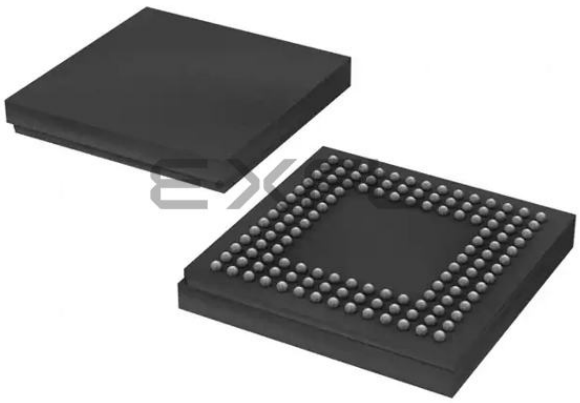

Welcome to E-XFL.COM

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	1164
Number of Logic Elements/Cells	10476
Total RAM Bits	368640
Number of I/O	92
Number of Gates	500000
Voltage - Supply	1.14V ~ 1.26V
Mounting Type	Surface Mount
Operating Temperature	0°C ~ 85°C (TJ)
Package / Case	132-TFBGA, CSPBGA
Supplier Device Package	132-CSPBGA (8x8)
Purchase URL	https://www.e-xfl.com/product-detail/xilinx/xc3s500e-5cpg132c

Architectural Overview

The Spartan-3E family architecture consists of five fundamental programmable functional elements:

- **Configurable Logic Blocks (CLBs)** contain flexible Look-Up Tables (LUTs) that implement logic plus storage elements used as flip-flops or latches. CLBs perform a wide variety of logical functions as well as store data.
- **Input/Output Blocks (IOBs)** control the flow of data between the I/O pins and the internal logic of the device. Each IOB supports bidirectional data flow plus 3-state operation. Supports a variety of signal standards, including four high-performance differential standards. Double Data-Rate (DDR) registers are included.
- **Block RAM** provides data storage in the form of 18-Kbit dual-port blocks.
- **Multiplier Blocks** accept two 18-bit binary numbers as inputs and calculate the product.
- **Digital Clock Manager (DCM) Blocks** provide self-calibrating, fully digital solutions for distributing, delaying, multiplying, dividing, and phase-shifting clock signals.

These elements are organized as shown in Figure 1. A ring of IOBs surrounds a regular array of CLBs. Each device has two columns of block RAM except for the XC3S100E, which has one column. Each RAM column consists of several 18-Kbit RAM blocks. Each block RAM is associated with a dedicated multiplier. The DCMs are positioned in the center with two at the top and two at the bottom of the device. The XC3S100E has only one DCM at the top and bottom, while the XC3S1200E and XC3S1600E add two DCMs in the middle of the left and right sides.

The Spartan-3E family features a rich network of traces that interconnect all five functional elements, transmitting signals among them. Each functional element has an associated switch matrix that permits multiple connections to the routing.

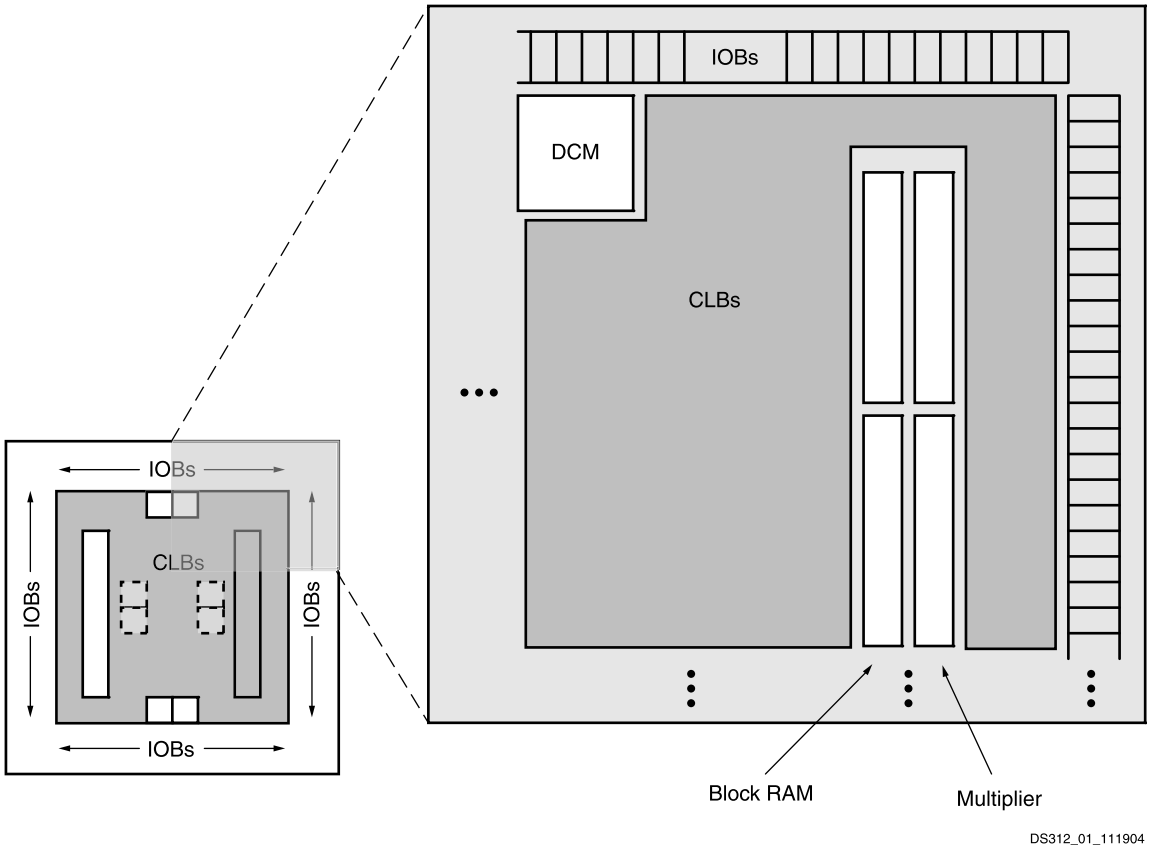

Figure 1: **Spartan-3E Family Architecture**

Introduction

As described in Architectural Overview, the Spartan-3E FPGA architecture consists of five fundamental functional elements:

- Input/Output Blocks (IOBs)
- Configurable Logic Block (CLB) and Slice Resources
- Block RAM
- Dedicated Multipliers
- Digital Clock Managers (DCMs)

The following sections provide detailed information on each of these functions. In addition, this section also describes the following functions:

- Clocking Infrastructure
- Interconnect
- Configuration
- Powering Spartan-3E FPGAs

Input/Output Blocks (IOBs)

For additional information, refer to the “Using I/O Resources” chapter in UG331.

IOB Overview

The Input/Output Block (IOB) provides a programmable, unidirectional or bidirectional interface between a package pin and the FPGA’s internal logic. The IOB is similar to that of the Spartan-3 family with the following differences:

- Input-only blocks are added
- Programmable input delays are added to all blocks
- DDR flip-flops can be shared between adjacent IOBs

The unidirectional input-only block has a subset of the full IOB capabilities. Thus there are no connections or logic for an output path. The following paragraphs assume that any reference to output functionality does not apply to the input-only blocks. The number of input-only blocks varies with device size, but is never more than 25% of the total IOB count.

Figure 5 is a simplified diagram of the IOB’s internal structure. There are three main signal paths within the IOB: the output path, input path, and 3-state path. Each path has its own pair of storage elements that can act as either registers or latches. For more information, see Storage Element Functions. The three main signal paths are as follows:

- The input path carries data from the pad, which is bonded to a package pin, through an optional programmable delay element directly to the I line. After the delay element, there are alternate routes through a pair of storage elements to the IQ1 and IQ2 lines. The IOB outputs I, IQ1, and IQ2 lead to the FPGA’s internal logic. The delay element can be set to ensure a hold time of zero (see Input Delay Functions).
- The output path, starting with the O1 and O2 lines, carries data from the FPGA’s internal logic through a multiplexer and then a three-state driver to the IOB pad. In addition to this direct path, the multiplexer provides the option to insert a pair of storage elements.
- The 3-state path determines when the output driver is high impedance. The T1 and T2 lines carry data from the FPGA’s internal logic through a multiplexer to the output driver. In addition to this direct path, the multiplexer provides the option to insert a pair of storage elements.
- All signal paths entering the IOB, including those associated with the storage elements, have an inverter option. Any inverter placed on these paths is automatically absorbed into the IOB.

