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	7776
Number of Logic Elements/Cells	34992
Total RAM Bits	589824
Number of I/O	660
Number of Gates	2188742
Voltage - Supply	1.71V ~ 1.89V
Mounting Type	Surface Mount
Operating Temperature	-40°C ~ 100°C (TJ)
Package / Case	860-BGA Exposed Pad
Supplier Device Package	860-FBGA (42.5x42.5)
Purchase URL	https://www.e-xfl.com/product-detail/xilinx/xcv1600e-6fg860i

Virtex-E Ordering Information

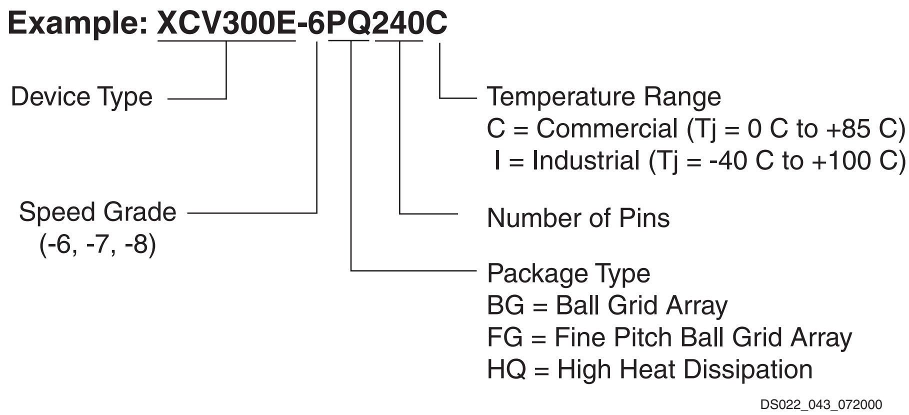

Figure 1: **Ordering Information**

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 “VREF 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“.

forces a storage element into the initialization state specified for it in the configuration. BY forces it into the opposite state. Alternatively, these signals can be configured to operate asynchronously. All of the control signals are independently invertible, and are shared by the two flip-flops within the slice.

Additional Logic

The F5 multiplexer in each slice combines the function generator outputs. This combination provides either a function generator that can implement any 5-input function, a 4:1 multiplexer, or selected functions of up to nine inputs.

Similarly, the F6 multiplexer combines the outputs of all four function generators in the CLB by selecting one of the F5-multiplexer outputs. This permits the implementation of any 6-input function, an 8:1 multiplexer, or selected functions of up to 19 inputs.

Each CLB has four direct feedthrough paths, two per slice. These paths provide extra data input lines or additional local routing that does not consume logic resources.

Arithmetic Logic

Dedicated carry logic provides fast arithmetic carry capability for high-speed arithmetic functions. The Virtex-E CLB supports two separate carry chains, one per Slice. The height of the carry chains is two bits per CLB.

The arithmetic logic includes an XOR gate that allows a 2-bit full adder to be implemented within a slice. In addition, a dedicated AND gate improves the efficiency of multiplier implementation. The dedicated carry path can also be used to cascade function generators for implementing wide logic functions.

BUFTs

Each Virtex-E CLB contains two 3-state drivers (BUFTs) that can drive on-chip buses. See **Dedicated Routing**. Each Virtex-E BUFT has an independent 3-state control pin and an independent input pin.

Block SelectRAM

Virtex-E FPGAs incorporate large block SelectRAM memories. These complement the Distributed SelectRAM memories that provide shallow RAM structures implemented in CLBs.

Block SelectRAM memory blocks are organized in columns, starting at the left (column 0) and right outside edges and inserted every 12 CLB columns (see notes for smaller devices). Each memory block is four CLBs high, and each memory column extends the full height of the chip, immediately adjacent (to the right, except for column 0) of the CLB column locations indicated in Table 3.

Table 3: **CLB/Block RAM Column Locations**

XCV Device /Col.	0	12	24	36	48	60	72	84	96	108	120	138	156
50E	Columns 0, 6, 18, & 24												
100E	Columns 0, 12, 18, & 30												
200E	Columns 0, 12, 30, & 42												
300E	√	√		√	√								
400E	√	√			√	√							
600E	√	√	√		√	√	√						
1000E	√	√	√				√	√	√				
1600E	√	√	√	√			√	√	√	√			
2000E	√	√	√	√				√	√	√	√		
2600E	√	√	√	√					√	√	√	√	
3200E	√	√	√	√						√	√	√	√

Table 4 shows the amount of block SelectRAM memory that is available in each Virtex-E device.

Table 4: **Virtex-E Block SelectRAM Amounts**

Virtex-E Device	# of Blocks	Block SelectRAM Bits
XCV50E	16	65,536
XCV100E	20	81,920
XCV200E	28	114,688
XCV300E	32	131,072
XCV400E	40	163,840
XCV600E	72	294,912
XCV1000E	96	393,216
XCV1600E	144	589,824
XCV2000E	160	655,360
XCV2600E	184	753,664
XCV3200E	208	851,968

As illustrated in Figure 6, each block SelectRAM cell is a fully synchronous dual-ported (True Dual Port) 4096-bit RAM with independent control signals for each port. The data widths of the two ports can be configured independently, providing built-in bus-width conversion.

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.

