IO_L02P_1/A14	IO_L02P_1/A14	IO_L02P_1/A14	P16	DUAL
1	N.C. (◆)	IO_L03N_1/VREF_1	IO_L03N_1/VREF_1	N15	***250E:*** N.C. ***500E:*** VREF ***1200E:*** VREF
1	N.C. (◆)	IO_L03P_1	IO_L03P_1	N14	***250E:*** N.C. ***500E:*** I/O ***1200E:*** I/O
1	IO_L04N_1/VREF_1	IO_L04N_1/VREF_1	IO_L04N_1/VREF_1	M16	VREF
1	IO_L04P_1	IO_L04P_1	IO_L04P_1	N16	I/O
1	N.C. (◆)	IO_L05N_1	IO_L05N_1	L13	***250E:*** N.C. ***500E:*** I/O ***1200E:*** I/O

Table 148: **FG320 Package Pinout *(Cont'd)***

Bank	XC3S500E Pin Name	XC3S1200E Pin Name	XC3S1600E Pin Name	FG320 Ball	Type
0	IO_L08P_0	IO_L08P_0	IO_L08P_0	E11	I/O
0	IO_L09N_0	IO_L09N_0	IO_L09N_0	D11	I/O
0	IO_L09P_0	IO_L09P_0	IO_L09P_0	C11	I/O
0	IO_L11N_0/GCLK5	IO_L11N_0/GCLK5	IO_L11N_0/GCLK5	E10	GCLK
0	IO_L11P_0/GCLK4	IO_L11P_0/GCLK4	IO_L11P_0/GCLK4	D10	GCLK
0	IO_L12N_0/GCLK7	IO_L12N_0/GCLK7	IO_L12N_0/GCLK7	A10	GCLK
0	IO_L12P_0/GCLK6	IO_L12P_0/GCLK6	IO_L12P_0/GCLK6	B10	GCLK
0	IO_L14N_0/GCLK11	IO_L14N_0/GCLK11	IO_L14N_0/GCLK11	D9	GCLK
0	IO_L14P_0/GCLK10	IO_L14P_0/GCLK10	IO_L14P_0/GCLK10	C9	GCLK
0	IO_L15N_0	IO_L15N_0	IO_L15N_0	F9	I/O
0	IO_L15P_0	IO_L15P_0	IO_L15P_0	E9	I/O
0	IO_L17N_0	IO_L17N_0	IO_L17N_0	F8	I/O
0	IO_L17P_0	IO_L17P_0	IO_L17P_0	E8	I/O
0	IO_L18N_0/VREF_0	IO_L18N_0/VREF_0	IO_L18N_0/VREF_0	D7	VREF
0	IO_L18P_0	IO_L18P_0	IO_L18P_0	C7	I/O
0	IO_L19N_0/VREF_0	IO_L19N_0/VREF_0	IO_L19N_0/VREF_0	E7	VREF
0	IO_L19P_0	IO_L19P_0	IO_L19P_0	F7	I/O
0	IO_L20N_0	IO_L20N_0	IO_L20N_0	A6	I/O
0	IO_L20P_0	IO_L20P_0	IO_L20P_0	B6	I/O
0	N.C. (◆)	IO_L21N_0	IO_L21N_0	E6	**500E:** N.C. **1200E:** I/O **1600E:** I/O
0	N.C. (◆)	IO_L21P_0	IO_L21P_0	D6	**500E:** N.C. **1200E:** I/O **1600E:** I/O
0	IO_L23N_0/VREF_0	IO_L23N_0/VREF_0	IO_L23N_0/VREF_0	D5	VREF
0	IO_L23P_0	IO_L23P_0	IO_L23P_0	C5	I/O
0	IO_L24N_0	IO_L24N_0	IO_L24N_0	B4	I/O
0	IO_L24P_0	IO_L24P_0	IO_L24P_0	A4	I/O
0	IO_L25N_0/HSWAP	IO_L25N_0/HSWAP	IO_L25N_0/HSWAP	B3	DUAL
0	IO_L25P_0	IO_L25P_0	IO_L25P_0	C3	I/O
0	IP	IP	IP	A3	INPUT
0	IP	IP	IP	C15	INPUT
0	IP_L02N_0	IP_L02N_0	IP_L02N_0	A15	INPUT
0	IP_L02P_0	IP_L02P_0	IP_L02P_0	B15	INPUT
0	IP_L07N_0	IP_L07N_0	IP_L07N_0	D12	INPUT
0	IP_L07P_0	IP_L07P_0	IP_L07P_0	C12	INPUT
0	IP_L10N_0	IP_L10N_0	IP_L10N_0	G10	INPUT
0	IP_L10P_0	IP_L10P_0	IP_L10P_0	F10	INPUT
0	IP_L13N_0/GCLK9	IP_L13N_0/GCLK9	IP_L13N_0/GCLK9	B9	GCLK
0	IP_L13P_0/GCLK8	IP_L13P_0/GCLK8	IP_L13P_0/GCLK8	B8	GCLK
0	IP_L16N_0	IP_L16N_0	IP_L16N_0	D8	INPUT
0	IP_L16P_0	IP_L16P_0	IP_L16P_0	C8	INPUT
0	IP_L22N_0	IP_L22N_0	IP_L22N_0	B5	INPUT
0	IP_L22P_0	IP_L22P_0	IP_L22P_0	A5	INPUT

