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	Obsolete
Number of LABs/CLBs	12696
Number of Logic Elements/Cells	57132
Total RAM Bits	753664
Number of I/O	804
Number of Gates	3263755
Voltage - Supply	1.71V ~ 1.89V
Mounting Type	Surface Mount
Operating Temperature	0°C ~ 85°C (TJ)
Package / Case	1156-BBGA
Supplier Device Package	1156-FBGA (35x35)
Purchase URL	https://www.e-xfl.com/product-detail/xilinx/xcv2600e-8fg1156c

Development System

Virtex-E FPGAs are supported by the Xilinx Foundation and Alliance Series CAE tools. The basic methodology for Virtex-E design consists of three interrelated steps: design entry, implementation, and verification. Industry-standard tools are used for design entry and simulation (for example, Synopsys FPGA Express), while Xilinx provides proprietary architecture-specific tools for implementation.

The Xilinx development system is integrated under the Xilinx Design Manager (XDM™) software, providing designers with a common user interface regardless of their choice of entry and verification tools. The XDM software simplifies the selection of implementation options with pull-down menus and on-line help.

Application programs ranging from schematic capture to Placement and Routing (PAR) can be accessed through the XDM software. The program command sequence is generated prior to execution, and stored for documentation.

Several advanced software features facilitate Virtex-E design. RPMs, for example, are schematic-based macros with relative location constraints to guide their placement. They help ensure optimal implementation of common functions.

For HDL design entry, the Xilinx FPGA Foundation development system provides interfaces to the following synthesis design environments.

- Synopsys (FPGA Compiler, FPGA Express)
- Exemplar (Spectrum)
- Synplicity (Synplify)

For schematic design entry, the Xilinx FPGA Foundation and Alliance development system provides interfaces to the following schematic-capture design environments.

- Mentor Graphics V8 (Design Architect, QuickSim II)
- Viewlogic Systems (Viewdraw)

Third-party vendors support many other environments.

A standard interface-file specification, Electronic Design Interchange Format (EDIF), simplifies file transfers into and out of the development system.

Virtex-E FPGAs are supported by a unified library of standard functions. This library contains over 400 primitives and macros, ranging from 2-input AND gates to 16-bit accumulators, and includes arithmetic functions, comparators, counters, data registers, decoders, encoders, I/O functions, latches, Boolean functions, multiplexers, shift registers, and barrel shifters.

The “soft macro” portion of the library contains detailed descriptions of common logic functions, but does not contain any partitioning or placement information. The performance of these macros depends, therefore, on the partitioning and placement obtained during implementation.

RPMs, on the other hand, do contain predetermined partitioning and placement information that permits optimal implementation of these functions. Users can create their own library of soft macros or RPMs based on the macros and primitives in the standard library.

The design environment supports hierarchical design entry, with high-level schematics that comprise major functional blocks, while lower-level schematics define the logic in these blocks. These hierarchical design elements are automatically combined by the implementation tools. Different design entry tools can be combined within a hierarchical design, thus allowing the most convenient entry method to be used for each portion of the design.

Design Implementation

The place-and-route tools (PAR) automatically provide the implementation flow described in this section. The partitioner takes the EDIF net list for the design and maps the logic into the architectural resources of the FPGA (CLBs and IOBs, for example). The placer then determines the best locations for these blocks based on their interconnections and the desired performance. Finally, the router interconnects the blocks.

The PAR algorithms support fully automatic implementation of most designs. For demanding applications, however, the user can exercise various degrees of control over the process. User partitioning, placement, and routing information is optionally specified during the design-entry process. The implementation of highly structured designs can benefit greatly from basic floor planning.

The implementation software incorporates Timing Wizard® timing-driven placement and routing. Designers specify timing requirements along entire paths during design entry. The timing path analysis routines in PAR then recognize these user-specified requirements and accommodate them.

Timing requirements are entered on a schematic in a form directly relating to the system requirements, such as the targeted clock frequency, or the maximum allowable delay between two registers. In this way, the overall performance of the system along entire signal paths is automatically tailored to user-generated specifications. Specific timing information for individual nets is unnecessary.

Design Verification

In addition to conventional software simulation, FPGA users can use in-circuit debugging techniques. Because Xilinx devices are infinitely reprogrammable, designs can be verified in real time without the need for extensive sets of software simulation vectors.

The development system supports both software simulation and in-circuit debugging techniques. For simulation, the system extracts the post-layout timing information from the design database, and back-annotates this information into the net list for use by the simulator. Alternatively, the user can verify timing-critical portions of the design using the TRCE® static timing analyzer.

the internal storage elements to begin changing state in response to the logic and the user clock.

The relative timing of these events can be changed. In addition, the GTS, GSR, and GWE events can be made dependent on the DONE pins of multiple devices all going High, forcing the devices to start synchronously. The sequence can also be paused at any stage until lock has been achieved on any or all DLLs.

Readback

The configuration data stored in the Virtex-E configuration memory can be readback for verification. Along with the configuration data it is possible to readback the contents all flip-flops/latches, LUT RAMs, and block RAMs. This capability is used for real-time debugging. For more detailed information, see application note XAPP138 "Virtex FPGA Series Configuration and Readback".

Design Considerations

This section contains more detailed design information on the following features.

- Delay-Locked Loop . . . see page 19
- BlockRAM . . . see page 24
- SelectI/O . . . see page 31

Using DLLs

The Virtex-E FPGA series provides up to eight fully digital dedicated on-chip Delay-Locked Loop (DLL) circuits which provide zero propagation delay, low clock skew between output clock signals distributed throughout the device, and advanced clock domain control. These dedicated DLLs can be used to implement several circuits which improve and simplify system level design.

Introduction

As FPGAs grow in size, quality on-chip clock distribution becomes increasingly important. Clock skew and clock delay impact device performance and the task of managing clock skew and clock delay with conventional clock trees becomes more difficult in large devices. The Virtex-E series of devices resolve this potential problem by providing up to eight fully digital dedicated on-chip DLL circuits, which provide zero propagation delay and low clock skew between output clock signals distributed throughout the device.

Each DLL can drive up to two global clock routing networks within the device. The global clock distribution network minimizes clock skews due to loading differences. By monitoring a sample of the DLL output clock, the DLL can compensate for the delay on the routing network, effectively eliminating the delay from the external input port to the individual clock loads within the device.

In addition to providing zero delay with respect to a user source clock, the DLL can provide multiple phases of the source clock. The DLL can also act as a clock doubler or it can divide the user source clock by up to 16.

Clock multiplication gives the designer a number of design alternatives. For instance, a 50 MHz source clock doubled by the DLL can drive an FPGA design operating at 100 MHz. This technique can simplify board design because the clock path on the board no longer distributes such a high-speed signal. A multiplied clock also provides designers the option of time-domain-multiplexing, using one circuit twice per clock cycle, consuming less area than two copies of the same circuit. Two DLLs in can be connected in series to increase the effective clock multiplication factor to four.

The DLL can also act as a clock mirror. By driving the DLL output off-chip and then back in again, the DLL can be used to deskew a board level clock between multiple devices.

In order to guarantee the system clock establishes prior to the device "waking up," the DLL can delay the completion of the device configuration process until after the DLL achieves lock.

By taking advantage of the DLL to remove on-chip clock delay, the designer can greatly simplify and improve system level design involving high-fanout, high-performance clocks.

Library DLL Symbols

Figure 21 shows the simplified Xilinx library DLL macro symbol, BUFGDLL. This macro delivers a quick and efficient way to provide a system clock with zero propagation delay throughout the device. Figure 22 and Figure 23 show the two library DLL primitives. These symbols provide access to the complete set of DLL features when implementing more complex applications.

