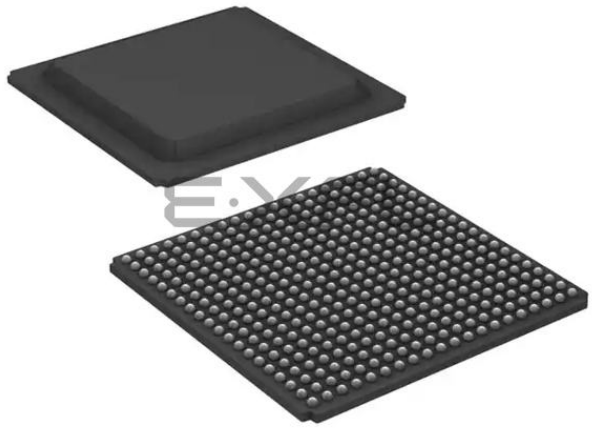

Welcome to E-XFL.COM

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	304
Number of Gates	1200000
Voltage - Supply	1.14V ~ 1.26V
Mounting Type	Surface Mount
Operating Temperature	-40°C ~ 100°C (TJ)
Package / Case	400-BGA
Supplier Device Package	400-FBGA (21x21)
Purchase URL	https://www.e-xfl.com/product-detail/xilinx/xc3s1200e-4fgg400i

Input Delay Functions

Each IOB has a programmable delay block that optionally delays the input signal. In Figure 6, the signal path has a coarse delay element that can be bypassed. The input signal then feeds a 6-tap delay line. The coarse and tap delays vary; refer to timing reports for specific delay values. All six taps are available via a multiplexer for use as an asynchronous input directly into the FPGA fabric. In this way, the delay is programmable in 12 steps. Three of the six taps are also available via a multiplexer to the D inputs of the synchronous storage elements. The delay inserted in the path to the storage element can be varied in six steps. The first, coarse delay element is common to both asynchronous and synchronous paths, and must be either used or not used for both paths.

The delay values are set up in the silicon once at configuration time—they are non-modifiable in device operation.

The primary use for the input delay element is to adjust the input delay path to ensure that there is no hold time requirement when using the input flip-flop(s) with a global clock. The default value is chosen automatically by the Xilinx software tools as the value depends on device size and the specific device edge where the flip-flop resides. The value set by the Xilinx ISE software is indicated in the Map report generated by the implementation tools, and the resulting effects on input timing are reported using the Timing Analyzer tool.

If the design uses a DCM in the clock path, then the delay element can be safely set to zero because the Delay-Locked Loop (DLL) compensation automatically ensures that there is still no input hold time requirement.

Both asynchronous and synchronous values can be modified, which is useful where extra delay is required on clock or data inputs, for example, in interfaces to various types of RAM.

These delay values are defined through the IBUF_DELAY_VALUE and the IFD_DELAY_VALUE parameters. The default IBUF_DELAY_VALUE is 0, bypassing the delay elements for the asynchronous input. The user can set this parameter to 0-12. The default IFD_DELAY_VALUE is AUTO. IBUF_DELAY_VALUE and IFD_DELAY_VALUE are independent for each input. If the same input pin uses both registered and non-registered input paths, both parameters can be used, but they must both be in the same half of the total delay (both either bypassing or using the coarse delay element).

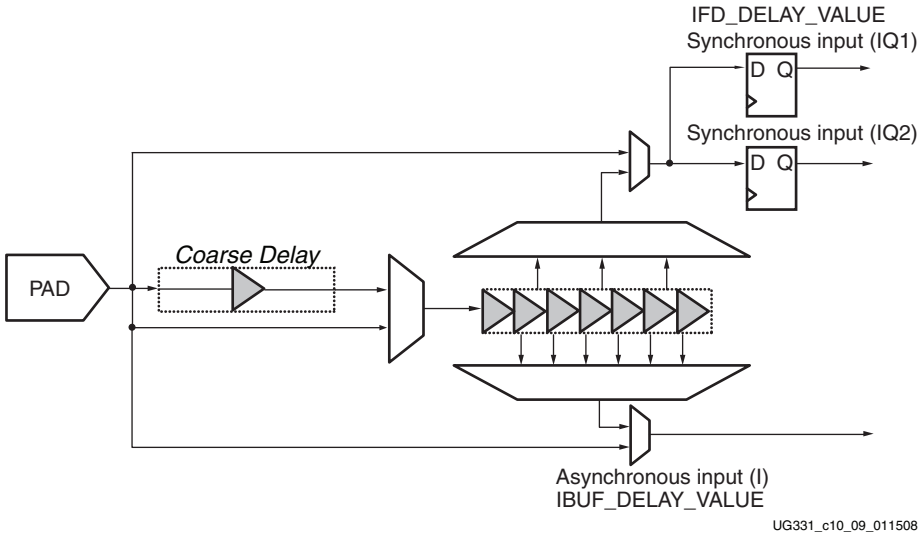

Figure 6: **Programmable Fixed Input Delay Elements**

The SLICEM pair supports two additional functions:

- Two 16x1 distributed RAM blocks, RAM16
- Two 16-bit shift registers, SRL16

Each of these elements is described in more detail in the following sections.

Logic Cells

The combination of a LUT and a storage element is known as a "Logic Cell". The additional features in a slice, such as the wide multiplexers, carry logic, and arithmetic gates, add to the capacity of a slice, implementing logic that would otherwise require additional LUTs. Benchmarks have shown that the overall slice is equivalent to 2.25 simple logic cells. This calculation provides the equivalent logic cell count shown in Table 9.

Slice Details

Figure 15 is a detailed diagram of the SLICEM. It represents a superset of the elements and connections to be found in all slices. The dashed and gray lines (blue when viewed in color) indicate the resources found only in the SLICEM and not in the SLICEL.

Each slice has two halves, which are differentiated as top and bottom to keep them distinct from the upper and lower slices in a CLB. The control inputs for the clock (CLK), Clock Enable (CE), Slice Write Enable (SLICEWE1), and Reset/Set (RS) are shared in common between the two halves.

The LUTs located in the top and bottom portions of the slice are referred to as "G" and "F", respectively, or the "G-LUT" and the "F-LUT". The storage elements in the top and bottom portions of the slice are called FFY and FFX, respectively.

Each slice has two multiplexers with F5MUX in the bottom portion of the slice and FiMUX in the top portion. Depending on the slice, the FiMUX takes on the name F6MUX, F7MUX, or F8MUX, according to its position in the multiplexer chain. The lower SLICEL and SLICEM both have an F6MUX. The upper SLICEM has an F7MUX, and the upper SLICEL has an F8MUX.

The carry chain enters the bottom of the slice as CIN and exits at the top as COUT. Five multiplexers control the chain: CYINIT, CY0F, and CYMUXF in the bottom portion and CY0G and CYMUXG in the top portion. The dedicated arithmetic logic includes the exclusive-OR gates XORF and XORG (bottom and top portions of the slice, respectively) as well as the AND gates FAND and GAND (bottom and top portions, respectively).

See Table 10 for a description of all the slice input and output signals.

Table 10: **Slice Inputs and Outputs**

Name	Location	Direction	Description
F[4:1]	SLICEL/M Bottom	Input	F-LUT and FAND inputs
G[4:1]	SLICEL/M Top	Input	G-LUT and GAND inputs or Write Address (SLICEM)
BX	SLICEL/M Bottom	Input	Bypass to or output (SLICEM) or storage element, or control input to F5MUX, input to carry logic, or data input to RAM (SLICEM)
BY	SLICEL/M Top	Input	Bypass to or output (SLICEM) or storage element, or control input to FiMUX, input to carry logic, or data input to RAM (SLICEM)
BXOUT	SLICEM Bottom	Output	BX bypass output
BYOUT	SLICEM Top	Output	BY bypass output
ALTDIG	SLICEM Top	Input	Alternate data input to RAM
DIG	SLICEM Top	Output	ALTDIG or SHIFTIN bypass output
SLICEWE1	SLICEM Common	Input	RAM Write Enable
F5	SLICEL/M Bottom	Output	Output from F5MUX; direct feedback to FiMUX
FXINA	SLICEL/M Top	Input	Input to FiMUX; direct feedback from F5MUX or another FiMUX
FXINB	SLICEL/M Top	Input	Input to FiMUX; direct feedback from F5MUX or another FiMUX
Fi	SLICEL/M Top	Output	Output from FiMUX; direct feedback to another FiMUX
CE	SLICEL/M Common	Input	FFX/Y Clock Enable
SR	SLICEL/M Common	Input	FFX/Y Set or Reset or RAM Write Enable (SLICEM)
CLK	SLICEL/M Common	Input	FFX/Y Clock or RAM Clock (SLICEM)
SHIFTIN	SLICEM Top	Input	Data input to G-LUT RAM

