



Welcome to [E-XFL.COM](https://www.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	1536
Number of Logic Elements/Cells	6912
Total RAM Bits	65536
Number of I/O	260
Number of Gates	322970
Voltage - Supply	2.375V ~ 2.625V
Mounting Type	Surface Mount
Operating Temperature	0°C ~ 85°C (TJ)
Package / Case	352-LBGA Exposed Pad, Metal
Supplier Device Package	352-MBGA (35x35)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/xilinx/xcv300-4bg352c">https://www.e-xfl.com/product-detail/xilinx/xcv300-4bg352c</a>

## Virtex Architecture

Virtex devices feature a flexible, regular architecture that comprises an array of configurable logic blocks (CLBs) surrounded by programmable input/output blocks (IOBs), all interconnected by a rich hierarchy of fast, versatile routing resources. The abundance of routing resources permits the Virtex family to accommodate even the largest and most complex designs.

Virtex FPGAs are SRAM-based, and are customized by loading configuration data into internal memory cells. In some modes, the FPGA reads its own configuration data from an external PROM (master serial mode). Otherwise, the configuration data is written into the FPGA (SelectMAP™, slave serial, and JTAG modes).

The standard Xilinx Foundation™ and Alliance Series™ Development systems deliver complete design support for Virtex, covering every aspect from behavioral and schematic entry, through simulation, automatic design translation and implementation, to the creation, downloading, and readback of a configuration bit stream.

## Higher Performance

Virtex devices provide better performance than previous generations of FPGA. Designs can achieve synchronous system clock rates up to 200 MHz including I/O. Virtex inputs and outputs comply fully with PCI specifications, and interfaces can be implemented that operate at 33 MHz or 66 MHz. Additionally, Virtex supports the hot-swapping requirements of Compact PCI.

Xilinx thoroughly benchmarked the Virtex family. While performance is design-dependent, many designs operated internally at speeds in excess of 100 MHz and can achieve 200 MHz. Table 2 shows performance data for representative circuits, using worst-case timing parameters.

**Table 2: Performance for Common Circuit Functions**

Function	Bits	Virtex -6
Register-to-Register		
Adder	16	5.0 ns
	64	7.2 ns
Pipelined Multiplier	8 x 8	5.1 ns
	16 x 16	6.0 ns
Address Decoder	16	4.4 ns
	64	6.4 ns
16:1 Multiplexer		5.4 ns
Parity Tree	9	4.1 ns
	18	5.0 ns
	36	6.9 ns
Chip-to-Chip		
HSTL Class IV		200 MHz
LVTTTL, 16mA, fast slew		180 MHz

### Virtex Device/Package Combinations and Maximum I/O

Table 3: Virtex Family Maximum User I/O by Device/Package (Excluding Dedicated Clock Pins)

Package	XCV50	XCV100	XCV150	XCV200	XCV300	XCV400	XCV600	XCV800	XCV1000
CS144	94	94							
TQ144	98	98							
PQ240	166	166	166	166	166				
HQ240						166	166	166	
BG256	180	180	180	180					
BG352			260	260	260				
BG432					316	316	316	316	
BG560						404	404	404	404
FG256	176	176	176	176					
FG456			260	284	312				
FG676						404	444	444	
FG680							512	512	512

### Virtex Ordering Information

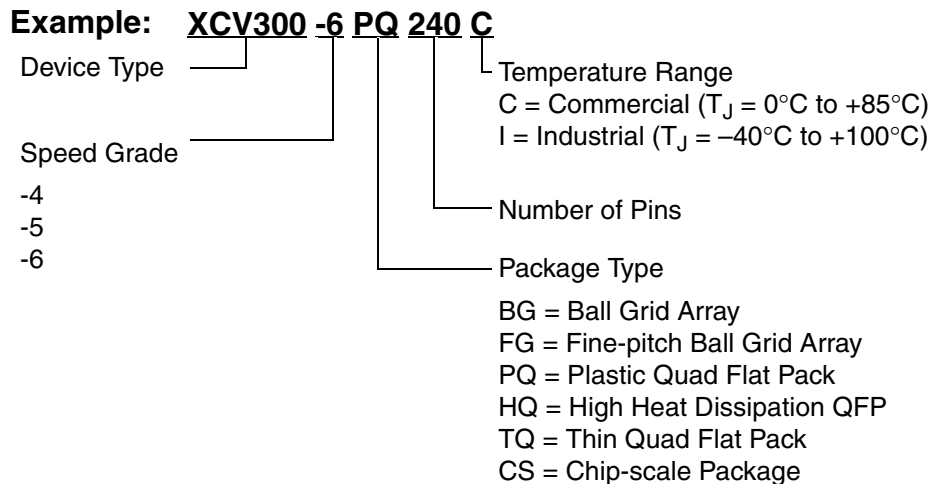


Figure 1: Virtex Ordering Information

Each block SelectRAM cell, as illustrated in Figure 6, is a fully synchronous dual-ported 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.

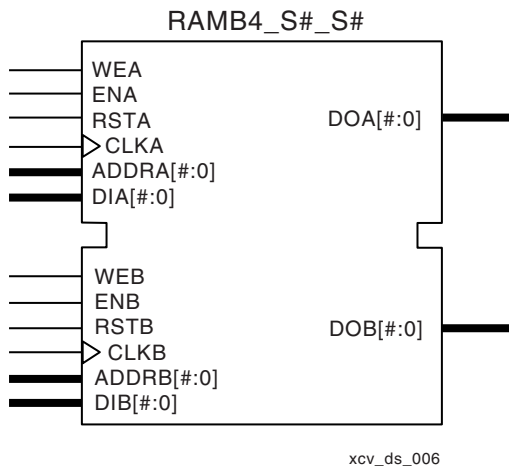


Figure 6: Dual-Port Block SelectRAM

Table 4 shows the depth and width aspect ratios for the block SelectRAM.

Table 4: 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>

The Virtex block SelectRAM also includes dedicated routing to provide an efficient interface with both CLBs and other block SelectRAMs. Refer to XAPP130 for block SelectRAM timing waveforms.

## Programmable Routing Matrix

It is the longest delay path that limits the speed of any worst-case design. Consequently, the Virtex routing architecture and its place-and-route software were defined in a single optimization process. This joint optimization minimizes long-path delays, and consequently, yields the best system performance.

The joint optimization also reduces design compilation times because the architecture is software-friendly. Design cycles are correspondingly reduced due to shorter design iteration times.