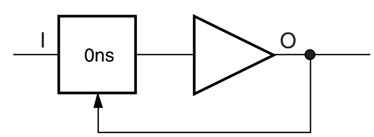

Figure 21: **Simplified DLL Macro Symbol BUFGDLL**

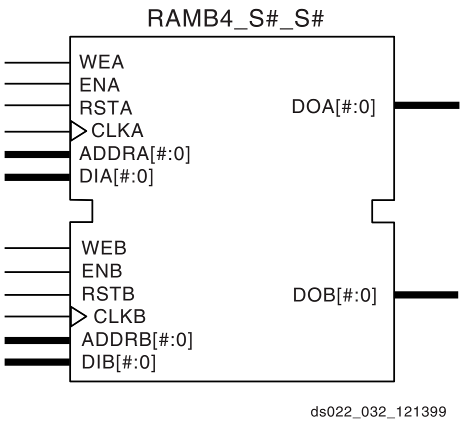

Figure 31: **Dual-Port Block SelectRAM+ Memory**

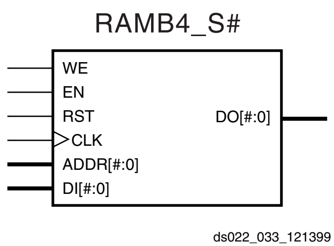

Figure 32: **Single-Port Block SelectRAM+ Memory**

Table 14: **Available Library Primitives**

Primitive	Port A Width	Port B Width
RAMB4_S1	1	N/A
RAMB4_S1_S1		1
RAMB4_S1_S2		2
RAMB4_S1_S4		4
RAMB4_S1_S8		8
RAMB4_S1_S16		16
RAMB4_S2	2	N/A
RAMB4_S2_S2		2
RAMB4_S2_S4		4
RAMB4_S2_S8		8
RAMB4_S2_S16		16
RAMB4_S4	4	N/A
RAMB4_S4_S4		4
RAMB4_S4_S8		8
RAMB4_S4_S16		16
RAMB4_S8	8	N/A
RAMB4_S8_S8		8
RAMB4_S8_S16		16
RAMB4_S16	16	N/A
RAMB4_S16_S16		16

Port Signals

Each block SelectRAM+ port operates independently of the others while accessing the same set of 4096 memory cells.

Table 15 describes the depth and width aspect ratios for the block SelectRAM+ memory.

Table 15: **Block SelectRAM+ Port Aspect Ratios**

Width	Depth	ADDR Bus	Data Bus
1	4096	ADDR<11:0>	DATA<0>
2	2048	ADDR<10:0>	DATA<1:0>
4	1024	ADDR<9:0>	DATA<3:0>
8	512	ADDR<8:0>	DATA<7:0>
16	256	ADDR<7:0>	DATA<15:0>

Clock—CLK[A|B]

Each port is fully synchronous with independent clock pins. All port input pins have setup time referenced to the port CLK pin. The data output bus has a clock-to-out time referenced to the CLK pin.

Enable—EN[A|B]

The enable pin affects the read, write and reset functionality of the port. Ports with an inactive enable pin keep the output pins in the previous state and do not write data to the memory cells.

Write Enable—WE[A|B]

Activating the write enable pin allows the port to write to the memory cells. When active, the contents of the data input bus are written to the RAM at the address pointed to by the address bus, and the new data also reflects on the data out bus. When inactive, a read operation occurs and the contents of the memory cells referenced by the address bus reflect on the data out bus.

Reset—RST[A|B]

The reset pin forces the data output bus latches to zero synchronously. This does not affect the memory cells of the RAM and does not disturb a write operation on the other port.

Address Bus—ADDR[A|B]<#:0>

The address bus selects the memory cells for read or write. The width of the port determines the required width of this bus as shown in Table 15.

Data In Bus—DI[A|B]<#:0>

The data in bus provides the new data value to be written into the RAM. This bus and the port have the same width, as shown in Table 15.

LVDS

Depending on whether the device is transmitting an LVDS signal or receiving an LVDS signal, there are two different circuits used for LVDS termination. A sample circuit illustrating a valid termination technique for transmitting LVDS signals appears in Figure 54. A sample circuit illustrating a valid termination for receiving LVDS signals appears in Figure 55. Table 38 lists DC voltage specifications. Further information on the specific termination resistor packs shown can be found on Table 40.

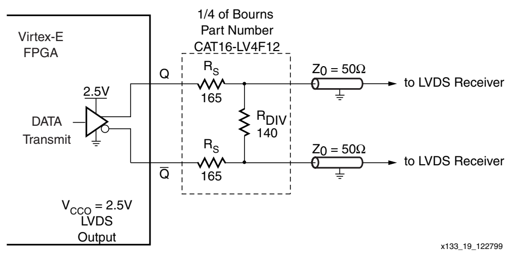

Figure 54: **Transmitting LVDS Signal Circuit**

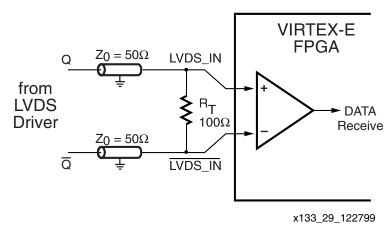

Figure 55: **Receiving LVDS Signal Circuit**

Table 38: **LVDS Voltage Specifications**

Parameter	Min	Typ	Max
V_{CCO}	2.375	2.5	2.625
V_{ICM}(2)	0.2	1.25	2.2
V_{OCM}(1)	1.125	1.25	1.375
V_{IDIFF} (1)	0.1	0.35	-
V_{ODIFF} (1)	0.25	0.35	0.45
V_{OH}(1)	1.25	-	-
V_{OL} (1)	-	-	1.25

Notes:
1. Measured with a 100 Ω resistor across Q and $\overline{Q}$.
2. Measured with a differential input voltage = +/– 350 mV.

LVPECL

Depending on whether the device is transmitting or receiving an LVPECL signal, two different circuits are used for LVPECL termination. A sample circuit illustrating a valid termination technique for transmitting LVPECL signals appears in Figure 56. A sample circuit illustrating a valid termination for receiving LVPECL signals appears in Figure 57. Table 39 lists DC voltage specifications. Further information on the specific termination resistor packs shown can be found on Table 40.

Table 39: **LVPECL Voltage Specifications**

Parameter	Min	Typ	Max
V_{CCO}	3.0	3.3	3.6
V_{REF}	-	-	-
V_{TT}	-	-	-
V_{IH}	1.49	-	2.72
V_{IL}	0.86	-	2.125
V_{OH}	1.8	-	-
V_{OL}	-	-	1.57

Notes:
1. For more detailed information, see **DS022-3**: *Virtex-E 1.8V FPGA DC and Switching Characteristics*, Module 3, LVPECL DC Specifications section.