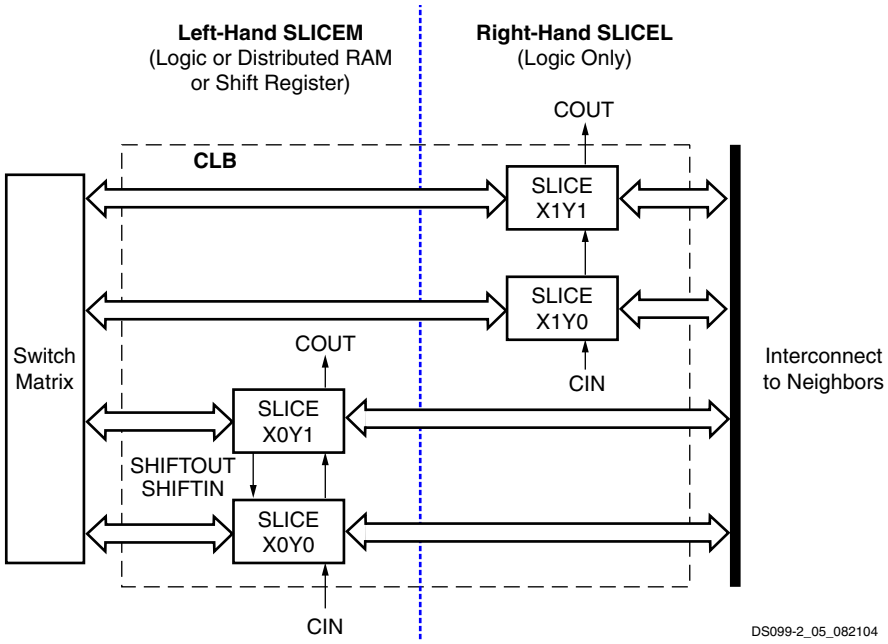

Figure 16: **Arrangement of Slices within the CLB**

Slice Location Designations

The Xilinx development software designates the location of a slice according to its X and Y coordinates, starting in the bottom left corner, as shown in Figure 14. The letter 'X' followed by a number identifies columns of slices, incrementing from the left side of the die to the right. The letter 'Y' followed by a number identifies the position of each slice in a pair as well as indicating the CLB row, incrementing from the bottom of the die. Figure 16 shows the CLB located in the lower left-hand corner of the die. The SLICEM always has an even 'X' number, and the SLICEL always has an odd 'X' number.

Slice Overview

A slice includes two LUT function generators and two storage elements, along with additional logic, as shown in Figure 17.

Both SLICEM and SLICEL have the following elements in common to provide logic, arithmetic, and ROM functions:

- Two 4-input LUT function generators, F and G
- Two storage elements
- Two wide-function multiplexers, F5MUX and FiMUX
- Carry and arithmetic logic

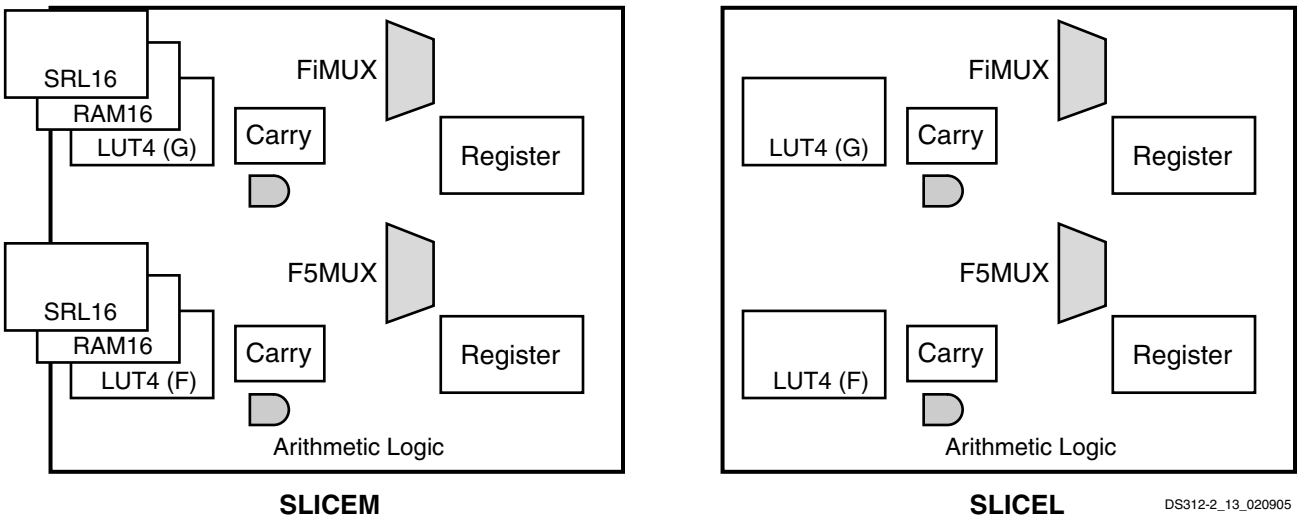

Figure 17: **Resources in a Slice**

Table 22: **Port Aspect Ratios**

Total Data Path Width (w bits)	DI/DO Data Bus Width (w-p bits)(1)	DIP/DOP Parity Bus Width (p bits)	ADDR Bus Width (r bits)(2)	DI/DO [w-p-1:0]	DIP/DOP [p-1:0]	ADDR [r-1:0]	No. of Addressable Locations (n)(3)	Block RAM Capacity (w*n bits)(4)
1	1	0	14	[0:0]	-	[13:0]	16,384	16,384
2	2	0	13	[1:0]	-	[12:0]	8,192	16,384
4	4	0	12	[3:0]	-	[11:0]	4,096	16,384
9	8	1	11	[7:0]	[0:0]	[10:0]	2,048	18,432
18	16	2	10	[15:0]	[1:0]	[9:0]	1,024	18,432
36	32	4	9	[31:0]	[3:0]	[8:0]	512	18,432

Notes:

1. The width of the total data path (w) is the sum of the DI/DO bus width (w-p) and any parity bits (p).
2. The width selection made for the DI/DO bus determines the number of address lines (r) according to the relationship expressed as: r = 14 – [log(w–p)/log9(2)].
3. The number of address lines delimits the total number (n) of addressable locations or depth according to the following equation: $n = 2^r$.
4. The product of w and n yields the total block RAM capacity.

If the data bus width of Port A differs from that of Port B, the block RAM automatically performs a bus-matching function as described in Figure 31. When data is written to a port with a narrow bus and then read from a port with a wide bus, the latter port effectively combines “narrow” words to form “wide” words. Similarly, when data is written into a port with a wide bus and then read from a port with a narrow bus, the latter port divides “wide” words to form “narrow” words. Parity bits are not available if the data port width is configured as x4, x2, or x1. For example, if a x36 data word (32 data, 4 parity) is addressed as two x18 halfwords (16 data, 2 parity), the parity bits associated with each data byte are mapped within the block RAM to the appropriate parity bits. The same effect happens when the x36 data word is mapped as four x9 words.