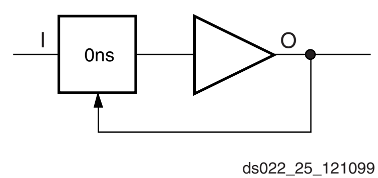

Figure 21: **Simplified DLL Macro Symbol BUFGDLL**

Table 21: **Guidelines for Max Number of Simultaneously Switching Outputs per Power/Ground Pair *(Continued)***

Standard	Package		
	BGA, CS, FGA	HQ	PQ, TQ
HSTL Class I	18	13	9
HSTL Class III	9	7	5
HSTL Class IV	5	4	3
SSTL2 Class I	15	11	8
SSTL2 Class II	10	7	5
SSTL3 Class I	11	8	6
SSTL3 Class II	7	5	4
CTT	14	10	7
AGP	9	7	5

Note: This analysis assumes a 35 pF load for each output.

Table 22: **Virtex-E Equivalent Power/Ground Pairs**

Pkg/Part	XCV100E	XCV200E	XCV300E	XCV400E	XCV600E	XCV1000E	XCV1600E	XCV2000E
CS144	12	12						
PQ240	20	20	20	20				
HQ240					20	20		
BG352	20	32	32					
BG432			32	40	40			
BG560				40	40	56	58	60
FG256(1)	20	24	24					
FG456		40	40					
FG676				54	56			
FG680(2)					46	56	56	56
FG860						58	60	64
FG900					56	58		60
FG1156						96	104	120

Notes:

1. Virtex-E devices in FG256 packages have more V_{CCO} than Virtex series devices.
2. FG680 numbers are preliminary.

Low Voltage Differential Signals

The Virtex-E family incorporates low-voltage signalling (LVDS and LVPECL). Two pins are utilized for these signals to be connected to a Virtex-E device. These are known as differential pin pairs. Each differential pin pair has a Positive (P) and a Negative (N) pin. These pairs are labeled in the following manner.

IO_L#[P/N]

where

L = LVDS or LVPECL pin
# = Pin Pair Number
P = Positive
N = Negative

I/O pins for differential signals can either be synchronous or asynchronous, input or output. The pin pairs can be used for synchronous input and output signals as well as asynchronous input signals. However, only some of the low-voltage pairs can be used for asynchronous output signals.

DIfferential signals require the pins of a pair to switch almost simultaneously. If the signals driving the pins are from IOB flip-flops, they are synchronous. If the signals driving the pins are from internal logic, they are asynchronous. Table 2 defines the names and function of the different types of low-voltage pin pairs in the Virtex-E family.

Table 2: **LVDS Pin Pairs**

Pin Name	Description
IO_L#[P/N] Example: IO_L22N	Represents a general IO or a synchronous input/output differential signal. When used as a differential signal, N means Negative I/O and P means Positive I/O.
IO_L#[P/N]_Y Example: IO_L22N_Y	Represents a general IO or a synchronous input/output differential signal, or a part-dependent asynchronous output differential signal.
IO_L#[P/N]_YY Example: O_L22N_YY	Represents a general IO or a synchronous input/output differential signal, or an asynchronous output differential signal.
IO_LVDS_DLL_L#[P/N] Example: IO_LVDS_DLL_L16N	Represents a general IO or a synchronous input/output differential signal, a differential clock input signal, or a DLL input. When used as a differential clock input, this pin is paired with the adjacent GCK pin. The GCK pin is always the positive input in the differential clock input configuration.

Virtex-E Package Pinouts

The Virtex-E family of FPGAs is available in 12 popular packages, including chip-scale, plastic and high heat-dissipation quad flat packs, and ball grid and fine-pitch ball grid arrays. Family members have footprint compatibility across devices provided in the same package. The pinout tables in this section indicate function, pin, and bank information for each package/device combination. Following each pinout table is an additional table summarizing information specific to differential pin pairs for all devices provided in that package.

Table 16: **FG256 Package — XCV50E, XCV100E, XCV200E, XCV300E**

Bank	Pin Description	Pin #
4	IO_L43P_Y	P12
4	IO_VREF_L43N_Y	R13[2]
4	IO_L44P_YY	N12
4	IO_L44N_YY	T13
4	IO_VREF_L45P_YY	T12
4	IO_L45N_YY	P11
4	IO_L46P_Y	R12
4	IO_L46N_Y	N11
4	IO_VREF_L47P_YY	T11[1]
4	IO_L47N_YY	M11
4	IO_L48P_YY	R11
4	IO_L48N_YY	T10
4	IO_L49P_Y	R10
4	IO_L49N_Y	M10
4	IO_VREF_L50P_Y	P9
4	IO_L50N_Y	T9
4	IO_L51P_Y	N10
4	IO_L51N_Y	R9
4	IO_LVDS_DLL_L52P	N9
5	GCK1	R8
5	IO	N7
5	IO	T7
5	IO_LVDS_DLL_L52N	T8
5	IO_L53P_Y	R7
5	IO_VREF_L53N_Y	P8
5	IO_L54P_Y	P7
5	IO_L54N_Y	T6
5	IO_L55P_YY	M7
5	IO_L55N_YY	R6
5	IO_L56P_YY	P6
5	IO_VREF_L56N_YY	R5[1]
5	IO_L57P_Y	N6
5	IO_L57N_Y	T5
5	IO_L58P_YY	M6