Table 14: **Carry Logic Functions *(Cont'd)***

Function	Description
CY0G	Carry generation for top half of slice. Fixed selection of: · G1 or G2 inputs to the LUT (both equal 1 when a carry is to be generated) · GAND gate for multiplication · BY input for carry initialization · Fixed 1 or 0 input for use as a simple Boolean function
CYMUXF	Carry generation or propagation mux for bottom half of slice. Dynamic selection via CYSELF of: · CYINIT carry propagation (CYSELF = 1) · CY0F carry generation (CYSELF = 0)
CYMUXG	Carry generation or propagation mux for top half of slice. Dynamic selection via CYSELF of: · CYMUXF carry propagation (CYSELG = 1) · CY0G carry generation (CYSELG = 0)
CYSELF	Carry generation or propagation select for bottom half of slice. Fixed selection of: · F-LUT output (typically XOR result) · Fixed 1 to always propagate
CYSELG	Carry generation or propagation select for top half of slice. Fixed selection of: · G-LUT output (typically XOR result) · Fixed 1 to always propagate
XORF	Sum generation for bottom half of slice. Inputs from: · F-LUT · CYINIT carry signal from previous stage Result is sent to either the combinatorial or registered output for the top of the slice.
XORG	Sum generation for top half of slice. Inputs from: · G-LUT · CYMUXF carry signal from previous stage Result is sent to either the combinatorial or registered output for the top of the slice.
FAND	Multiplier partial product for bottom half of slice. Inputs: · F-LUT F1 input · F-LUT F2 input Result is sent through CY0F to become the carry generate signal into CYMUXF
GAND	Multiplier partial product for top half of slice. Inputs: · G-LUT G1 input · G-LUT G2 input Result is sent through CY0G to become the carry generate signal into CYMUXG

The basic usage of the carry logic is to generate a half-sum in the LUT via an XOR function, which generates or propagates a carry out COUT via the carry mux CYMUXF (or CYMUXG), and then complete the sum with the dedicated XORF (or XORG) gate and the carry input CIN. This structure allows two bits of an arithmetic function in each slice. The CYMUXF (or CYMUXG) can be instantiated using the MUXCY element, and the XORF (or XORG) can be instantiated using the XORCY element.

The FAND (or GAND) gate is used for partial product multiplication and can be instantiated using the MULT_AND component. Partial products are generated by two-input AND gates and then added. The carry logic is efficient for the adder, but one of the inputs must be outside the LUT as shown in Figure 23.

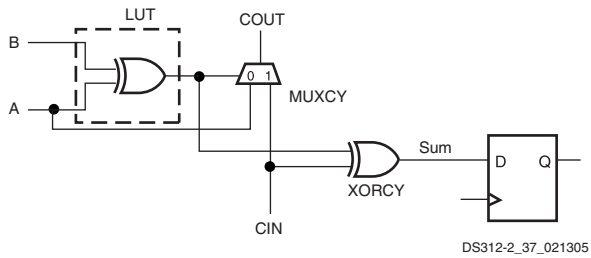

Figure 23: **Using the MUXCY and XORCY in the Carry Logic**

The FAND (or GAND) gate is used to duplicate one of the partial products, while the LUT generates both partial products and the XOR function, as shown in Figure 24.

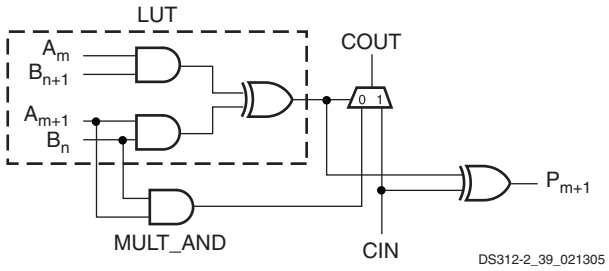

Figure 24: **Using the MULT_AND for Multiplication in Carry Logic**

The MULT_AND is useful for small multipliers. Larger multipliers can be built using the dedicated 18x18 multiplier blocks (see Dedicated Multipliers).

Storage Elements

The storage element, which is programmable as either a D-type flip-flop or a level-sensitive transparent latch, provides a means for synchronizing data to a clock signal, among other uses. The storage elements in the top and bottom portions of the slice are called FFY and FFX, respectively. FFY has a fixed multiplexer on the D input selecting either the combinatorial output Y or the bypass signal BY. FFX selects between the combinatorial output X or the bypass signal BX.

The functionality of a slice storage element is identical to that described earlier for the I/O storage elements. All signals have programmable polarity; the default active-High function is described.

Table 15: **Storage Element Signals**

Signal	Description
D	Input. For a flip-flop data on the D input is loaded when R and S (or CLR and PRE) are Low and CE is High during the Low-to-High clock transition. For a latch, Q reflects the D input while the gate (G) input and gate enable (GE) are High and R and S (or CLR and PRE) are Low. The data on the D input during the High-to-Low gate transition is stored in the latch. The data on the Q output of the latch remains unchanged as long as G or GE remains Low.
Q	Output. Toggles after the Low-to-High clock transition for a flip-flop and immediately for a latch.
C	Clock for edge-triggered flip-flops.
G	Gate for level-sensitive latches.
CE	Clock Enable for flip-flops.
GE	Gate Enable for latches.
S	Synchronous Set (Q = High). When the S input is High and R is Low, the flip-flop is set, output High, during the Low-to-High clock (C) transition. A latch output is immediately set, output High.
R	Synchronous Reset (Q = Low); has precedence over Set.
PRE	Asynchronous Preset (Q = High). When the PRE input is High and CLR is Low, the flip-flop is set, output High, during the Low-to-High clock (C) transition. A latch output is immediately set, output High.
CLR	Asynchronous Clear (Q = Low); has precedence over Preset to reset Q output Low
SR	CLB input for R, S, CLR, or PRE
REV	CLB input for opposite of SR. Must be asynchronous or synchronous to match SR.

The control inputs R, S, CE, and C are all shared between the two flip-flops in a slice.

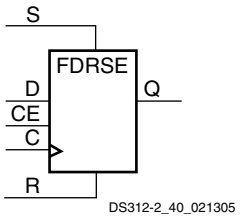

Figure 25: **FD Flip-Flop Component with Synchronous Reset, Set, and Clock Enable**

Table 16: **FD Flip-Flop Functionality with Synchronous Reset, Set, and Clock Enable**

Inputs					Outputs
R	S	CE	D	C	Q
1	X	X	X	↑	0
0	1	X	X	↑	1
0	0	0	X	X	No Change
0	0	1	1	↑	1
0	0	1	0	↑	0

initialized distributed RAM contents are not disturbed during the configuration process.

The distributed RAM is useful for smaller amounts of memory. Larger memory requirements can use the dedicated 18Kbit RAM blocks (see Block RAM).

Shift Registers

For additional information, refer to the “Using Look-Up Tables as Shift Registers (SRL16)” chapter in UG331.

It is possible to program each SLICEM LUT as a 16-bit shift register (see Figure 28). Used in this way, each LUT can delay serial data anywhere from 1 to 16 clock cycles without using any of the dedicated flip-flops. The resulting programmable delays can be used to balance the timing of data pipelines.

The SLICEM LUTs cascade from the G-LUT to the F-LUT through the DIFMUX (see Figure 15). SHIFTIN and SHIFTOUT lines cascade a SLICEM to the SLICEM below to form larger shift registers. The four SLICEM LUTs of a single CLB can be combined to produce delays up to 64 clock cycles. It is also possible to combine shift registers across more than one CLB.

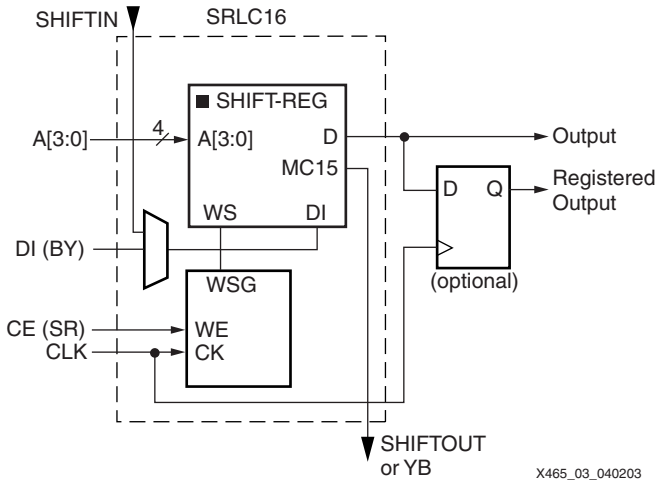

Figure 28: **Logic Cell SRL16 Structure**

Each shift register provides a shift output MC15 for the last bit in each LUT, in addition to providing addressable access to any bit in the shift register through the normal D output. The address inputs A[3:0] are the same as the distributed RAM address lines, which come from the LUT inputs F[4:1] or G[4:1]. At the end of the shift register, the CLB flip-flop can be used to provide one more shift delay for the addressable bit.

The shift register element is known as the SRL16 (Shift Register LUT 16-bit), with a ‘C’ added to signify a cascade ability (Q15 output) and ‘E’ to indicate a Clock Enable. See Figure 29 for an example of the SRLC16E component.

