Editorial contact:
Ann Duft
Xilinx, Inc.
(408) 879-4726
publicrelations@xilinx.com

Kathy Keller
Oak Ridge Public Relations
(408) 253-5042
kathy.keller@oakridge.com

Product Marketing contact:
Bruce Jorgens
Xilinx, Inc.
(408) 879-5236
bruce.jorgens@xilinx.com

FOR IMMEDIATE RELEASE

XILINX BREAKS ONE MILLION-GATE BARRIER

WITH DELIVERY OF NEW VIRTEX SERIES

Next generation architecture couples unparalleled performance and

unprecedented system integration capabilities in a broad density range family

SAN JOSE, Calif., October 26, 1998—Dramatically expanding on the traditional uses for programmable logic, Xilinx, Inc. (NASDAQ: XLNX) today announced the first delivery of the Virtex FPGA series as a new FPGA platform to truly address system-level design issues. Starting at $10, the family encompasses a robust feature set across a full density range from 50,000 to one million system gates. The million-gate Virtex FPGAs—an industry first at this density—and a 300,000 gate device are available now.

The Virtex FPGA has built-in capabilities to solve designers' challenges throughout the system: solutions for chip-to-chip communication needs amidst multiple I/O standards, numerous clock signal synchronization inside and outside the device, and management of a variety of memory needs. No other FPGA solves these system-level problems that designers face. Moreover, these features are available at new lower industry price points, offering ASIC-competitive performance advantages at densities as low as 50,000 system gates.

Xilinx has worked with several customers worldwide, including Adtech, Hawaii; Hughes Space and Communications Company, Calif.; and NDS (News Data Services) Ltd., UK. Software support for Virtex FPGAs has been available since November 1997, allowing customers to have working systems today based on Virtex FPGAs.

"The Virtex FPGAs are not merely a collection of gates; they are intelligent devices that deliver a value beyond the socket, changing the way entire systems are designed," said Wim Roelandts, Xilinx president and CEO. "Systems architects will choose Virtex FPGAs not just because of the unmatched density, performance, and features, but because they address the breadth of their systems needs better

—more—

Using the Virtex Delay-Locked Loop

XAPP132 October 11, 1999 (Version 1.4) Application Note

Summary

The Virtex 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.

Xilinx Family

Virtex™ FPGAs (XCV series)

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 series of devices resolve this potential problem by providing up to eight fully digital dedicated on-chip Delay-Locked Loop (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 de-skew 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.

Fundamentals

Two basic types of circuits remove clock delay: Phased-Locked Loops (PLLs), and DLLs. In addition to the primary function of removing clock distribution delay, DLLs and PLLs typically provide some additional functionality such as frequency synthesis (clock multiplication and clock division) and clock conditioning (duty cycle correction and phase shifting). Multiple clock outputs can also be de-skewed with respect to one another, to take advantage of multiple clock domains.

Delay-Locked Loop

As shown in Figure 1, a DLL in its simplest form consists of a variable delay line and control logic. The delay line produces a delayed version of the input clock CLKIN. The clock distribution network routes the clock to all internal registers and to the clock feedback CLKFB pin. The control logic must sample the input clock as well as the feedback clock in order to adjust the delay line.

Delay lines can be built using a voltage controlled delay or as a series of discrete delay elements. For optimum performance the Virtex DLL uses a discrete digital delay line.

A DLL works by inserting delay between the input clock and the feedback clock until the two rising edges align, putting the two clocks 360 degrees out of phase (meaning they are in phase). After the edges from the input clock line up with the edges from the feedback clock, the DLL "locks." As long as the circuit is not evaluated until after the DLL locks, the two clocks have no discernible difference. Thus, the DLL output clock compensates for the delay in the clock distribution network, effectively removing the delay between the source clock and its loads.

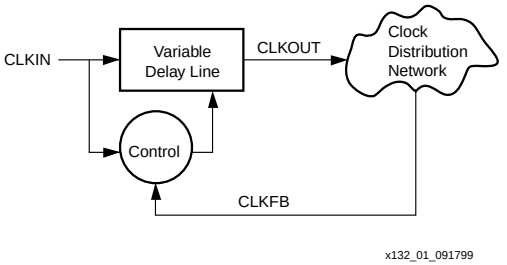


Figure 1: Delay-Locked Loop Block Diagram

Phase-Locked Loop

While designed for the same basic function, the PLL uses a different architecture to accomplish the task. As shown in Figure 2, the fundamental difference between the PLL and DLL is that instead of a delay line, the PLL uses a voltage controlled oscillator which generates a clock signal that approximates the input clock CLKIN. The control logic, consisting of a phase detector and filter, adjusts the oscillator frequency and phase to compensate for the clock distribution delay.

The PLL control logic compares the input clock to the feedback clock CLKFB and adjusts the oscillator clock until the rising edge of the input clock aligns with the feedback clock. The PLL then "locks."

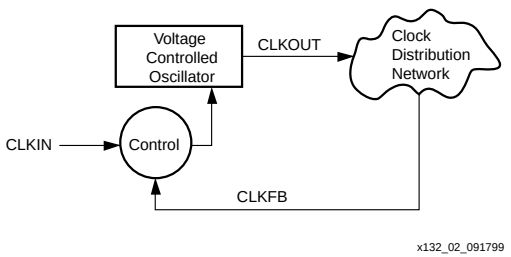


Figure 2: Phase-Locked Loop Block Diagram

Implementation

Implementation of the DLL or PLL can be accomplished using either analog or digital circuitry; each holds its own advantages. An analog implementation with careful design can produce a DLL or PLL with a finer timing resolution. Analog implementations can additionally take less silicon area.

Conversely, digital implementations offer advantages in noise sensitivity, lower power consumption and jitter performance. Digital implementations also provide the ability to stop the clock, facilitating power management. Analog implementations can require additional power supplies, require close control of the power supply, and pose problems in migration to new process technologies.

DLL vs. PLL

When it comes to choosing between a PLL or a DLL for a particular application, understand the differences in the architectures. The oscillator used in the PLL inherently introduces instability and an accumulation of phase error. This in turn degrades the performance of the PLL when attempting to compensate for the delay of the clock distribution network. Conversely, the unconditionally stable DLL architecture does not accumulate phase error.

For this reason, for delay compensation and clock conditioning, choose the DLL as the architecture. On the other hand, the PLL typically has an advantage when it comes to frequency synthesis.

Library DLL Symbols

Figure 3 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 4 and Figure 5 show the two library DLL primitives. These symbols provide access to the complete set of DLL features when implementing more complex applications.

BUFGDLL Pin Descriptions

Use the BUFGDLL macro as the simplest way to provide zero propagation delay for a high-fanout on-chip clock from an external input. This macro uses the IBUFG, CLKDLL and BUFG primitives to implement the most basic DLL application as shown in Figure 6.

This symbol does not provide access to the advanced clock domain controls or to the clock multiplication or clock division features of the DLL. This symbol also does not provide access to the RST, or LOCKED pins of the DLL. For access to these features, a designer must use the library DLL primitives described in the following sections.

Source Clock Input — I

The I pin provides the user source clock, the clock signal on which the DLL operates, to the BUFGDLL. For the BUFGDLL macro the source clock frequency must fall in the low frequency range as specified in the datasheet. The BUFGDLL requires an external signal source clock. Therefore, only an external input port can source the signal that drives the BUFGDLL I pin.

Clock Output — O

The clock output pin O represents a delay-compensated version of the source clock (I) signal. This signal, sourced by a global clock buffer BUFG symbol, takes advantage of the dedicated global clock routing resources of the device.

The output clock has a 50-50 duty cycle unless you deactivate the duty cycle correction property.

CLKDLL Primitive Pin Descriptions

The library CLKDLL primitives provide access to the complete set of DLL features needed when implementing more complex applications with the DLL.

Source Clock Input — CLKIN

The CLKIN pin provides the user source clock (the clock signal on which the DLL operates) to the DLL. The CLKIN frequency must fall in the ranges specified in the datasheet. Either a global clock buffer (BUFG) driven from another CLKDLL or one of the global clock input buffers (IBUFG) on the same edge of the device (top or bottom) must source this clock signal.

Feedback Clock Input — CLKFB

The DLL requires a reference or feedback signal to provide the delay-compensated output. Connect only the CLK0 or CLK2X DLL outputs to the feedback clock input (CLKFB) pin to provide the necessary feedback to the DLL. Either a global clock buffer (BUFG) or one of the global clock input buffers (IBUFG) on the same edge of the device (top or bottom) must source this clock signal.