Table 23: **Block RAM Port Signals**

Signal Description	Port A Signal Name	Port B Signal Name	Direction	Function
Address Bus	ADDRA	ADDRB	Input	The Address Bus selects a memory location for read or write operations. The width (w) of the port's associated data path determines the number of available address lines (r), as per Table 22. Whenever a port is enabled (ENA or ENB = High), address transitions must meet the data sheet setup and hold times with respect to the port clock (CLKA or CLKB), as shown in Table 103, page 138.This requirement must be met even if the RAM read output is of no interest.
Data Input Bus	DIA	DIB	Input	Data at the DI input bus is written to the RAM location specified by the address input bus (ADDR) during the active edge of the CLK input, when the clock enable (EN) and write enable (WE) inputs are active. It is possible to configure a port's DI input bus width (w-p) based on Table 22. This selection applies to both the DI and DO paths of a given port.
Parity Data Input(s)	DIPA	DIPB	Input	Parity inputs represent additional bits included in the data input path. Although referred to herein as "parity" bits, the parity inputs and outputs have no special functionality for generating or checking parity and can be used as additional data bits. The number of parity bits 'p' included in the DI (same as for the DO bus) depends on a port's total data path width (w). See Table 22.
Data Output Bus	DOA	DOB	Output	Data is written to the DO output bus from the RAM location specified by the address input bus, ADDR. See the DI signal description for DO port width configurations. Basic data access occurs on the active edge of the CLK when WE is inactive and EN is active. The DO outputs mirror the data stored in the address ADDR memory location. Data access with WE active if the WRITE_MODE attribute is set to the value: WRITE_FIRST, which accesses data after the write takes place. READ_FIRST accesses data before the write occurs. A third attribute, NO_CHANGE, latches the DO outputs upon the assertion of WE. See Block RAM Data Operations for details on the WRITE_MODE attribute.
Parity Data Output(s)	DOPA	DOPB	Output	Parity outputs represent additional bits included in the data input path. The number of parity bits 'p' included in the DI bus (same as for the DO bus) depends on a port's total data path width (w). See the DIP signal description for configuration details.
Write Enable	WEA	WEB	Input	When asserted together with EN, this input enables the writing of data to the RAM. When WE is inactive with EN asserted, read operations are still possible. In this case, a latch passes data from the addressed memory location to the DO outputs.
Clock Enable	ENA	ENB	Input	When asserted, this input enables the CLK signal to perform read and write operations to the block RAM. When inactive, the block RAM does not perform any read or write operations.
Set/Reset	SSRA	SSRB	Input	When asserted, this pin forces the DO output latch to the value of the SRVAL attribute. It is synchronized to the CLK signal.
Clock	CLKA	CLKB	Input	This input accepts the clock signal to which read and write operations are synchronized. All associated port inputs are required to meet setup times with respect to the clock signal's active edge. The data output bus responds after a clock-to-out delay referenced to the clock signal's active edge.

Block RAM Attribute Definitions

A block RAM has a number of attributes that control its behavior as shown in Table 24.

Table 24: **Block RAM Attributes**

Function	Attribute	Possible Values
Initial Content for Data Memory, Loaded during Configuration	INITxx (INIT_00 through INIT3F)	Each initialization string defines 32 hex values of the 16384-bit data memory of the block RAM.
Initial Content for Parity Memory, Loaded during Configuration	INITPxx (INITP_00 through INITP0F)	Each initialization string defines 32 hex values of the 2048-bit parity data memory of the block RAM.
Data Output Latch Initialization	INIT (single-port) INITA, INITB (dual-port)	Hex value the width of the chosen port.
Data Output Latch Synchronous Set/Reset Value	SRVAL (single-port) SRVAL_A, SRVAL_B (dual-port)	Hex value the width of the chosen port.
Data Output Latch Behavior during Write (see Block RAM Data Operations)	WRITE_MODE	WRITE_FIRST, READ_FIRST, NO_CHANGE

Block RAM Data Operations

Writing data to and accessing data from the block RAM are synchronous operations that take place independently on each of the two ports. Table 25 describes the data operations of each port as a result of the block RAM control signals in their default active-High edges.

The waveforms for the write operation are shown in the top half of Figure 33, Figure 34, and Figure 35. When the WE and EN signals enable the active edge of CLK, data at the DI input bus is written to the block RAM location addressed by the ADDR lines.

Table 25: **Block RAM Function Table**

Input Signals								Output Signals		RAM Data	
GSR	EN	SSR	WE	CLK	ADDR	DIP	DI	DOP	DO	Parity	Data
Immediately After Configuration											
Loaded During Configuration								X	X	INITP_xx	INIT_xx
Global Set/Reset Immediately After Configuration											
1	X	X	X	X	X	X	X	INIT	INIT	No Chg	No Chg
RAM Disabled											
0	0	X	X	X	X	X	X	No Chg	No Chg	No Chg	No Chg
Synchronous Set/Reset											
0	1	1	0	↑	X	X	X	SRVAL	SRVAL	No Chg	No Chg
Synchronous Set/Reset During Write RAM											
0	1	1	1	↑	addr	pdata	Data	SRVAL	SRVAL	RAM(addr) ← pdata	RAM(addr) ← data
Read RAM, no Write Operation											
0	1	0	0	↑	addr	X	X	RAM(pdata)	RAM(data)	No Chg	No Chg
Write RAM, Simultaneous Read Operation											
0	1	0	1	↑	addr	pdata	Data	WRITE_MODE = WRITE_FIRST			
								pdata	data	RAM(addr) ← pdata	RAM(addr) ← data
								WRITE_MODE = READ_FIRST			
								RAM(data)	RAM(data)	RAM(addr) ← pdata	RAM(addr) ← pdata
								WRITE_MODE = NO_CHANGE			
								No Chg	No Chg	RAM(addr) ← pdata	RAM(addr) ← pdata

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.

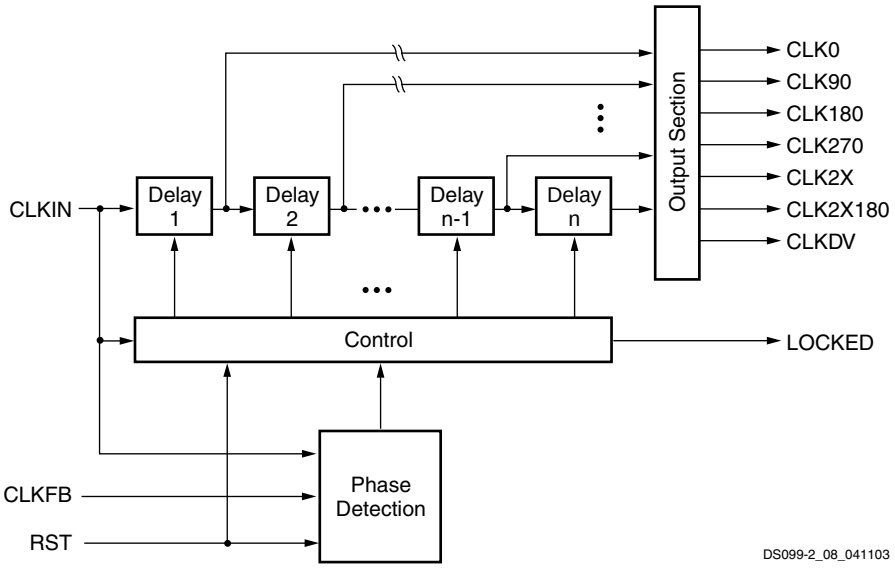

Figure 41: **Simplified Functional Diagram of DLL**

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

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.