FG400 Footprint

Left Half of Package (top view)

156 **I/O:** Unrestricted, general-purpose user I/O

62 **INPUT:** Unrestricted, general-purpose input pin

46 **DUAL:** Configuration pin, then possible user I/O

24 **VREF:** User I/O or input voltage reference for bank

16 **CLK:** User I/O, input, or clock buffer input

2 **CONFIG:** Dedicated configuration pins

4 **JTAG:** Dedicated JTAG port pins

42 **GND:** Ground

24 **VCCO:** Output voltage supply for bank

16 **VCCINT:** Internal core supply voltage (+1.2V)

8 **VCCAUX:** Auxiliary supply voltage (+2.5V)

0 **N.C.:** Not connected

Bank 0

	1	2	3	4	5	6	7	8	9	10
A	GND	I/O L31N_0	I/O	INPUT L29N_0	INPUT L29P_0	I/O L25N_0	I/O L25P_0	I/O	I/O L18N_0 GCLK11	I/O L18P_0 GCLK10
B	I/O L03P_3	I/O L31P_0	TDI	VCCO_0	I/O L27N_0	I/O L27P_0	GND	I/O L22N_0 VREF_0	I/O L22P_0	VCCO_0
C	I/O L03N_3	PROG_B	GND	I/O L32P_0	I/O L30N_0 VREF_0	I/O L28P_0	I/O	INPUT L23N_0	I/O L21N_0	I/O
D	I/O L04P_3	I/O L01N_3	I/O L01P_3	I/O L32N_0 HSWAP	I/O L30P_0	I/O L28N_0	VCCO_0	INPUT L23P_0	I/O L21P_0	GND
E	I/O L04N_3	VCCO_3	I/O L02N_3 VREF_3	I/O L02P_3	INPUT	INPUT L26N_0	INPUT L26P_0	I/O	I/O L19P_0	I/O L16N_0 GCLK7
F	I/O L06N_3	I/O L06P_3	I/O L05N_3	I/O L05P_3	INPUT	GND	I/O L24N_0 VREF_0	I/O L24P_0	I/O L19N_0	VCCO_0
G	INPUT	GND	I/O L07P_3	I/O L07N_3	I/O L08N_3	INPUT	I/O	INPUT L20P_0	INPUT L20N_0	INPUT L17N_0 GCLK9
H	INPUT	I/O L09P_3	I/O L09N_3 VREF_3	VCCO_3	I/O L08P_3	I/O L10P_3	I/O L10N_3	GND	VCCINT	INPUT L17P_0 GCLK8
J	I/O L12N_3	I/O L12P_3	I/O L11P_3	I/O L11N_3	INPUT	I/O L13N_3	VCCAUX	VCCINT	GND	VCCINT
K	GND	I/O L14N_3 LHCLK1	I/O L14P_3 LHCLK0	VCCAUX	INPUT VREF_3	I/O L13P_3	I/O L15P_3 LHCLK2	GND	VCCINT	GND
L	I/O L16N_3 LHCLK5	VCCO_3	I/O L17N_3 LHCLK7	GND	INPUT	VCCO_3	I/O L15N_3 LHCLK3 IRDY2	INPUT	GND	VCCINT