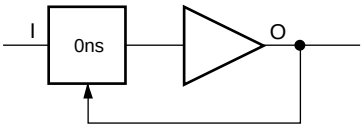
If an IBUFG sources the CLKFB pin, the following special rules apply.

1. An external input port must source the signal that drives the IBUFG I pin.
2. The CLK2X output must feedback to the device if both the CLK0 and CLK2X outputs are driving off chip devices.
3. That signal must directly drive only OBUFs and nothing else.

These rules enable the software determine which DLL clock output sources the CLKFB pin.

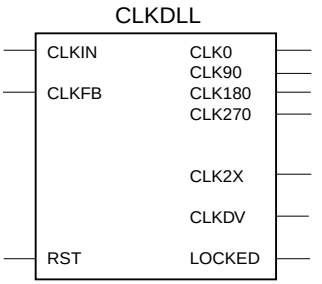
Reset Input — RST

When the reset pin RST activates the LOCKED signal deactivates within four source clock cycles. The RST pin, active High, must either connect to a dynamic signal or tied to ground. As the DLL delay taps reset to zero, glitches can occur on the DLL clock output pins. Activation of the RST pin can also severely affect the duty cycle of the clock output pins. Furthermore, the DLL output clocks no longer de-skew with respect to one another. For these reasons, rarely use the reset pin unless reconfiguring the device or changing the input frequency.



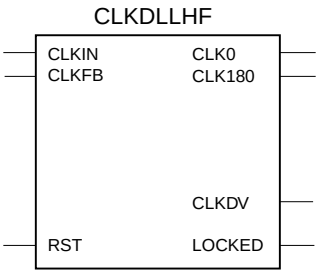
x132_03_092499

Figure 3: Simplified DLL Macro Symbol BUFGDLL



x132_04_091799

Figure 4: Standard DLL Symbol CLKDLL



x132_05_091799

Figure 5: High Frequency DLL Symbol CLKDLLHF

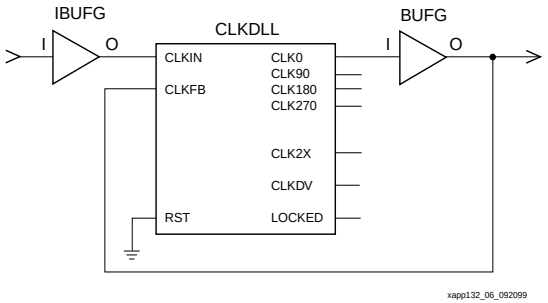


Figure 6: BUFGDLL Schematic

2x Clock Output — CLK2X

The output pin CLK2X provides a frequency-doubled clock with an automatic 50/50 duty-cycle correction. Until the CLKDLL has achieved lock, the CLK2X output appears as a 1x version of the input clock with a 25/75 duty cycle. This behavior allows the DLL to lock on the correct edge with respect to source clock. This pin is not available on the CLKDLLHF primitive.

Clock Divide Output — CLKDV

The clock divide output pin CLKDV provides a lower frequency version of the source clock. The CLKDV_DIVIDE property controls CLKDV such that the source clock is divided by N where N is either 1.5, 2, 2.5, 3, 4, 5, 8, or 16.

This feature provides automatic duty cycle correction such that the CLKDV output pin always has a 50/50 duty cycle.

1x Clock Outputs — CLK[0|90|180|270]

The 1x clock output pin CLK0 represents a delay-compensated version of the source clock (CLKIN) signal. The CLKDLL primitive provides three phase-shifted versions of the CLK0 signal while CLKDLLHF provides only the 180 phase-shifted version. The relationship between phase shift and the corresponding period shift appears in Table 1.

The timing diagrams in Figure 7 illustrate the DLL clock output characteristics.

The DLL provides duty cycle correction on all 1x clock outputs such that all 1x clock outputs by default have a 50/50 duty cycle. The DUTY_CYCLE_CORRECTION property (TRUE by default), controls this feature. In order to deactivate the DLL duty cycle correction, attach the DUTY_CYCLE_CORRECTION=FALSE property to the DLL symbol. When duty cycle correction deactivates, the output clock has the same duty cycle as the source clock.

The DLL clock outputs can drive an OBUF, a BUFG, or they can route directly to destination clock pins. The DLL clock outputs can only drive the BUFGs that reside on the same edge (top or bottom).

Table 1: Relationship of Phase-Shifted Output Clock to Period Shift

Phase (degrees)	% Period Shift
0	0%
90	25%
180	50%
270	75%

Locked Output — LOCKED

In order to achieve lock, the DLL may need to sample several thousand clock cycles. After the DLL achieves lock the LOCKED signal activates. The DLL timing parameter section of the datasheet provides estimates for locking times.

In order to guarantee that the system clock is established prior to the device “waking up,” the DLL can delay the completion of the device configuration process until after the DLL locks. The STARTUP_WAIT property activates this feature.

Until the LOCKED signal activates, the DLL output clocks are not valid and can exhibit glitches, spikes, or other spurious movement. In particular the CLK2X output will appear as a 1x clock with a 25/75 duty cycle.

DLL Properties

Properties provide access to some of the Virtex series DLL features, (for example, clock division and duty cycle correction).

Duty Cycle Correction Property

The 1x clock outputs, CLK0, CLK90, CLK180, and CLK270, use the duty cycle corrected default such that they exhibit a 50/50 duty cycle. The DUTY_CYCLE_CORRECTION property (by default TRUE) controls this feature. In order to deactivate the DLL duty cycle correction for the 1x clock outputs, attach the DUTY_CYCLE_CORRECTION=FALSE property to the DLL symbol. When duty cycle correction deactivates, the output clock has the same duty cycle as the source clock.

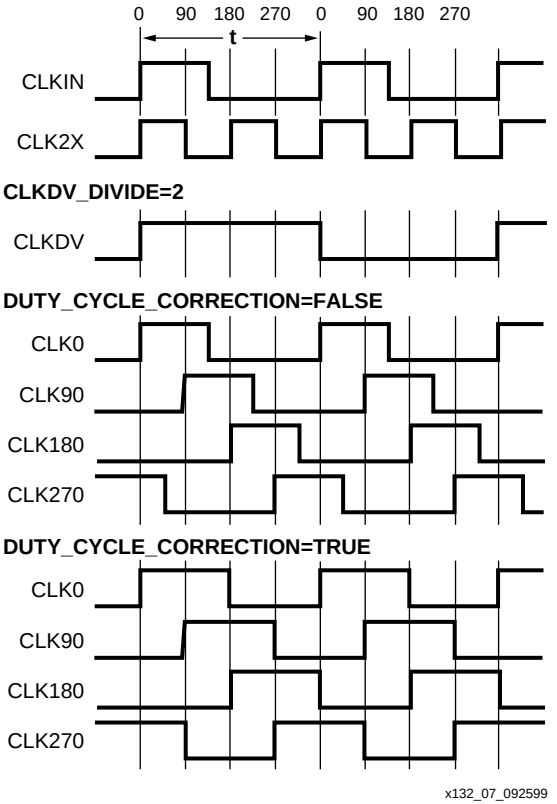


Figure 7: DLL Output Characteristics

Clock Divide Property

The CLKDV_DIVIDE property specifies how the signal on the CLKDV pin is frequency divided with respect to the CLK0 pin. The values allowed for this property are 1.5, 2, 2.5, 3, 4, 5, 8, or 16; the default value is 2.

Startup Delay Property

This property, STARTUP_WAIT, takes on a value of TRUE or FALSE (the default value). When TRUE the device configuration DONE signal waits until the DLL locks before going to High.

Virtex Series DLL Location Constraints

The DLLs are distributed such that there is one DLL in each corner of the device. The location constraint LOC, attached to the DLL symbol with the numeric identifier 0, 1, 2, or 3, controls DLL location. The orientation of the four DLLs and their corresponding clock resources appears in Figure 8.

The LOC property uses the following form.

LOC = DLL2

See Appendix A for other Virtex series DLL locations.