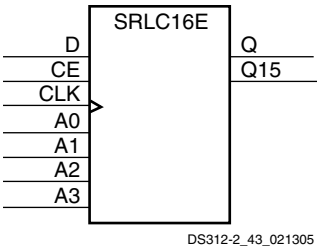

Figure 29: **SRL16 Shift Register Component with Cascade and Clock Enable**

The functionality of the shift register is shown in Table 20. The SRL16 shifts on the rising edge of the clock input when the Clock Enable control is High. This shift register cannot be initialized either during configuration or during operation except by shifting data into it. The clock enable and clock inputs are shared between the two LUTs in a SLICEM. The clock enable input is automatically kept active if unused.

Table 20: **SRL16 Shift Register Function**

Inputs				Outputs	
Am	**CLK**	**CE**	**D**	**Q**	**Q15**
Am	X	0	X	Q[Am]	Q[15]
Am	↑	1	D	Q[Am-1]	Q[15]

Notes:

1. m = 0, 1, 2, 3.

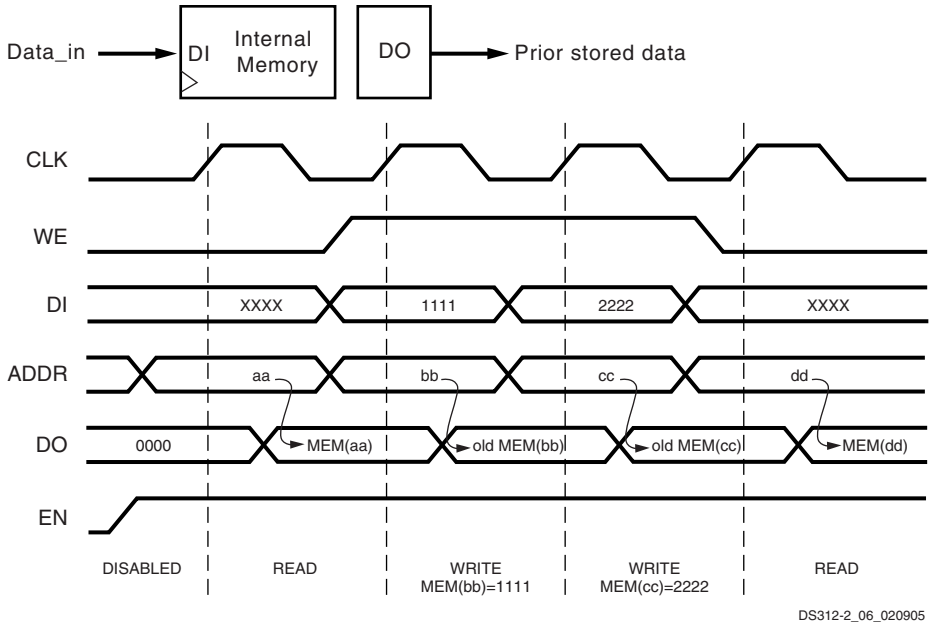

Figure 34: **Waveforms of Block RAM Data Operations with READ_FIRST Selected**

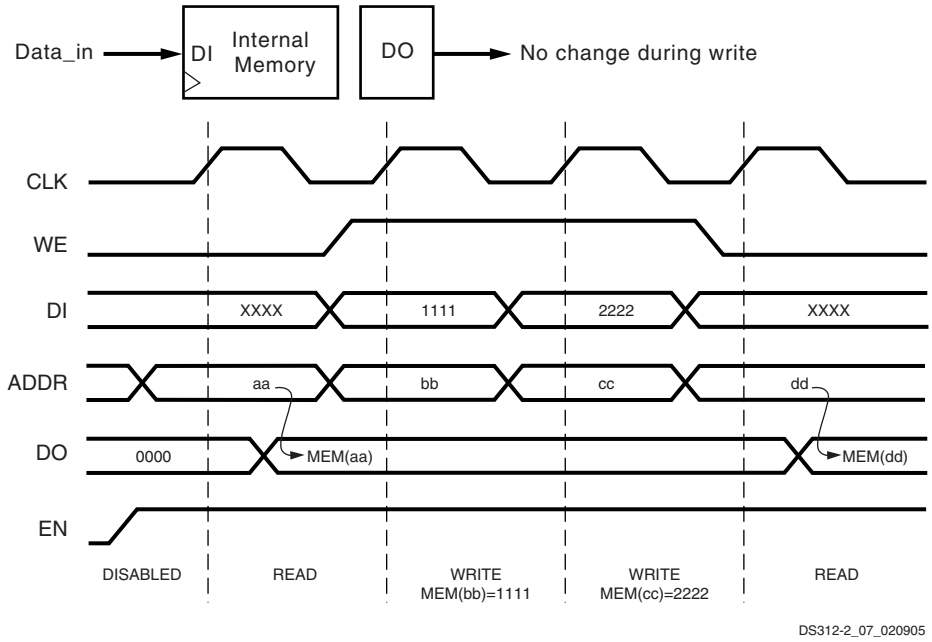

Figure 35: **Waveforms of Block RAM Data Operations with NO_CHANGE Selected**

Setting the WRITE_MODE attribute to a value of NO_CHANGE, puts the DO outputs in a latched state when asserting WE. Under this condition, the DO outputs retain the data driven just before WE is asserted. NO_CHANGE timing is shown in the portion of Figure 35 during which WE is High.

The connections for the bottom-edge BUFGMUX elements are similar to the top-edge connections (see Figure 46).

On the left and right edges, only two clock inputs feed each pair of BUFGMUX elements.

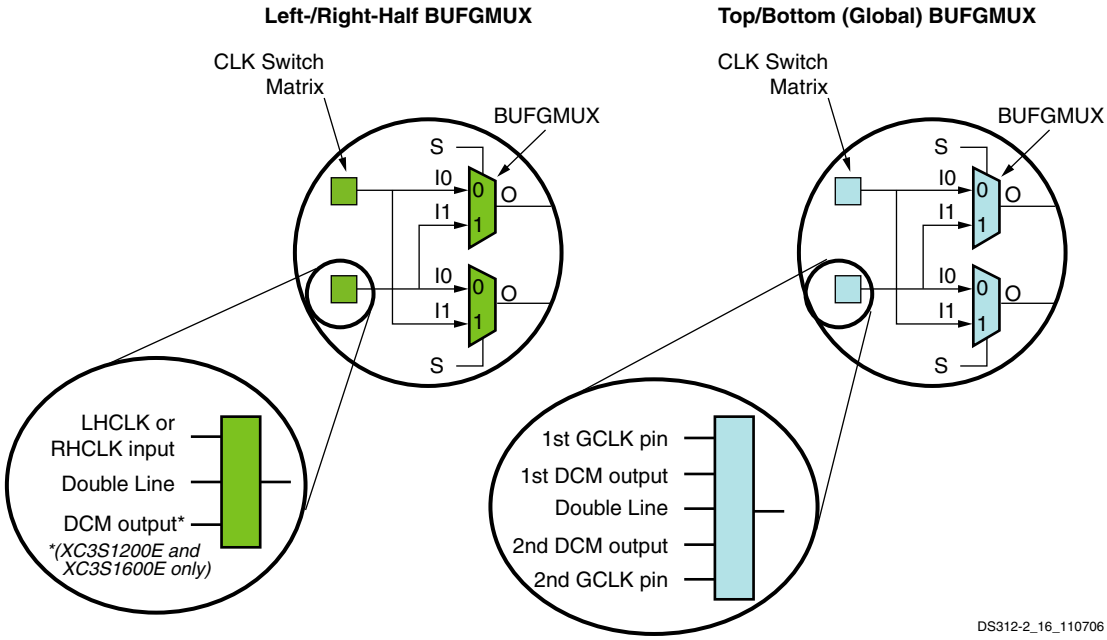

Figure 46: **Clock Switch Matrix to BUFGMUX Pair Connectivity**

Quadrant Clock Routing

The clock routing within the FPGA is quadrant-based, as shown in Figure 45. Each clock quadrant supports eight total clock signals, labeled 'A' through 'H' in Table 41 and Figure 47. The clock source for an individual clock line originates either from a global BUFGMUX element along the top and bottom edges or from a BUFGMUX element along the associated edge, as shown in Figure 47. The clock lines feed the synchronous resource elements (CLBs, IOBs, block RAM, multipliers, and DCMs) within the quadrant.

The four quadrants of the device are:

- Top Right (TR)
- Bottom Right (BR)
- Bottom Left (BL)
- Top Left (TL)

Note that the quadrant clock notation (TR, BR, BL, TL) is separate from that used for similar IOB placement constraints.

To estimate the quadrant location for a particular I/O, see the footprint diagrams in Module 4, Pinout Descriptions. For exact quadrant locations, use the floorplanning tool. In the QFP packages (VQ100, TQ144 and PQ208) the quadrant borders fall in the middle of each side of the package, at a GND pin. The clock inputs fall on the quadrant boundaries, as indicated in Table 42.