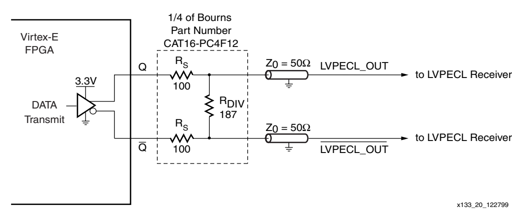

Figure 56: **Transmitting LVPECL Signal Circuit**

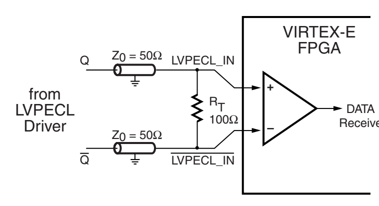

Figure 57: **Receiving LVPECL Signal Circuit**

Table 42: **Input Library Macros**

Name	Inputs	Outputs
IBUFDS_FD_LVDS	I, IB, C	Q
IBUFDS_FDE_LVDS	I, IB, CE, C	Q
IBUFDS_FDC_LVDS	I, IB, C, CLR	Q
IBUFDS_FDCE_LVDS	I, IB, CE, C, CLR	Q
IBUFDS_FDP_LVDS	I, IB, C, PRE	Q
IBUFDS_FDPE_LVDS	I, IB, CE, C, PRE	Q
IBUFDS_FDR_LVDS	I, IB, C, R	Q
IBUFDS_FDRE_LVDS	I, IB, CE, C, R	Q
IBUFDS_FDS_LVDS	I, IB, C, S	Q
IBUFDS_FDSE_LVDS	I, IB, CE, C, S	Q
IBUFDS_LD_LVDS	I, IB, G	Q
IBUFDS_LDE_LVDS	I, IB, GE, G	Q
IBUFDS_LDC_LVDS	I, IB, G, CLR	Q
IBUFDS_LDCE_LVDS	I, IB, GE, G, CLR	Q
IBUFDS_LDP_LVDS	I, IB, G, PRE	Q
IBUFDS_LDPE_LVDS	I, IB, GE, G, PRE	Q

Creating LVDS Output Buffers

LVDS output buffers can be placed in a wide number of IOB locations. The exact locations are dependent on the package used. The Virtex-E package information lists the possible locations as IO_L#P for the P-side and IO_L#N for the N-side, where # is the pair number.

HDL Instantiation

Both output buffers are required to be instantiated in the design and placed on the correct IO_L#P and IO_L#N locations. The IOB must have the same net source the following pins, clock (C), set/reset (SR), output (O), output clock enable (OCE). In addition, the output (O) pins must be inverted with respect to each other, and if output registers are used, the INIT states must be opposite values (one HIGH and one LOW). Failure to follow these rules leads to DRC errors in software.

VHDL Instantiation

```
data0_p   : OBUF_LVDS port map
(I=>data_int(0),   O=>data_p(0));

data0_inv: INV       port map
(I=>data_int(0),   O=>data_n_int(0));

data0_n   : OBUF_LVDS port map
(I=>data_n_int(0), O=>data_n(0));
```

Verilog Instantiation

```
OBUF_LVDS data0_p   (.I(data_int[0]),
.O(data_p[0]));

INV       data0_inv (.I(data_int[0],
.O(data_n_int[0]);

OBUF_LVDS data0_n   (.I(data_n_int[0]),
.O(data_n[0]));
```

Location Constraints

All LVDS buffers must be explicitly placed on a device. For the output buffers this can be done with the following constraint in the .ucf or .ncf file.

```
NET data_p<0> LOC = D28; # IO_L0P

NET data_n<0> LOC = B29; # IO_L0N
```

Synchronous vs. Asynchronous Outputs

If the outputs are synchronous (registered in the IOB) then any IO_L#P|N pair can be used. If the outputs are asynchronous (no output register), then they must use one of the pairs that are part of the same IOB group at the end of a ROW or COLUMN in the device.

The LVDS pairs that can be used as asynchronous outputs are listed in the Virtex-E pinout tables. Some pairs are marked as asynchronous-capable for all devices in that package, and others are marked as available only for that device in the package. If the device size might change at some point in the product lifetime, then only the common pairs for all packages should be used.

Adding an Output Register

All LVDS buffers can have an output register in the IOB. The output registers must be in both the P-side and N-side IOBs. All the normal IOB register options are available (FD, FDE, FDC, FDCE, FDP, FDPE, FDR, FDRE, FDS, FDSE, LD, LDE, LDC, LDCE, LDP, LDPE). The register elements can be inferred or explicitly instantiated in the HDL code.

Special care must be taken to insure that the D pins of the registers are inverted and that the INIT states of the registers are opposite. The clock pin (C), clock enable (CE) and set/reset (CLR/PRE or S/R) pins must connect to the same source. Failure to do this leads to a DRC error in the software.

The register elements can be packed in the IOB using the IOB property to TRUE on the register or by using the “map -pr [i|o|b]” where “i” is inputs only, “o” is outputs only and “b” is both inputs and outputs.

To improve design coding times VHDL and Verilog synthesis macro libraries have been developed to explicitly create these structures. The output library macros are listed in Table 43. The O and OB inputs to the macros are the external net connections.

Table 43: **Output Library Macros**

Name	Inputs	Outputs
OBUFDS_FD_LVDS	D, C	O, OB
OBUFDS_FDE_LVDS	DD, CE, C	O, OB
OBUFDS_FDC_LVDS	D, C, CLR	O, OB
OBUFDS_FDCE_LVDS	D, CE, C, CLR	O, OB
OBUFDS_FDP_LVDS	D, C, PRE	O, OB
OBUFDS_FDPE_LVDS	D, CE, C, PRE	O, OB
OBUFDS_FDR_LVDS	D, C, R	O, OB
OBUFDS_FDRE_LVDS	D, CE, C, R	O, OB
OBUFDS_FDS_LVDS	D, C, S	O, OB
OBUFDS_FDSE_LVDS	D, CE, C, S	O, OB
OBUFDS_LD_LVDS	D, G	O, OB
OBUFDS_LDE_LVDS	D, GE, G	O, OB
OBUFDS_LDC_LVDS	D, G, CLR	O, OB
OBUFDS_LDCE_LVDS	D, GE, G, CLR	O, OB
OBUFDS_LDP_LVDS	D, G, PRE	O, OB
OBUFDS_LDPE_LVDS	D, GE, G, PRE	O, OB

Creating LVDS Output 3-State Buffers

LVDS output 3-state buffers can be placed in a wide number of IOB locations. The exact locations are dependent on the package used. The Virtex-E package information lists the possible locations as IO_L#P for the P-side and IO_L#N for the N-side, where # is the pair number.

HDL Instantiation

Both output 3-state buffers are required to be instantiated in the design and placed on the correct IO_L#P and IO_L#N locations. The IOB must have the same net source the following pins, clock (C), set/reset (SR), 3-state (T), 3-state clock enable (TCE), output (O), output clock enable (OCE). In addition, the output (O) pins must be inverted with respect to each other, and if output registers are used, the INIT states must be opposite values (one High and one Low). If 3-state registers are used, they must be initialized to the same state. Failure to follow these rules leads to DRC errors in the software.

VHDL Instantiation

```
data0_p:   OBUFT_LVDS port map
(I=>data_int(0), T=>data_tri,
O=>data_p(0));

data0_inv: INV port map
(I=>data_int(0), O=>data_n_int(0));

data0_n:    OBUFT_LVDS port map
(I=>data_n_int(0), T=>data_tri,
O=>data_n(0));
```

Verilog Instantiation

```
OBUFT_LVDS data0_p   (.I(data_int[0]),
.T(data_tri), .O(data_p[0]));

INV        data0_inv (.I(data_int[0],
.O(data_n_int[0]);

OBUFT_LVDS data0_n   (.I(data_n_int[0]),
.T(data_tri), .O(data_n[0]));
```

Location Constraints

All LVDS buffers must be explicitly placed on a device. For the output buffers this can be done with the following constraint in the .ucf or .ncf file.

```
NET data_p<0> LOC = D28; # IO_L0P

NET data_n<0> LOC = B29; # IO_L0N
```

Synchronous vs. Asynchronous 3-State Outputs

If the outputs are synchronous (registered in the IOB), then any IO_L#P|N pair can be used. If the outputs are asynchronous (no output register), then they must use one of the pairs that are part of the same IOB group at the end of a ROW or COLUMN in the device. This applies for either the 3-state pin or the data out pin.