Table 55: **Serial Peripheral Interface (SPI) Connections *(Cont'd)***

Pin Name	FPGA Direction	Description	During Configuration	After Configuration
VS[2:0] Ⓢ	Input	**Variant Select**. Instructs the FPGA how to communicate with the attached SPI Flash PROM. See Design Considerations for the HSWAP, M[2:0], and VS[2:0] Pins.	Must be at the logic levels shown in Table 53. Sampled when INIT_B goes High.	User I/O
MOSI	Output	**Serial Data Output**.	FPGA sends SPI Flash memory read commands and starting address to the PROM's serial data input.	User I/O
DIN	Input	**Serial Data Input**.	FPGA receives serial data from PROM's serial data output.	User I/O
CSO_B	Output	**Chip Select Output**. Active Low.	Connects to the SPI Flash PROM's chip-select input. If HSWAP = 1, connect this signal to a 4.7 kΩ pull-up resistor to 3.3V.	Drive CSO_B High after configuration to disable the SPI Flash and reclaim the MOSI, DIN, and CCLK pins. Optionally, re-use this pin and MOSI, DIN, and CCLK to continue communicating with SPI Flash.
CCLK	Output	**Configuration Clock**. Generated by FPGA internal oscillator. Frequency controlled by ***ConfigRate*** bitstream generator option. If CCLK PCB trace is long or has multiple connections, terminate this output to maintain signal integrity. See CCLK Design Considerations.	Drives PROM's clock input.	User I/O
DOUT	Output	**Serial Data Output**.	Actively drives. Not used in single-FPGA designs. In a daisy-chain configuration, this pin connects to DIN input of the next FPGA in the chain.	User I/O
INIT_B	Open-drain bidirectional I/O	**Initialization Indicator**. Active Low. Goes Low at start of configuration during Initialization memory clearing process. Released at end of memory clearing, when mode select pins are sampled. In daisy-chain applications, this signal requires an external 4.7 kΩ pull-up resistor to VCCO_2.	Active during configuration. If SPI Flash PROM requires > 2 ms to awake after powering on, hold INIT_B Low until PROM is ready. If CRC error detected during configuration, FPGA drives INIT_B Low.	User I/O. If unused in the application, drive INIT_B High.
DONE	Open-drain bidirectional I/O	**FPGA Configuration Done**. Low during configuration. Goes High when FPGA successfully completes configuration. Requires external 330 Ω pull-up resistor to 2.5V.	Low indicates that the FPGA is not yet configured.	Pulled High via external pull-up. When High, indicates that the FPGA successfully configured.
PROG_B	Input	**Program FPGA**. Active Low. When asserted Low for 500 ns or longer, forces the FPGA to restart its configuration process by clearing configuration memory and resetting the DONE and INIT_B pins once PROG_B returns High. Recommend external 4.7 kΩ pull-up resistor to 2.5V. Internal pull-up value may be weaker (see Table 78). If driving externally with a 3.3V output, use an open-drain or open-collector driver or use a current limiting series resistor.	Must be High to allow configuration to start.	Drive PROG_B Low and release to reprogram FPGA. Hold PROG_B to force FPGA I/O pins into Hi-Z, allowing direct programming access to SPI Flash PROM pins.

Daisy-Chaining

If the application requires multiple FPGAs with different configurations, then configure the FPGAs using a daisy chain, as shown in Figure 57. Daisy-chaining from a single SPI serial Flash PROM is supported in Stepping 1 devices. It is not supported in Stepping 0 devices. Use SPI Flash mode (M[2:0] = <0:0:1>) for the FPGA connected to the Platform Flash PROM and Slave Serial mode (M[2:0] = <1:1:1>) for all other FPGAs in the daisy-chain. After the master FPGA—the FPGA on the left in the diagram—finishes loading its configuration data from the SPI Flash PROM, the master device uses its DOUT output pin to supply data to the next device in the daisy-chain, on the falling CCLK edge.

Design Note

SPI mode daisy chains are supported only in Stepping 1 silicon versions.

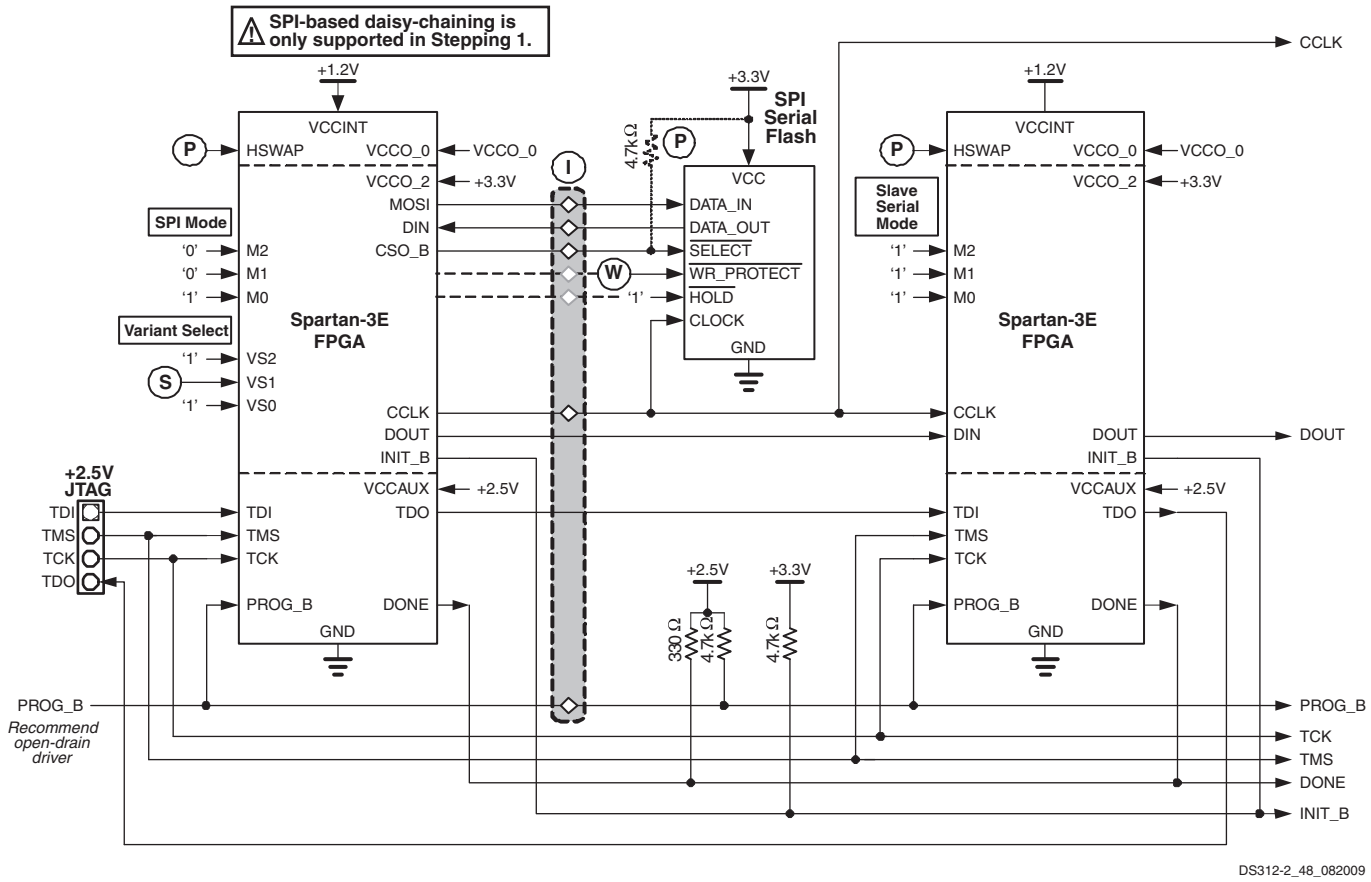

Figure 57: **Daisy-Chaining from SPI Flash Mode (Stepping 1)**

Programming Support

For successful daisy-chaining, the ***DONE_cycle*** configuration option must be set to cycle 5 or sooner. The default cycle is 4. See Table 69 and the Start-Up section for additional information.

(I) In production applications, the SPI Flash PROM is usually pre-programmed before it is mounted on the printed circuit board. The Xilinx ISE development software produces industry-standard programming files that can be used with third-party gang programmers. Consult your specific SPI Flash vendor for recommended production programming solutions.

In-system programming support is available from some third-party PROM programmers using a socket adapter with attached wires. To gain access to the SPI 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 SPI 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 SPI Flash pins. The programming access points are highlighted in the gray box in Figure 53, Figure 54, and Figure 57.