Table 42: **QFP Package Clock Quadrant Locations**

Clock Pins	Quadrant
GCLK[3:0]	BR
GCLK[7:4]	TR
GCLK[11:8]	TL
GCLK[15:12]	BL
RHCLK[3:0]	BR
RHCLK[7:4]	TR
LHCLK[3:0]	TL
LHCLK[7:4]	BL

In a few cases, a dedicated input is physically in one quadrant of the device but connects to a different clock quadrant:

- FT256, H16 is in clock quadrant BR
- FG320, K2 is in clock quadrant BL
- FG400, L8 is in clock quadrant TL and the I/O at N11 is in clock quadrant BL
- FG484, M2 is in clock quadrant TL and L15 is in clock quadrant BR

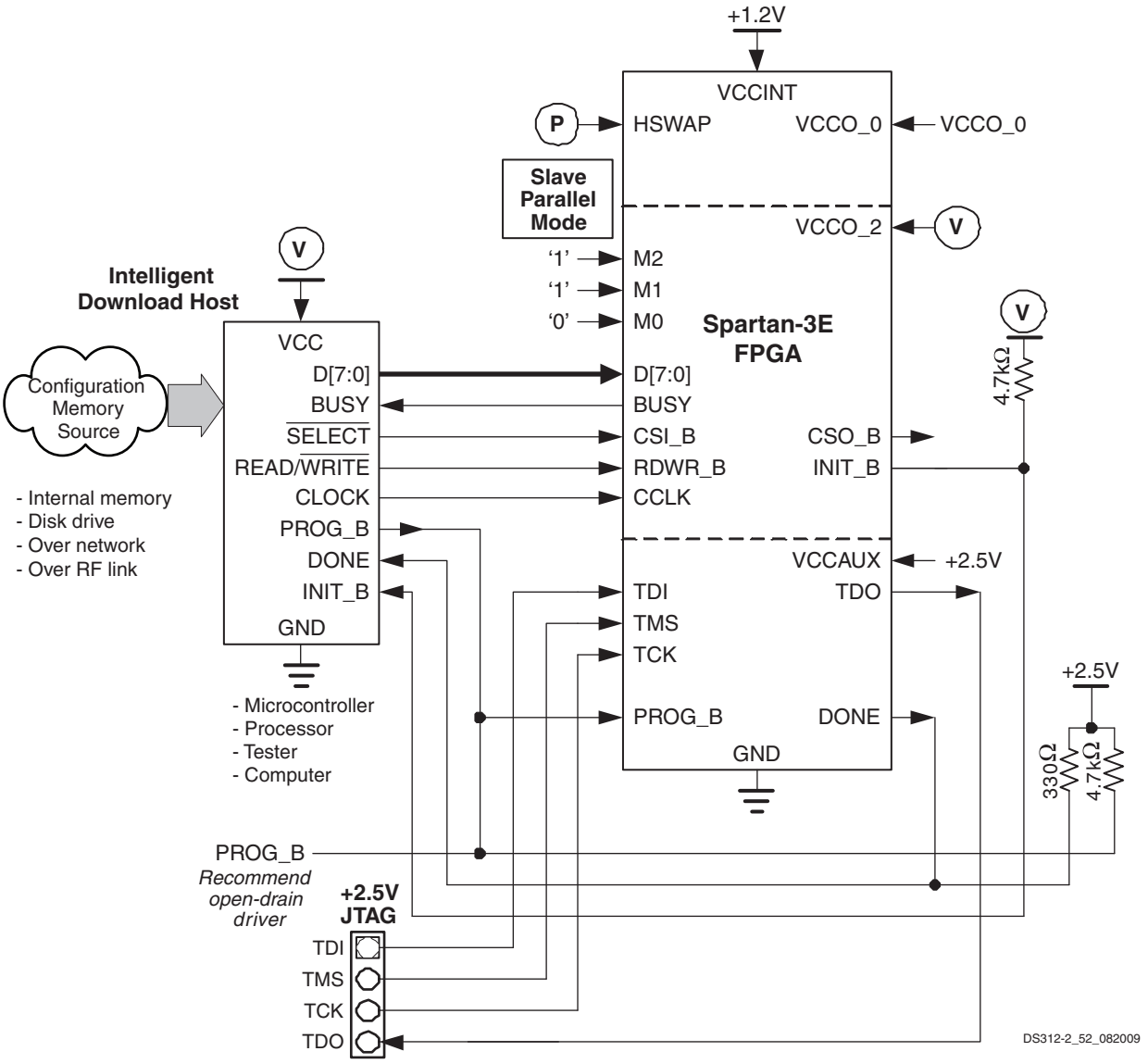

Figure 61: **Slave Parallel Configuration Mode**

Slave Parallel Mode

For additional information, refer to the "Slave Parallel (SelectMAP) Mode" chapter in UG332.

In Slave Parallel mode (M[2:0] = <1:1:0>), an external host, such as a microprocessor or microcontroller, writes byte-wide configuration data into the FPGA, using a typical peripheral interface as shown in Figure 61.

The external download host starts the configuration process by pulsing PROG_B and monitoring that the INIT_B pin goes High, indicating that the FPGA is ready to receive its first data. The host asserts the active-Low chip-select signal (CSI_B) and the active-Low Write signal (RDWR_B). The host then continues supplying data and clock signals until either the FPGA's DONE pin goes High, indicating a successful configuration, or until the FPGA's INIT_B pin goes Low, indicating a configuration error.

The FPGA captures data on the rising CCLK edge. If the CCLK frequency exceeds 50 MHz, then the host must also monitor the FPGA's BUSY output. If the FPGA asserts BUSY High, the host must hold the data for an additional clock cycle, until BUSY returns Low. If the CCLK frequency is 50 MHz or below, the BUSY pin may be ignored but actively drives during configuration.

The configuration process requires more clock cycles than indicated from the configuration file size. Additional clocks are required during the FPGA's start-up sequence, especially if the FPGA is programmed to wait for selected Digital Clock Managers (DCMs) to lock to their respective clock inputs (see Start-Up, page 105).

If the Slave Parallel interface is only used to configure the FPGA, never to read data back, then the RDWR_B signal

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.

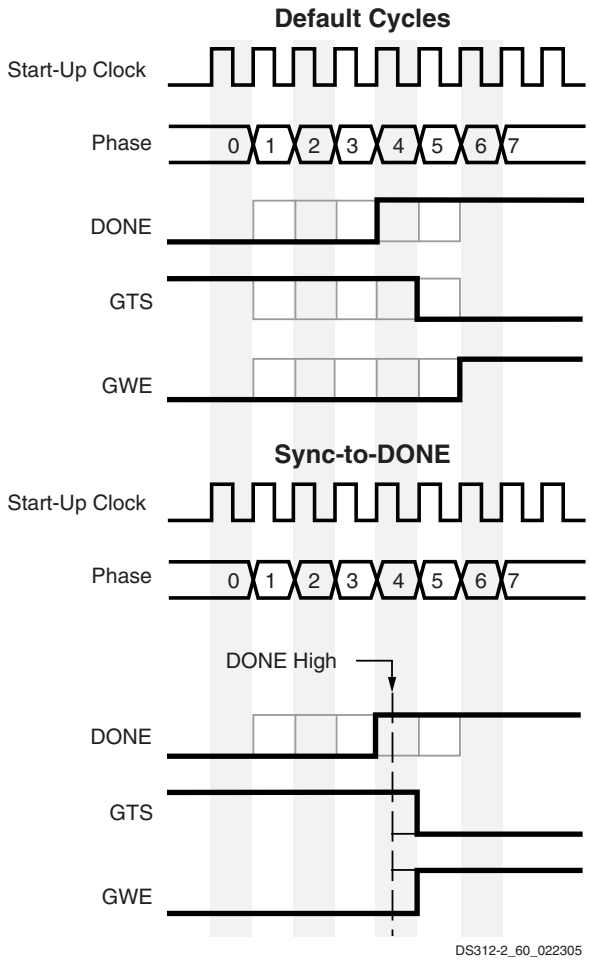

Figure 68: **Default Start-Up Sequence**

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.

Production Stepping

The Spartan-3E FPGA family uses production stepping to indicate improved capabilities or enhanced features.

Stepping 1 is, by definition, a functional superset of Stepping 0. Furthermore, configuration bitstreams generated for Stepping 0 are compatible with Stepping 1.

Designs operating on the Stepping 0 devices perform similarly on a Stepping 1 device.

Differences Between Steppings

Table 71 summarizes the feature and performance differences between Stepping 0 devices and Stepping 1 devices.