LVDS pairs that can be used as asynchronous outputs are listed in the Virtex-E pinout tables. Some pairs are marked as “asynchronous capable” for all devices in that package, and others are marked as available only for that device in the package. If the device size might be changed at some point in the product lifetime, then only the common pairs for all packages should be used.

Adding Output and 3-State Registers

All LVDS buffers can have an output register in the IOB. The output registers must be in both the P-side and N-side IOBs. All the normal IOB register options are available (FD, FDE, FDC, FDCE, FDP, FDPE, FDR, FDRE, FDS, FDSE, LD, LDE, LDC, LDCE, LDP, LDPE). The register elements can be inferred or explicitly instantiated in the HDL code.

Special care must be taken to insure that the D pins of the registers are inverted and that the INIT states of the registers are opposite. The 3-state (T), 3-state clock enable (CE), clock pin (C), output clock enable (CE) and set/reset (CLR/PRE or S/R) pins must connect to the same source. Failure to do this leads to a DRC error in the software.

Virtex-E Switching Characteristics

All devices are 100% functionally tested. Internal timing parameters are derived from measuring internal test patterns. Listed below are representative values. For more specific, more precise, and worst-case guaranteed data, use the values reported by the static timing analyzer (TRCE in the Xilinx Development System) and back-annotated to the simulation net list. All timing parameters assume worst-case operating conditions (supply voltage and junction temperature). Values apply to all Virtex-E devices unless otherwise noted.

IOB Input Switching Characteristics

Input delays associated with the pad are specified for LVTTL levels in Table 2. For other standards, adjust the delays with the values shown in **IOB Input Switching Characteristics Standard Adjustments**, page 8.

Table 2: **IOB Input Switching Characteristics**

			Speed Grade(1)				
Description(2)	**Symbol**	**Device**	**Min**	**-8**	**-7**	**-6**	**Units**
Propagation Delays							
Pad to I output, no delay	T_{IOPI}	All	0.43	0.8	0.8	0.8	ns, max
Pad to I output, with delay	T_{IOPID}	XCV50E	0.51	1.0	1.0	1.0	ns, max
		XCV100E	0.51	1.0	1.0	1.0	ns, max
		XCV200E	0.51	1.0	1.0	1.0	ns, max
		XCV300E	0.51	1.0	1.0	1.0	ns, max
		XCV400E	0.51	1.0	1.0	1.0	ns, max
		XCV600E	0.51	1.0	1.0	1.0	ns, max
		XCV1000E	0.55	1.1	1.1	1.1	ns, max
		XCV1600E	0.55	1.1	1.1	1.1	ns, max
		XCV2000E	0.55	1.1	1.1	1.1	ns, max
		XCV2600E	0.55	1.1	1.1	1.1	ns, max
		XCV3200E	0.55	1.1	1.1	1.1	ns, max
Pad to output IQ via transparent latch, no delay	T_{IOPLI}	All	0.8	1.4	1.5	1.6	ns, max
Pad to output IQ via transparent latch, with delay	T_{IOPLID}	XCV50E	1.31	2.9	3.0	3.1	ns, max
		XCV100E	1.31	2.9	3.0	3.1	ns, max
		XCV200E	1.39	3.1	3.2	3.3	ns, max
		XCV300E	1.39	3.1	3.2	3.3	ns, max
		XCV400E	1.43	3.2	3.3	3.4	ns, max
		XCV600E	1.55	3.5	3.6	3.7	ns, max
		XCV1000E	1.55	3.5	3.6	3.7	ns, max
		XCV1600E	1.59	3.6	3.7	3.8	ns, max
		XCV2000E	1.59	3.6	3.7	3.8	ns, max
		XCV2600E	1.59	3.6	3.7	3.8	ns, max
		XCV3200E	1.59	3.6	3.7	3.8	ns, max

CLB Distributed RAM Switching Characteristics

Description	Symbol	Speed Grade[(1)] Min	-8	-7	-6	Units
Sequential Delays						
Clock CLK to X/Y outputs (WE active) 16 x 1 mode	$T_{SHCKO16}$	0.67	1.38	1.5	1.7	ns, max
Clock CLK to X/Y outputs (WE active) 32 x 1 mode	$T_{SHCKO32}$	0.84	1.66	1.9	2.1	ns, max
Shift-Register Mode						
Clock CLK to X/Y outputs	T_{REG}	1.25	2.39	2.9	3.2	ns, max
Setup and Hold Times before/after Clock CLK						
F/G address inputs	T_{AS}/T_{AH}	0.19 / 0	0.38 / 0	0.42 / 0	0.47 / 0	ns, min
BX/BY data inputs (DIN)	T_{DS}/T_{DH}	0.44 / 0	0.87 / 0	0.97 / 0	1.09 / 0	ns, min
SR input (WE)	T_{WS}/T_{WH}	0.29 / 0	0.57 / 0	0.7 / 0	0.8 / 0	ns, min
Clock CLK						
Minimum Pulse Width, High	T_{WPH}	0.96	1.9	2.1	2.4	ns, min
Minimum Pulse Width, Low	T_{WPL}	0.96	1.9	2.1	2.4	ns, min
Minimum clock period to meet address write cycle time	T_{WC}	1.92	3.8	4.2	4.8	ns, min
Shift-Register Mode						
Minimum Pulse Width, High	T_{SRPH}	1.0	1.9	2.1	2.4	ns, min
Minimum Pulse Width, Low	T_{SRPL}	1.0	1.9	2.1	2.4	ns, min

Notes:

1. A Zero "0" Hold Time listing indicates no hold time or a negative hold time. Negative values can not be guaranteed "best-case", but if a "0" is listed, there is no positive hold time.

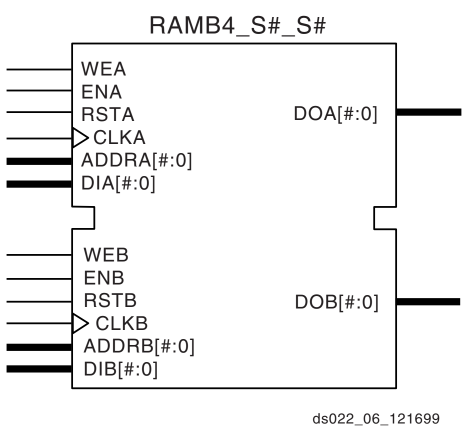

Figure 3: **Dual-Port Block SelectRAM**

Table 4: **CS144 — XCV50E, XCV100E, XCV200E**

Bank	Pin Description	Pin #
1	VCCO	A13
1	VCCO	D7
2	VCCO	B12
3	VCCO	G11
3	VCCO	M13
4	VCCO	N13
5	VCCO	N1
5	VCCO	N7
6	VCCO	M2
7	VCCO	B2
7	VCCO	G2
NA	GND	A1
NA	GND	B9
NA	GND	B11
NA	GND	C7
NA	GND	D5
NA	GND	E4
NA	GND	E11
NA	GND	F1
NA	GND	G10
NA	GND	J1
NA	GND	J12
NA	GND	L3
NA	GND	L5
NA	GND	L7
NA	GND	L9
NA	GND	N12

Notes:

1. V_{REF} or I/O option only in the XCV200E; otherwise, I/O option only.
2. V_{REF} or I/O option only in the XCV100E, 200E; otherwise, I/O option only.

CS144 Differential Pin Pairs

Virtex-E devices have differential pin pairs that can also provide other functions when not used as a differential pair. A √ in the AO column indicates that the pin pair can be used as an asynchronous output for all devices provided in this package. Pairs with a note number in the AO column are device dependent. They can have asynchronous outputs if the pin pair are in the same CLB row and column in the device. Numbers in this column refer to footnotes that indicate which devices have pin pairs than can be asynchronous outputs. The Other Functions column indicates alternative function(s) not available when the pair is used as a differential pair or differential clock.