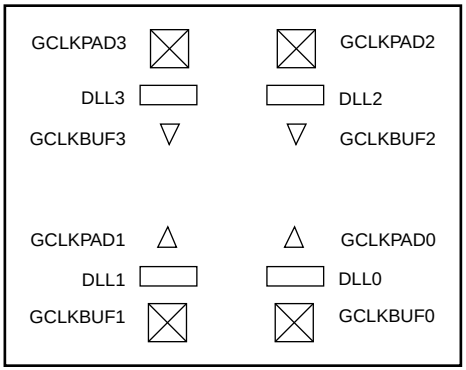


Figure 8: Orientation of Virtex Series DLLs

Design Considerations

Use the following design considerations to avoid pitfalls and improve success designing with Xilinx devices.

Input Clock

The output clock signal of a DLL, essentially a delayed version of the input clock signal, reflects any instability on the input clock in the output waveform. For this reason the quality of the DLL input clock relates directly to the quality of the output clock waveforms generated by the DLL. The DLL input clock requirements are specified in the datasheet.

In most systems a crystal oscillator generates the system clock. The DLL can be used with any commercially available quartz crystal oscillator. For example, most crystal oscillators produce an output waveform with a frequency tolerance of 100 PPM, meaning 0.01 percent change in the clock period. The DLL operates reliably on an input waveform with a frequency drift of up to 1 ns — orders of magnitude in excess of that needed to support any crystal oscillator in the industry. However, the cycle-to-cycle jitter must be kept to less than 300 ps in the low frequencies and 150 ps for the high frequencies.

Input Clock Changes

Changing the period of the input clock beyond the maximum drift amount requires a manual reset of the CLKDLL. Failure to reset the DLL will produce an unreliable lock signal and output clock.

It is possible to stop the input clock with little impact to the DLL. Stopping the clock should be limited to less than 100 µs to keep device cooling to a minimum. The clock should be stopped during a Low phase, and when restored the full High period should be seen. During this time LOCKED will stay High and remain High when the clock is restored.

When the clock is stopped, one to four more clocks will still be observed as the delay line is flushed. When the clock is restarted, the output clocks will not be observed for one to four clocks as the delay line is filled. The most common case will be two or three clocks.

In a similar manner, a phase shift of the input clock is also possible. The phase shift will propagate to the output one to four clocks after the original shift, with no disruption to the CLKDLL control.

Output Clocks

As mentioned earlier in the DLL pin descriptions, some restrictions apply regarding the connectivity of the output pins. The DLL clock outputs can drive an OBUF, a global clock buffer BUFG, or they can route directly to destination clock pins. The only BUFGs that the DLL clock outputs can drive are the two on the same edge of the device (top or bottom).

Do not use the DLL output clock signals until after activation of the LOCKED signal. Prior to the activation of the LOCKED signal, the DLL output clocks are not valid and can exhibit glitches, spikes, or other spurious movement.

Useful Application Examples

The Virtex DLL can be used in a variety of creative and useful applications. The following examples show some of the more common applications. The Verilog and VHDL example files are available at:

ftp://ftp.xilinx.com/pub/applications/xapp/xapp132.zip

Standard Usage

The circuit shown in Figure 9 resembles the BUFGDLL macro implemented in such a way as to provide access to the RST and LOCKED pins of the CLKDLL.

The dll_standard files in the xapp132.zip file show the VHDL and Verilog implementation of this circuit.

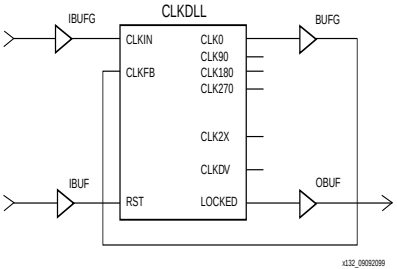


Figure 9: Standard DLL Implementation

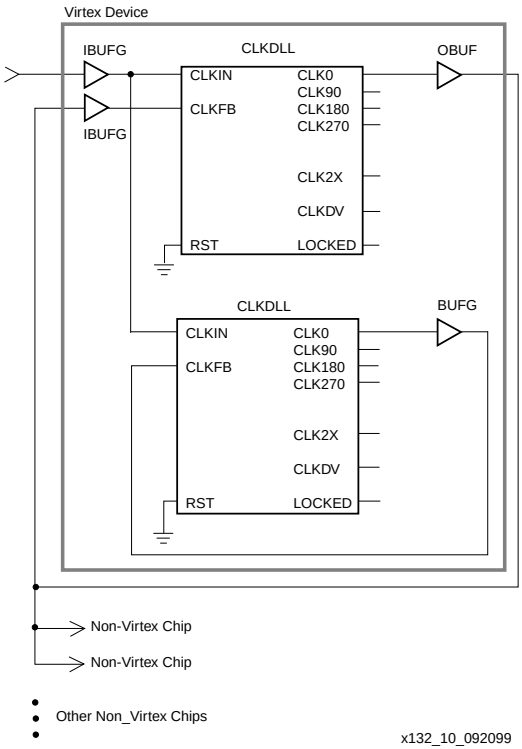


Figure 10: DLL De-skew of Board Level Clock

Board Level De-skew of Multiple Non-Virtex Devices

The circuit shown in Figure 10 can be used to de-skew a system clock between a Virtex chip and other non-Virtex chips on the same board. This application is commonly used when the Virtex device is used in conjunction with other standard products such as SRAM or DRAM devices. While designing the board level route, ensure that the return net delay to the source equals the delay to the other chips involved.

Do not use the DLL output clock signals until after activation of the LOCKED signal. Prior to the activation of the LOCKED signal, the DLL output clocks are not valid and can exhibit glitches, spikes, or other spurious movement.

The dll_mirror_1 files in the xapp132.zip file show the VHDL and Verilog implementation of this circuit.

Board Level De-skew of Multiple Virtex Devices

The circuit shown in Figure 11 can be used to de-skew a system clock between multiple Virtex chips on the same board. While designing the board level route, ensure that the return net delay to the source equals the delay to the other chips involved.

Do not use the DLL output clock signals until after activation of the LOCKED signal. Prior to the activation of the LOCKED signal, the DLL output clocks are not valid and can exhibit glitches, spikes, or other spurious movement.

The dll_mirror_2 files in the xapp132.zip file show the VHDL and Verilog implementation of this circuit.

De-skew of Clock and Its 2x Multiple

The circuit shown in Figure 12 implements a 2x clock multiplier and also uses the CLK0 clock output with zero ns skew between registers on the same chip. A clock divider circuit could alternatively be implemented using similar connections

Because any single DLL can only access at most two BUFGs, any additional output clock signals must be routed from the DLL in this example on the high speed backbone routing.

The dll_2x files in the xapp132.zip file show the VHDL and Verilog implementation of this circuit.

Generating a 4x Clock

By connecting two DLL circuits each implementing a 2x clock multiplier in series as shown in Figure 13, a 4x clock multiply can be implemented with zero ns skew between registers in the same device.

If other clock output is needed, the clock could access a BUFG only if the DLLs are constrained to exist on opposite edges (Top or Bottom) of the device.

When using this circuit it is vital to use the SRL16 cell to reset the second DLL after the initial chip reset. If this is not done, the second DLL may not recognize the change of frequencies from when the input changes from a 1x (25/75) waveform to a 2x (50/50) waveform.

The dll_4x files in the xapp132.zip file show the VHDL and Verilog implementation of this circuit.