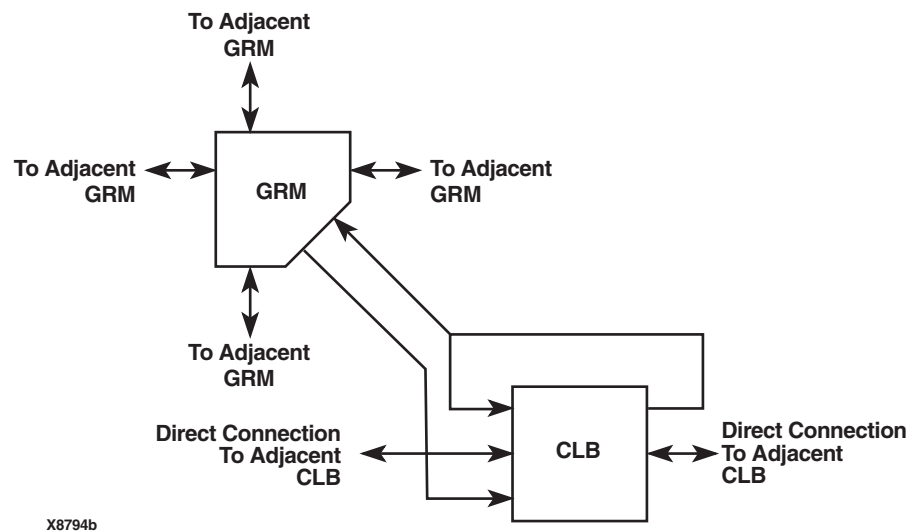


Figure 7: Virtex Local Routing

## Local Routing

The VersaBlock provides local routing resources, as shown in Figure 7, providing the following three types of connections.

- Interconnections among the LUTs, flip-flops, and GRM

- Internal CLB feedback paths that provide high-speed connections to LUTs within the same CLB, chaining them together with minimal routing delay
- Direct paths that provide high-speed connections between horizontally adjacent CLBs, eliminating the delay of the GRM.

## General Purpose Routing

Most Virtex signals are routed on the general purpose routing, and consequently, the majority of interconnect resources are associated with this level of the routing hierarchy. The general routing resources are located in horizontal and vertical routing channels associated with the rows and columns CLBs. The general-purpose routing resources are listed below.

- Adjacent to each CLB is a General Routing Matrix (GRM). The GRM is the switch matrix through which horizontal and vertical routing resources connect, and is also the means by which the CLB gains access to the general purpose routing.
- 24 single-length lines route GRM signals to adjacent GRMs in each of the four directions.
- 12 buffered Hex lines route GRM signals to another GRMs six-blocks away in each one of the four directions. Organized in a staggered pattern, Hex lines can be driven only at their endpoints. Hex-line signals can be accessed either at the endpoints or at the midpoint (three blocks from the source). One third of the Hex lines are bidirectional, while the remaining ones are uni-directional.

- 12 Longlines are buffered, bidirectional wires that distribute signals across the device quickly and efficiently. Vertical Longlines span the full height of the device, and horizontal ones span the full width of the device.

## I/O Routing

Virtex devices have additional routing resources around their periphery that form an interface between the CLB array and the IOBs. This additional routing, called the VersaRing, facilitates pin-swapping and pin-locking, such that logic redesigns can adapt to existing PCB layouts. Time-to-market is reduced, since PCBs and other system components can be manufactured while the logic design is still in progress.

## Dedicated Routing

Some classes of signal require dedicated routing resources to maximize performance. In the Virtex architecture, dedicated routing resources are provided for two classes of signal.

- Horizontal routing resources are provided for on-chip 3-state busses. Four partitionable bus lines are provided per CLB row, permitting multiple busses within a row, as shown in Figure 8.
- Two dedicated nets per CLB propagate carry signals vertically to the adjacent CLB.

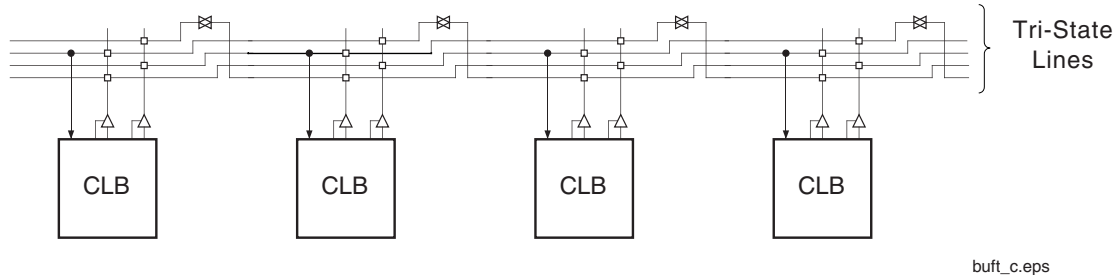


Figure 8: BUFT Connections to Dedicated Horizontal Bus Lines

## Global Routing

Global Routing resources distribute clocks and other signals with very high fanout throughout the device. Virtex devices include two tiers of global routing resources referred to as primary global and secondary local clock routing resources.

- The primary global routing resources are four dedicated global nets with dedicated input pins that are designed to distribute high-fanout clock signals with minimal skew. Each global clock net can drive all CLB, IOB, and block RAM clock pins. The primary global nets can only be driven by global buffers. There are four global buffers, one for each global net.

- The secondary local clock routing resources consist of 24 backbone lines, 12 across the top of the chip and 12 across bottom. From these lines, up to 12 unique signals per column can be distributed via the 12 longlines in the column. These secondary resources are more flexible than the primary resources since they are not restricted to routing only to clock pins.

## Clock Distribution

Virtex provides high-speed, low-skew clock distribution through the primary global routing resources described above. A typical clock distribution net is shown in Figure 9.

Four global buffers are provided, two at the top center of the device and two at the bottom center. These drive the four primary global nets that in turn drive any clock pin.

In addition to the test instructions outlined above, the boundary-scan circuitry can be used to configure the FPGA, and also to read back the configuration data.

**Figure 10** is a diagram of the Virtex Series boundary scan logic. It includes three bits of Data Register per IOB, the IEEE 1149.1 Test Access Port controller, and the Instruction Register with decodes.

### Instruction Set

The Virtex Series boundary scan instruction set also includes instructions to configure the device and read back configuration data (CFG\_IN, CFG\_OUT, and JSTART). The complete instruction set is coded as shown in **Table 5**.

### Data Registers

The primary data register is the boundary scan register. For each IOB pin in the FPGA, bonded or not, it includes three bits for In, Out, and 3-State Control. Non-IOB pins have appropriate partial bit population if input-only or output-only. Each EXTEST CAPTURED-OR state captures all In, Out, and 3-state pins.

The other standard data register is the single flip-flop BYPASS register. It synchronizes data being passed through the FPGA to the next downstream boundary scan device.

The FPGA supports up to two additional internal scan chains that can be specified using the BSCAN macro. The macro provides two user pins (SEL1 and SEL2) which are decoded of the USER1 and USER2 instructions respectively. For these instructions, two corresponding pins (TDO1 and TDO2) allow user scan data to be shifted out of TDO.

Likewise, there are individual clock pins (DRCK1 and DRCK2) for each user register. There is a common input pin (TDI) and shared output pins that represent the state of the TAP controller (RESET, SHIFT, and UPDATE).

### Bit Sequence

The order within each IOB is: In, Out, 3-State. The input-only pins contribute only the In bit to the boundary scan I/O data register, while the output-only pins contributes all three bits.

From a cavity-up view of the chip (as shown in EPIC), starting in the upper right chip corner, the boundary scan data-register bits are ordered as shown in **Figure 11**.

BSDL (Boundary Scan Description Language) files for Virtex Series devices are available on the Xilinx web site in the File Download area.