Table 5: **CS144 Differential Pin Pair Summary XCV50E, XCV100E, XCV200E**

Pair	Bank	P Pin	N Pin	AO	Other Functions
Global Differential Clock					
0	4	K7	N8	NA	IO_DLL_L18P
1	5	M7	M6	NA	IO_DLL_L18N
2	1	A7	B7	NA	IO_DLL_L2P
3	0	A6	C6	NA	IO_DLL_L2N
IO LVDS Total Pairs: 30, Asynchronous Output Pairs: 18					
0	0	A4	B4	√	VREF
1	0	A5	B5	√	-
2	1	B7	C6	NA	IO_LVDS_DLL
3	1	D8	C8	√	-
4	1	D9	C9	√	VREF
5	1	D10	C10	√	CS, WRITE
6	2	C11	C12	√	DIN, D0
7	2	D13	E10	1	D1, VREF
8	2	E12	E13	√	D2
9	2	F10	F11	1	D3, VREF
10	3	F13	G13	NA	-
11	3	H12	H11	1	D4, VREF
12	3	H10	J13	√	D5
13	3	J11	J10	1	D6, VREF
14	3	K10	L13	√	INIT
15	4	L11	M11	√	-
16	4	N10	K9	√	VREF
17	4	N9	K8	√	-

Table 6: **PQ240 — XCV50E, XCV100E, XCV200E, XCV300E, XCV400E**

Pin #	Pin Description	Bank
P74	IO_L43P_YY	5
P73[1]	IO_VREF_L43N_YY	5
P72	IO	5
P71	IO_L44P_YY	5
P70	IO_VREF_L44N_YY	5
P68	IO_L45P_YY	5
P67	IO_L45N_YY	5
P66[2]	IO_VREF_L46P_Y	5
P65	IO_L46N_Y	5
P64	IO_L47P_YY	5
P63	IO_L47N_YY	5
P57	IO_L48N_YY	6
P56	IO_L48P_YY	6
P54[2]	IO_VREF	6
P53	IO_L49N_Y	6
P52	IO_L49P_Y	6
P50	IO_VREF_L50N_Y	6
P49	IO_L50P_Y	6
P48	IO	6
P47[1]	IO_VREF_L51N_Y	6
P46	IO_L51P_Y	6
P42	IO_L52N_YY	6
P41	IO_L52P_YY	6
P40	IO	6
P39	IO_L53N_Y	6
P38	IO_L53P_Y	6
P36	IO_VREF_L54N_Y	6
P35	IO_L54P_Y	6
P34	IO_L55N_Y	6
P33[3]	IO_VREF_L55P_Y	6
P31	IO	6
P28	IO_L56N_YY	7
P27	IO_L56P_YY	7

Table 6: **PQ240 — XCV50E, XCV100E, XCV200E, XCV300E, XCV400E**

Pin #	Pin Description	Bank
P26[3]	IO_VREF	7
P24	IO_L57N_Y	7
P23	IO_VREF_L57P_Y	7
P21	IO_L58N_Y	7
P20	IO_L58P_Y	7
P19	IO	7
P18	IO_L59N_YY	7
P17	IO_L59P_YY	7
P13	IO_L60N_Y	7
P12[1]	IO_VREF_L60P_Y	7
P11	IO	7
P10	IO_L61N_Y	7
P9	IO_VREF_L61P_Y	7
P7	IO_L62N_Y	7
P6	IO_L62P_Y	7
P5[2]	IO_VREF_L63N_Y	7
P4	IO_L63P_Y	7
P3	IO	7
P179	CCLK	2
P120	DONE	3
P60	M0	NA
P58	M1	NA
P62	M2	NA
P122	PROGRAM	NA
P183	TDI	NA
P239	TCK	NA
P181	TDO	2
P2	TMS	NA
P225	VCCINT	NA
P214	VCCINT	NA
P198	VCCINT	NA
P164	VCCINT	NA
P148	VCCINT	NA

Table 10: **BG352 — XCV100E, XCV200E, XCV300E**

Bank	Pin Description	Pin #
2	IO_D3_L30N _Y	M3
2	IO_L31P	M2
2	IO_L31N	M1
2	IO	N3[1]
2	IO_L32P_YY	N4
2	IO_L32N_YY	N2
3	IO	P1
3	IO	P3[1]
3	IO_L33P	R1
3	IO_L33N	R2
3	IO_D4_L34P _Y	R3
3	IO_VREF_3_L34N _Y	R4
3	IO_L35P_YY	T2
3	IO_L35N_YY	U2
3	IO	T3[1]
3	IO_L36P	T4
3	IO_L36N	V1
3	IO	V2[1]
3	IO_L37P_YY	U3
3	IO_D5_L37N_YY	U4
3	IO_D6_L38P _Y	V3
3	IO_VREF_3_L38N _Y	V4
3	IO_L39P _Y	Y1
3	IO_L39N _Y	Y2
3	IO	W3
3	IO	W4[1]
3	IO	AA1[1]
3	IO_L40P_Y	AA2
3	IO_VREF_3_L40N_Y	Y3
3	IO_L41P_YY	AC1
3	IO_L41N_YY	AB2
3	IO	AA3[1]
3	IO_L42P_YY	AA4

Table 10: **BG352 — XCV100E, XCV200E, XCV300E**

Bank	Pin Description	Pin #
3	IO_VREF_3_L42N_YY	AC2[2]
3	IO	AB3
3	IO	AD1[1]
3	IO	AB4[1]
3	IO_D7_L43P_YY	AC3
3	IO_INIT_L43N_YY	AD2
4	IO_L44P_YY	AC5
4	IO_L44N_YY	AD4
4	IO	AE3[1]
4	IO	AD5[1]
4	IO	AC6
4	IO_VREF_4_L45P_YY	AE4[2]
4	IO_L45N_YY	AF3
4	IO	AF4[1]
4	IO_L46P_YY	AC7
4	IO_L46N_YY	AD6
4	IO_VREF_4_L47P_YY	AE5
4	IO_L47N_YY	AE6
4	IO	AD7[1]
4	IO	AE7[1]
4	IO_L48P	AF6
4	IO_L48N	AC9
4	IO	AD8
4	IO_VREF_4_L49P_YY	AE8
4	IO_L49N_YY	AF7
4	IO_L50P_YY	AD9
4	IO_L50N_YY	AE9
4	IO	AD10[1]
4	IO_L51P	AF9
4	IO_L51N	AC11
4	IO	AE10[1]
4	IO_L52P_Y	AD11
4	IO_L52N_Y	AE11

Table 14: **BG560 — XCV400E, XCV600E, XCV1000E, XCV1600E, XCV2000E**

Bank	Pin Description	Pin#	See Note
7	IO_L165N_YY	P32	
7	IO_VREF_L165P_YY	P31	
7	IO_L166N_Y	P30	
7	IO_L166P_Y	P29	
7	IO_L167N_Y	M32	
7	IO_L167P_Y	N31	
7	IO_L168N_Y	N30	
7	IO_VREF_L168P_Y	L33	3
7	IO_L169N_Y	M31	
7	IO_L169P_Y	L32	
7	IO_L170N_Y	M30	
7	IO_L170P_Y	L31	
7	IO_L171N_YY	M29	
7	IO_L171P_YY	J33	
7	IO_L172N_YY	L30	
7	IO_VREF_L172P_YY	K31	
7	IO_L173N_Y	L29	
7	IO_L173P_Y	H33	
7	IO_L174N_Y	J31	
7	IO_VREF_L174P_Y	H32	4
7	IO_L175N_Y	K29	
7	IO_L175P_Y	H31	
7	IO_L176N_Y	J30	
7	IO_VREF_L176P_Y	G32	1
7	IO_L177N_YY	J29	
7	IO_VREF_L177P_YY	G31	
7	IO_L178N_Y	E33	
7	IO_L178P_Y	E32	
7	IO_L179N_Y	H29	
7	IO_L179P_Y	F31	
7	IO_L180N_Y	D32	
7	IO_VREF_L180P_Y	E31	
7	IO_L181N_Y	G29	
7	IO_L181P_Y	C33	
7	IO_L182N_Y	F30	