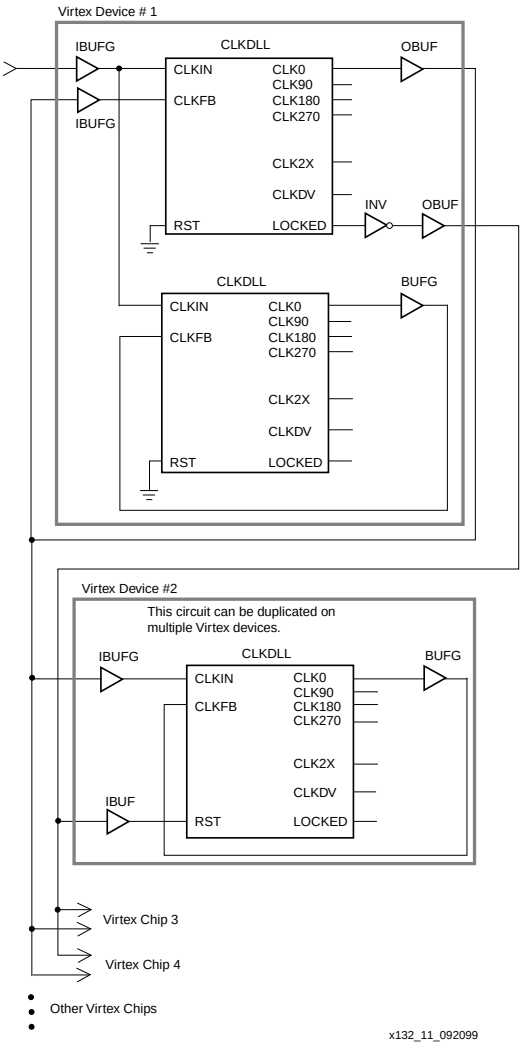


Figure 11: DLL De-skew of Board Level Clock

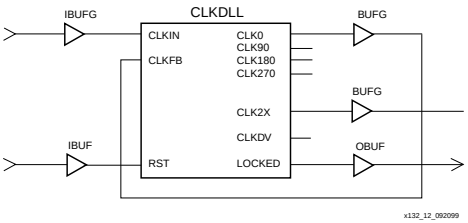


Figure 12: DLL De-skew of Clock and 2x Multiple

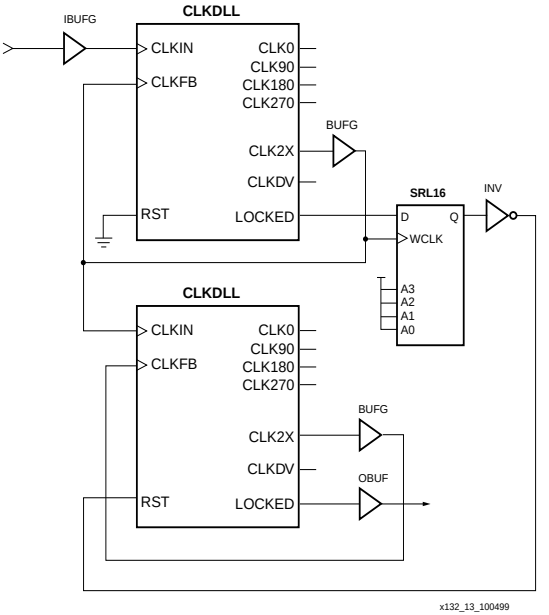


Figure 13: DLL Generation of 4x Clock

Appendix A

DLLs in Virtex-E devices

This appendix describes the DLL differences in the Virtex-E devices. The basic operations of the DLLs in the Virtex-E devices remains the same as in the Virtex devices. For example, board level de-skew of multiple non-Virtex devices, board level de-skew of multiple Virtex devices, de-skew of clock and its 2x multiple, or generating a 4x clock. The number of DLLs and their connectivity has changed in the Virtex-E devices.

Virtex-E DLL Location Constraints

As shown in the Figure 14, there are four additional DLLs in the Virtex-E devices, for a total of eight per Virtex-E device. These DLLs are located in silicon, at the top and bottom of the two innermost Block SelectRAM columns. The location constraint LOC, attached to the DLL symbol with the identifier DLL0S, DLL0P, DLL1S, DLL2S, DLL2P, DLL3S, or DLL3P controls the DLL location.

The LOC property uses the following form:

LOC = DLL0P

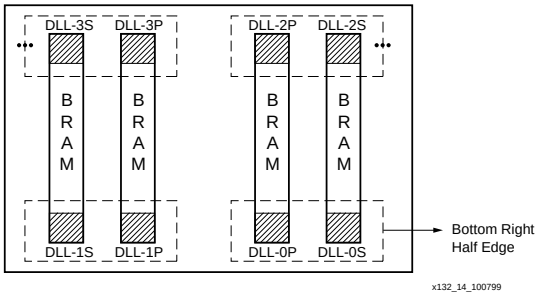


Figure 14: DLLs in Virtex-E devices

Virtex-E 4x Clock

Two DLLs located in the same half-edge (top-left, top-right, bottom-right, bottom-left) can be connected together, without using a BUFG between the CLKDLLs, to generate a 4x clock as shown in Figure 15. Virtex-E devices, like the Virtex devices, have four clock networks that are available for internal de-skewing of the clock. Each of the eight DLLs have access to two of the four clock networks. Although all the DLLs can be used for internal de-skewing, the presence of two GCLKBUFs on the top and two on the bottom indicate that only two of the four DLLs on the top (and two of the four DLLs on the bottom) can be used for this purpose.

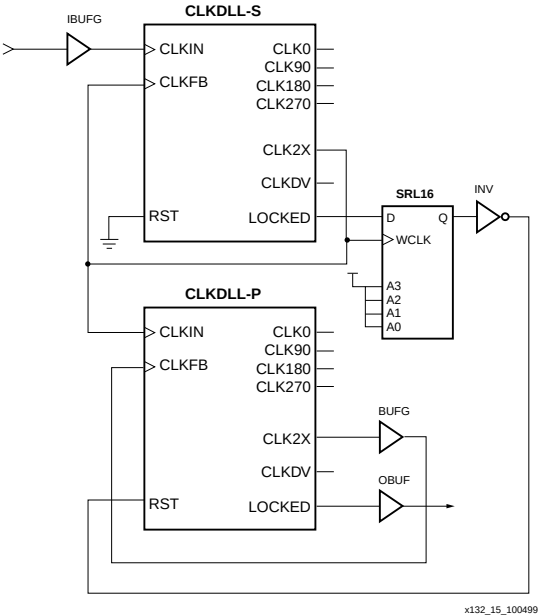


Figure 15: DLL Generation of 4x Clock in Virtex-E devices

The dll_4xe files in the XAPP132.zip file show the DLL implementation in Verilog for Virtex-E devices. These files can be found at:

ftp://ftp.xilinx.com/pub/applications/xapp/xapp132.zip

DLL Input Pins

There are four additional input pins (IO_LVDS_DLL) that can be used as inputs to the DLLs. This makes a total of eight usable inputs for DLLs in the Virtex-E family.

Revision:

Date	Version	Revision
10/21/98	1.31	Initial Release
10/11/99	1.4	Included Virtex-E devices in a new Appendix A.

FOR IMMEDIATE RELEASE

Exemplar Logic announces support for Xilinx Virtex Series FPGAs

Fremont, California – October 26, 1998 – Exemplar Logic, the world leader in FPGA synthesis today announced the immediate support for Xilinx Virtex Series FPGAs in LeonardoSpectrum.

LeonardoSpectrum is a synthesis environment designed to handle the industries largest and most complex FPGA devices such as Virtex, which now offers over 1 million system gates of logic. "Virtex will have a profound effect on present HDL design methodologies," says Ellis Smith, CEO of Exemplar Logic. "Previous devices are mostly designed by a single engineer using a simple top-down approach. Virtex's immense gate counts will result in multi-engineer FPGA design teams using more sophisticated team and bottom-up design techniques." LeonardoSpectrum fully supports these large device design methodologies by giving users the power to group, flatten, constrain, optimize and assemble on a hierarchical block by block basis. In addition to high control, Exemplar Logic's FPGA Architecture Specific Technology (F.A.S.T.) optimization technology provides tremendous throughput and quality of results easily capable of handling Virtex's' high gate counts.

Both Exemplar Logic and Xilinx are focusing on the gate array market segment. "This is clearly a device designed to attract the gate array designer into the world of FPGAs," says Tom Feist vice president of marketing for Exemplar Logic. Volume pricing for the XCV50 device matches gate array pricing up to 50,000 gates and isn't that far off at 100K gates. If these parts can handle the design they are going to be tough to beat and with up to 660 user I/Os and clock frequencies exceeding 100 MHz they can handle a lot of designs. Add to that the reduced risk associated with FPGA devices and you've got a win/win solution!"

Exemplar has experienced a sharp increase in interest in its ASIC synthesis products over the past six months. According to Rich Sevcik, vice president of software and cores at Xilinx, engineers increasingly want the option to choose between ASICs and FPGAs later in the design process which is driving the need for a synthesis environment that does both. LeonardoSpectrum is the only true seamless ASIC and FPGA synthesis tool on the market today offering the ease of use and run times FPGA designers have come to expect plus the power and control demanded by ASIC designers on both UNIX and PC platforms.