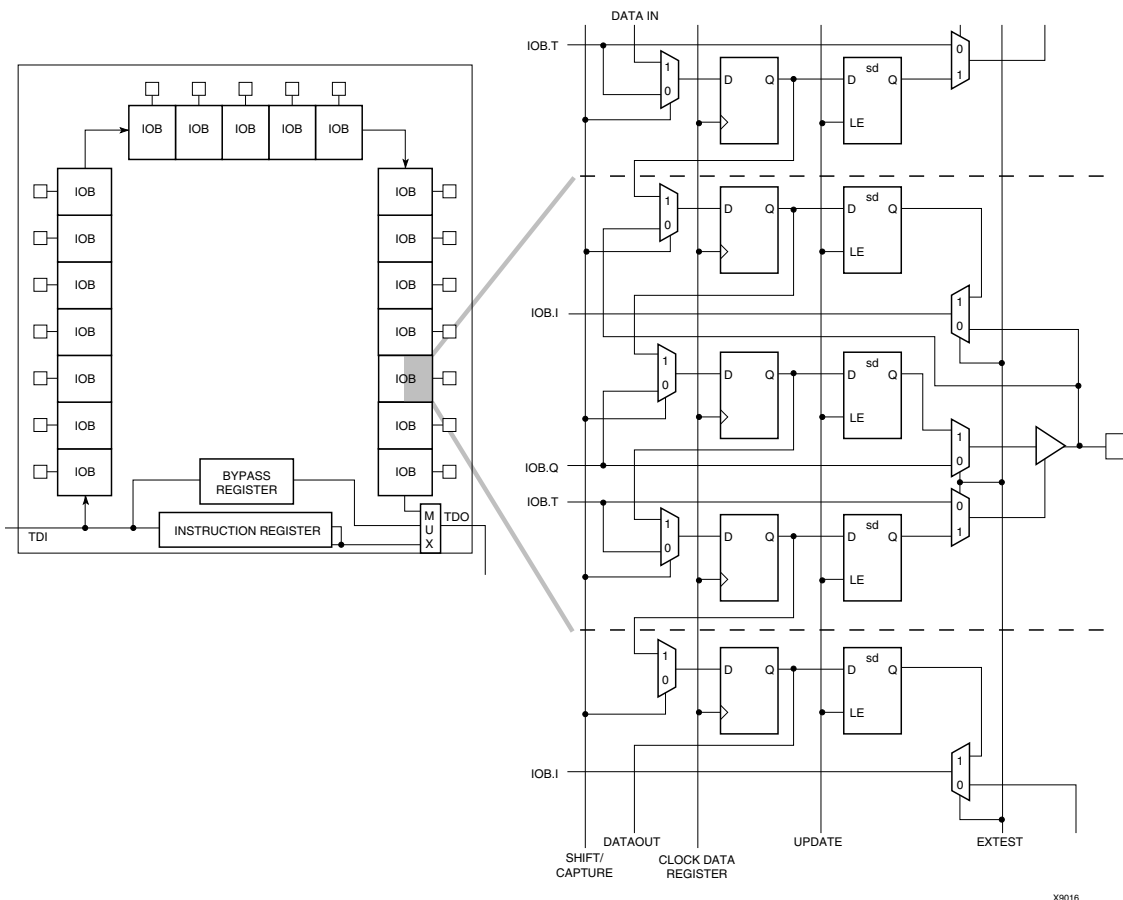


Figure 10: Virtex Series Boundary Scan Logic

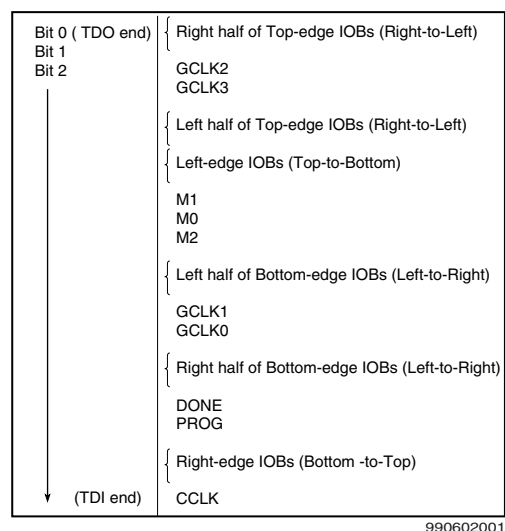


Figure 11: Boundary Scan Bit Sequence

Table 5: Boundary Scan Instructions

Boundary-Scan Command	Binary Code(4:0)	Description
EXTEST	00000	Enables boundary-scan EXTEST operation
SAMPLE/PRELOAD	00001	Enables boundary-scan SAMPLE/PRELOAD operation
USER 1	00010	Access user-defined register 1
USER 2	00011	Access user-defined register 2
CFG_OUT	00100	Access the configuration bus for read operations.
CFG_IN	00101	Access the configuration bus for write operations.
INTEST	00111	Enables boundary-scan INTEST operation
USERCODE	01000	Enables shifting out USER code
IDCODE	01001	Enables shifting out of ID Code
HIGHZ	01010	3-states output pins while enabling the Bypass Register
JSTART	01100	Clock the start-up sequence when StartupClk is TCK
BYPASS	11111	Enables BYPASS
RESERVED	All other codes	Xilinx reserved instructions

## Identification Registers

The IDCODE register is supported. By using the IDCODE, the device connected to the JTAG port can be determined.

The IDCODE register has the following binary format:

vvvv:ffff:ffa:aaaa:aaaa:cccc:cccc:ccc1

where

v = the die version number

f = the family code (03h for Virtex family)

a = the number of CLB rows (ranges from 010h for XCV50 to 040h for XCV1000)

c = the company code (49h for Xilinx)

The USERCODE register is supported. By using the USERCODE, a user-programmable identification code can be loaded and shifted out for examination. The identification code is embedded in the bitstream during bitstream generation and is valid only after configuration.

Table 6: IDCODEs Assigned to Virtex FPGAs

FPGA	IDCODE
XCV50	v0610093h
XCV100	v0614093h
XCV150	v0618093h
XCV200	v061C093h
XCV300	v0620093h
XCV400	v0628093h
XCV600	v0630093h
XCV800	v0638093h
XCV1000	v0640093h

## Including Boundary Scan in a Design

Since the boundary scan pins are dedicated, no special element needs to be added to the design unless an internal data register (USER1 or USER2) is desired.

If an internal data register is used, insert the boundary scan symbol and connect the necessary pins as appropriate.

## Development System

Virtex FPGAs are supported by the Xilinx Foundation and Alliance CAE tools. The basic methodology for Virtex 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 design-



ers 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 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 FPGAs 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 TRACE® static timing analyzer.