Table 16: **FG256 Package — XCV50E, XCV100E, XCV200E, XCV300E**

Bank	Pin Description	Pin #
5	IO_VREF_L58N_YY	T4
5	IO_L59P_YY	T3
5	IO_L59N_YY	P5
5	IO_VREF_L60P_Y	T2[2]
5	IO_L60N_Y	N5
6	IO_L61N_YY	M3
6	IO_L61P_YY	R1
6	IO_L62N	M4
6	IO_VREF_L62P	N2[2]
6	IO_L63N_YY	L5
6	IO_L63P_YY	P1
6	IO_VREF_L64N_Y	N1
6	IO_L64P_Y	L3
6	IO_L65N	M2
6	IO_L65P	L4
6	IO_VREF_L66N_Y	M1[1]
6	IO_L66P_Y	K4
6	IO_L67N_YY	L2
6	IO_L67P_YY	L1
6	IO_L68N	K3
6	IO_L68P	K1
6	IO_L69N_YY	K2
6	IO_L69P_YY	K5
6	IO_VREF_L70N_Y	J3
6	IO_L70P_Y	J1
6	IO_L71N	J4
6	IO_L71P	H1
6	IO	J2
7	IO	C2
7	IO_L72N_YY	G1
7	IO_L72P_YY	H4
7	IO_L73N	G5
7	IO_L73P	H2

Table 19: **FG456 Differential Pin Pair Summary XCV200E, XCV300E**

Pair	Bank	P Pin	N Pin	AO	Other Functions
88	5	V7	AB3	√	-
89	6	Y2	W3	√	-
90	6	V3	V4	√	-
91	6	U4	Y1	√	VREF
92	6	W1	V2	√	-
93	6	U2	T3	√	VREF
94	6	V1	T5	2	-
95	6	U1	R5	1	-
96	6	T1	R4	2	VREF
97	6	P3	R2	√	-
98	6	R1	P5	√	-
99	6	N5	P2	√	-
100	6	N4	P1	2	-
101	6	N2	N3	1	VREF
102	6	M4	N1	2	-
103	6	M6	M3	√	-
104	7	L4	L3	√	-
105	7	L1	L5	√	-
106	7	K2	L6	2	-
107	7	K3	K4	2	VREF
108	7	K5	K1	√	-
109	7	J2	J3	√	-
110	7	H1	J5	√	-
111	7	H3	H2	√	-
112	7	H4	G1	2	VREF
113	7	F2	F1	2	-
114	7	G3	H5	√	-
115	7	E2	E1	√	VREF
116	7	G5	F3	√	-
117	7	D2	E3	√	VREF
118	7	C1	F5	√	-

Notes:
1. AO in the XCV200E.
2. AO in the XCV300E.

FG676 Fine-Pitch Ball Grid Array Package

XCV400E and XCV600E devices in the FG676 fine-pitch Ball Grid Array package have footprint compatibility. Pins labeled I0_VREF can be used as either in all parts unless device-dependent as indicated in the footnotes. If the pin is not used as V_{REF}, it can be used as general I/O. Immediately following Table 20, see Table 21 for Differential Pair information.

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

Bank	Pin Description	Pin #
0	GCK3	E13
0	IO	A6
0	IO	A9[1]
0	IO	A10[1]
0	IO	B3
0	IO	B4[1]
0	IO	B12[1]
0	IO	C6
0	IO	C8
0	IO	D5
0	IO	D13[1]
0	IO	G13
0	IO_L0N_Y	C4
0	IO_L0P_Y	F7
0	IO_L1N_YY	G8
0	IO_L1P_YY	C5
0	IO_VREF_L2N_YY	D6
0	IO_L2P_YY	E7
0	IO_L3N	A4
0	IO_L3P	F8
0	IO_L4N	B5
0	IO_L4P	D7
0	IO_VREF_L5N_YY	E8
0	IO_L5P_YY	G9
0	IO_L6N_YY	A5
0	IO_L6P_YY	F9
0	IO_L7N_Y	D8
0	IO_L7P_Y	C7
0	IO_VREF_L8N_Y	B7[2]
0	IO_L8P_Y	E9

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

Bank	Pin Description	Pin #
5	IO_L129N_Y	AB9
5	IO_L130P_YY	AA9
5	IO_L130N_YY	AF6
5	IO_L131P_YY	AC8
5	IO_VREF_L131N_YY	AC7
5	IO_L132P_YY	AD6
5	IO_L132N_YY	Y9
5	IO_L133P_YY	AE5
5	IO_L133N_YY	AA8
5	IO_L134P_YY	AC6
5	IO_VREF_L134N_YY	AB8
5	IO_L135P_YY	AD5
5	IO_L135N_YY	AA7
5	IO_L136P_Y	AF4
5	IO_L136N_Y	AC5
6	IO	P3
6	IO	AA3
6	IO	AC1[1]
6	IO	P1[1]
6	IO	R2[1]
6	IO	T1[1]
6	IO	V1[1]
6	IO	W3
6	IO	Y2
6	IO	Y6
6	IO_L137N_YY	AA5
6	IO_L137P_YY	AC3
6	IO_L138N_YY	AC2
6	IO_L138P_YY	AB4
6	IO_L139N_Y	W6
6	IO_L139P_Y	AA4
6	IO_VREF_L140N_Y	AB3
6	IO_L140P_Y	Y5
6	IO_L141N_Y	AB2
6	IO_L141P_Y	V7
6	IO_L142N_YY	AB1