Press Contacts:

Jeff Garrison
Synplicity, Inc.
(408) 548-6031
jeff@synplicity.com

Lisa Neitzel
Tsantes & Associates
(408) 369-1500
lisa@tsantes.com

HOLD FOR RELEASE UNTIL OCTOBER 26

SYNPLICITY ADDS ENHANCED SUPPORT FOR VIRTEX; XILINX'S MILLION-GATE FPGAS

Synplify Speeds Time-to-Market For High-Density Virtex FPGA Designs

SUNNYVALE, Calif., October 26, 1998—In conjunction with the shipping of Xilinx's (Nasdaq: XLNX) Virtex FPGAs, Synplicity®, Inc., a leading supplier of logic synthesis software for programmable logic design, today announced it has enhanced its Synplify® logic synthesis tool to further support this high-density family of FPGAs. Developed specifically for Xilinx's million-gate architecture, Synplify's innovative timing-driven technology mapper allows designers to quickly solve the complex development requirements of these high-performance FPGAs. With compile times that increase linearly with design size, rather than exponentially, Virtex users can employ Synplify to obtain high quality of results in a fraction of the time of traditional FPGA synthesis tools.

"Today, million-gate FPGAs are a reality, and to successfully design these devices synthesis tools are a requirement," said Rich Sevcik, vice president of software for Xilinx. "This announcement represents another milestone in Xilinx and Synplicity's commitment to provide

(more)

The Xilinx Virtex™ Series:

Redefining FPGAs

A Product Backgrounder

Redefining the FPGA

New FPGA platform first to offer system designers powerful board-level I/O, clock, and memory functions on a chip for under $10

XILINX®

CUSTOMER TESTIMONIALS FOR VIRTEX

AdTech, Inc., Hawaii

"These million gate devices enable our test modules to provide thousands of continuous measurements at telecom speeds up to 2.4 gigabits per second. We especially like the fact that Virtex devices are programmable. This allows us to use a single test module for multiple transmission technologies such as ATM and frame relay, which provides huge cost savings and convenience for our customers plus future enhancement capability."

Carl Uyehara, vice president of Engineering

Hughes Space and Communication Company, Los Angeles, CA

"The critical features of Virtex, such as the segmented routing and 0.22 micron feature size, allowed new levels of performance for our high-speed digital designs. The lower voltage and higher performance logic of the Virtex process are unmatched from any other supplier. The level of support from the design development and the hotline assistance that Xilinx offered with the new family also impressed us."

Ted Pascaru, Senior Staff Engineer

News Data Services (NDS)

"Virtex FPGAs have allowed us to implement our next generation digital TV broadcast systems in record time. A key time saver was the availability of multiple DLLs that allowed us to synchronize a 74 MHz clock to more than 30 devices including multiple FPGAs, SDRAMs, and other components. Designing a no-skew clock system from scratch would take months. Xilinx delivered a ready-made solution to us with Virtex FPGAs."

John Simmons, Project Manager

Nortel Networks

"In our next generation networking product, we specifically needed block RAM with true dual-port capabilities. We investigated various programmable solutions available and found that no other vendor could provide a single chip solution containing block memory with the ability to read and write to one port and simultaneously read from the other. Additionally, the Virtex DLLs performance was required in order to interface directly to an ASIC running at 78 MHz. Not only do Virtex devices meet our 78 MHz internal performance requirements, they meet our external interface performance requirements by providing very fast 'clock-to-out' timing on even the slowest device."

Ranvir Chitkara, Director of Engineering

Virtex™ 2.5 V Field Programmable Gate Arrays

July 13, 1999 (Version 1.6)

Advance Product Specification

Features

- Fast, high-density Field-Programmable Gate Arrays
  - Densities from 50k to 1M system gates
  - System performance up to 200 MHz
  - 66-MHz PCI Compliant
  - Hot-swappable for Compact PCI
- Multi-standard SelectIO™ interfaces
  - 16 high-performance interface standards
  - Connects directly to ZBTRAM devices
- Built-in clock-management circuitry
  - Four dedicated delay-locked loops (DLLs) for advanced clock control
  - Four primary low-skew global clock distribution nets, plus 24 secondary local clock nets
- Hierarchical memory system
  - LUTs configurable as 16-bit RAM, 32-bit RAM, 16-bit dual-ported RAM, or 16-bit Shift Register
  - Configurable synchronous dual-ported 4k-bit RAMs
  - Fast interfaces to external high-performance RAMs
- Flexible architecture that balances speed and density
  - Dedicated carry logic for high-speed arithmetic
  - Dedicated multiplier support
  - Cascade chain for wide-input functions
  - Abundant registers/latches with clock enable, and dual synchronous/asynchronous set and reset
  - Internal 3-state bussing
  - IEEE 1149.1 boundary-scan logic
  - Die-temperature sensor diode
- Supported by FPGA Foundation™ and Alliance Development Systems
  - Complete support for Unified Libraries, Relationally Placed Macros, and Design Manager
  - Wide selection of PC and workstation platforms
- SRAM-based in-system configuration
  - Unlimited reprogrammability
  - Four programming modes
- 0.22 µm 5-layer metal process
- 100% factory tested

3

Description

The Virtex FPGA family delivers high-performance, high-capacity programmable logic solutions. Dramatic increases in silicon efficiency result from optimizing the new architecture for place-and-route efficiency and exploiting an aggressive 5-layer-metal 0.22-µm CMOS process. These advances make Virtex FPGAs powerful and flexible alternatives to mask-programmed gate arrays. The Virtex family comprises the nine members shown in Table 1.

Building on experience gained from previous generations of FPGAs, the Virtex family represents a revolutionary step forward in programmable logic design. Combining a wide variety of programmable system features, a rich hierarchy of fast, flexible interconnect resources, and advanced process technology, the Virtex family delivers a high-speed and high-capacity programmable logic solution that enhances design flexibility while reducing time-to-market.

Table 1: Virtex Field-Programmable Gate Array Family Members.

Device	System Gates	CLB Array	Logic Cells	Maximum Available I/O	BlockRAM Bits	Max Select RAM Bits
XCV50	57,906	16x24	1,728	180	32,768	24,576
XCV100	108,904	20x30	2,700	180	40,960	38,400
XCV150	164,674	24x36	3,888	260	49,152	55,296
XCV200	236,666	28x42	5,292	284	57,344	75,264
XCV300	322,970	32x48	6,912	316	65,536	98,304
XCV400	468,252	40x60	10,800	404	81,920	153,600
XCV600	661,111	48x72	15,552	512	98,304	221,184
XCV800	888,439	56x84	21,168	512	114,688	301,056
XCV1000	1,124,022	64x96	27,648	512	131,072	393,216

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

Architectural Description

Virtex Array

The Virtex user-programmable gate array, shown in Figure 1, comprises two major configurable elements: configurable logic blocks (CLBs) and input/output blocks (IOBs).

- CLBs provide the functional elements for constructing logic
- IOBs provide the interface between the package pins and the CLBs

CLBs interconnect through a general routing matrix (GRM). The GRM comprises an array of routing switches located at the intersections of horizontal and vertical routing channels. Each CLB nests into a VersaBlock™ that also provides local routing resources to connect the CLB to the GRM.

The VersaRing™ I/O interface provides additional routing resources around the periphery of the device. This routing improves I/O routability and facilitates pin locking.

The Virtex architecture also includes the following circuits that connect to the GRM.

- Dedicated block memories of 4096 bits each
- Clock DLLs for clock-distribution delay compensation and clock domain control
- 3-State buffers (BUFTs) associated with each CLB that drive dedicated segmentable horizontal routing resources

Values stored in static memory cells control the configurable logic elements and interconnect resources. These values load into the memory cells on power-up, and can reload if necessary to change the function of the device.

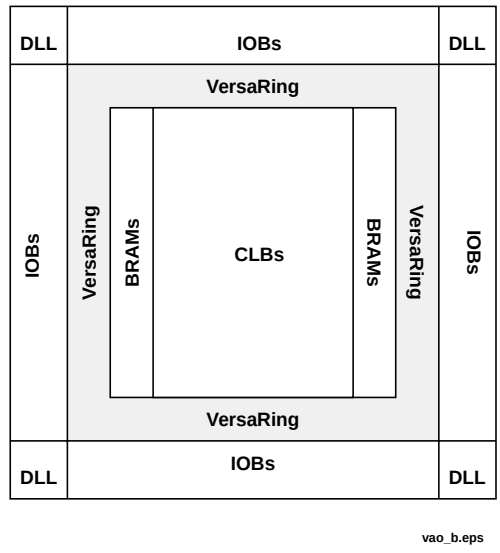


Figure 1: Virtex Architecture Overview

Input/Output Block

The Virtex IOB, Figure 2, features SelectIO™ inputs and outputs that support a wide variety of I/O signalling standards, see Table 3.