Table 71: **Differences between Spartan-3E Production Stepping Levels**

	Stepping 0	Stepping 1
Production status	Production from 2005 to 2007	Production starting March 2006
Speed grade and operating conditions	-4C only	-4C, -4I, -5C
JTAG ID code	Different revision fields. See Table 67.	
DCM DLL maximum input frequency	90 MHz (200 MHz for XC3S1200E)	240 MHz (-4 speed grade) 275 MHz (-5 speed grade)
DCM DFS output frequency range(s)	Split ranges at 5 – 90 MHz and 220 – 307 MHz (single range 5 – 307 MHz for XC3S1200E)	Continuous range: 5 – 311 MHz (-4) 5 – 333 MHz (-5)
Supports multi-FPGA daisy-chain configurations from SPI Flash	No, single FPGA only	Yes
JTAG configuration supported when FPGA in BPI mode with a valid image in the attached parallel NOR Flash PROM	No(1)	Yes
JTAG EXTEST, INTEST, SAMPLE support	Yes: XC3S100E, XC3S250E, XC3S500E No(2): XC3S1200E, XC3S1600E	Yes All Devices
Power sequencing when using HSWAP Pull-Up	Requires V_{CCINT} before V_{CCAUX}	Any sequence
PCI compliance	No	Yes

Notes:

1. Workarounds exist. See Stepping 0 Limitations when Reprogramming via JTAG if FPGA Set for BPI Configuration.
2. JTAG BYPASS and JTAG configuration are supported

Ordering a Later Stepping

-5C and -4I devices, and -4C devices (with date codes 0901 (2009) and later) always support the Stepping 1 feature set independent of the stepping code. Optionally, to order only Stepping 1 for the -4C devices, append an “S1” suffix to the standard ordering code, where ‘1’ is the stepping number, as indicated in Table 72.

Table 72: **Spartan-3E Optional Stepping Ordering**

Stepping Number	Suffix Code	Status
0	None	Production
1	S1	Production

Software Version Requirements

Production Spartan-3E applications must be processed using the Xilinx ISE 8.1i, Service Pack 3 or later development software, using the v1.21 or later speed files. The ISE 8.1i software implements critical bitstream generator updates.

For additional information on Spartan-3E development software and known issues, see the following Answer Record:

- Xilinx Answer #22253
  http://www.xilinx.com/support/answers/22253.htm

Table 94: **Output Timing Adjustments for IOB**

Convert Output Time from LVCMOS25 with 12mA Drive and Fast Slew Rate to the Following Signal Standard (IOSTANDARD)			Add the Adjustment Below		Units
			Speed Grade		
			-5	-4	
Single-Ended Standards					
LVTTL	Slow	2 mA	5.20	5.41	ns
		4 mA	2.32	2.41	ns
		6 mA	1.83	1.90	ns
		8 mA	0.64	0.67	ns
		12 mA	0.68	0.70	ns
		16 mA	0.41	0.43	ns
	Fast	2 mA	4.80	5.00	ns
		4 mA	1.88	1.96	ns
		6 mA	1.39	1.45	ns
		8 mA	0.32	0.34	ns
		12 mA	0.28	0.30	ns
		16 mA	0.28	0.30	ns
LVCMOS33	Slow	2 mA	5.08	5.29	ns
		4 mA	1.82	1.89	ns
		6 mA	1.00	1.04	ns
		8 mA	0.66	0.69	ns
		12 mA	0.40	0.42	ns
		16 mA	0.41	0.43	ns
	Fast	2 mA	4.68	4.87	ns
		4 mA	1.46	1.52	ns
		6 mA	0.38	0.39	ns
		8 mA	0.33	0.34	ns
		12 mA	0.28	0.30	ns
		16 mA	0.28	0.30	ns
LVCMOS25	Slow	2 mA	4.04	4.21	ns
		4 mA	2.17	2.26	ns
		6 mA	1.46	1.52	ns
		8 mA	1.04	1.08	ns
		12 mA	0.65	0.68	ns
	Fast	2 mA	3.53	3.67	ns
		4 mA	1.65	1.72	ns
		6 mA	0.44	0.46	ns
		8 mA	0.20	0.21	ns
		12 mA	0	0	ns

Table 94: **Output Timing Adjustments for IOB *(Cont'd)***

Convert Output Time from LVCMOS25 with 12mA Drive and Fast Slew Rate to the Following Signal Standard (IOSTANDARD)			Add the Adjustment Below		Units
			Speed Grade		
			-5	-4	
LVCMOS18	Slow	2 mA	5.03	5.24	ns
		4 mA	3.08	3.21	ns
		6 mA	2.39	2.49	ns
		8 mA	1.83	1.90	ns
	Fast	2 mA	3.98	4.15	ns
		4 mA	2.04	2.13	ns
		6 mA	1.09	1.14	ns
		8 mA	0.72	0.75	ns
LVCMOS15	Slow	2 mA	4.49	4.68	ns
		4 mA	3.81	3.97	ns
		6 mA	2.99	3.11	ns
	Fast	2 mA	3.25	3.38	ns
		4 mA	2.59	2.70	ns
		6 mA	1.47	1.53	ns
LVCMOS12	Slow	2 mA	6.36	6.63	ns
	Fast	2 mA	4.26	4.44	ns
HSTL_I_18			0.33	0.34	ns
HSTL_III_18			0.53	0.55	ns
PCI33_3			0.44	0.46	ns
PCI66_3			0.44	0.46	ns
SSTL18_I			0.24	0.25	ns
SSTL2_I			–0.20	–0.20	ns
Differential Standards					
LVDS_25			–0.55	–0.55	ns
BLVDS_25			0.04	0.04	ns
MINI_LVDS_25			–0.56	–0.56	ns
LVPECL_25			Input Only		ns
RSDS_25			–0.48	–0.48	ns
DIFF_HSTL_I_18			0.42	0.42	ns
DIFF_HSTL_III_18			0.53	0.55	ns
DIFF_SSTL18_I			0.40	0.40	ns
DIFF_SSTL2_I			0.44	0.44	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 output- and three-state-path times originally specified for the LVCMOS25 standard with 12 mA drive and Fast slew rate to times that correspond to other signal standards. Do not adjust times that measure when outputs go into a high-impedance state.

Configuration Clock (CCLK) Characteristics

Table 112: **Master Mode CCLK Output Period by *ConfigRate* Option Setting**

Symbol	Description	*ConfigRate* Setting	Temperature Range	Minimum	Maximum	Units
T_{CCLK1}	CCLK clock period by ***ConfigRate*** setting	1 *(power-on value and default value)*	Commercial	570	1,250	ns
			Industrial	485		ns
T_{CCLK3}		3	Commercial	285	625	ns
			Industrial	242		ns
T_{CCLK6}		6	Commercial	142	313	ns
			Industrial	121		ns
T_{CCLK12}		12	Commercial	71.2	157	ns
			Industrial	60.6		ns
T_{CCLK25}		25	Commercial	35.5	78.2	ns
			Industrial	30.3		ns
T_{CCLK50}		50	Commercial	17.8	39.1	ns
			Industrial	15.1		ns

Notes:

1. Set the ***ConfigRate*** option value when generating a configuration bitstream. See Bitstream Generator (BitGen) Options in Module 2.

Table 113: **Master Mode CCLK Output Frequency by *ConfigRate* Option Setting**

Symbol	Description	*ConfigRate* Setting	Temperature Range	Minimum	Maximum	Units
F_{CCLK1}	Equivalent CCLK clock frequency by ***ConfigRate*** setting	1 *(power-on value and default value)*	Commercial	0.8	1.8	MHz
			Industrial		2.1	MHz
F_{CCLK3}		3	Commercial	1.6	3.6	MHz
			Industrial		4.2	MHz
F_{CCLK6}		6	Commercial	3.2	7.1	MHz
			Industrial		8.3	MHz
F_{CCLK12}		12	Commercial	6.4	14.1	MHz
			Industrial		16.5	MHz
F_{CCLK25}		25	Commercial	12.8	28.1	MHz
			Industrial		33.0	MHz
F_{CCLK50}		50	Commercial	25.6	56.2	MHz
			Industrial		66.0	MHz

Table 114: **Master Mode CCLK Output Minimum Low and High Time**

Symbol	Description		*ConfigRate* Setting						Units
			1	3	6	12	25	50	
T_{MCCL}, T_{MCCH}	Master mode CCLK minimum Low and High time	Commercial	276	138	69	34.5	17.1	8.5	ns
		Industrial	235	117	58	29.3	14.5	7.3	ns

Table 115: **Slave Mode CCLK Input Low and High Time**

Symbol	Description	Min	Max	Units
T_{SCCL}, T_{SCCH}	CCLK Low and High time	5	∞	ns