Table 14: **BG560 — XCV400E, XCV600E, XCV1000E, XCV1600E, XCV2000E**

Bank	Pin Description	Pin#	See Note
7	IO_VREF_L182P_Y	D31	3
2	CCLK	C4	
3	DONE	AJ5	
NA	DXN	AK29	
NA	DXP	AJ28	
NA	M0	AJ29	
NA	M1	AK30	
NA	M2	AN32	
NA	PROGRAM	AM1	
NA	TCK	E29	
NA	TDI	D5	
2	TDO	E6	
NA	TMS	B33	
NA	NC	C31	
NA	NC	AC2	
NA	NC	AK4	
NA	NC	AL3	
NA	VCCINT	A21	
NA	VCCINT	B12	
NA	VCCINT	B14	
NA	VCCINT	B18	
NA	VCCINT	B28	
NA	VCCINT	C22	
NA	VCCINT	C24	
NA	VCCINT	E9	
NA	VCCINT	E12	
NA	VCCINT	F2	
NA	VCCINT	H30	
NA	VCCINT	J1	
NA	VCCINT	K32	
NA	VCCINT	M3	
NA	VCCINT	N1	

Table 20: **FG676 — XCV400E, XCV600E**

Bank	Pin Description	Pin #
1	IO_L40P_YY	D20
1	IO_L41N_YY	F19
1	IO_VREF_L41P_YY	C21
1	IO_L42N_YY	B22
1	IO_L42P_YY	E20
1	IO_L43N_Y	A23
1	IO_L43P_Y	D21
1	IO_WRITE_L44N_YY	C22
1	IO_CS_L44P_YY	E21
2	IO	D25[1]
2	IO	D26
2	IO	E26
2	IO	F26
2	IO	H26[1]
2	IO	K26[1]
2	IO	M25[1]
2	IO	N26[1]
2	IO_D1	K24
2	IO_DOUT_BUSY_L45P_YY	E23
2	IO_DIN_D0_L45N_YY	F22
2	IO_L46P_YY	E24
2	IO_L46N_YY	F20
2	IO_L47P_Y	G21
2	IO_L47N_Y	G22
2	IO_VREF_L48P_Y	F24
2	IO_L48N_Y	H20
2	IO_L49P_Y	E25
2	IO_L49N_Y	H21
2	IO_L50P_YY	F23
2	IO_L50N_YY	G23
2	IO_VREF_L51P_YY	H23
2	IO_L51N_YY	J20
2	IO_L52P_YY	G24
2	IO_L52N_YY	H22
2	IO_L53P_Y	J21
2	IO_L53N_Y	G25

Table 20: **FG676 — XCV400E, XCV600E**

Bank	Pin Description	Pin #
2	IO_VREF_L54P_Y	G26[2]
2	IO_L54N_Y	J22
2	IO_L55P_YY	H24
2	IO_L55N_YY	J23
2	IO_L56P_YY	J24
2	IO_VREF_L56N_YY	K20
2	IO_D2_L57P_YY	K22
2	IO_L57N_YY	K21
2	IO_L58P_YY	H25
2	IO_L58N_YY	K23
2	IO_L59P_Y	L20
2	IO_L59N_Y	J26
2	IO_L60P_Y	K25
2	IO_L60N_Y	L22
2	IO_L61P_Y	L21
2	IO_L61N_Y	L23
2	IO_L62P_Y	M20
2	IO_L62N_Y	L24
2	IO_VREF_L63P_YY	M23
2	IO_D3_L63N_YY	M22
2	IO_L64P_YY	L26
2	IO_L64N_YY	M21
2	IO_L65P_Y	N19
2	IO_L65N_Y	M24
2	IO_VREF_L66P_Y	M26
2	IO_L66N_Y	N20
2	IO_L67P_YY	N24
2	IO_L67N_YY	N21
2	IO_L68P_YY	N23
2	IO_L68N_YY	N22
3	IO	P24
3	IO	P26[1]
3	IO	R26[1]
3	IO	T26[1]
3	IO	U26[1]
3	IO	W25

Table 22: **FG680 - XCV600E, XCV1000E, XCV1600E, XCV2000E**

Bank	Pin Description	Pin #
6	IO_VREF_L200N_YY	AH39
6	IO_L200P_YY	AG38
6	IO_L201N_YY	AG36
6	IO_L201P_YY	AG39
6	IO_L202N_Y	AG37
6	IO_L202P_Y	AF39
6	IO_L203N	AF36
6	IO_L203P	AE38
6	IO_L204N	AF37
6	IO_L204P	AF38
6	IO_VREF_L205N_Y	AE39[1]
6	IO_L205P_Y	AE36
6	IO_L206N_YY	AD38
6	IO_L206P_YY	AE37
6	IO_L207N	AD39
6	IO_L207P	AD36
6	IO_L208N_Y	AC38
6	IO_L208P_Y	AC39
6	IO_VREF_L209N_YY	AD37
6	IO_L209P_YY	AB38
6	IO_L210N_YY	AC35
6	IO_L210P_YY	AB39
6	IO_L211N	AC36
6	IO_L211P	AA38
6	IO_L212N	AC37
6	IO_L212P	AA39
6	IO_VREF_L213N_YY	AB35
6	IO_L213P_YY	Y38
6	IO_L214N_YY	AB36
6	IO_L214P_YY	Y39
6	IO_VREF_L215N	AB37[2]
6	IO_L215P	AA36
7	IO	C38
7	IO	B37
7	IO	F37

Table 22: **FG680 - XCV600E, XCV1000E, XCV1600E, XCV2000E**

Bank	Pin Description	Pin #
7	IO_L216N_YY	AA37
7	IO_L216P_YY	W38
7	IO_L217N	W37
7	IO_VREF_L217P	V39[2]
7	IO_L218N_YY	W36
7	IO_L218P_YY	U39
7	IO_L219N_YY	V38
7	IO_VREF_L219P_YY	U38
7	IO_L220N	V37
7	IO_L220P	T39
7	IO_L221N	V36
7	IO_L221P	T38
7	IO_L222N_YY	V35
7	IO_L222P_YY	R39
7	IO_L223N_YY	U37
7	IO_VREF_L223P_YY	U36
7	IO_L224N_Y	R38
7	IO_L224P_Y	U35
7	IO_L225N	P39
7	IO_L225P	T37
7	IO_L226N_YY	P38
7	IO_L226P_YY	T36
7	IO_L227N_Y	N39
7	IO_VREF_L227P_Y	N38[1]
7	IO_L228N	R37
7	IO_L228P	M39
7	IO_L229N	R36
7	IO_L229P	M38
7	IO_L230N_Y	P37
7	IO_L230P_Y	L39
7	IO_L231N_YY	P36
7	IO_L231P_YY	N37
7	IO_L232N_YY	L38
7	IO_VREF_L232P_YY	N36
7	IO_L233N	K39
7	IO_L233P	M37