The three IOB storage elements function either as edge-triggered D-type flip-flops or as level sensitive latches. Each IOB has a clock signal (CLK) shared by the three flip-flops and independent clock enable signals for each flip-flop.

In addition to the CLK and CE control signals, the three flip-flops share a Set/Reset (SR). For each flip-flop, this signal can be independently configured as a synchronous Set, a synchronous Reset, an asynchronous Preset, or an asynchronous Clear.

The output buffer and all of the IOB control signals have independent polarity controls.

All pads are protected against damage from electrostatic discharge (ESD) and from over-voltage transients. Two forms of over-voltage protection are provided, one that permits 5 V compliance, and one that does not. For 5 V compliance, a Zener-like structure connected to ground turns on when the output rises to approximately 6.5 V. When PCI 3.3 V compliance is required, a conventional clamp diode is connected to the output supply voltage, V_{CCO}.

Optional pull-up and pull-down resistors and an optional weak-keeper circuit are attached to each pad. Prior to configuration, all pins not involved in configuration are forced into their high-impedance state. The pull-down resistors and the weak-keeper circuits are inactive, but inputs may optionally be pulled up.

The activation of pull-up resistors prior to configuration is controlled on a global basis by the configuration mode pins. If the pull-up resistors are not activated, all the pins will float. Consequently, external pull-up or pull-down resistors must be provided on pins required to be at a well-defined logic level prior to configuration.

All Virtex IOBs support IEEE 1149.1-compatible boundary scan testing.

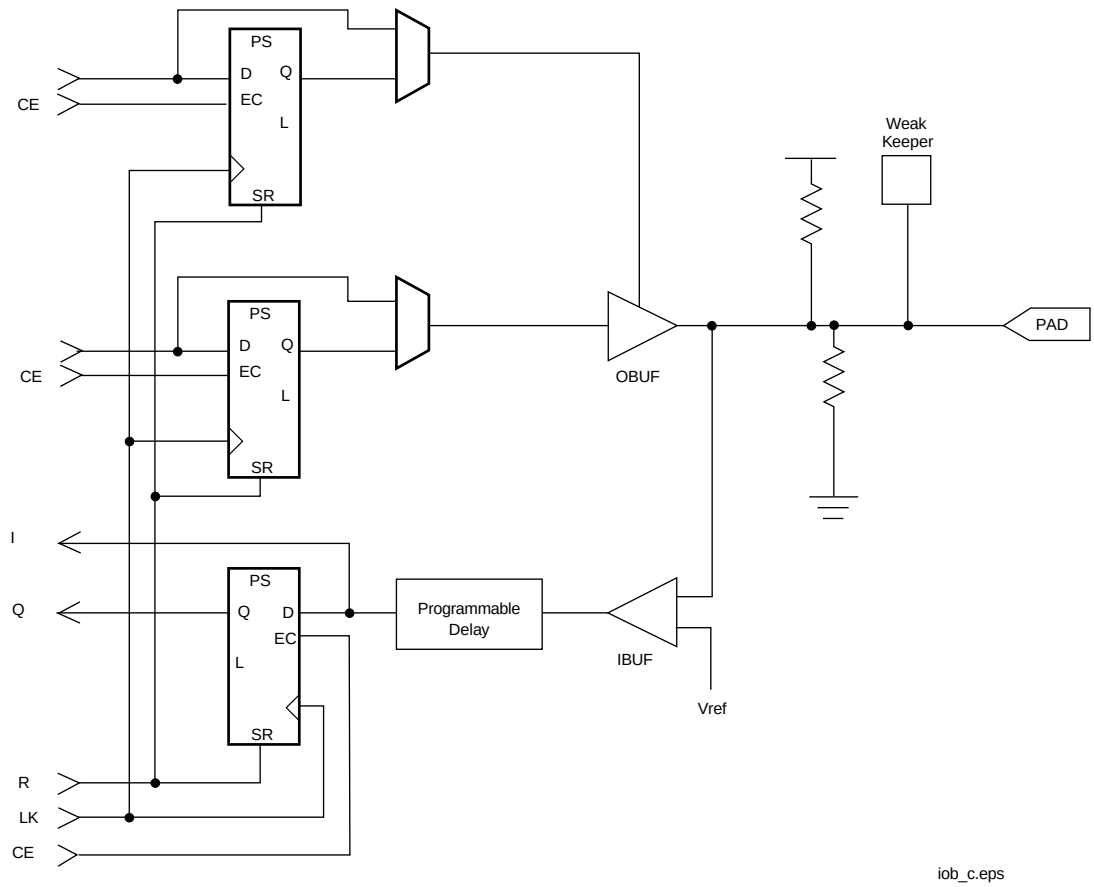


Figure 2: Virtex Input/Output Block (IOB)

Table 3: Supported Select I/O Standards

I/O Standard	Input Reference Voltage (V_{REF})	Output Source Voltage (V_{CCO})	Board Termination Voltage (V_{TT})	5 V Tolerant
LVTTL 2 – 24 mA	N/A	3.3	N/A	Yes
LVCMOS2	N/A	2.5	N/A	Yes
PCI, 5 V	N/A	3.3	N/A	Yes
PCI, 3.3 V	N/A	3.3	N/A	No
GTL	0.8	N/A	1.2	No
GTL+	1.0	N/A	1.5	No
HSTL Class I	0.75	1.5	0.75	No
HSTL Class III	0.9	1.5	1.5	No
HSTL Class IV	0.9	1.5	1.5	No
SSTL3 Class I &II	1.5	3.3	1.5	No
SSTL2 Class I & II	1.25	2.5	1.25	No
CTT	1.5	3.3	1.5	No
AGP	1.32	3.3	N/A	No

Input Path

A buffer In the Virtex IOB input path routes the input signal either directly to internal logic or through an optional input flip-flop.

An optional delay element at the D-input of this flip-flop eliminates pad-to-pad hold time. The delay is matched to the internal clock-distribution delay of the FPGA, and when used, assures that the pad-to-pad hold time is zero.

Each input buffer can be configured to conform to any of the low-voltage signalling standards supported. In some of these standards the input buffer utilizes a user-supplied threshold voltage, V_{REF}. The need to supply V_{REF} imposes constraints on which standards can used in close proximity to each other. See “I/O Banking” on page 4.

There are optional pull-up and pull-down resistors at each input for use after configuration. Their value is in the range 50 – 100 kΩ.

Output Path

The output path includes a 3-state output buffer that drives the output signal onto the pad. The output signal can be routed to the buffer directly from the internal logic or through an optional IOB output flip-flop.

The 3-state control of the output can also be routed directly from the internal logic or through a flip-flip that provides synchronous enable and disable.

Each output driver can be individually programmed for a wide range of low-voltage signalling standards. Each output buffer can source up to 24 mA and sink up to 48mA. Drive strength and slew rate controls minimize bus transients.

In most signalling standards, the output High voltage depends on an externally supplied V_{CCO} voltage. The need to supply V_{CCO} imposes constraints on which standards can be used in close proximity to each other. See “I/O Banking” on page 4.

An optional weak-keeper circuit is connected to each output. When selected, the circuit monitors the voltage on the pad and weakly drives the pin High or Low to match the input signal. If the pin is connected to a multiple-source signal, the weak keeper holds the signal in its last state if all drivers are disabled. Maintaining a valid logic level in this way eliminates bus chatter.

Because the weak-keeper circuit uses the IOB input buffer to monitor the input level, an appropriate V_{REF} voltage must be provided if the signalling standard requires one. The provision of this voltage must comply with the I/O banking rules.

I/O Banking

Some of the I/O standards described above require V_{CCO} and/or V_{REF} voltages. These voltages externally and connected to device pins that serve groups of IOBs, called banks. Consequently, restrictions exist about which I/O standards can be combined within a given bank.

Eight I/O banks result from separating each edge of the FPGA into two banks, as shown in Figure 3. Each bank has multiple V_{CCO} pins, all of which must be connected to the same voltage. This voltage is determined by the output standards in use.

Within a bank, output standards may be mixed only if they use the same V_{CCO}. Compatible standards are shown in Table 4. GTL and GTL+ appear under all voltages because their open-drain outputs do not depend on V_{CCO}.