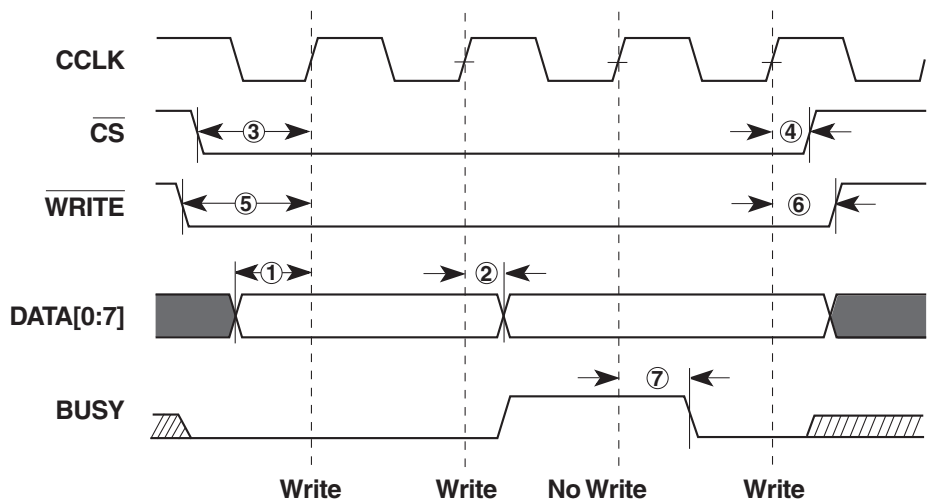
For in-circuit debugging, the development system includes a download and readback cable. This cable connects the FPGA in the target system to a PC or workstation. After downloading the design into the FPGA, the designer can single-step the logic, readback the contents of the flip-flops, and so observe the internal logic state. Simple modifications can be downloaded into the system in a matter of minutes.



3. At the rising edge of CCLK: If BUSY is Low, the data is accepted on this clock. If BUSY is High (from a previous write), the data is not accepted. Acceptance will instead occur on the first clock after BUSY goes Low, and the data must be held until this has happened.
4. Repeat steps 2 and 3 until all the data has been sent.

5. De-assert  $\overline{\text{CS}}$  and  $\overline{\text{WRITE}}$ .

A flowchart for the write operation appears in [Figure 17](#). Note that if CCLK is slower than  $f_{\text{CCNH}}$ , the FPGA never asserts BUSY. In this case, the above handshake is unnecessary, and data can simply be entered into the FPGA every CCLK cycle.



ds003\_16\_071902

Figure 16: Write Operations

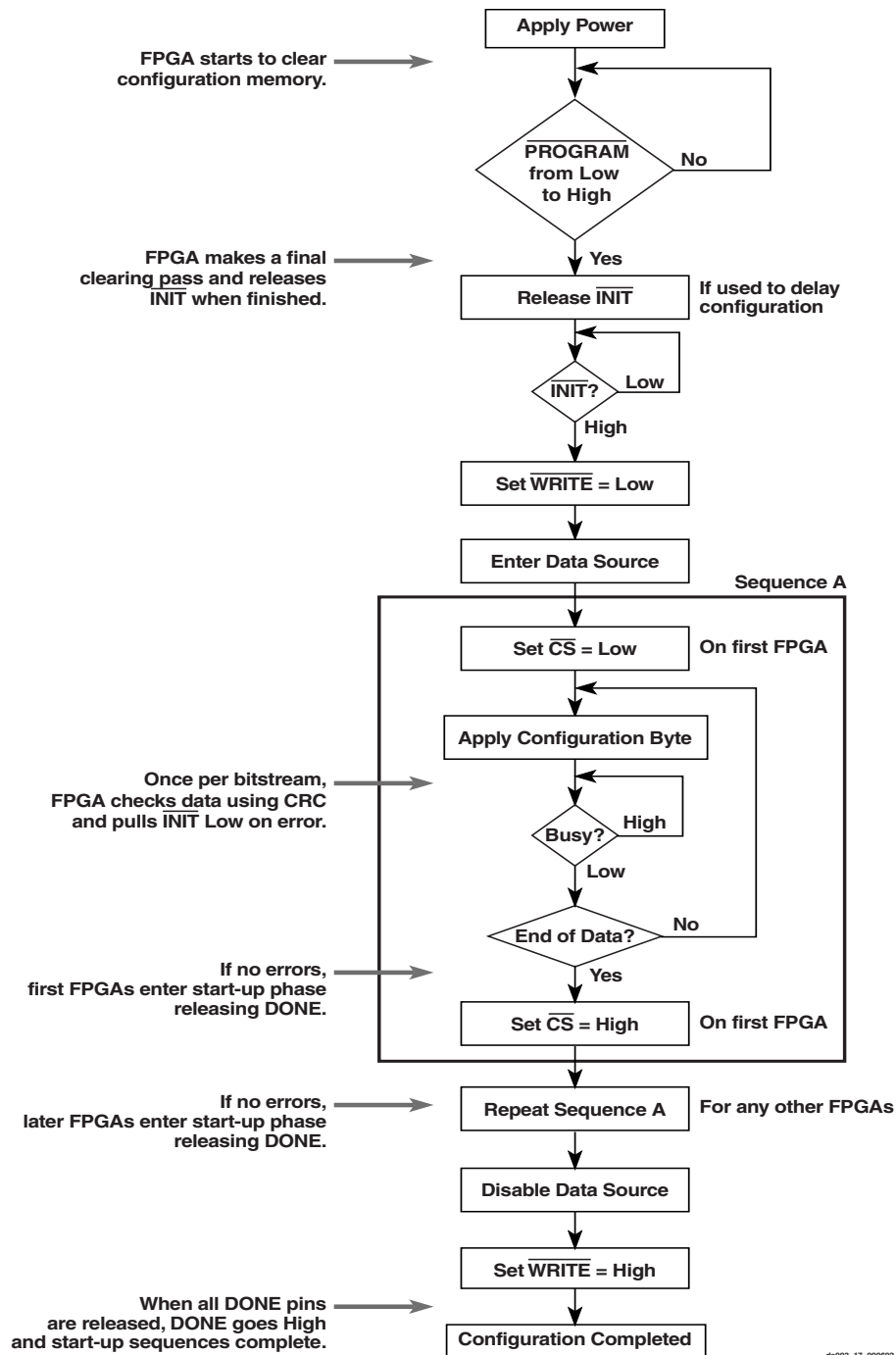


Figure 17: SelectMAP Flowchart for Write Operation

**Abort**

During a given assertion of  $\overline{CS}$ , the user cannot switch from a write to a read, or vice-versa. This action causes the current packet command to be aborted. The device will remain BUSY until the aborted operation has completed. Following an abort, data is assumed to be unaligned to word boundar-

ies, and the FPGA requires a new synchronization word prior to accepting any new packets.

To initiate an abort during a write operation, de-assert  $\overline{WRITE}$ . At the rising edge of CCLK, an abort is initiated, as shown in Figure 18.

## CLB Arithmetic Switching Characteristics

Setup times not listed explicitly can be approximated by decreasing the combinatorial delays by the setup time adjustment listed. Precise values are provided by the timing analyzer.