Table 24: **FG860 — XCV1000E, XCV1600E, XCV2000E**

Bank	Pin Description	Pin #
3	IO_L117N_Y	AJ5
3	IO_L118P	AG2
3	IO_L118N	AK4
3	IO_L119P_Y	AG3
3	IO_L119N_Y	AL4
3	IO_L120P_Y	AH1
3	IO_L120N_Y	AL5
3	IO_L121P_Y	AH2
3	IO_L121N_Y	AM4
3	IO_L122P_YY	AH3
3	IO_D5_L122N_YY	AM5
3	IO_D6_L123P_YY	AJ1
3	IO_VREF_L123N_YY	AN3
3	IO_L124P_Y	AN4
3	IO_L124N_Y	AJ3
3	IO_L125P_YY	AN5
3	IO_L125N_YY	AK1
3	IO_L126P_YY	AK2
3	IO_VREF_L126N_YY	AP4
3	IO_L127P_Y	AK3
3	IO_L127N_Y	AP5
3	IO_L128P_Y	AR3
3	IO_VREF_L128N_Y	AL2[2]
3	IO_L129P_YY	AR4
3	IO_L129N_YY	AL3
3	IO_L130P_YY	AM1
3	IO_VREF_L130N_YY	AT3
3	IO_L131P_Y	AM2
3	IO_L131N_Y	AT4
3	IO_L132P_Y	AT5
3	IO_L132N_Y	AN1
3	IO_L133P_YY	AU3
3	IO_L133N_YY	AN2
3	IO_L134P_Y	AP1
3	IO_VREF_L134N_Y	AP2
3	IO_L135P_Y	AR1
3	IO_L135N_Y	AV3
3	IO_L136P	AR2
3	IO_L136N	AT1
3	IO_L137P_Y	AV4
3	IO_VREF_L137N_Y	AT2
3	IO_L138P_Y	AU1
3	IO_L138N_Y	AU5
3	IO_L139P_Y	AU2
3	IO_L139N_Y	AW3
3	IO_D7_L140P_YY	AV1
3	IO_INIT_L140N_YY	AW5
4	GCK0	BA22
4	IO	AV17
4	IO	AY11
4	IO	AY12
4	IO	AY13
4	IO	AY14
4	IO	BA8
4	IO	BA17
4	IO	BA19
4	IO	BA20
4	IO	BA21
4	IO	BB9
4	IO	BB18
4	IO_L141P_YY	AV6
4	IO_L141N_YY	BA4
4	IO_L142P_Y	AY4
4	IO_L142N_Y	BA5
4	IO_L143P_Y	AW6
4	IO_L143N_Y	BB5
4	IO_VREF_L144P_Y	BA6
4	IO_L144N_Y	AY5
4	IO_L145P_Y	BB6
4	IO_L145N_Y	AY6
4	IO_L146P_YY	BA7
4	IO_L146N_YY	AV7
4	IO_VREF_L147P_YY	BB7

Table 24: **FG860 — XCV1000E, XCV1600E, XCV2000E**

Bank	Pin Description	Pin #
7	IO_L275N_Y	G38
7	IO_VREF_L275P_Y	G42
7	IO_L276N_Y	G41
7	IO_L276P_Y	F40
7	IO_L277N	F42
7	IO_L277P	F41
7	IO_L278N_Y	F39
7	IO_VREF_L278P_Y	E42
7	IO_L279N_Y	E40
7	IO_L279P_Y	E41
7	IO_L280N_Y	E39
7	IO_L280P_Y	D41
2	CCLK	B4
3	DONE	AW2
NA	DXN	BA38
NA	DXP	AW38
NA	M0	AW41
NA	M1	AV37
NA	M2	BA39
NA	PROGRAM	AV2
NA	TCK	B38
NA	TDI	B5
2	TDO	D5
NA	TMS	B39
NA	VCCINT	F9
NA	VCCINT	F10
NA	VCCINT	F17
NA	VCCINT	F18
NA	VCCINT	F25
NA	VCCINT	F26
NA	VCCINT	F33
NA	VCCINT	F34
NA	VCCINT	J6
NA	VCCINT	J37
NA	VCCINT	K6

Table 24: **FG860 — XCV1000E, XCV1600E, XCV2000E**

Bank	Pin Description	Pin #
NA	VCCINT	K37
NA	VCCINT	T6
NA	VCCINT	T37
NA	VCCINT	U6
NA	VCCINT	U37
NA	VCCINT	V6
NA	VCCINT	V37
NA	VCCINT	AE6
NA	VCCINT	AE37
NA	VCCINT	AF6
NA	VCCINT	AF37
NA	VCCINT	AG6
NA	VCCINT	AG37
NA	VCCINT	AN6
NA	VCCINT	AN37
NA	VCCINT	AP6
NA	VCCINT	AP37
NA	VCCINT	AU9
NA	VCCINT	AU10
NA	VCCINT	AU17
NA	VCCINT	AU18
NA	VCCINT	AU25
NA	VCCINT	AU26
NA	VCCINT	AU33
NA	VCCINT	AU34
NA	VCCO_0	F23
NA	VCCO_0	F24
NA	VCCO_0	F28
NA	VCCO_0	F29
NA	VCCO_0	F31
NA	VCCO_0	F32
NA	VCCO_0	F35
NA	VCCO_0	F36
NA	VCCO_1	F11
NA	VCCO_1	F12
NA	VCCO_1	F14

Table 26: **FG900 — XCV600E, XCV1000E, XCV1600E**

Bank	Pin Description	Pin #
6	IO	AC5[4]
6	IO	AD1[4]
6	IO	AE5[5]
6	IO_L212N_YY	AF3
6	IO_L212P_YY	AC6
6	IO_L213N	AH2[4]
6	IO_L213P	AG2[3]
6	IO_L214N	AB9
6	IO_L214P	AE4
6	IO_VREF_L215N_YY	AE3[1]
6	IO_L215P_YY	AH1
6	IO_L216N_Y	AB8[4]
6	IO_L216P_Y	AD6[3]
6	IO_L217N_YY	AG1
6	IO_L217P_YY	AA10
6	IO_VREF_L218N	AA9
6	IO_L218P	AD4
6	IO_L219N_YY	AD5
6	IO_L219P_YY	AD2
6	IO_L220N_YY	AD3
6	IO_L220P_YY	AF2
6	IO_L221N	AA8
6	IO_L221P	AA7
6	IO_VREF_L222N_YY	AF1
6	IO_L222P_YY	Y9
6	IO_L223N_YY	AB6
6	IO_L223P_YY	AC4
6	IO_L224N	AE1
6	IO_L224P	W8
6	IO_L225N_YY	Y8
6	IO_L225P_YY	AB4
6	IO_VREF_L226N_YY	AB3
6	IO_L226P_YY	W9
6	IO_L227N_YY	AA5[4]
6	IO_L227P_YY	W10[3]
6	IO_L228N_YY	AB1
6	IO_L228P_YY	V10

Table 26: **FG900 — XCV600E, XCV1000E, XCV1600E**

Bank	Pin Description	Pin #
6	IO_L229N_YY	Y7[4]
6	IO_VREF_L229P_YY	AC1
6	IO_L230N	V11
6	IO_L230P	AA3
6	IO_L231N_YY	AA2[3]
6	IO_L231P_YY	U10[4]
6	IO_L232N	W7
6	IO_L232P	AA6
6	IO_L233N_YY	Y6
6	IO_L233P_YY	Y4
6	IO_L234N_Y	AA1[4]
6	IO_L234P_Y	V7[4]
6	IO_L235N_YY	Y3
6	IO_L235P_YY	Y2
6	IO_VREF_L236N	Y5[1]
6	IO_L236P	W5
6	IO_L237N_YY	W4
6	IO_L237P_YY	W6
6	IO_L238N_YY	V6
6	IO_L238P_YY	W2
6	IO_L239N	U9
6	IO_L239P	V4
6	IO_VREF_L240N_YY	AB2
6	IO_L240P_YY	T8
6	IO_L241N_YY	U5
6	IO_L241P_YY	W1
6	IO_L242N	Y1
6	IO_L242P	T9
6	IO_L243N_YY	T7
6	IO_L243P_YY	U3
6	IO_VREF_L244N_YY	T5
6	IO_L244P_YY	V2
6	IO_L245N_YY	R9[4]
6	IO_L245P_YY	T6[3]
6	IO_VREF_L246N_YY	T4[2]
6	IO_L246P_YY	U2
6	IO_L247N	T1