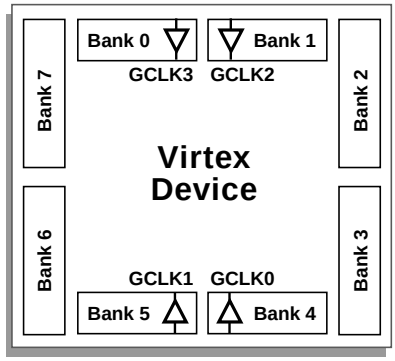


Figure 3: Virtex I/O Banks

Table 4: Compatible Output Standards

V_{CCO}	Compatible Standards
3.3 V	PCI, LVTTL, SSTL3 I, SSTL3 II, CTT, AGP, GTL, GTL+
2.5 V	SSTL2 I, SSTL2 II, LVCMOS2, GTL, GTL+
1.5 V	HSTL I, HSTL III, HSTL IV, GTL, GTL+

Some input standards require a user-supplied threshold voltage, V_{REF}. In this case, certain user-I/O pins are automatically configured as inputs for the V_{REF} voltage. Approximately one in six of the I/O pins in the bank assume this role.

The V_{REF} pins within a bank are interconnected internally and consequently only one V_{REF} voltage can be used within each bank. All V_{REF} pins in the bank, however, must be connected to the external voltage source for correct operation.

Within a bank, inputs that require V_{REF} can be mixed with those that do not. However, only one V_{REF} voltage may be used within a bank. Input buffers that use V_{REF} are not 5 V tolerant. LVTTL, LVCMOS2, and PCI 33 MHz 5 V, are 5 V tolerant.

The V_{CCO} and V_{REF} pins for each bank appear in the device pin-out tables and diagrams. The diagrams also show the bank affiliation of each I/O.

Within a given package, the number of V_{REF} and V_{CCO} pins can vary depending on the size of device. In larger devices, more I/O pins convert to V_{REF} pins. Since these are always a superset of the V_{REF} pins used for smaller devices, it is possible to design a PCB that permits migration to a larger device if necessary. All the V_{REF} pins for the largest device anticipated must be connected to the V_{REF} voltage, and not used for I/O.

In smaller devices, some V_{CCO} pins used in larger devices do not connect within the package. These unconnected pins may be left unconnected externally, or may be connected to the V_{CCO} voltage to permit migration to a larger device if necessary.

In TQ144 and PQ/HQ240 packages, all V_{CCO} pins are bonded together internally, and consequently the same V_{CCO} voltage must be connected to all of them. In the CS144 package, bank pairs that share a side are interconnected internally, pemitting four choices for VCCO. In both cases, the V_{REF} pins remain internally connected as eight banks, and may be used as described previously.

Configurable Logic Block

The basic building block of the Virtex CLB is the logic cell (LC). An LC includes a 4-input function generator, carry logic, and a storage element. The output from the function generator in each LC drives both the CLB output and the D input of the flip-flop. Each Virtex CLB contains four LCs, organized in two similar slices, as shown in Figure 4. Figure 5 shows a more detailed view of a single slice.

In addition to the four basic LCs, the Virtex CLB contains logic that combines function generators to provide functions of five or six inputs. Consequently, when estimating the number of system gates provided by a given device, each CLB counts as 4.5 LCs.

Look-Up Tables

Virtex function generators are implemented as 4-input look-up tables (LUTs). In addition to operating as a function generator, each LUT can provide a 16 x 1-bit synchronous RAM. Furthermore, the two LUTs within a slice can be combined to create a 16 x 2-bit or 32 x 1-bit synchronous RAM, or a 16x1-bit dual-port synchronous RAM.

The Virtex LUT can also provide a 16-bit shift register that is ideal for capturing high-speed or burst-mode data. This mode can also be used to store data in applications such as Digital Signal Processing.

Storage Elements

The storage elements in the Virtex slice can be configured either as edge-triggered D-type flip-flops or as level-sensitive latches. The D inputs can be driven either by the function generators within the slice or directly from slice inputs, bypassing the function generators.

In addition to Clock and Clock Enable signals, each Slice has synchronous set and reset signals (SR and BY). SR forces a storage element into the initialization state specified for it in the configuration. BY forces it into the opposite state. Alternatively, these signals may be configured to operate asynchronously. All of the control signals are independently invertible, and are shared by the two flip-flops within the slice.

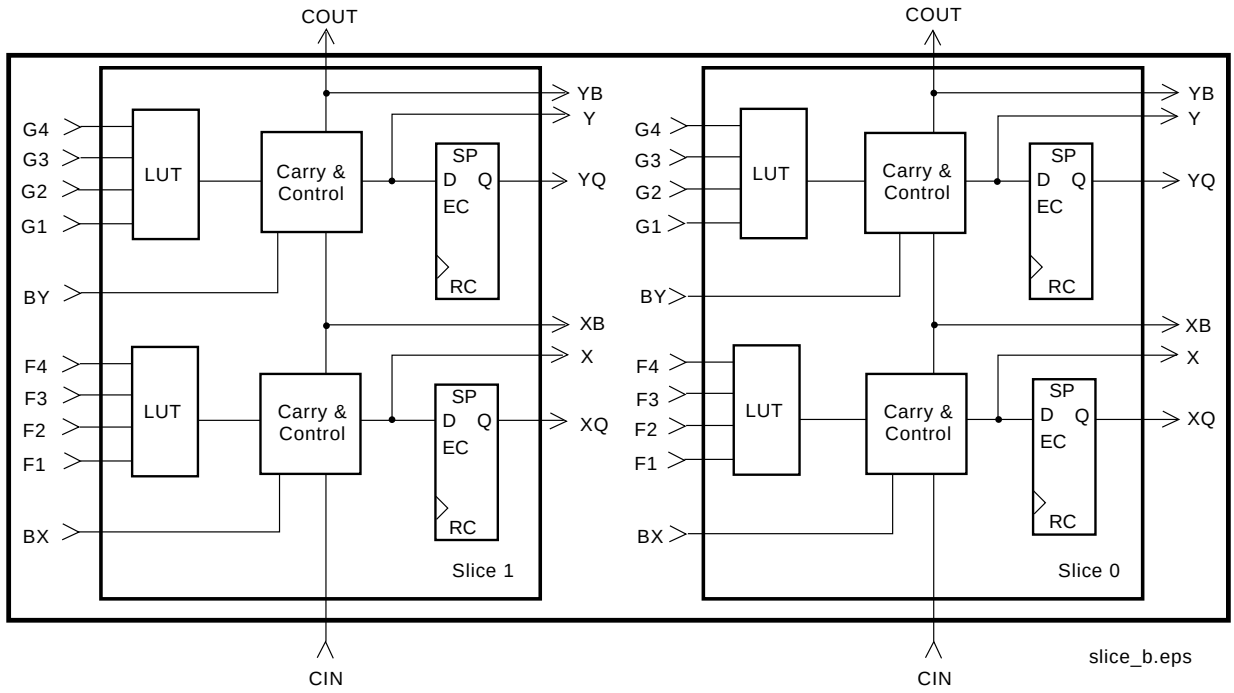


Figure 4: 2-Slice Virtex CLB

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, one per LC. 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 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 1-bit full adder to be implemented within an LC. 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 CLB contains two 3-state drivers (BUFTs) that can drive on-chip busses. See “Dedicated Routing” on page 9. Each Virtex BUFT has an independent 3-state control pin and an independent input pin.

Block SelectRAM

Virtex FPGAs incorporate several large Block SelectRAM memories. These complement the distributed LUT SelectRAMs that provide shallow RAM structures implemented in CLBs.

Block SelectRAM memory blocks are organized in columns. All Virtex devices contain two such columns, one along each vertical edge. These columns extend the full height of the chip. Each memory block is four CLBs high, and consequently, a Virtex device 64 CLBs high contains 16 memory blocks per column, and a total of 32 blocks.