Description	Symbol	Speed Grade				Units
		Min	-6	-5	-4	
Combinatorial Delays						
F operand inputs to X via XOR	T <sub>OPX</sub>	0.37	0.8	0.9	1.0	ns, max
F operand input to XB output	T <sub>OPXB</sub>	0.54	1.1	1.3	1.4	ns, max
F operand input to Y via XOR	T <sub>OPY</sub>	0.8	1.5	1.7	2.0	ns, max
F operand input to YB output	T <sub>OPYB</sub>	0.8	1.5	1.7	2.0	ns, max
F operand input to COUT output	T <sub>OPCYF</sub>	0.6	1.2	1.3	1.5	ns, max
G operand inputs to Y via XOR	T <sub>OPGY</sub>	0.46	1.0	1.1	1.2	ns, max
G operand input to YB output	T <sub>OPGYB</sub>	0.8	1.6	1.8	2.1	ns, max
G operand input to COUT output	T <sub>OPCYG</sub>	0.7	1.3	1.4	1.6	ns, max
BX initialization input to COUT	T <sub>BXCY</sub>	0.41	0.9	1.0	1.1	ns, max
CIN input to X output via XOR	T <sub>CINX</sub>	0.21	0.41	0.46	0.53	ns, max
CIN input to XB	T <sub>CINXB</sub>	0.02	0.04	0.05	0.06	ns, max
CIN input to Y via XOR	T <sub>CINY</sub>	0.23	0.46	0.52	0.6	ns, max
CIN input to YB	T <sub>CINYB</sub>	0.23	0.45	0.51	0.6	ns, max
CIN input to COUT output	T <sub>BYP</sub>	0.05	0.09	0.10	0.11	ns, max
Multiplier Operation						
F1/2 operand inputs to XB output via AND	T <sub>FANDXB</sub>	0.18	0.36	0.40	0.46	ns, max
F1/2 operand inputs to YB output via AND	T <sub>FANDYB</sub>	0.40	0.8	0.9	1.1	ns, max
F1/2 operand inputs to COUT output via AND	T <sub>FANDCY</sub>	0.22	0.43	0.48	0.6	ns, max
G1/2 operand inputs to YB output via AND	T <sub>GANDYB</sub>	0.25	0.50	0.6	0.7	ns, max
G1/2 operand inputs to COUT output via AND	T <sub>GANDCY</sub>	0.07	0.13	0.15	0.17	ns, max
Setup and Hold Times before/after Clock CLK <sup>(1)</sup>	Setup Time / Hold Time					
CIN input to FFX	T <sub>CCKX</sub> /T <sub>CKCX</sub>	0.50 / 0	1.0 / 0	1.2 / 0	1.3 / 0	ns, min
CIN input to FFY	T <sub>CCKY</sub> /T <sub>CKCY</sub>	0.53 / 0	1.1 / 0	1.2 / 0	1.4 / 0	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.

## Block RAM Switching Characteristics

Description	Symbol	Speed Grade				Units
		Min	-6	-5	-4	
Sequential Delays						
Clock CLK to DOUT output	T <sub>BCKO</sub>	1.7	3.4	3.8	4.3	ns, max
Setup and Hold Times before/after Clock CLK <sup>(1)</sup>	Setup Time / Hold Time					
ADDR inputs	T <sub>BACK</sub> /T <sub>BCKA</sub>	0.6 / 0	1.2 / 0	1.3 / 0	1.5 / 0	ns, min
DIN inputs	T <sub>BDCK</sub> /T <sub>BCKD</sub>	0.6 / 0	1.2 / 0	1.3 / 0	1.5 / 0	ns, min
EN input	T <sub>BECK</sub> /T <sub>BCKE</sub>	1.3 / 0	2.6 / 0	3.0 / 0	3.4 / 0	ns, min
RST input	T <sub>BRCK</sub> /T <sub>BCKR</sub>	1.3 / 0	2.5 / 0	2.7 / 0	3.2 / 0	ns, min
WEN input	T <sub>BWCK</sub> /T <sub>BCKW</sub>	1.2 / 0	2.3 / 0	2.6 / 0	3.0 / 0	ns, min
Clock CLK						
Minimum Pulse Width, High	T <sub>BPWH</sub>	0.8	1.5	1.7	2.0	ns, min
Minimum Pulse Width, Low	T <sub>BPWL</sub>	0.8	1.5	1.7	2.0	ns, min
CLKA -> CLKB setup time for different ports	T <sub>BCCS</sub>		3.0	3.5	4.0	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.

## TBUF Switching Characteristics

Description	Symbol	Speed Grade				Units
		Min	-6	-5	-4	
Combinatorial Delays						
IN input to OUT output	T <sub>IO</sub>	0	0	0	0	ns, max
TRI input to OUT output high-impedance	T <sub>OFF</sub>	0.05	0.09	0.10	0.11	ns, max
TRI input to valid data on OUT output	T <sub>ON</sub>	0.05	0.09	0.10	0.11	ns, max

## JTAG Test Access Port Switching Characteristics

Description	Symbol	Speed Grade			Units
		-6	-5	-4	
TMS and TDI Setup times before TCK	$T_{TAPTCK}$	4.0	4.0	4.0	ns, min
TMS and TDI Hold times after TCK	$T_{TCKTAP}$	2.0	2.0	2.0	ns, min
Output delay from clock TCK to output TDO	$T_{TCKTDO}$	11.0	11.0	11.0	ns, max
Maximum TCK clock frequency	$F_{TCK}$	33	33	33	MHz, max

## Global Clock Set-Up and Hold for LVTTL Standard, *without* DLL

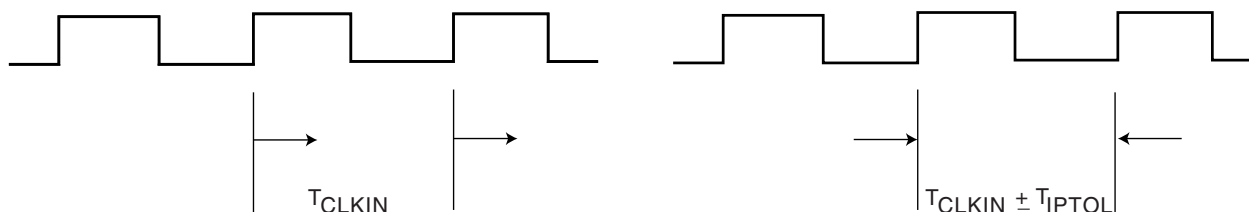
Description	Symbol	Device	Speed Grade				Units
			Min	-6	-5	-4	
Input Setup and Hold Time Relative to Global Clock Input Signal for LVTTL Standard. <sup>(2)</sup> For data input with different standards, adjust the setup time delay by the values shown in Input Delay Adjustments.							
Full Delay Global Clock and IFF, without DLL	T <sub>PSFD</sub> /T <sub>PHFD</sub>	XCV50	0.6 / 0	2.3 / 0	2.6 / 0	2.9 / 0	ns, min
		XCV100	0.6 / 0	2.3 / 0	2.6 / 0	3.0 / 0	ns, min
		XCV150	0.6 / 0	2.4 / 0	2.7 / 0	3.1 / 0	ns, min
		XCV200	0.7 / 0	2.5 / 0	2.8 / 0	3.2 / 0	ns, min
		XCV300	0.7 / 0	2.5 / 0	2.8 / 0	3.2 / 0	ns, min
		XCV400	0.7 / 0	2.6 / 0	2.9 / 0	3.3 / 0	ns, min
		XCV600	0.7 / 0	2.6 / 0	2.9 / 0	3.3 / 0	ns, min
		XCV800	0.7 / 0	2.7 / 0	3.1 / 0	3.5 / 0	ns, min
		XCV1000	0.7 / 0	2.8 / 0	3.1 / 0	3.6 / 0	ns, min