Beginning with the Xilinx ISE 8.2i software release, the iMPACT programming utility provides direct, in-system prototype programming support for STMicro M25P-series

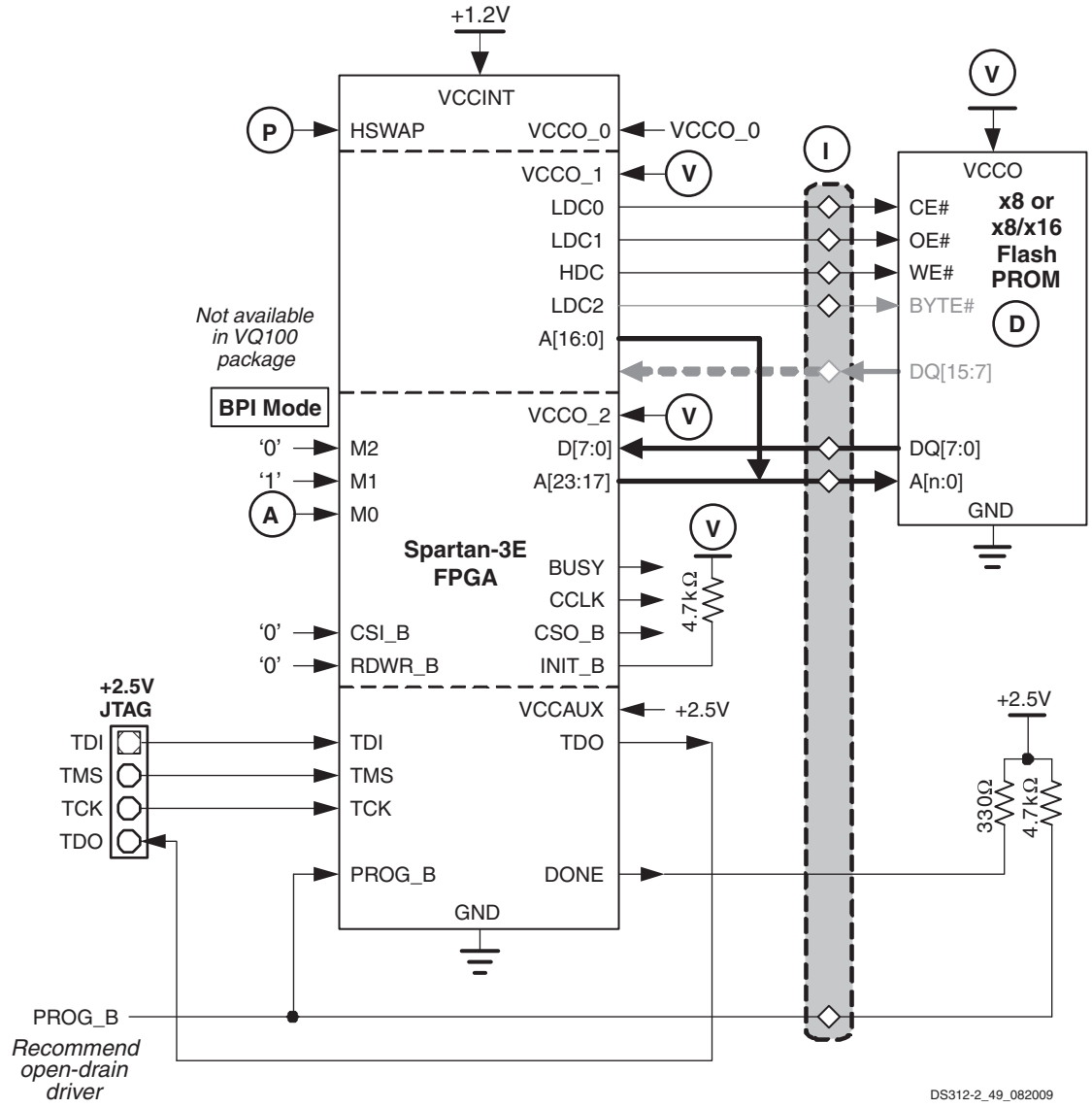

Figure 58: **Byte-wide Peripheral Interface (BPI) Mode Configured from Parallel NOR Flash PROMs**

(A) During configuration, the value of the M0 mode pin determines how the FPGA generates addresses, as shown Table 58. When M0 = 0, the FPGA generates addresses starting at 0 and increments the address on every falling CCLK edge. Conversely, when M0 = 1, the FPGA generates addresses starting at `0xFF_FFFF` (all ones) and decrements the address on every falling CCLK edge.

Table 58: **BPI Addressing Control**

M2	M1	M0	Start Address	Addressing
0	1	0	0	Incrementing
		1	`0xFF_FFFF`	Decrementing

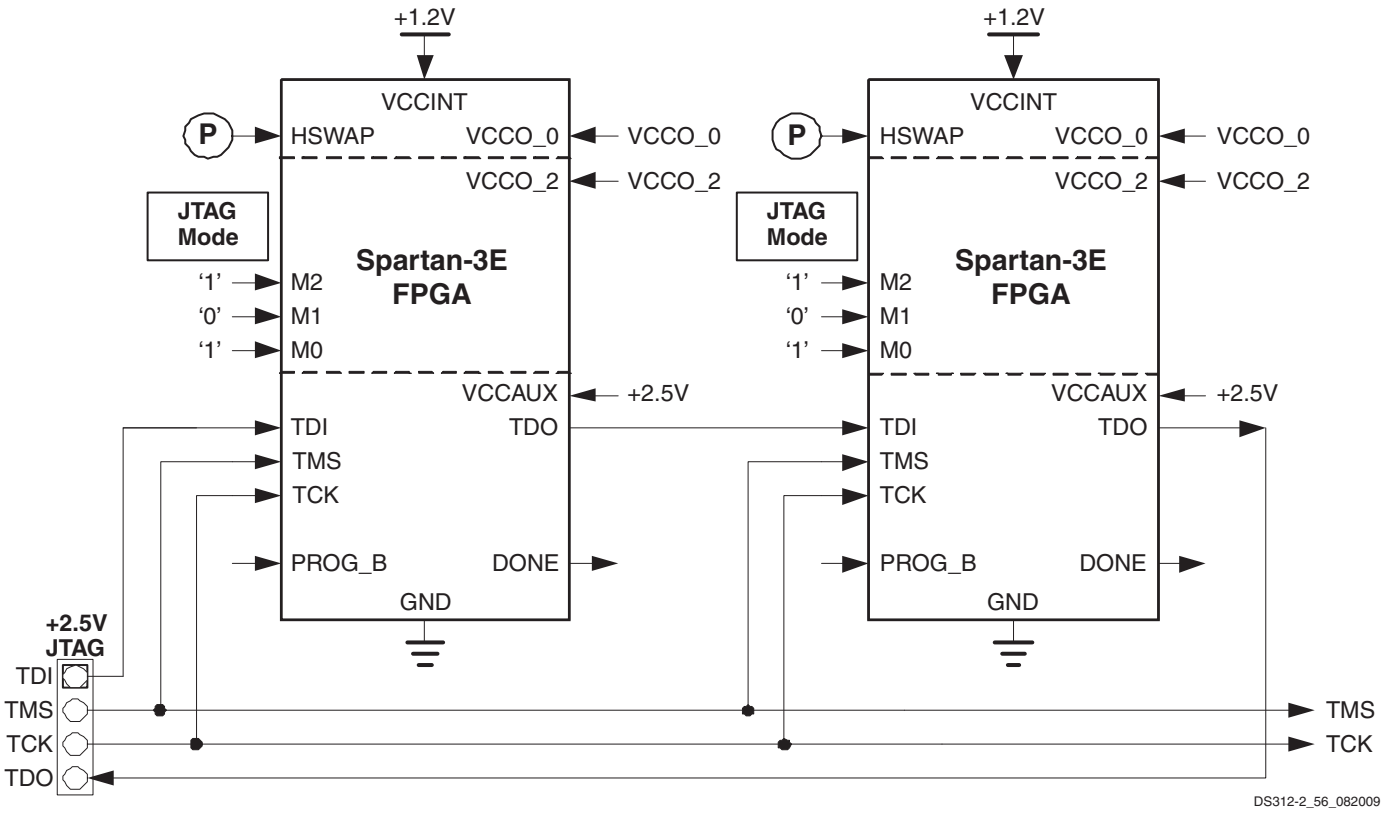

Figure 65: **JTAG Configuration Mode**

Voltage Compatibility

The 2.5V V_{CCAUX} supply powers the JTAG interface. All of the user I/Os are separately powered by their respective VCCO_# supplies.

When connecting the Spartan-3E JTAG port to a 3.3V interface, the JTAG input pins must be current-limited to 10 mA or less using series resistors. Similarly, the TDO pin is a CMOS output powered from +2.5V. The TDO output can directly drive a 3.3V input but with reduced noise immunity. See XAPP453: *The 3.3V Configuration of Spartan-3 FPGAs* for additional information.