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

Bank	Pin Description	Pin #
6	IO_L142P_YY	Y4
6	IO_VREF_L143N_YY	V5
6	IO_L143P_YY	W5
6	IO_L144N_YY	AA1
6	IO_L144P_YY	V6
6	IO_L145N_Y	W4
6	IO_L145P_Y	Y3
6	IO_VREF_L146N_Y	Y1[2]
6	IO_L146P_Y	U7
6	IO_L147N_YY	W1
6	IO_L147P_YY	V4
6	IO_L148N_YY	W2
6	IO_VREF_L148P_YY	U6
6	IO_L149N_YY	V3
6	IO_L149P_YY	T5
6	IO_L150N_YY	U5
6	IO_L150P_YY	U4
6	IO_L151N_Y	T7
6	IO_L151P_Y	U3
6	IO_L152N_Y	U2
6	IO_L152P_Y	T6
6	IO_L153N_Y	U1
6	IO_L153P_Y	T4
6	IO_L154N_Y	R7
6	IO_L154P_Y	T3
6	IO_VREF_L155N_YY	R4
6	IO_L155P_YY	R6
6	IO_L156N_YY	R3
6	IO_L156P_YY	R5
6	IO_L157N_Y	P8
6	IO_L157P_Y	P7
6	IO_VREF_L158N_Y	R1
6	IO_L158P_Y	P6
6	IO_L159N_YY	P5
6	IO_L159P_YY	P4
7	IO	D1[1]

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

Bank	Pin Description	Pin #
NA	NC	L2
NA	NC	F6
NA	NC	F25
NA	NC	F21
NA	NC	F2
NA	NC	C26
NA	NC	C25
NA	NC	C2
NA	NC	C1
NA	NC	B6
NA	NC	B26
NA	NC	B24
NA	NC	B21
NA	NC	B16
NA	NC	B11
NA	NC	B1
NA	NC	AF25
NA	NC	AF24
NA	NC	AF2
NA	NC	AE6
NA	NC	AE3
NA	NC	AE26
NA	NC	AE24
NA	NC	AE21
NA	NC	AE16
NA	NC	AE14
NA	NC	AE11
NA	NC	AE1
NA	NC	AD25
NA	NC	AD2
NA	NC	AD1
NA	NC	AA6
NA	NC	AA25
NA	NC	AA21
NA	NC	AA2
NA	NC	A3
NA	NC	A25

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

Bank	Pin Description	Pin #
NA	NC	A2
NA	NC	A15
NA	VCCINT	G7
NA	VCCINT	G20
NA	VCCINT	H8
NA	VCCINT	H19
NA	VCCINT	J9
NA	VCCINT	J10
NA	VCCINT	J11
NA	VCCINT	J16
NA	VCCINT	J17
NA	VCCINT	J18
NA	VCCINT	K9
NA	VCCINT	K18
NA	VCCINT	L9
NA	VCCINT	L18
NA	VCCINT	T9
NA	VCCINT	T18
NA	VCCINT	U9
NA	VCCINT	U18
NA	VCCINT	V9
NA	VCCINT	V10
NA	VCCINT	V11
NA	VCCINT	V16
NA	VCCINT	V17
NA	VCCINT	V18
NA	VCCINT	Y7
NA	VCCINT	Y20
NA	VCCINT	W8
NA	VCCINT	W19
0	VCCO	J13
0	VCCO	J12
0	VCCO	H9
0	VCCO	H12
0	VCCO	H11

FG676 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 21: **FG676 Differential Pin Pair Summary XCV400E, XCV600E**

Pair	Bank	P Pin	N Pin	AO	Other Functions
Global Differential Clock					
3	0	E13	B13	NA	IO_DLL_L21N
2	1	C13	F14	NA	IO_DLL_L21P
1	5	AB13	AF13	NA	IO_DLL_L115N
0	4	AA14	AC14	NA	IO_DLL_L115P
IOLVDS Total Pairs: 183, Asynchronous Output Pairs: 97					
0	0	F7	C4	1	-
1	0	C5	G8	√	-
2	0	E7	D6	√	VREF
3	0	F8	A4	NA	-
4	0	D7	B5	NA	-
5	0	G9	E8	√	VREF
6	0	F9	A5	√	-
7	0	C7	D8	1	-
8	0	E9	B7	1	VREF
9	0	D9	A7	NA	-
10	0	G10	B8	NA	VREF
11	0	F10	C9	√	-
12	0	E10	A8	1	-
13	0	D10	G11	√	-
14	0	F11	B10	√	-
15	0	E11	C10	NA	-
16	0	D11	G12	√	-
17	0	F12	C11	√	VREF
18	0	E12	A11	√	-
19	0	C12	D12	1	-
20	0	H13	A12	1	VREF
21	1	F14	B13	NA	IO_LVDS_DLL
22	1	F13	E14	NA	-
23	1	A14	D14	1	VREF
24	1	H14	C14	1	-
25	1	C15	G14	√	-
26	1	D15	E15	√	VREF
27	1	F15	C16	√	-
28	1	D16	G15	-	-
29	1	A17	E16	√	-
30	1	E17	C17	√	-
31	1	D17	F16	1	-
32	1	C18	F17	√	-
33	1	G16	A18	√	VREF
34	1	G17	C19	√	-
35	1	B19	D18	1	VREF
36	1	E18	D19	1	-
37	1	B20	F18	√	-
38	1	C20	G19	√	VREF
39	1	E19	G18	√	-
40	1	D20	A21	√	-
41	1	C21	F19	√	VREF
42	1	E20	B22	√	-
43	1	D21	A23	2	-
44	1	E21	C22	√	CS
45	2	E23	F22	√	DIN, D0
46	2	E24	F20	√	-
47	2	G21	G22	2	-
48	2	F24	H20	1	VREF
49	2	E25	H21	1	-
50	2	F23	G23	√	-
51	2	H23	J20	√	VREF