IFF = Input Flip-Flop or Latch

**Notes: Notes:**

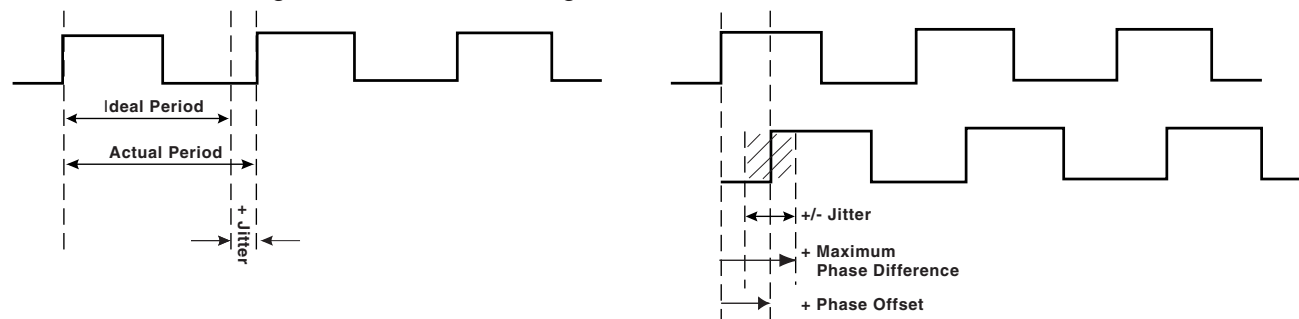
1. Set-up time is measured relative to the Global Clock input signal with the fastest route and the lightest load. Hold time is measured relative to the Global Clock input signal with the slowest route and heaviest load.
2. A Zero "0" Hold Time listing indicates no hold time or a negative hold time. Negative values can not be guaranteed "best-case", but if a "0" is listed, there is no positive hold time.

**Period Tolerance:** the allowed input clock period change in nanoseconds.



**Output Jitter:** the difference between an ideal reference clock edge and the actual design.

**Phase Offset and Maximum Phase Difference**



ds003\_20c\_110399

Figure 1: Frequency Tolerance and Clock Jitter

## Revision History

Date	Version	Revision
11/98	1.0	Initial Xilinx release.
01/99	1.2	Updated package drawings and specs.
02/99	1.3	Update of package drawings, updated specifications.
05/99	1.4	Addition of package drawings and specifications.
05/99	1.5	Replaced FG 676 & FG680 package drawings.
07/99	1.6	Changed Boundary Scan Information and changed Figure 11, Boundary Scan Bit Sequence. Updated IOB Input & Output delays. Added Capacitance info for different I/O Standards. Added 5 V tolerant information. Added DLL Parameters and waveforms and new Pin-to-pin Input and Output Parameter tables for Global Clock Input to Output and Setup and Hold. Changed Configuration Information including Figures 12, 14, 17 & 19. Added device-dependent listings for quiescent currents ICCINTQ and ICCOQ. Updated IOB Input and Output Delays based on default standard of LVTTTL, 12 mA, Fast Slew Rate. Added IOB Input Switching Characteristics Standard Adjustments.
09/99	1.7	Speed grade update to preliminary status, Power-on specification and Clock-to-Out Minimums additions, "0" hold time listing explanation, quiescent current listing update, and Figure 6 ADDRA input label correction. Added T <sub>IJITCC</sub> parameter, changed T <sub>OJIT</sub> to T <sub>OPHASE</sub> .
01/00	1.8	Update to speed.txt file 1.96. Corrections for CRs 111036, 111137, 112697, 115479, 117153, 117154, and 117612. Modified notes for Recommended Operating Conditions (voltage and temperature). Changed Bank information for V <sub>CCO</sub> in CS144 package on p.43.



Table 4: Virtex Pinout Tables (Fine-Pitch BGA)

Pin Name	Device	FG256	FG456	FG676	FG680
GCK0	All	N8	W12	AA14	AW19
GCK1	All	R8	Y11	AB13	AU22
GCK2	All	C9	A11	C13	D21
GCK3	All	B8	C11	E13	A20
M0	All	N3	AB2	AD4	AT37
M1	All	P2	U5	W7	AU38
M2	All	R3	Y4	AB6	AT35
CCLK	All	D15	B22	D24	E4
PROGRAM	All	P15	W20	AA22	AT5
DONE	All	R14	Y19	AB21	AU5
INIT	All	N15	V19	Y21	AU2
BUSY/DOUT	All	C15	C21	E23	E3
D0/DIN	All	D14	D20	F22	C2
D1	All	E16	H22	K24	P4
D2	All	F15	H20	K22	P3
D3	All	G16	K20	M22	R1
D4	All	J16	N22	R24	AD3
D5	All	M16	R21	U23	AG2
D6	All	N16	T22	V24	AH1
D7	All	N14	Y21	AB23	AR4
WRITE	All	C13	A20	C22	B4
CS	All	B13	C19	E21	D5
TDI	All	A15	B20	D22	B3
TDO	All	B14	A21	C23	C4
TMS	All	D3	D3	F5	E36
TCK	All	C4	C4	E6	C36
DXN	All	R4	Y5	AB7	AV37
DXP	All	P4	V6	Y8	AU35

Table 4: Virtex Pinout Tables (Fine-Pitch BGA) (Continued)