Table 26: **FG900 — XCV600E, XCV1000E, XCV1600E**

Bank	Pin Description	Pin #
7	IO_L275N_YY	G3
7	IO_L275P_YY	E1
7	IO_L276N_YY	H6
7	IO_L276P_YY	E2
7	IO_L277N	E4
7	IO_VREF_L277P	K9
7	IO_L278N_YY	J8
7	IO_L278P_YY	F4
7	IO_L279N_Y	D1[3]
7	IO_L279P_Y	H7[4]
7	IO_L280N_YY	G6
7	IO_VREF_L280P_YY	C2[1]
7	IO_L281N	D2
7	IO_L281P	F5
7	IO_L282N_YY	D3[4]
7	IO_L282P_YY	K10[3]
2	CCLK	F26
3	DONE	AJ28
NA	DXN	AJ3
NA	DXP	AH4
NA	M0	AF4
NA	M1	AC7
NA	M2	AK3
NA	PROGRAM	AG28
NA	TCK	B3
NA	TDI	H22
2	TDO	D26
NA	TMS	C1
NA	VCCINT	L11
NA	VCCINT	L12
NA	VCCINT	L19
NA	VCCINT	L20
NA	VCCINT	M11
NA	VCCINT	M12
NA	VCCINT	M19

Table 26: **FG900 — XCV600E, XCV1000E, XCV1600E**

Bank	Pin Description	Pin #
NA	VCCINT	M20
NA	VCCINT	N13
NA	VCCINT	N14
NA	VCCINT	N15
NA	VCCINT	N16
NA	VCCINT	N17
NA	VCCINT	N18
NA	VCCINT	P13
NA	VCCINT	P18
NA	VCCINT	R13
NA	VCCINT	R18
NA	VCCINT	T13
NA	VCCINT	T18
NA	VCCINT	U13
NA	VCCINT	U18
NA	VCCINT	V13
NA	VCCINT	V14
NA	VCCINT	V15
NA	VCCINT	V16
NA	VCCINT	V17
NA	VCCINT	V18
NA	VCCINT	W11
NA	VCCINT	W12
NA	VCCINT	W19
NA	VCCINT	W20
NA	VCCINT	Y11
NA	VCCINT	Y12
NA	VCCINT	Y19
NA	VCCINT	Y20
NA	VCCO_0	B6
NA	VCCO_0	M15
NA	VCCO_0	M14
NA	VCCO_0	L15
NA	VCCO_0	L14
NA	VCCO_0	H14
NA	VCCO_0	M13

Table 28: **FG1156 — XCV1000E, XCV1600E, XCV2000E, XCV2600E, XCV3200E**

Bank	Pin Description	Pin #
NA	GND	AP2
NA	GND	AN3
NA	GND	AM20
NA	GND	AK30
NA	GND	AG8
NA	GND	AC29
NA	GND	Y3
NA	GND	Y32
NA	GND	W21
NA	GND	V21
NA	GND	T8
NA	GND	T27
NA	GND	R21
NA	GND	P21
NA	GND	H19
NA	GND	F29
NA	GND	C11
NA	GND	B3
NA	GND	A32
NA	GND	AP3
NA	GND	AN32
NA	GND	AM24
NA	GND	AJ6
NA	GND	AG16
NA	GND	AA14
NA	GND	Y14
NA	GND	W8
NA	GND	W27
NA	GND	U14
NA	GND	T14
NA	GND	R3
NA	GND	R32
NA	GND	M6
NA	GND	H27

Table 28: **FG1156 — XCV1000E, XCV1600E, XCV2000E, XCV2600E, XCV3200E**

Bank	Pin Description	Pin #
NA	GND	E5
NA	GND	C15
NA	GND	B32
NA	GND	A33
NA	GND	AP7
NA	GND	AN33
NA	GND	AM32
NA	GND	AJ12
NA	GND	AG19
NA	GND	AA15
NA	GND	Y15
NA	GND	W14
NA	GND	V14
NA	GND	U15
NA	GND	T15
NA	GND	R14
NA	GND	P14
NA	GND	M29
NA	GND	G1
NA	GND	E18
NA	GND	C20
NA	GND	B33
NA	GND	A34
NA	GND	AP28
NA	GND	AN34
NA	GND	AM33
NA	GND	AJ23
NA	GND	AG27
NA	GND	AA16
NA	GND	Y16
NA	GND	W15
NA	GND	V15
NA	GND	U16
NA	GND	T16

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
12/7/99	1.0	Initial Xilinx release.
1/10/00	1.1	Re-released with spd.txt v. 1.18, FG860/900/1156 package information, and additional DLL, Select RAM and SelectI/O information.
1/28/00	1.2	Added Delay Measurement Methodology table, updated SelectI/O section, Figures 30, 54, & 55, text explaining Table 5, T_{BYP} values, buffered Hex Line info, p. 8, I/O Timing Measurement notes, notes for Tables 15, 16, and corrected F1156 pinout table footnote references.
2/29/00	1.3	Updated pinout tables, V_{CC} page 20, and corrected Figure 20.
5/23/00	1.4	Correction to table on p. 22.
7/10/00	1.5	• Numerous minor edits. • Data sheet upgraded to Preliminary. • Preview -8 numbers added to **Virtex-E Electrical Characteristics** tables.
8/1/00	1.6	• Reformatted entire document to follow new style guidelines. • Changed speed grade values in tables on pages 35-37.
9/20/00	1.7	• Min values added to **Virtex-E Electrical Characteristics** tables. • XCV2600E and XCV3200E numbers added to **Virtex-E Electrical Characteristics** tables (Module 3). • Corrected user I/O count for XCV100E device in Table 1 (Module 1). • Changed several pins to "No Connect in the XCV100E" and removed duplicate V_{CCINT} pins in Table ~ (Module 4). • Changed pin J10 to "No connect in XCV600E" in Table 74 (Module 4). • Changed pin J30 to "V_{REF} or I/O option only in the XCV600E" in Table 74 (Module 4). • Corrected pair 18 in Table 75 (Module 4) to be "AO in the XCV1000E, XCV1600E".
11/20/00	1.8	• Upgraded speed grade -8 numbers in **Virtex-E Electrical Characteristics** tables to Preliminary. • Updated minimums in Table 13 and added notes to Table 14. • Added to note 2 to **Absolute Maximum Ratings**. • Changed speed grade -8 numbers for $T_{SHCKO32}$, T_{REG}, T_{BCCS}, and T_{ICKOF}. • Changed all minimum hold times to –0.4 under **Global Clock Set-Up and Hold for LVTTL Standard, with DLL.** • Revised maximum T_{DLLPW} in -6 speed grade for **DLL Timing Parameters**. • Changed GCLK0 to BA22 for FG860 package in Table 46.
2/12/01	1.9	• Revised footnote for Table 14. • Added numbers to **Virtex-E Electrical Characteristics** tables for XCV1000E and XCV2000E devices. • Updated Table 27 and Table 78 to include values for XCV400E and XCV600E devices. • Revised Table 62 to include pinout information for the XCV400E and XCV600E devices in the BG560 package. • Updated footnotes 1 and 2 for Table 76 to include XCV2600E and XCV3200E devices.