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

Bank	Pin Description	Pin #
4	IO_VREF_L132P_YY	AV8
4	IO_L132N_YY	AU9
4	IO_L133P_Y	AW8
4	IO_L133N_Y	AT10
4	IO_VREF_L134P_Y	AV9[3]
4	IO_L134N_Y	AU10
4	IO_L135P_YY	AW9
4	IO_L135N_YY	AT11
4	IO_VREF_L136P_YY	AV10
4	IO_L136N_YY	AU11
4	IO_L137P_Y	AW10
4	IO_L137N_Y	AU12
4	IO_L138P_Y	AV11
4	IO_L138N_Y	AT13
4	IO_VREF_L139P_YY	AW11
4	IO_L139N_YY	AU13
4	IO_L140P_YY	AT14
4	IO_L140N_YY	AV12
4	IO_L141P_Y	AU14
4	IO_L141N_Y	AW12
4	IO_L142P_Y	AT15
4	IO_L142N_Y	AV13
4	IO_L143P_YY	AU15
4	IO_L143N_YY	AW13
4	IO_VREF_L144P_YY	AV14[1]
4	IO_L144N_YY	AT16
4	IO_L145P_Y	AW14
4	IO_L145N_Y	AU16
4	IO_L146P_Y	AV15
4	IO_L146N_Y	AR17
4	IO_L147P_YY	AW15
4	IO_L147N_YY	AT17
4	IO_VREF_L148P_YY	AU17
4	IO_L148N_YY	AV16
4	IO_L149P_Y	AR18
4	IO_L149N_Y	AW16
4	IO_L150P_Y	AT18
4	IO_L150N_Y	AV17
4	IO_L151P_YY	AU18
4	IO_L151N_YY	AW17
4	IO_VREF_L152P_YY	AT19
4	IO_L152N_YY	AV18
4	IO_L153P_Y	AU19
4	IO_L153N_Y	AW18
4	IO_VREF_L154P	AU21[2]
4	IO_L154N	AV19
4	IO_LVDS_DLL_L155P	AT21
5	GCK1	AU22
5	IO	AT34
5	IO	AW20
5	IO_LVDS_DLL_L155N	AT22
5	IO_VREF_L156P_Y	AV20[2]
5	IO_L156N_Y	AR22
5	IO_L157P_YY	AV23
5	IO_VREF_L157N_YY	AW21
5	IO_L158P_YY	AU23
5	IO_L158N_YY	AV21
5	IO_L159P_Y	AT23
5	IO_L159N_Y	AW22
5	IO_L160P_Y	AR23
5	IO_L160N_Y	AV22
5	IO_L161P_YY	AV24
5	IO_VREF_L161N_YY	AW23
5	IO_L162P_YY	AW24
5	IO_L162N_YY	AU24
5	IO_L163P_Y	AW25
5	IO_L163N_Y	AT24
5	IO_L164P_Y	AV25
5	IO_L164N_Y	AU25
5	IO_L165P_YY	AW26
5	IO_VREF_L165N_YY	AT25[1]

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

Bank	Pin Description	Pin #
7	IO_L234N_YY	K38
7	IO_L234P_YY	L37
7	IO_L235N_YY	J39
7	IO_VREF_L235P_YY	L36
7	IO_L236N	J38
7	IO_L236P	K37
7	IO_L237N	H39
7	IO_VREF_L237P	K36[3]
7	IO_L238N_YY	H38
7	IO_L238P_YY	J37
7	IO_L239N_YY	G39
7	IO_VREF_L239P_YY	G38
7	IO_L240N_Y	J36
7	IO_L240P_Y	F39
7	IO_L241N	H37
7	IO_L241P	F38
7	IO_L242N_YY	H36
7	IO_L242P_YY	E39
7	IO_L243N_Y	G37
7	IO_VREF_L243P_Y	E38
7	IO_L244N	G36
7	IO_L244P	D39
7	IO_L245N	D38
7	IO_VREF_L245P	F36[1]
7	IO_L246N_Y	D37
7	IO_L246P_Y	E37
2	CCLK	E4
3	DONE	AU5
NA	DXN	AV37
NA	DXP	AU35
NA	M0	AT37
NA	M1	AU38
NA	M2	AT35
NA	PROGRAM	AT5
NA	TCK	C36

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

Bank	Pin Description	Pin #
NA	TDI	B3
2	TDO	C4
NA	TMS	E36
NA	VCCINT	E8
NA	VCCINT	E9
NA	VCCINT	E15
NA	VCCINT	E16
NA	VCCINT	E24
NA	VCCINT	E25
NA	VCCINT	E31
NA	VCCINT	E32
NA	VCCINT	H5
NA	VCCINT	H35
NA	VCCINT	J5
NA	VCCINT	J35
NA	VCCINT	R5
NA	VCCINT	R35
NA	VCCINT	T5
NA	VCCINT	T35
NA	VCCINT	AD5
NA	VCCINT	AD35
NA	VCCINT	AE5
NA	VCCINT	AE35
NA	VCCINT	AL5
NA	VCCINT	AL35
NA	VCCINT	AM5
NA	VCCINT	AM35
NA	VCCINT	AR8
NA	VCCINT	AR9
NA	VCCINT	AR15
NA	VCCINT	AR16
NA	VCCINT	AR24
NA	VCCINT	AR25
NA	VCCINT	AR31
NA	VCCINT	AR32