Table 129: **Maximum User I/O by Package**

Device	Package	Maximum User I/Os and Input-Only	Maximum Input-Only	Maximum Differential Pairs	All Possible I/Os by Type					
					I/O	INPUT	DUAL	VREF[1]	CLK[2]	N.C.
XC3S100E	VQ100	66	7	30	16	1	21	4	24	0
XC3S250E		66	7	30	16	1	21	4	24	0
XC3S500E		66	7	30	16	1	21	4	24	0
XC3S100E	CP132	83	11	35	16	2	42	7	16	9
XC3S250E		92	7	41	22	0	46	8	16	0
XC3S500E		92	7	41	22	0	46	8	16	0
XC3S100E	TQ144	108	28	40	22	19	42	9	16	0
XC3S250E		108	28	40	20	21	42	9	16	0
XC3S250E	PQ208	158	32	65	58	25	46	13	16	0
XC3S500E		158	32	65	58	25	46	13	16	0
XC3S250E	FT256	172	40	68	62	33	46	15	16	18
XC3S500E		190	41	77	76	33	46	19	16	0
XC3S1200E		190	40	77	78	31	46	19	16	0
XC3S500E	FG320	232	56	92	102	48	46	20	16	18
XC3S1200E		250	56	99	120	47	46	21	16	0
XC3S1600E		250	56	99	120	47	46	21	16	0
XC3S1200E	FG400	304	72	124	156	62	46	24	16	0
XC3S1600E		304	72	124	156	62	46	24	16	0
XC3S1600E	FG484	376	82	156	214	72	46	28	16	0

Notes:

1. Some VREF pins are on INPUT pins. See pinout tables for details.
2. All devices have 24 possible global clock and right- and left-half side clock inputs. The right-half and bottom-edge clock pins have shared functionality in some FPGA configuration modes. Consequently, some clock pins are counted in the DUAL column. 4 GCLK pins, including 2 DUAL pins, are on INPUT pins.

Electronic versions of the package pinout tables and footprints are available for download from the Xilinx website. Download the files from the following location:

http://www.xilinx.com/support/documentation/data_sheets/s3e_pin.zip

Using a spreadsheet program, the data can be sorted and reformatted according to any specific needs. Similarly, the ASCII-text file is easily parsed by most scripting programs.

PQ208 Footprint (Right)

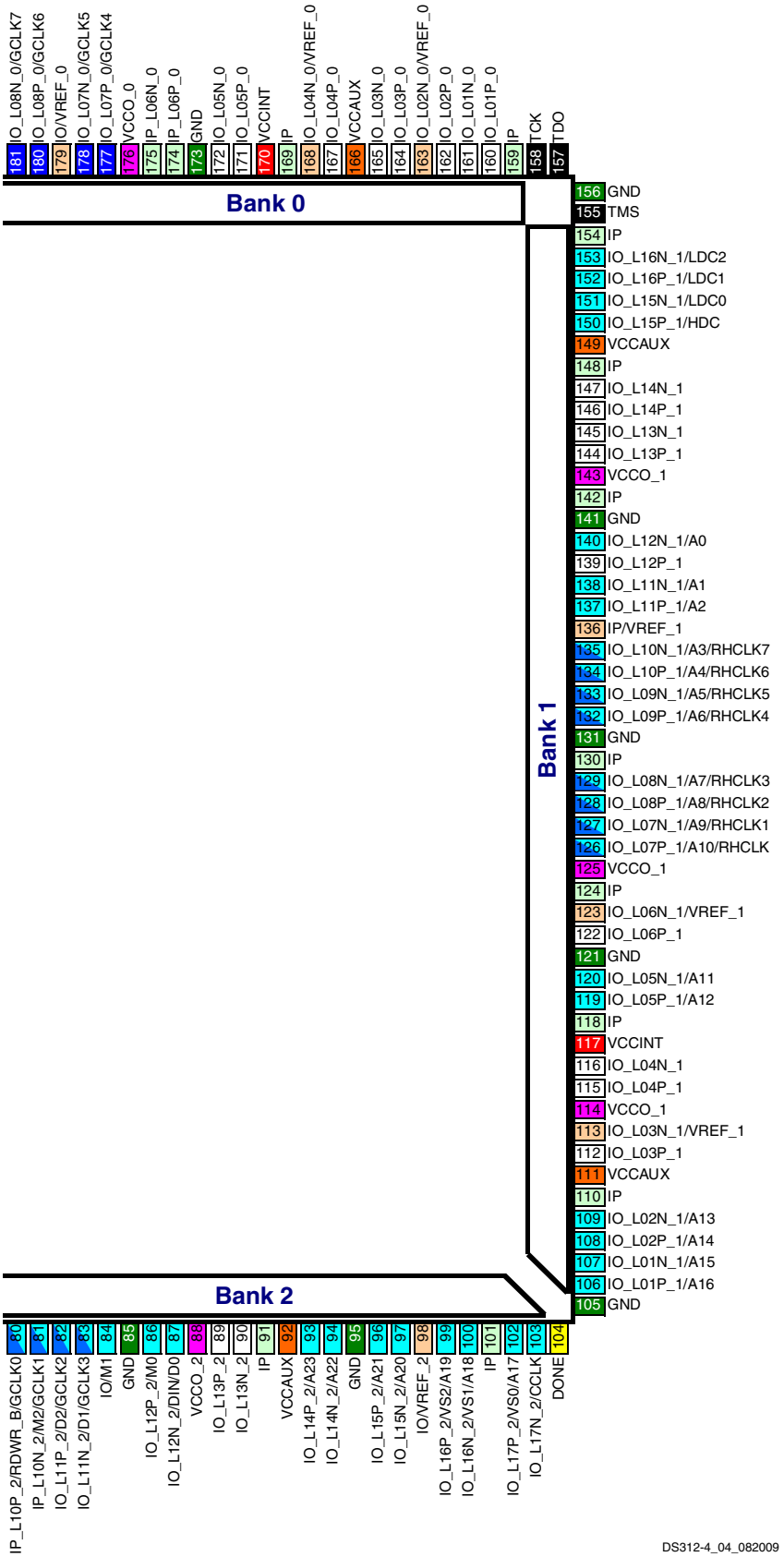

Figure 84: **PQ208 Footprint (Right)**

Footprint Migration Differences

Table 147 summarizes any footprint and functionality differences between the XC3S250E, the XC3S500E, and the XC3S1200E FPGAs that may affect easy migration between devices in the FG256 package. There are 26 such balls. All other pins not listed in Table 147 unconditionally migrate between Spartan-3E devices available in the FT256 package.

The XC3S250E is duplicated on both the left and right sides of the table to show migrations to and from the XC3S500E and the XC3S1200E. The arrows indicate the direction for easy migration. A double-ended arrow (←→) indicates that the two pins have identical functionality. A left-facing arrow (←) indicates that the pin on the device on the right unconditionally migrates to the pin on the device on the left. It may be possible to migrate the opposite direction depending on the I/O configuration. For example, an I/O pin (Type = I/O) can migrate to an input-only pin (Type = INPUT) if the I/O pin is configured as an input.

Table 147: **FT256 Footprint Migration Differences**

FT256 Ball	Bank	XC3S250E Type	Migration	XC3S500E Type	Migration	XC3S1200E Type	Migration	XC3S250E Type
B6	0	INPUT	←→	INPUT	→	I/O	←	INPUT
B7	0	N.C.	→	I/O	←→	I/O	←	N.C.
B10	0	INPUT	←→	INPUT	→	I/O	←	INPUT
C7	0	N.C.	→	I/O	←→	I/O	←	N.C.
D16	1	VREF(I/O)	←	VREF(INPUT)	←→	VREF(INPUT)	→	VREF(I/O)
E13	1	N.C.	→	I/O	←→	I/O	←	N.C.
E16	1	N.C.	→	I/O	←→	I/O	←	N.C.
F3	3	N.C.	→	I/O	←→	I/O	←	N.C.
F4	3	N.C.	→	VREF	←→	VREF	←	N.C.
F5	3	I/O	←→	I/O	←	INPUT	→	I/O
L2	3	N.C.	→	VREF	←→	VREF	←	N.C.
L3	3	N.C.	→	I/O	←→	I/O	←	N.C.
L4	3	N.C.	→	I/O	←→	I/O	←	N.C.
L12	1	N.C.	→	I/O	←→	I/O	←	N.C.
L13	1	N.C.	→	I/O	←→	I/O	←	N.C.
M4	3	N.C.	→	I/O	←→	I/O	←	N.C.
M7	2	INPUT	←→	INPUT	→	I/O	←	INPUT
M14	1	I/O	←→	I/O	←	INPUT	→	I/O
N2	3	VREF(I/O)	←→	VREF(I/O)	←	VREF(INPUT)	→	VREF(I/O)
N7	2	N.C.	→	I/O	←→	I/O	←	N.C.
N14	1	N.C.	→	I/O	←→	I/O	←	N.C.
N15	1	N.C.	→	VREF	←→	VREF	←	N.C.
P7	2	N.C.	→	I/O	←→	I/O	←	N.C.
P10	2	N.C.	→	I/O	←→	I/O	←	N.C.
R10	2	N.C.	→	VREF	←→	VREF	←	N.C.
T12	2	INPUT	←→	INPUT	→	I/O	←	INPUT
DIFFERENCES			**19**		**7**		**26**	

Legend:

←→ This pin is identical on the device on the left and the right.