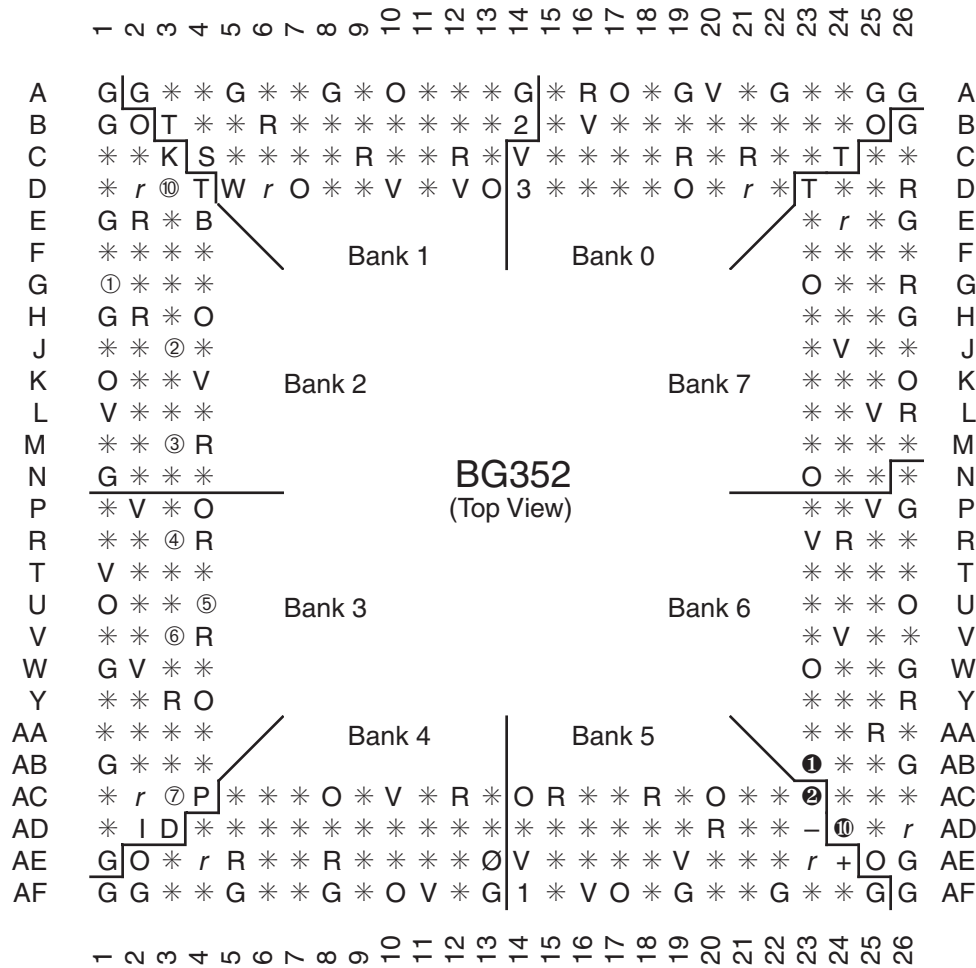
Pin Name	Device	FG256	FG456	FG676	FG680
V <sub>CCINT</sub>	All	C3, C14, D4, D13, E5, E12, M5, M12, N4, N13, P3, P14	E5, E18, F6, F17, G7, G8, G9, G14, G15, G16, H7, H16, J7, J16, P7, P16, R7, R16, T7, T8, T9, T14, T15, T16, U6, U17, V5, V18	G7, G20, H8, H19, J9, J10, J11, J16, J17, J18, K9, K18, L9, L18, T9, T18, U9, U18, V9, V10, V11, V16, V17, V18, W8, W19, Y7, Y20	AD5, AD35, AE5, AE35, AL5, AL35, AM5, AM35, AR8, AR9, AR15, AR16, AR24, AR25, AR31, AR32, E8, E9, E15, E16, E24, E25, E31, E32, H5, H35, J5, J35, R5, R35, T5, T35
V <sub>CCO</sub> , Bank 0	All	E8, F8	F7, F8, F9, F10, G10, G11	H9, H10, H11, H12, J12, J13	E26, E27, E29, E30, E33, E34
V <sub>CCO</sub> , Bank 1	All	E9, F9	F13, F14, F15, F16, G12, G13	H15, H16, H17, H18, J14, J15	E6, E7, E10, E11, E13, E14
V <sub>CCO</sub> , Bank 2	All	H11, H12	G17, H17, J17, K16, K17, L16	J19, K19, L19, M18, M19, N18	F5, G5, K5, L5, N5, P5
V <sub>CCO</sub> , Bank 3	All	J11, J12	M16, N16, N17, P17, R17, T17	P18, R18, R19, T19, U19, V19	AF5, AG5, AN5, AK5, AJ5, AP5
V <sub>CCO</sub> , Bank 4	All	L9, M9	T12, T13, U13, U14, U15, U16,	V14, V15, W15, W16, W17, W18	AR6, AR7, AR10, AR11, AR13, AR14
V <sub>CCO</sub> , Bank 5	All	L8, M8	T10, T11, U7, U8, U9, U10	V12, V13, W9, W10, W11, W12	AR26, AR27, AR29, AR30, AR33, AR34
V <sub>CCO</sub> , Bank 6	All	J5, J6	M7, N6, N7, P6, R6, T6	P9, R8, R9, T8, U8, V8	AF35, AG35, AJ35, AK35, AN35, AP35
V <sub>CCO</sub> , Bank 7	All	H5, H6	G6, H6, J6, K6, K7, L7	J8, K8, L8, M8, M9, N9	F35, G35, K35, L35, N35, P35
V <sub>REF</sub> , Bank 0 (VREF pins are listed incrementally. Connect all pins listed for both the required device and all smaller devices listed in the same package.) Within each bank, if input reference voltage is not required, all V <sub>REF</sub> pins are general I/O.	XCV50	B4, B7	N/A	N/A	N/A
	XCV100/150	... + C6	A9, C6, E8	N/A	N/A
	XCV200/300	... + A3	... + B4	N/A	N/A
	XCV400	N/A	N/A	A12, C11, D6, E8, G10	
	XCV600	N/A	N/A	... + B7	A33, B28, B30, C23, C24, D33
	XCV800	N/A	N/A	... + B10	... + A26
	XCV1000	N/A	N/A	N/A	... + D34

Table 4: Virtex Pinout Tables (Fine-Pitch BGA) (Continued)

Pin Name	Device	FG256	FG456	FG676	FG680
<b>V<sub>REF</sub> Bank 1</b> (V <sub>REF</sub> pins are listed incrementally. Connect all pins listed for both the required device and all smaller devices listed in the same package.) Within each bank, if input reference voltage is not required, all V <sub>REF</sub> pins are general I/O.	XCV50	B9, C11	N/A	N/A	N/A
	XCV100/150	... + E11	A18, B13, E14	N/A	N/A
	XCV200/300	... + A14	... + A19	N/A	N/A
	XCV400	N/A	N/A	A14, C20, C21, D15, G16	N/A
	XCV600	N/A	N/A	... + B19	B6, B8, B18, D11, D13, D17
	XCV800	N/A	N/A	... + A17	... + B14
	XCV1000	N/A	N/A	N/A	... + B5
<b>V<sub>REF</sub> Bank 2</b> (V <sub>REF</sub> pins are listed incrementally. Connect all pins listed for both the required device and all smaller devices listed in the same package.) Within each bank, if input reference voltage is not required, all V <sub>REF</sub> pins are general I/O.	XCV50	F13, H13	N/A	N/A	N/A
	XCV100/150	... + F14	F21, H18, K21	N/A	N/A
	XCV200/300	... + E13	... + D22	N/A	N/A
	XCV400	N/A	N/A	F24, H23, K20, M23, M26	N/A
	XCV600	N/A	N/A	... + G26	G1, H4, J1, L2, V5, W3
	XCV800	N/A	N/A	... + K25	... + N1
	XCV1000	N/A	N/A	N/A	... + D2
<b>V<sub>REF</sub> Bank 3</b> (V <sub>REF</sub> pins are listed incrementally. Connect all pins listed for both the required device and all smaller devices listed in the same package.) Within each bank, if input reference voltage is not required, all V <sub>REF</sub> pins are general I/O.	XCV50	K16, L14	N/A	N/A	N/A
	XCV100/150	... + L13	N21, R19, U21	N/A	N/A
	XCV200/300	... + M13	... + U20	N/A	N/A
	XCV400	N/A	N/A	R23, R25, U21, W22, W23	N/A
	XCV600	N/A	N/A	... + W26	AC1, AJ2, AK3, AL4, AR1, Y1
	XCV800	N/A	N/A	... + U25	... + AF3
	XCV1000	N/A	N/A	N/A	... + AP4