Table 67: **Spartan-3E JTAG Device Identifiers**

Spartan-3E FPGA	4-Bit Revision Code		28-Bit Vendor/Device Identifier
	Step 0	Step 1	
XC3S100E	0x0	0x1	0x1C 10 093
XC3S250E	0x0	0x1	0x1C 1A 093
XC3S500E	0x0 0x2	0x4	0x1C 22 093
XC3S1200E	0x0 0x1	0x2	0x1C 2E 093
XC3S1600E	0x0 0x1	0x2	0x1C 3A 093

JTAG Device ID

Each Spartan-3E FPGA array type has a 32-bit device-specific JTAG device identifier as shown in Table 67. The lower 28 bits represent the device vendor (Xilinx) and device identifer. The upper four bits, ignored by most tools, represent the revision level of the silicon mounted on the printed circuit board. Table 67 associates the revision code with a specific stepping level.

JTAG User ID

The Spartan-3E JTAG interface also provides the option to store a 32-bit User ID, loaded during configuration. The User ID value is specified via the ***UserID*** configuration bitstream option, shown in Table 69, page 107.

Using JTAG Interface to Communicate to a Configured FPGA Design

After the FPGA is configured, using any of the available modes, the JTAG interface offers a possible communications channel to internal FPGA logic. The BSCAN_SPARTAN3 design primitive provides two private JTAG instructions to create an internal boundary scan chain.

Voltage Regulators

Various power supply manufacturers offer complete power solutions for Xilinx FPGAs including some with integrated three-rail regulators specifically designed for Spartan-3 and Spartan-3E FPGAs. The Xilinx Power Corner website provides links to vendor solution guides and Xilinx power estimation and analysis tools.

Power Distribution System (PDS) Design and Decoupling/Bypass Capacitors

Good power distribution system (PDS) design is important for all FPGA designs, but especially so for high performance applications, greater than 100 MHz. Proper design results in better overall performance, lower clock and DCM jitter, and a generally more robust system. Before designing the printed circuit board (PCB) for the FPGA design, please review XAPP623: *Power Distribution System (PDS) Design: Using Bypass/Decoupling Capacitors*.

Power-On Behavior

For additional power-on behavior information, including I/O behavior before and during configuration, refer to the "Sequence of Events" chapter in UG332.

Spartan-3E FPGAs have a built-in Power-On Reset (POR) circuit that monitors the three power rails required to successfully configure the FPGA. At power-up, the POR circuit holds the FPGA in a reset state until the V_{CCINT}, V_{CCAUX}, and V_{CCO} Bank 2 supplies reach their respective input threshold levels (see Table 74 in Module 3). After all three supplies reach their respective thresholds, the POR reset is released and the FPGA begins its configuration process.

Supply Sequencing

Because the three FPGA supply inputs must be valid to release the POR reset and can be supplied in any order, there are no FPGA-specific voltage sequencing requirements. Applying the FPGA's V_{CCAUX} supply before the V_{CCINT} supply uses the least I_{CCINT} current.

Although the FPGA has no specific voltage sequence requirements, be sure to consider any potential sequencing requirement of the configuration device attached to the FPGA, such as an SPI serial Flash PROM, a parallel NOR Flash PROM, or a microcontroller. For example, Flash PROMs have a minimum time requirement before the PROM can be selected and this must be considered if the 3.3V supply is the last in the sequence. See Power-On Precautions if 3.3V Supply is Last in Sequence for more details.

When all three supplies are valid, the minimum current required to power-on the FPGA equals the worst-case quiescent current, specified in Table 79. Spartan-3E FPGAs do not require Power-On Surge (POS) current to successfully configure.

Surplus I_{CCINT} if V_{CCINT} Applied before V_{CCAUX}

If the V_{CCINT} supply is applied before the V_{CCAUX} supply, the FPGA might draw a surplus I_{CCINT} current in addition to the I_{CCINT} quiescent current levels specified in Table 79, page 118. The momentary additional I_{CCINT} surplus current might be a few hundred milliamperes under nominal conditions, significantly less than the instantaneous current consumed by the bypass capacitors at power-on. However, the surplus current immediately disappears when the V_{CCAUX} supply is applied, and, in response, the FPGA's I_{CCINT} quiescent current demand drops to the levels specified in Table 79. The FPGA does not use or require the surplus current to successfully power-on and configure. If applying V_{CCINT} before V_{CCAUX}, ensure that the regulator does not have a foldback feature that could inadvertently shut down in the presence of the surplus current.

Configuration Data Retention, Brown-Out

The FPGA's configuration data is stored in robust CMOS configuration latches. The data in these latches is retained even when the voltages drop to the minimum levels necessary to preserve RAM contents, as specified in Table 76.

If, after configuration, the V_{CCAUX} or V_{CCINT} supply drops below its data retention voltage, the current device configuration must be cleared using one of the following methods:

- Force the V_{CCAUX} or V_{CCINT} supply voltage below the minimum Power On Reset (POR) voltage threshold (Table 74).
- Assert PROG_B Low.

The POR circuit does not monitor the VCCO_2 supply after configuration. Consequently, dropping the VCCO_2 voltage does not reset the device by triggering a Power-On Reset (POR) event.

No Internal Charge Pumps or Free-Running Oscillators

Some system applications are sensitive to sources of analog noise. Spartan-3E FPGA circuitry is fully static and does not employ internal charge pumps.

The CCLK configuration clock is active during the FPGA configuration process. After configuration completes, the CCLK oscillator is automatically disabled unless the Bitstream Generator (BitGen) option ***Persist=Yes***.

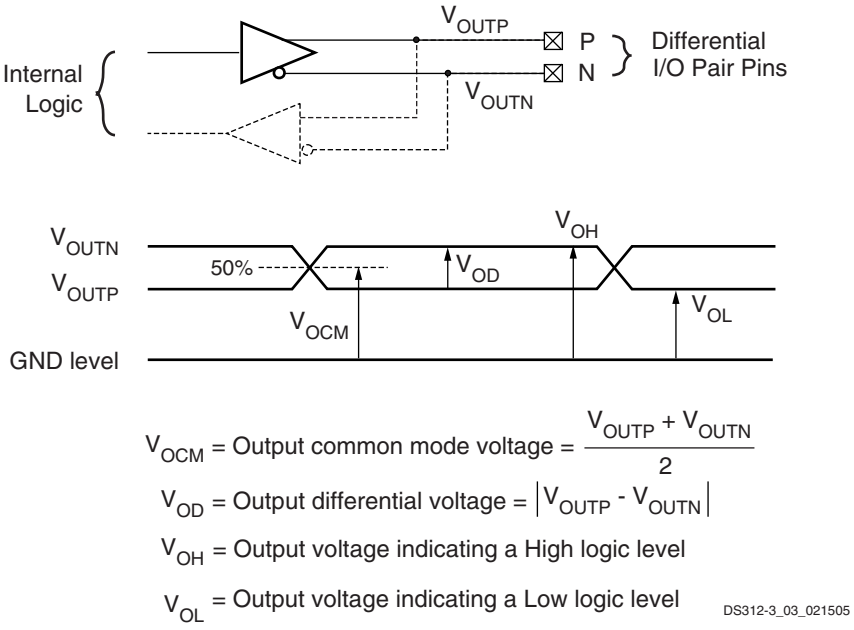

Figure 70: **Differential Output Voltages**

Table 83: **DC Characteristics of User I/Os Using Differential Signal Standards**

IOSTANDARD Attribute	V_{OD}			ΔV_{OD}		V_{OCM}			ΔV_{OCM}		V_{OH}	V_{OL}
	Min (mV)	Typ (mV)	Max (mV)	Min (mV)	Max (mV)	Min (V)	Typ (V)	Max (V)	Min (mV)	Max (mV)	Min (V)	Max (V)
LVDS_25	250	350	450	–	–	1.125	–	1.375	–	–	–	–
BLVDS_25	250	350	450	–	–	–	1.20	–	–	–	–	–
MINI_LVDS_25	300	–	600	–	50	1.0	–	1.4	–	50	–	–
RSDS_25	100	–	400	–	–	1.1	–	1.4	–	–	–	–
DIFF_HSTL_I_18	–	–	–	–	–	–	–	–	–	–	V_{CCO} – 0.4	0.4
DIFF_HSTL_III_18	–	–	–	–	–	–	–	–	–	–	V_{CCO} – 0.4	0.4
DIFF_SSTL18_I	–	–	–	–	–	–	–	–	–	–	V_{TT} + 0.475	V_{TT} – 0.475
DIFF_SSTL2_I	–	–	–	–	–	–	–	–	–	–	V_{TT} + 0.61	V_{TT} – 0.61