→ This pin can unconditionally migrate from the device on the left to the device on the right. Migration in the other direction may be possible depending on how the pin is configured for the device on the right.

← This pin can unconditionally migrate from the device on the right to the device on the left. Migration in the other direction may be possible depending on how the pin is configured for the device on the left.

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

Bank	XC3S500E Pin Name	XC3S1200E Pin Name	XC3S1600E Pin Name	FG320 Ball	Type
2	IO_L20P_2	IO_L20P_2	IO_L20P_2	T12	I/O
2	N.C. (◆)	IO_L21N_2	IO_L21N_2	P12	**500E:** N.C. **1200E:** I/O **1600E:** I/O
2	N.C. (◆)	IO_L21P_2	IO_L21P_2	N12	**500E:** N.C. **1200E:** I/O **1600E:** I/O
2	IO_L22N_2/A22	IO_L22N_2/A22	IO_L22N_2/A22	R13	DUAL
2	IO_L22P_2/A23	IO_L22P_2/A23	IO_L22P_2/A23	P13	DUAL
2	IO_L24N_2/A20	IO_L24N_2/A20	IO_L24N_2/A20	R14	DUAL
2	IO_L24P_2/A21	IO_L24P_2/A21	IO_L24P_2/A21	T14	DUAL
2	IO_L25N_2/VS1/A18	IO_L25N_2/VS1/A18	IO_L25N_2/VS1/A18	U15	DUAL
2	IO_L25P_2/VS2/A19	IO_L25P_2/VS2/A19	IO_L25P_2/VS2/A19	V15	DUAL
2	IO_L26N_2/CCLK	IO_L26N_2/CCLK	IO_L26N_2/CCLK	U16	DUAL
2	IO_L26P_2/VS0/A17	IO_L26P_2/VS0/A17	IO_L26P_2/VS0/A17	T16	DUAL
2	IP	IP	IP	V2	INPUT
2	IP	IP	IP	V16	INPUT
2	IP_L02N_2	IP_L02N_2	IP_L02N_2	V3	INPUT
2	IP_L02P_2	IP_L02P_2	IP_L02P_2	V4	INPUT
2	IP_L08N_2	IP_L08N_2	IP_L08N_2	R7	INPUT
2	IP_L08P_2	IP_L08P_2	IP_L08P_2	T7	INPUT
2	IP_L11N_2/VREF_2	IP_L11N_2/VREF_2	IP_L11N_2/VREF_2	V8	VREF
2	IP_L11P_2	IP_L11P_2	IP_L11P_2	U8	INPUT
2	IP_L14N_2/M2/GCLK1	IP_L14N_2/M2/GCLK1	IP_L14N_2/M2/GCLK1	T10	DUAL/GCLK
2	IP_L14P_2/RDWR_B/ GCLK0	IP_L14P_2/RDWR_B/ GCLK0	IP_L14P_2/RDWR_B/ GCLK0	U10	DUAL/GCLK
2	IP_L17N_2	IP_L17N_2	IP_L17N_2	U11	INPUT
2	IP_L17P_2	IP_L17P_2	IP_L17P_2	T11	INPUT
2	IP_L23N_2	IP_L23N_2	IP_L23N_2	U14	INPUT
2	IP_L23P_2	IP_L23P_2	IP_L23P_2	V14	INPUT
2	VCCO_2	VCCO_2	VCCO_2	M8	VCCO
2	VCCO_2	VCCO_2	VCCO_2	M11	VCCO
2	VCCO_2	VCCO_2	VCCO_2	T6	VCCO
2	VCCO_2	VCCO_2	VCCO_2	T13	VCCO
2	VCCO_2	VCCO_2	VCCO_2	V10	VCCO
3	N.C. (◆)	IO	IO	D4	**500E:** N.C. **1200E:** I/O **1600E:** I/O
3	IO_L01N_3	IO_L01N_3	IO_L01N_3	C2	I/O
3	IO_L01P_3	IO_L01P_3	IO_L01P_3	C1	I/O
3	IO_L02N_3/VREF_3	IO_L02N_3/VREF_3	IO_L02N_3/VREF_3	D2	VREF
3	IO_L02P_3	IO_L02P_3	IO_L02P_3	D1	I/O
3	IO_L03N_3	IO_L03N_3	IO_L03N_3	E1	I/O
3	IO_L03P_3	IO_L03P_3	IO_L03P_3	E2	I/O

Footprint Migration Differences

Table 151 summarizes any footprint and functionality differences between the XC3S500E, the XC3S1200E, and the XC3S1600E FPGAs that may affect easy migration between devices available in the FG320 package. There are 26 such balls. All other pins not listed in Table 151 unconditionally migrate between Spartan-3E devices available in the FG320 package.

The XC3S500E is duplicated on both the left and right sides of the table to show migrations to and from the XC3S1200E and the XC3S1600E. The arrows indicate the direction for easy migration. A double-ended arrow (←→) indicates that the two pins have identical functionality. A left-facing arrow (←) indicates that the pin on the device on the right unconditionally migrates to the pin on the device on the left. It may be possible to migrate the opposite direction depending on the I/O configuration. For example, an I/O pin (Type = I/O) can migrate to an input-only pin (Type = INPUT) if the I/O pin is configured as an input.

Table 151: **FG320 Footprint Migration Differences**

Pin	Bank	XC3S500E	Migration	XC3S1200E	Migration	XC3S1600E	Migration	XC3S500E
A7	0	INPUT	→	I/O	←→	I/O	←	INPUT
A12	0	N.C.	→	I/O	←→	I/O	←	N.C.
D4	3	N.C.	→	I/O	←→	I/O	←	N.C.
D6	0	N.C.	→	I/O	←→	I/O	←	N.C.
D13	0	INPUT	→	I/O	←→	I/O	←	INPUT
E3	3	N.C.	→	I/O	←→	I/O	←	N.C.
E4	3	N.C.	→	I/O	←→	I/O	←	N.C.
E6	0	N.C.	→	I/O	←→	I/O	←	N.C.
E15	1	N.C.	→	I/O	←→	I/O	←	N.C.
E16	1	N.C.	→	I/O	←→	I/O	←	N.C.
E17	1	I/O	←	INPUT	←→	INPUT	→	I/O
F4	3	I/O	←	INPUT	←→	INPUT	→	I/O
N12	2	N.C.	→	I/O	←→	I/O	←	N.C.
N14	1	N.C.	→	I/O	←→	I/O	←	N.C.
N15	1	N.C.	→	I/O	←→	I/O	←	N.C.
P3	3	N.C.	→	I/O	←→	I/O	←	N.C.
P4	3	N.C.	→	I/O	←→	I/O	←	N.C.
P12	2	N.C.	→	I/O	←→	I/O	←	N.C.
P15	1	I/O	←	INPUT	←→	INPUT	→	I/O
P16	1	N.C.	→	I/O	←→	I/O	←	N.C.
R4	3	VREF(I/O)	←	VREF(INPUT)	←→	VREF(INPUT)	→	VREF(I/O)
U6	2	INPUT	→	I/O	←→	I/O	←	INPUT
U13	2	INPUT	→	I/O	←→	I/O	←	INPUT
V5	2	N.C.	→	I/O	←→	I/O	←	N.C.
V6	2	N.C.	→	VREF	←→	VREF	←	N.C.
V7	2	N.C.	→	I/O	←→	I/O	←	N.C.
DIFFERENCES			**26**		**0**		**26**	

Legend:

←→ This pin is identical on the device on the left and the right.

→ This pin can unconditionally migrate from the device on the left to the device on the right. Migration in the other direction may be possible depending on how the pin is configured for the device on the right.

← This pin can unconditionally migrate from the device on the right to the device on the left. Migration in the other direction may be possible depending on how the pin is configured for the device on the left.