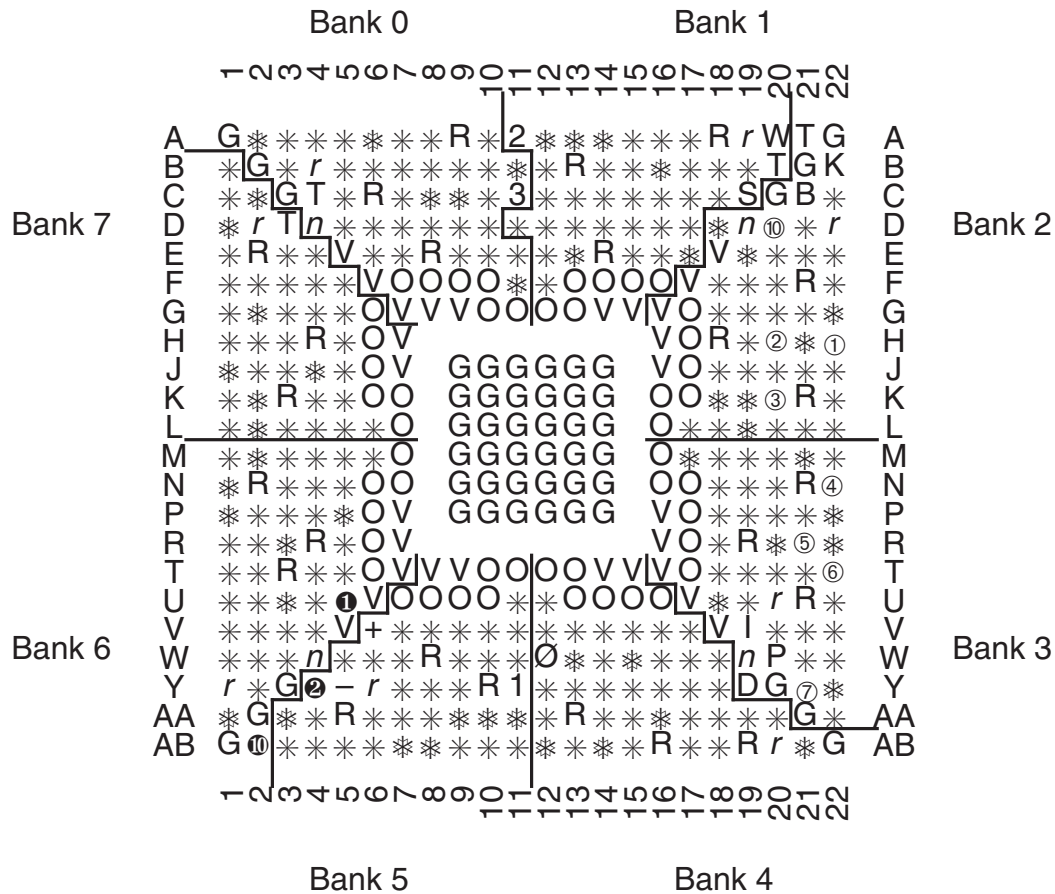
## BG352 Pin Function Diagram



DS003\_19\_100600

Figure 5: BG352 Pin Function Diagram

## FG456 Pin Function Diagram



## FG456 (Top view)

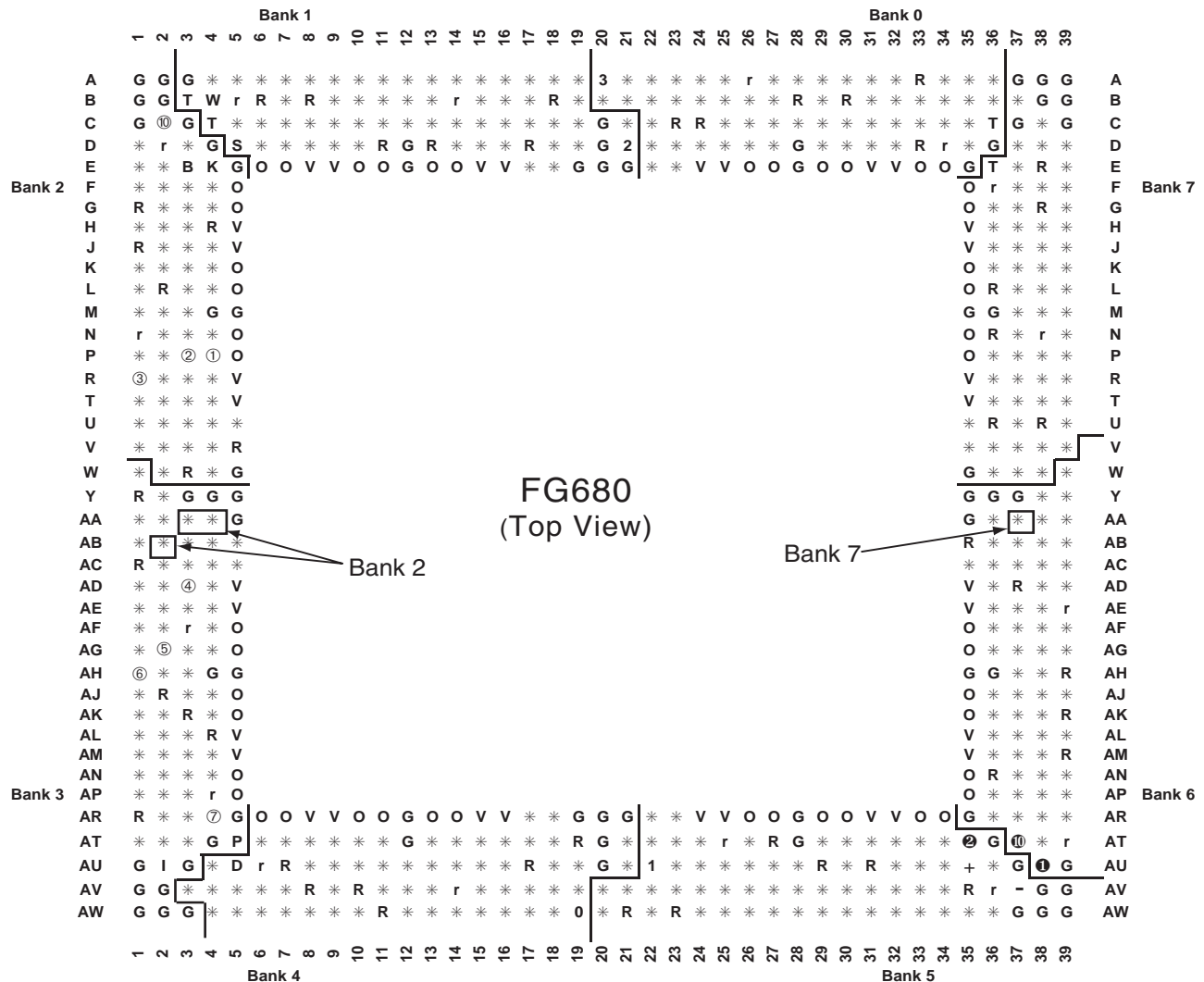
Figure 9: FG456 Pin Function Diagram

### Notes:

Packages FG456 and FG676 are layout compatible.



# FG680 Pin Function Diagram



Note: AA3, AA4, and AB2 are in Bank 2

Note: AA37 is in Bank 7

fg680\_12a

Figure 11: FG680 Pin Function Diagram