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

Bank	Pin Description	Pin #
0	IO_VREF_L27N_YY	D27
0	IO_L27P_YY	B25
0	IO_L28N_Y	A25
0	IO_L28P_Y	D26
0	IO_L29N_Y	A24
0	IO_L29P_Y	E25
0	IO_L30N_YY	D25
0	IO_L30P_YY	B24
0	IO_VREF_L31N_YY	E24
0	IO_L31P_YY	A23
0	IO_L32N_Y	C23
0	IO_L32P_Y	E23
0	IO_VREF_L33N_Y	B23[1]
0	IO_L33P_Y	D23
0	IO_LVDS_DLL_L34N	A22
1	GCK2	B22
1	IO	A14
1	IO	A20
1	IO	B11
1	IO	B13
1	IO	C8
1	IO	C18
1	IO	C21
1	IO	D7
1	IO	D10
1	IO	D15
1	IO	D17
1	IO	E20
1	IO_LVDS_DLL_L34P	D22
1	IO_L35N_Y	D21
1	IO_VREF_L35P_Y	B21[1]
1	IO_L36N_Y	D20
1	IO_L36P_Y	A21
1	IO_L37N_YY	C20
1	IO_VREF_L37P_YY	D19
1	IO_L38N_YY	B20

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

Bank	Pin Description	Pin #
1	IO_L38P_YY	E19
1	IO_L39N_Y	D18
1	IO_L39P_Y	A19
1	IO_L40N_Y	E18
1	IO_L40P_Y	C19
1	IO_L41N_YY	B19
1	IO_VREF_L41P_YY	E17
1	IO_L42N_YY	A18
1	IO_L42P_YY	D16
1	IO_L43N_Y	E16
1	IO_L43P_Y	B18
1	IO_L44N_Y	F16
1	IO_L44P_Y	A17
1	IO_L45N_YY	C17
1	IO_VREF_L45P_YY	E15
1	IO_L46N_YY	B17
1	IO_L46P_YY	D14
1	IO_L47N_Y	A16
1	IO_L47P_Y	E14
1	IO_L48N_Y	C16
1	IO_L48P_Y	D13
1	IO_L49N_Y	B16
1	IO_L49P_Y	D12
1	IO_L50N_Y	A15
1	IO_L50P_Y	E12
1	IO_L51N_YY	C15
1	IO_L51P_YY	C11
1	IO_L52N_YY	B15
1	IO_VREF_L52P_YY	D11
1	IO_L53N_Y	E11
1	IO_L53P_Y	C14
1	IO_L54N_Y	C10
1	IO_L54P_Y	B14
1	IO_L55N_YY	A13
1	IO_VREF_L55P_YY	E10
1	IO_L56N_YY	C13
1	IO_L56P_YY	C9

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

Bank	Pin Description	Pin #
6	IO_VREF_L245N_Y	AB40[1]
6	IO_L245P_Y	AC39
7	IO	F38
7	IO	H40
7	IO	H41
7	IO	J42
7	IO	K39
7	IO	L42
7	IO	N40
7	IO	T40
7	IO	U40
7	IO	V38
7	IO	W42
7	IO	Y42
7	IO	AA42
7	IO_L246N_YY	AA41
7	IO_L246P_YY	AB39
7	IO_L247N_Y	Y41
7	IO_VREF_L247P_Y	AA39[1]
7	IO_L248N_YY	Y40
7	IO_L248P_YY	Y39
7	IO_L249N_YY	Y38
7	IO_VREF_L249P_YY	W41
7	IO_L250N_Y	W40
7	IO_L250P_Y	W39
7	IO_L251N_Y	W38
7	IO_L251P_Y	V41
7	IO_L252N_YY	V39
7	IO_L252P_YY	V40
7	IO_L253N_YY	V42
7	IO_VREF_L253P_YY	U39
7	IO_L254N_Y	U41
7	IO_L254P_Y	U38
7	IO_L255N_Y	U42
7	IO_L255P_Y	T39
7	IO_L256N_YY	T41
7	IO_L256P_YY	T38
7	IO_L257N_Y	R39
7	IO_VREF_L257P_Y	T42
7	IO_L258N_Y	R42
7	IO_L258P_Y	R38
7	IO_L259N	R40
7	IO_L259P	P39
7	IO_L260N_Y	R41
7	IO_L260P_Y	P38
7	IO_L261N_Y	P42
7	IO_L261P_Y	N39
7	IO_L262N_Y	P40
7	IO_L262P_Y	M39
7	IO_L263N_YY	P41
7	IO_L263P_YY	M38
7	IO_L264N_YY	N42
7	IO_VREF_L264P_YY	L39
7	IO_L265N_Y	L38
7	IO_L265P_Y	N41
7	IO_L266N_YY	K40
7	IO_L266P_YY	M42
7	IO_L267N_YY	M40
7	IO_VREF_L267P_YY	K38
7	IO_L268N_Y	M41
7	IO_L268P_Y	J40
7	IO_L269N_Y	J39
7	IO_VREF_L269P_Y	L40
7	IO_L270N_YY	J38
7	IO_L270P_YY	L41
7	IO_L271N_YY	K42
7	IO_VREF_L271P_YY	H39
7	IO_L272N_Y	K41
7	IO_L272P_Y	H38
7	IO_L273N_Y	J41
7	IO_L273P_Y	G40
7	IO_L274N_YY	H42
7	IO_L274P_YY	G39