Table 152: **FG400 Package Pinout *(Cont'd)***

Bank	XC3S1200E XC3S1600E Pin Name	FG400 Ball	Type
2	IP_L14P_2	T9	INPUT
2	IP_L17N_2/M2/GCLK1	P12	DUAL/ GCLK
2	IP_L17P_2/RDWR_B/ GCLK0	P11	DUAL/ GCLK
2	IP_L20N_2	T12	INPUT
2	IP_L20P_2	R12	INPUT
2	IP_L23N_2/VREF_2	T13	VREF
2	IP_L23P_2	T14	INPUT
2	IP_L26N_2	V14	INPUT
2	IP_L26P_2	V15	INPUT
2	IP_L29N_2	W16	INPUT
2	IP_L29P_2	Y16	INPUT
2	VCCO_2	R11	VCCO
2	VCCO_2	U8	VCCO
2	VCCO_2	U14	VCCO
2	VCCO_2	W5	VCCO
2	VCCO_2	W11	VCCO
2	VCCO_2	W17	VCCO
3	IO_L01N_3	D2	I/O
3	IO_L01P_3	D3	I/O
3	IO_L02N_3/VREF_3	E3	VREF
3	IO_L02P_3	E4	I/O
3	IO_L03N_3	C1	I/O
3	IO_L03P_3	B1	I/O
3	IO_L04N_3	E1	I/O
3	IO_L04P_3	D1	I/O
3	IO_L05N_3	F3	I/O
3	IO_L05P_3	F4	I/O
3	IO_L06N_3	F1	I/O
3	IO_L06P_3	F2	I/O
3	IO_L07N_3	G4	I/O
3	IO_L07P_3	G3	I/O
3	IO_L08N_3	G5	I/O
3	IO_L08P_3	H5	I/O
3	IO_L09N_3/VREF_3	H3	VREF
3	IO_L09P_3	H2	I/O
3	IO_L10N_3	H7	I/O
3	IO_L10P_3	H6	I/O
3	IO_L11N_3	J4	I/O
3	IO_L11P_3	J3	I/O
3	IO_L12N_3	J1	I/O
3	IO_L12P_3	J2	I/O
3	IO_L13N_3	J6	I/O

Table 152: **FG400 Package Pinout *(Cont'd)***

Bank	XC3S1200E XC3S1600E Pin Name	FG400 Ball	Type
3	IO_L13P_3	K6	I/O
3	IO_L14N_3/LHCLK1	K2	LHCLK
3	IO_L14P_3/LHCLK0	K3	LHCLK
3	IO_L15N_3/LHCLK3/IRDY2	L7	LHCLK
3	IO_L15P_3/LHCLK2	K7	LHCLK
3	IO_L16N_3/LHCLK5	L1	LHCLK
3	IO_L16P_3/LHCLK4/TRDY2	M1	LHCLK
3	IO_L17N_3/LHCLK7	L3	LHCLK
3	IO_L17P_3/LHCLK6	M3	LHCLK
3	IO_L18N_3	M7	I/O
3	IO_L18P_3	M8	I/O
3	IO_L19N_3	M4	I/O
3	IO_L19P_3	M5	I/O
3	IO_L20N_3/VREF_3	N6	VREF
3	IO_L20P_3	M6	I/O
3	IO_L21N_3	N2	I/O
3	IO_L21P_3	N1	I/O
3	IO_L22N_3	P7	I/O
3	IO_L22P_3	N7	I/O
3	IO_L23N_3	N4	I/O
3	IO_L23P_3	N3	I/O
3	IO_L24N_3	R1	I/O
3	IO_L24P_3	P1	I/O
3	IO_L25N_3	R5	I/O
3	IO_L25P_3	P5	I/O
3	IO_L26N_3	T2	I/O
3	IO_L26P_3	R2	I/O
3	IO_L27N_3	R4	I/O
3	IO_L27P_3	R3	I/O
3	IO_L28N_3/VREF_3	T1	VREF
3	IO_L28P_3	U1	I/O
3	IO_L29N_3	T3	I/O
3	IO_L29P_3	U3	I/O
3	IO_L30N_3	V1	I/O
3	IO_L30P_3	V2	I/O
3	IP	F5	INPUT
3	IP	G1	INPUT
3	IP	G6	INPUT
3	IP	H1	INPUT
3	IP	J5	INPUT
3	IP	L5	INPUT
3	IP	L8	INPUT
3	IP	M2	INPUT

Table 152: **FG400 Package Pinout *(Cont'd)***

Bank	XC3S1200E XC3S1600E Pin Name	FG400 Ball	Type
3	IP	N5	INPUT
3	IP	P3	INPUT
3	IP	T4	INPUT
3	IP	W1	INPUT
3	IP/VREF_3	K5	VREF
3	IP/VREF_3	P6	VREF
3	VCCO_3	E2	VCCO
3	VCCO_3	H4	VCCO
3	VCCO_3	L2	VCCO
3	VCCO_3	L6	VCCO
3	VCCO_3	P4	VCCO
3	VCCO_3	U2	VCCO
GND	GND	A1	GND
GND	GND	A11	GND
GND	GND	A20	GND
GND	GND	B7	GND
GND	GND	B14	GND
GND	GND	C3	GND
GND	GND	C18	GND
GND	GND	D10	GND
GND	GND	F6	GND
GND	GND	F15	GND
GND	GND	G2	GND
GND	GND	G12	GND
GND	GND	G19	GND
GND	GND	H8	GND
GND	GND	J9	GND
GND	GND	J11	GND
GND	GND	K1	GND
GND	GND	K8	GND
GND	GND	K10	GND
GND	GND	K12	GND
GND	GND	K17	GND
GND	GND	L4	GND
GND	GND	L9	GND
GND	GND	L11	GND
GND	GND	L13	GND
GND	GND	L20	GND
GND	GND	M10	GND
GND	GND	M12	GND
GND	GND	N13	GND
GND	GND	P2	GND
GND	GND	P9	GND

Table 152: **FG400 Package Pinout *(Cont'd)***

Bank	XC3S1200E XC3S1600E Pin Name	FG400 Ball	Type
GND	GND	P19	GND
GND	GND	R6	GND
GND	GND	R15	GND
GND	GND	U11	GND
GND	GND	V3	GND
GND	GND	V18	GND
GND	GND	W7	GND
GND	GND	W14	GND
GND	GND	Y1	GND
GND	GND	Y10	GND
GND	GND	Y20	GND
VCCAUX	DONE	V17	CONFIG
VCCAUX	PROG_B	C2	CONFIG
VCCAUX	TCK	D17	JTAG
VCCAUX	TDI	B3	JTAG
VCCAUX	TDO	B19	JTAG
VCCAUX	TMS	E17	JTAG
VCCAUX	VCCAUX	D11	VCCAUX
VCCAUX	VCCAUX	H12	VCCAUX
VCCAUX	VCCAUX	J7	VCCAUX
VCCAUX	VCCAUX	K4	VCCAUX
VCCAUX	VCCAUX	L17	VCCAUX
VCCAUX	VCCAUX	M14	VCCAUX
VCCAUX	VCCAUX	N9	VCCAUX
VCCAUX	VCCAUX	U10	VCCAUX
VCCINT	VCCINT	H9	VCCINT
VCCINT	VCCINT	H11	VCCINT
VCCINT	VCCINT	H13	VCCINT
VCCINT	VCCINT	J8	VCCINT
VCCINT	VCCINT	J10	VCCINT
VCCINT	VCCINT	J12	VCCINT
VCCINT	VCCINT	K9	VCCINT
VCCINT	VCCINT	K11	VCCINT
VCCINT	VCCINT	L10	VCCINT
VCCINT	VCCINT	L12	VCCINT
VCCINT	VCCINT	M9	VCCINT
VCCINT	VCCINT	M11	VCCINT
VCCINT	VCCINT	M13	VCCINT
VCCINT	VCCINT	N8	VCCINT
VCCINT	VCCINT	N10	VCCINT
VCCINT	VCCINT	N12	VCCINT

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
03/01/05	1.0	Initial Xilinx release.
03/21/05	1.1	Added XC3S250E in the CP132 package to Table 129. Corrected number of differential I/O pairs on CP132. Added pinout and footprint information for the CP132, FG400, and FG484 packages. Removed IRDY and TRDY pins from the VQ100, TQ144, and PQ208 packages.
11/23/05	2.0	Corrected title of Table 153. Updated differential pair numbering for some pins in Bank 0 of the FG400 package, affecting Table 152 and Figure 87. Pin functionality and ball assignment were not affected. Added Package Thermal Characteristics section. Added package mass values to Table 125.
03/22/06	3.0	Included I/O pins, not just input-only pins under the VREF description in Table 124. Clarified that some global clock inputs are Input-only pins in Table 124. Added information on the XC3S100E in the CP132 package, affecting Table 129, Table 130, Table 133, Table 134, Table 136, and Figure 81. Ball A12 on the XC3S1600E in the FG320 package a full I/O pin, not an Input-only pin. Corrected the I/O counts for the XC3S1600E in the FG320 package, affecting Table 129, Table 150, Table 151, and Figure 86. Corrected pin type for XC3S1600E balls N14 and N15 in Table 148.
05/19/06	3.1	Minor text edits.
11/09/06	3.4	Added package thermal data for the XC3S100E in the CP132 package to Table 130. Corrected pin migration arrows for balls E17 and F4 between the XC3S500E and XC3S1600E in Table 151. Promoted Module 4 to Production status. Synchronized all modules to v3.4.
03/16/07	3.5	Minor formatting changes.
05/29/07	3.6	Corrected ‘Lxx’ to ‘Lxxy’ in Table 124. Noted that some GCLK and VREF pins are on INPUT pins in Table 124 and Table 129. Added link before Table 127 to Material Declaration Data Sheets.
04/18/08	3.7	Added XC3S500E VQG100 package. Added Material Declaration Data Sheet links in Table 127. Updated Thermal Characteristics in Table 130. Updated links.
08/26/09	3.8	Minor typographical updates.
10/29/12	4.0	Added Notice of Disclaimer. This product is not recommended for new designs. Updated the XC3S250E-FT256 in Table 129.

Notice of Disclaimer