M	I/O L16P_3 LHCLK4 TRDY2	INPUT	I/O L17P_3 LHCLK6	I/O L19N_3	I/O L19P_3	I/O L20P_3	I/O L18N_3	I/O L18P_3	VCCINT	GND
N	I/O L21P_3	I/O L21N_3	I/O L23P_3	I/O L23N_3	INPUT	I/O L20N_3 VREF_3	I/O L22P_3	VCCINT	VCCAUX	VCCINT
P	I/O L24P_3	GND	INPUT	VCCO_3	I/O L25P_3	INPUT VREF_3	I/O L22N_3	I/O	GND	I/O L16N_2 D3 GCLK15
R	I/O L24N_3	I/O L26P_3	I/O L27P_3	I/O L27N_3	I/O L25N_3	GND	I/O L09N_2 VREF_2	INPUT L11N_2	I/O	I/O L16P_2 D4 GCLK14
T	I/O L28N_3 VREF_3	I/O L26N_3	I/O L29N_3	INPUT	I/O L06P_2	I/O L06N_2	I/O L09P_2	INPUT L11P_2	INPUT L14P_2	INPUT L14N_2 VREF_2
U	I/O L28P_3	VCCO_3	I/O L29P_3	I/O L01P_2 CSO_B	I/O L03P_2 DOUT BUSY	INPUT L05P_2	I/O L07N_2	VCCO_2	I/O L12N_2	VCCAUX
V	I/O L30N_3	I/O L30P_3	GND	I/O L01N_2 INIT_B	I/O L03N_2 MOSI CSI_B	INPUT L05N_2	I/O L07P_2	I/O L10N_2	I/O L12P_2	I/O L18P_2 D2 GCLK2
W	INPUT	INPUT L02P_2	INPUT	I/O L04P_2	VCCO_2	INPUT L08P_2	GND	I/O L10P_2	I/O L15P_2 D7 GCLK12	I/O L15N_2 D6 GCLK13
Y	GND	INPUT L02N_2	I/O VREF_2	I/O L04N_2	I/O	INPUT L08N_2	I/O	I/O L13N_2	I/O L13P_2	GND

Bank 3

Bank 2

DS312-4_08_101905

Figure 87: **FG400 Package Footprint (top view)**

FG484 Footprint

Left Half of Package (top view)

214 **I/O:** Unrestricted, general-purpose user I/O

72 **INPUT:** User I/O or reference resistor input for bank

46 **DUAL:** Configuration pin, then possible user I/O

28 **VREF:** User I/O or input voltage reference for bank

16 **CLK:** User I/O, input, or clock buffer input

2 **CONFIG:** Dedicated configuration pins

4 **JTAG:** Dedicated JTAG port pins

48 **GND:** Ground

28 **VCCO:** Output voltage supply for bank

16 **VCCINT:** Internal core supply voltage (+1.2V)

10 **VCCAUX:** Auxiliary supply voltage (+2.5V)