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 25: **FG860 Differential Pin Pair Summary XCV1000E, XCV1600E, XCV2000E**

Pair	Bank	P Pin	N Pin	AO	Other Functions
256	7	T38	T41	√	-
257	7	T42	R39	1	VREF
258	7	R38	R42	2	-
259	7	P39	R40	4	-
260	7	P38	R41	2	-
261	7	N39	P42	1	-
262	7	M39	P40	3	-
263	7	M38	P41	√	-
264	7	L39	N42	√	VREF
265	7	N41	L38	2	-
266	7	M42	K40	√	-
267	7	K38	M40	√	VREF
268	7	J40	M41	2	-
269	7	L40	J39	5	VREF
270	7	L41	J38	√	-
271	7	H39	K42	√	VREF
272	7	H38	K41	1	-
273	7	G40	J41	2	-
274	7	G39	H42	√	-
275	7	G42	G38	1	VREF
276	7	F40	G41	2	-
277	7	F41	F42	4	-
278	7	E42	F39	2	VREF
279	7	E41	E40	1	-
280	7	D41	E39	3	-

Notes:
1. AO in the XCV1000E, 2000E.
2. AO in the XCV1000E, 1600E.
3. AO in the XCV2000E.
4. AO in the XCV1600E.
5. AO in the XCV1000E.

FG900 Fine-Pitch Ball Grid Array Package

XCV600E, XCV1000E, and XCV1600E devices in the FG900 fine-pitch Ball Grid Array package have footprint compatibility. Pins labeled I0_VREF can be used as either in all parts unless device-dependent as indicated in the footnotes. If the pin is not used as V_{REF}, it can be used as general I/O. Immediately following Table 26, see Table 27 for Differential Pair information.

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

Bank	Pin Description	Pin #
0	GCK3	C15
0	IO	A7[4]
0	IO	A13[4]
0	IO	C5[4]
0	IO	C6[4]
0	IO	C14[4]
0	IO	D8[5]
0	IO	D10
0	IO	D13[4]
0	IO	E6
0	IO	E9[5]
0	IO	E14[5]
0	IO	F9[4]
0	IO	F14[5]
0	IO	G15
0	IO	K11[5]
0	IO	K12
0	IO	L13[4]
0	IO_L0N_YY	C4[4]
0	IO_L0P_YY	F7[3]
0	IO_L1N_Y	D5
0	IO_L1P_Y	G8
0	IO_VREF_L2N_Y	A3[1]
0	IO_L2P_Y	H9
0	IO_L3N_Y	B4[4]
0	IO_L3P_Y	J10[4]
0	IO_L4N_YY	A4
0	IO_L4P_YY	D6
0	IO_VREF_L5N_YY	E7
0	IO_L5P_YY	B5

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

Bank	Pin Description	Pin #
4	IO_L154N	AG23
4	IO_L155P_YY	AF22
4	IO_L155N_YY	AE22
4	IO_VREF_L156P_YY	AJ22
4	IO_L156N_YY	AG22
4	IO_L157P	AK24[4]
4	IO_L157N	AD20[3]
4	IO_L158P_YY	AA19
4	IO_L158N_YY	AF21
4	IO_L159P	AH22[4]
4	IO_VREF_L159N	AA18
4	IO_L160P	AG21
4	IO_L160N	AK23
4	IO_L161P_YY	AH21[4]
4	IO_L161N_YY	AD19[4]
4	IO_L162P	AE20
4	IO_L162N	AJ21
4	IO_L163P	AG20
4	IO_L163N	AF20
4	IO_L164P	AC18[4]
4	IO_L164N	AF19[4]
4	IO_L165P_YY	AJ20
4	IO_L165N_YY	AE19
4	IO_VREF_L166P_YY	AK22[1]
4	IO_L166N_YY	AH20
4	IO_L167P	AG19
4	IO_L167N	AB17
4	IO_L168P	AJ19
4	IO_L168N	AD17
4	IO_L169P_YY	AA16
4	IO_L169N_YY	AA17
4	IO_VREF_L170P_YY	AK21
4	IO_L170N_YY	AB16
4	IO_L171P	AG18
4	IO_L171N	AK20
4	IO_L172P	AK19
4	IO_L172N	AD16

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

Bank	Pin Description	Pin #
4	IO_L173P_YY	AE16
4	IO_L173N_YY	AE17
4	IO_VREF_L174P_YY	AG17
4	IO_L174N_YY	AJ17
4	IO_L175P	AD15[4]
4	IO_L175N	AH17[3]
4	IO_VREF_L176P_YY	AG16[2]
4	IO_L176N_YY	AK17
4	IO_LVDS_DLL_L177P	AF16
5	GCK1	AK16
5	IO	AA11[4]
5	IO	AA14[4]
5	IO	AD14[4]
5	IO	AE7[5]
5	IO	AE8[5]
5	IO	AE10[4]
5	IO	AF6[4]
5	IO	AF10[4]
5	IO	AG9[4]
5	IO	AG12[4]
5	IO	AG14[5]
5	IO	AH8[4]
5	IO	AK6[5]
5	IO	AK14[5]
5	IO	AJ13[4]
5	IO	AJ15[4]
5	IO_LVDS_DLL_L177N	AH16
5	IO_L178P_YY	AC15[4]
5	IO_VREF_L178N_YY	AG15[2,3]
5	IO_L179P_YY	AB15
5	IO_L179N_YY	AF15
5	IO_L180P_YY	AA15
5	IO_VREF_L180N_YY	AF14
5	IO_L181P_YY	AH15
5	IO_L181N_YY	AK15
5	IO_L182P	AB14