Notes:

1. The numbers in this table are based on the conditions set forth in Table 77 and Table 82.
2. Output voltage measurements for all differential standards are made with a termination resistor (R_T) of 100Ω across the N and P pins of the differential signal pair. The exception is for BLVDS, shown in Figure 71.
3. At any given time, no more than two of the following differential output standards may be assigned to an I/O bank: LVDS_25, RSDS_25, MINI_LVDS_25

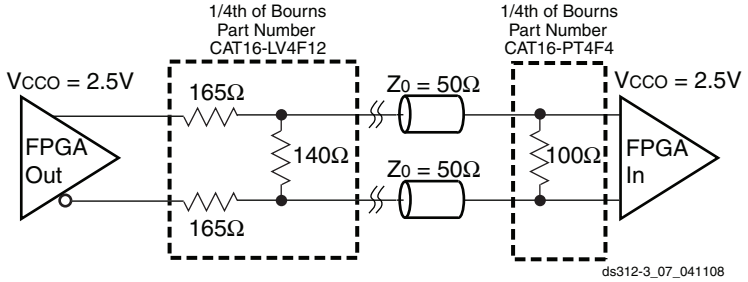

Figure 71: **External Termination Resistors for BLVDS Transmitter and BLVDS Receiver**

Table 117: **Timing for the Slave Parallel Configuration Mode *(Cont'd)***

Symbol	Description			All Speed Grades		Units
				Min	Max	
Clock Timing						
T_{CCH}	The High pulse width at the CCLK input pin			5	-	ns
T_{CCL}	The Low pulse width at the CCLK input pin			5	-	ns
F_{CCPAR}	Frequency of the clock signal at the CCLK input pin	No bitstream compression	Not using the BUSY pin(2)	0	50	MHz
			Using the BUSY pin	0	66	MHz
		With bitstream compression		0	20	MHz

Notes:

1. The numbers in this table are based on the operating conditions set forth in Table 77.
2. In the Slave Parallel mode, it is necessary to use the BUSY pin when the CCLK frequency exceeds this maximum specification.
3. Some Xilinx documents refer to Parallel modes as "SelectMAP" modes.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
03/01/05	1.0	Initial Xilinx release.
11/23/05	2.0	Added AC timing information and additional DC specifications.
03/22/06	3.0	Upgraded data sheet status to Preliminary. Finalized production timing parameters. All speed grades for all Spartan-3E FPGAs are now Production status using the v1.21 speed files, as shown in Table 84. Expanded description in Note 2, Table 78. Updated pin-to-pin and clock-to-output timing based on final characterization, shown in Table 86. Updated system-synchronous input setup and hold times based on final characterization, shown in Table 87 and Table 88. Updated other I/O timing in Table 90. Provided input and output adjustments for LVPECL_25, DIFF_SSTL and DIFF_HSTL I/O standards that supersede the v1.21 speed file values, in Table 91 and Table 94. Reduced I/O three-state and set/reset delays in Table 93. Added XC3S100E FPGA in CP132 package to Table 96. Increased T_{AS} slice flip-flop timing by 100 ps in Table 98. Updated distributed RAM timing in Table 99 and SRL16 timing in Table 100. Updated global clock timing, removed left/right clock buffer limits in Table 101. Updated block RAM timing in Table 103. Added DCM parameters for remainder of Step 0 device; added improved Step 1 DCM performance to Table 104, Table 105, Table 106, and Table 107. Added minimum INIT_B pulse width specification, T_{INIT}, in Table 111. Increased data hold time for Slave Parallel mode to 1.0 ns (T_{SMCCD}) in Table 117. Improved the DCM performance for the XC3S1200E, Stepping 0 in Table 104, Table 105, Table 106, and Table 107. Corrected links in Table 118 and Table 120. Added MultiBoot timing specifications to Table 122.
04/07/06	3.1	Improved SSO limits for LVDS_25, MINI_LVDS_25, and RSDS_25 I/O standards in the QFP packages (Table 97). Removed potentially confusing Note 2 from Table 78.
05/19/06	3.2	Clarified that 100 mV of hysteresis applies to LVCMOS33 and LVCMOS25 I/O standards (Note 4, Table 80). Other minor edits.
05/30/06	3.2.1	Corrected various typos and incorrect links.
11/09/06	3.4	Improved absolute maximum voltage specifications in Table 73, providing additional overshoot allowance. Widened the recommended voltage range for PCI and PCI-X standards in Table 80. Clarified Note 2, Table 83. Improved various timing specifications for v1.26 speed file. Added Table 85 to summarize the history of speed file releases after which time all devices became Production status. Added absolute minimum values for Table 86, Table 92, and Table 93. Updated pin-to-pin setup and hold timing based on default IFD_DELAY_VALUE settings in Table 87, Table 88, and Table 90. Added Table 89 about source-synchronous input capture sample window. Promoted Module 3 to Production status. Synchronized all modules to v3.4.
03/16/07	3.5	Based on extensive 90 nm production data, improved (reduced) the maximum quiescent current limits for the I_{CCINTQ}, I_{CCAUXQ}, and I_{CCOQ} specifications in Table 79 by an average of 50%.
05/29/07	3.6	Added note to Table 74 and Table 75 regarding HSWAP in step 0 devices. Updated t_{RPW_CLB} in Table 98 to match value in speed file. Improved CLKOUT_FREQ_CLK90 to 200 MHz for Stepping 1 in Table 105.
04/18/08	3.7	Clarified that Stepping 0 was offered only for -4C and removed Stepping 0 -5 specifications. Added reference to XAPP459 in Table 73 and Table 77. Improved recommended max V_{CCO} to 3.465V (3.3V + 5%) in Table 77. Removed minimum input capacitance from Table 78. Updated Recommended Operating Conditions for LVCMOS and PCI I/O standards in Table 80. Removed Absolute Minimums from Table 86, Table 92 and Table 93 and added footnote recommending use of Timing Analyzer for minimum values. Updated T_{PSFD} and T_{PHFD} in Table 87 to match current speed file. Update T_{RPW_IOB} in Table 88 to match current speed file and CLB equivalent spec. Added XC3S500E VQG100 to Table 96. Replaced $T_{MULCKID}$ with T_{MSCKD} for A, B, and P registers in Table 102. Updated CLKOUT_PER_JITT_FX in Table 107. Updated MAX_STEPS equation in Table 109. Updated Figure 77 and Table 120 to correct CCLK active edge. Updated links.

Table 137: **TQ144 Package Pinout *(Cont'd)***

Bank	XC3S100E Pin Name	XC3S250E Pin Name	TQ144 Pin	Type
GND	GND	GND	P27	GND
GND	GND	GND	P37	GND
GND	GND	GND	P46	GND
GND	GND	GND	P55	GND
GND	GND	GND	P61	GND
GND	GND	GND	P73	GND
GND	GND	GND	P90	GND
GND	GND	GND	P99	GND
GND	GND	GND	P118	GND
GND	GND	GND	P127	GND
GND	GND	GND	P133	GND
VCCAUX	DONE	DONE	P72	CONFIG
VCCAUX	PROG_B	PROG_B	P1	CONFIG
VCCAUX	TCK	TCK	P110	JTAG
VCCAUX	TDI	TDI	P144	JTAG
VCCAUX	TDO	TDO	P109	JTAG
VCCAUX	TMS	TMS	P108	JTAG
VCCAUX	VCCAUX	VCCAUX	P30	VCCAUX
VCCAUX	VCCAUX	VCCAUX	P65	VCCAUX
VCCAUX	VCCAUX	VCCAUX	P102	VCCAUX
VCCAUX	VCCAUX	VCCAUX	P137	VCCAUX
VCCINT	VCCINT	VCCINT	P9	VCCINT
VCCINT	VCCINT	VCCINT	P45	VCCINT
VCCINT	VCCINT	VCCINT	P80	VCCINT
VCCINT	VCCINT	VCCINT	P115	VCCINT

User I/Os by Bank

Table 142 indicates how the 158 available user-I/O pins are distributed between the four I/O banks on the PQ208 package.