0 **N.C.:** Not connected

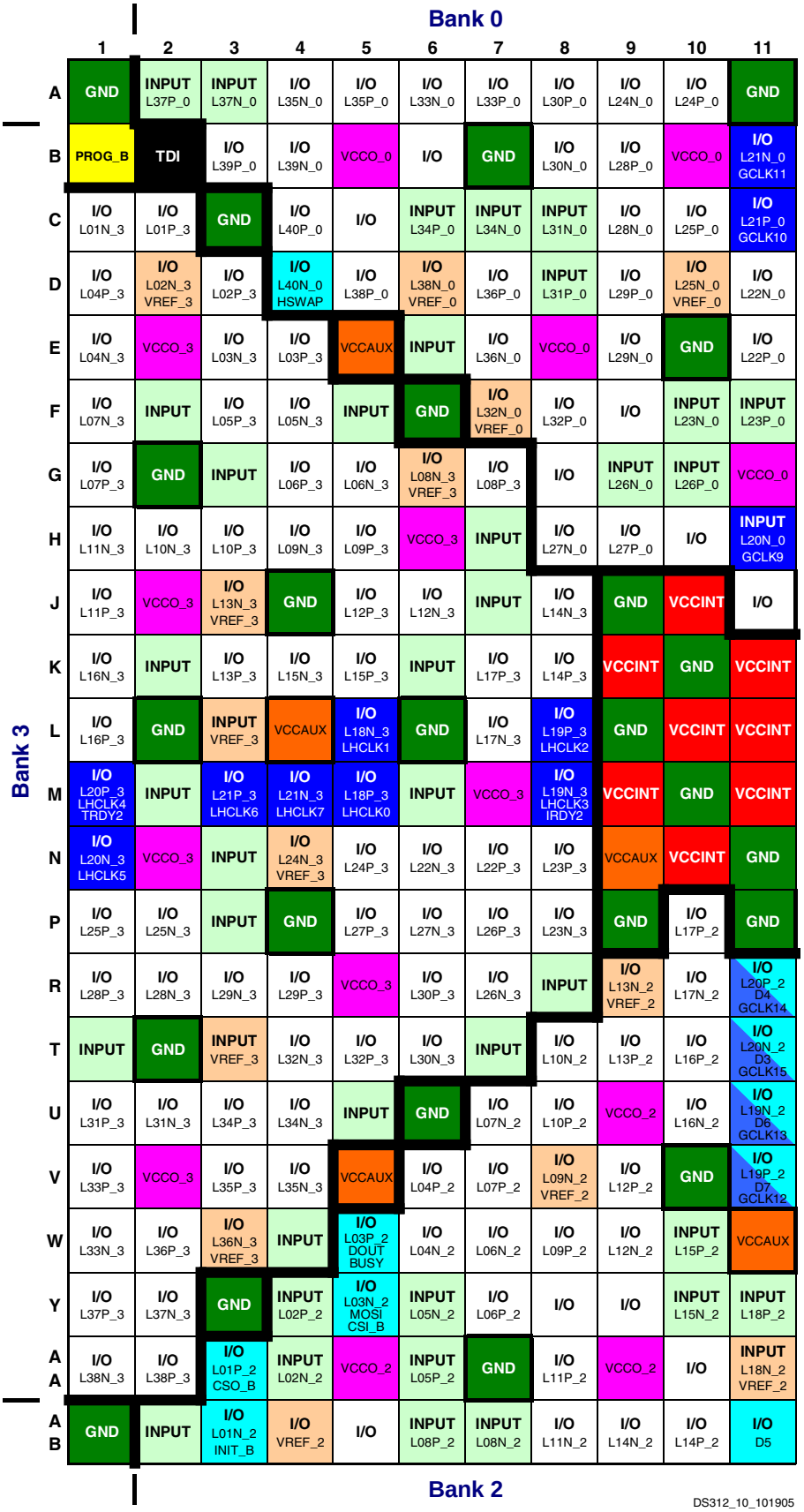


Figure 88: **FG484 Package Footprint (top view)**