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

Bank	Pin Description	Pin #
0	IO_L40P_Y	A17
0	IO_VREF_L41N_Y	G17[1]
0	IO_L41P_Y	B17
0	IO_LVDS_DLL_L42N	C17
1	GCK2	D17
1	IO	A18
1	IO	B18[3]
1	IO	B24
1	IO	B25
1	IO	E22[3]
1	IO	E23[3]
1	IO	D18[3]
1	IO	D19
1	IO	D25[3]
1	IO	D26[3]
1	IO	D28[3]
1	IO	D29[3]
1	IO	G23[3]
1	IO	J23[3]
1	IO_LVDS_DLL_L42P	J18
1	IO_L43N_Y	G18
1	IO_VREF_L43P_Y	C18[1]
1	IO_L44N_Y	H18
1	IO_L44P_Y	F18
1	IO_L45N_YY	B19
1	IO_VREF_L45P_YY	A19
1	IO_L46N_YY	K19
1	IO_L46P_YY	C19
1	IO_L47N	F19[5]
1	IO_L47P	E19[4]
1	IO_L48N_Y	G19
1	IO_L48P_Y	J19
1	IO_L49N_Y	A20
1	IO_L49P_Y	G20
1	IO_L50N	B20[5]
1	IO_L50P	F20[4]
1	IO_L51N_YY	D20
1	IO_VREF_L51P_YY	E20
1	IO_L52N_YY	H20
1	IO_L52P_YY	A21
1	IO_L53N	E21[5]
1	IO_L53P	J20[4]
1	IO_L54N_Y	D21
1	IO_L54P_Y	K20
1	IO_L55N_Y	B21
1	IO_L55P_Y	H21
1	IO_L56N_YY	G21[5]
1	IO_L56P_YY	F21[4]
1	IO_L57N_YY	A22
1	IO_VREF_L57P_YY	B22
1	IO_L58N_YY	J21
1	IO_L58P_YY	C22
1	IO_L59N_Y	D22
1	IO_L59P_Y	G22
1	IO_L60N_Y	K21
1	IO_L60P_Y	A23
1	IO_L61N_Y	F22
1	IO_L61P_Y	B23
1	IO_L62N_Y	C23
1	IO_L62P_Y	H22
1	IO_L63N_YY	D23
1	IO_L63P_YY	K22
1	IO_L64N_YY	A24
1	IO_VREF_L64P_YY	J22
1	IO_L65N_Y	H23
1	IO_L65P_Y	D24
1	IO_L66N_Y	A25

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

Bank	Pin Description	Pin #
5	IO_L239P_Y	AP9
5	IO_L239N_Y	AK11
5	IO_L240P_YY	AL11
5	IO_VREF_L240N_YY	AL10
5	IO_L241P_YY	AE13
5	IO_L241N_YY	AM9
5	IO_L242P	AF12[5]
5	IO_L242N	AP8[4]
5	IO_L243P_Y	AL9
5	IO_VREF_L243N_Y	AH11[2]
5	IO_L244P_Y	AF11
5	IO_L244N_Y	AN8
5	IO_L245P_Y	AM8[5]
5	IO_L245N_Y	AG11[4]
5	IO_L246P_YY	AL8
5	IO_VREF_L246N_YY	AK9
5	IO_L247P_YY	AH10
5	IO_L247N_YY	AN7
5	IO_L248P	AE12[5]
5	IO_L248N	AJ9[4]
5	IO_L249P_Y	AM7
5	IO_L249N_Y	AL7
5	IO_L250P_Y	AG10
5	IO_L250N_Y	AN6
5	IO_L251P_YY	AK8[5]
5	IO_L251N_YY	AH9[4]
5	IO_L252P_YY	AP5
5	IO_VREF_L252N_YY	AJ8
5	IO_L253P_YY	AE11
5	IO_L253N_YY	AN5
5	IO_L254P_Y	AF10
5	IO_L254N_Y	AM6
5	IO_L255P_Y	AL6
5	IO_VREF_L255N_Y	AG9

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

Bank	Pin Description	Pin #
5	IO_L256P_Y	AH8
5	IO_L256N_Y	AP4
5	IO_L257P_Y	AN4
5	IO_L257N_Y	AJ7
5	IO_L258P_YY	AM5
5	IO_L258N_YY	AK6
6	IO	T1
6	IO	V2
6	IO	V3
6	IO	V5[3]
6	IO	V8[3]
6	IO	AA10[3]
6	IO	AB5[3]
6	IO	AB7[3]
6	IO	AB9[3]
6	IO	AD7[3]
6	IO	AD8[3]
6	IO	AE2
6	IO	AE4
6	IO	AJ4[3]
6	IO	AH5[3]
6	IO_L259N_YY	AH6
6	IO_L259P_YY	AF8
6	IO_L260N_Y	AE9
6	IO_L260P_Y	AK3
6	IO_L261N_Y	AD10
6	IO_L261P_Y	AL2
6	IO_VREF_L262N_Y	AL1
6	IO_L262P_Y	AH4
6	IO_L263N	AG6
6	IO_L263P	AK1
6	IO_L264N_Y	AF7
6	IO_L264P_Y	AK2