Footprint Migration Differences

The XC3S250E and XC3S500E FPGAs have identical footprints in the PQ208 package. Designs can migrate between the XC3S250E and XC3S500E without further consideration.

Table 142: **User I/Os Per Bank for the XC3S250E and XC3S500E in the PQ208 Package**

Package Edge	I/O Bank	Maximum I/O	All Possible I/O Pins by Type				
			I/O	INPUT	DUAL	VREF(1)	CLK(2)
Top	0	38	18	6	1	5	8
Right	1	40	9	7	21	3	0(2)
Bottom	2	40	8	6	24	2	0(2)
Left	3	40	23	6	0	3	8
TOTAL		**158**	**58**	**25**	**46**	**13**	**16**

Notes:

1. Some VREF and CLK pins are on INPUT pins.
2. The eight global clock pins in this bank have optional functionality during configuration and are counted in the DUAL column.

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

Bank	XC3S250E Pin Name	XC3S500E Pin Name	XC3S1200E Pin Name	FT256 Ball	Type
3	IP	IP	IP	N3	INPUT
3	IP/VREF_3	IP/VREF_3	IP/VREF_3	G1	VREF
3	IO/VREF_3	IO/VREF_3	IP/VREF_3	N2	***250E:*** VREF(I/O) ***500E:*** VREF(I/O) ***1200E:*** VREF(INPUT)
3	VCCO_3	VCCO_3	VCCO_3	E2	VCCO
3	VCCO_3	VCCO_3	VCCO_3	G6	VCCO
3	VCCO_3	VCCO_3	VCCO_3	K6	VCCO
3	VCCO_3	VCCO_3	VCCO_3	M2	VCCO
GND	GND	GND	GND	A1	GND
GND	GND	GND	GND	A16	GND
GND	GND	GND	GND	B9	GND
GND	GND	GND	GND	F6	GND
GND	GND	GND	GND	F11	GND
GND	GND	GND	GND	G7	GND
GND	GND	GND	GND	G8	GND
GND	GND	GND	GND	G9	GND
GND	GND	GND	GND	G10	GND
GND	GND	GND	GND	H2	GND
GND	GND	GND	GND	H7	GND
GND	GND	GND	GND	H8	GND
GND	GND	GND	GND	H9	GND
GND	GND	GND	GND	H10	GND
GND	GND	GND	GND	J7	GND
GND	GND	GND	GND	J8	GND
GND	GND	GND	GND	J9	GND
GND	GND	GND	GND	J10	GND
GND	GND	GND	GND	J15	GND
GND	GND	GND	GND	K7	GND
GND	GND	GND	GND	K8	GND
GND	GND	GND	GND	K9	GND
GND	GND	GND	GND	K10	GND
GND	GND	GND	GND	L6	GND
GND	GND	GND	GND	L11	GND
GND	GND	GND	GND	R8	GND
GND	GND	GND	GND	T1	GND
GND	GND	GND	GND	T16	GND
VCCAUX	DONE	DONE	DONE	T15	CONFIG
VCCAUX	PROG_B	PROG_B	PROG_B	D3	CONFIG
VCCAUX	TCK	TCK	TCK	A15	JTAG
VCCAUX	TDI	TDI	TDI	A2	JTAG
VCCAUX	TDO	TDO	TDO	C14	JTAG
VCCAUX	TMS	TMS	TMS	B15	JTAG
VCCAUX	VCCAUX	VCCAUX	VCCAUX	A6	VCCAUX

FG320 Footprint

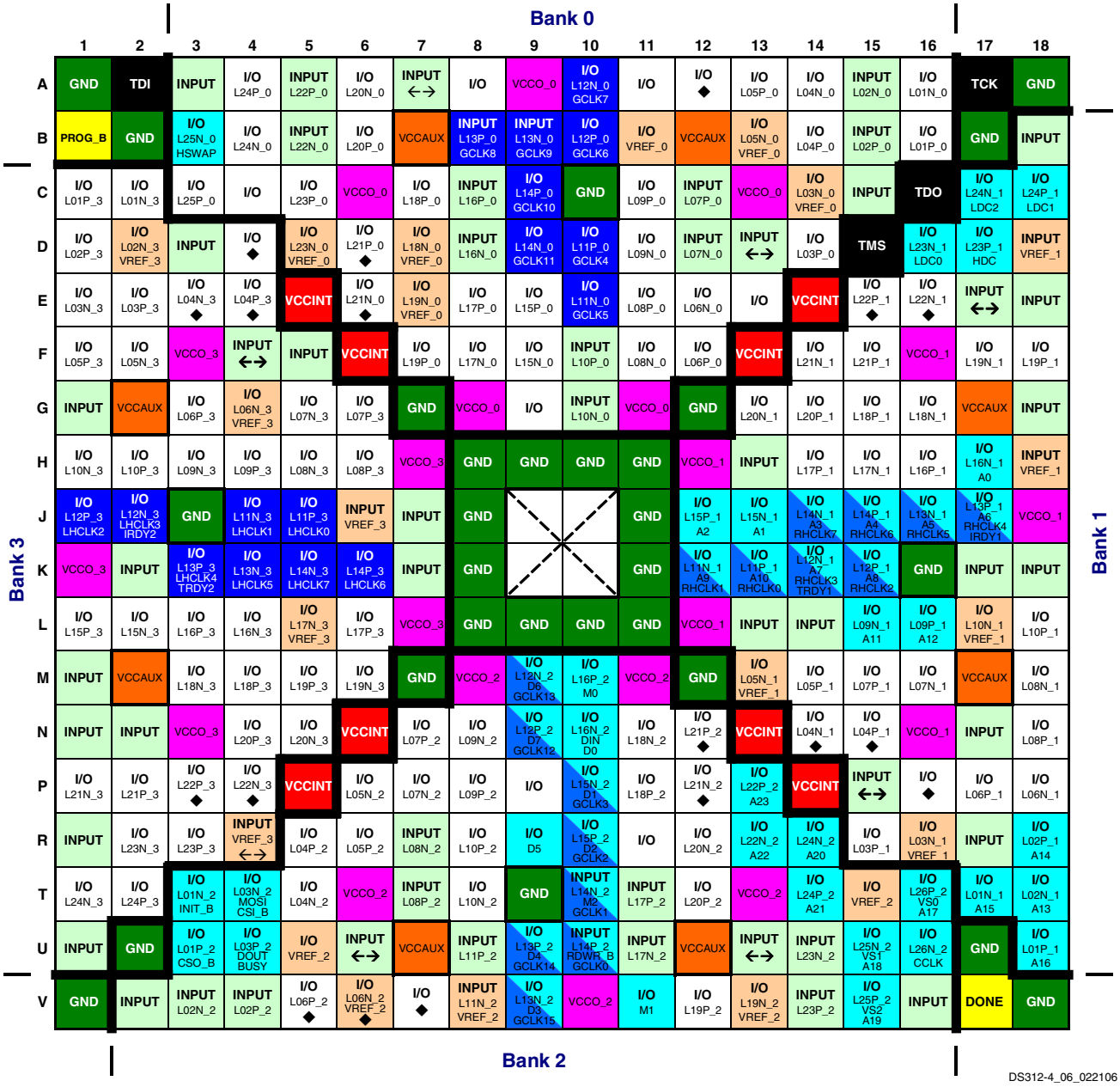

Figure 86: **FG320 Package Footprint (top view)**

102-120	**I/O:** Unrestricted, general-purpose user I/O	46	**DUAL:** Configuration pin, then possible user-I/O	20-21	**VREF:** User I/O or input voltage reference for bank
47-48	**INPUT:** Unrestricted, general-purpose input pin	16	**CLK:** User I/O, input, or global buffer input	20	**VCCO:** Output voltage supply for bank
2	**CONFIG:** Dedicated configuration pins	4	**JTAG:** Dedicated JTAG port pins	8	**VCCINT:** Internal core supply voltage (+1.2V)
18	**N.C.:** Not connected. Only the XC3S500E has these pins (◆).	28	**GND:** Ground	8	**VCCAUX:** Auxiliary supply voltage (+2.5V)