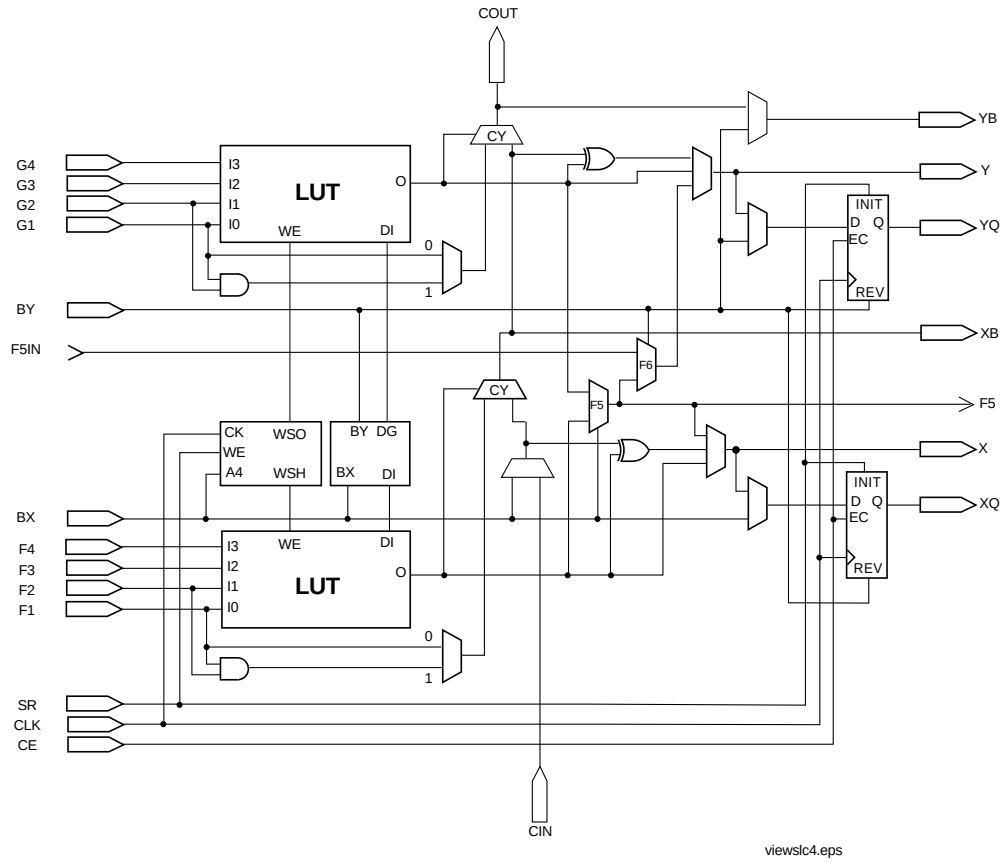


Figure 5: Detailed View of VIrtex Slice

Table 5 shows the amount of Block SelectRAM memory that is available in each Virtex device.

Table 5: Virtex Block SelectRAM Amounts

Virtex Device	# of Blocks	Total Block SelectRAM Bits
XCV50	8	32,768
XCV100	10	40,960
XCV150	12	49,152
XCV200	14	57,344
XCV300	16	65,536
XCV400	20	81,920
XCV600	24	98,304
XCV800	28	114,688
XCV1000	32	131,072

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.

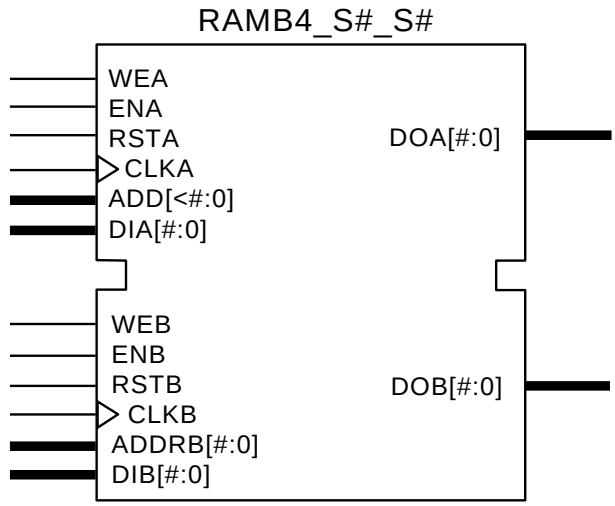


Figure 6: Dual-Port Block SelectRAM

.

Table 6 shows the depth and width aspect ratios for the Block SelectRAM

Table 6: 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.

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.

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.
- 96 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 may 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.

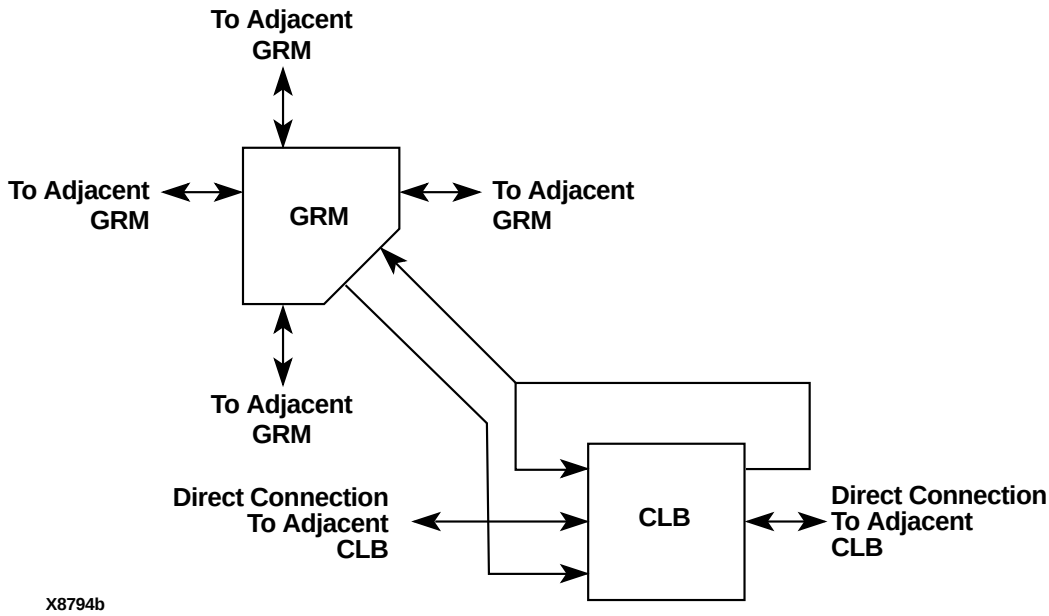


Figure 7: Virtex Local Routing

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.

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 may 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.

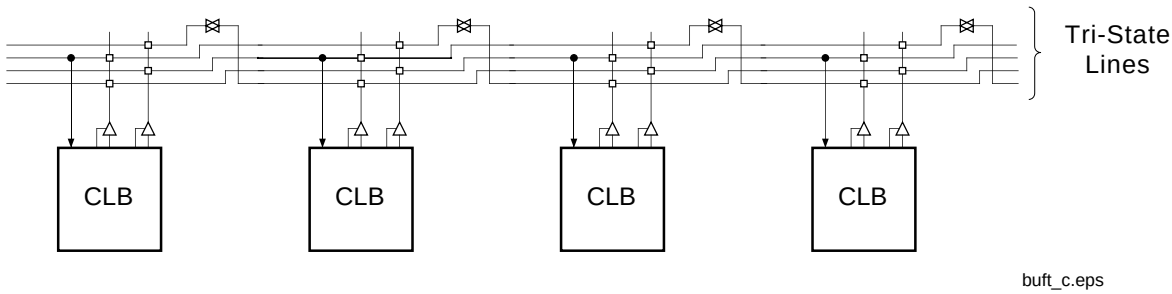


Figure 8: BUFT Connections to Dedicated Horizontal Bus Lines

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.

Four dedicated clock pads are provided, one adjacent to each of the global buffers. The input to the global buffer is selected either from these pads or from signals in the general purpose routing.

Delay-Locked Loop (DLL)

Associated with each global clock input buffer is a fully digital Delay-Locked Loop (DLL) that can eliminate skew between the clock input pad and internal clock-input pins throughout the device. Each DLL can drive two global clock networks.The DLL monitors the input clock and the distributed clock, and automatically adjusts a clock delay element. Additional delay is introduced such that clock edges reach internal flip-flops one to four clock periods after they arrive at the input. This closed-loop system effectively eliminates clock-distribution delay by ensuring that clock edges arrive at internal flip-flops in synchronism with clock edges arriving at the input.

In addition to eliminating clock-distribution delay, the DLL provides advanced control of multiple clock domains. The DLL provides four quadrature phases of the source clock, can double the clock, or divide the clock by 1.5, 2, 2.5, 3, 4, 5, 8, or 16.

The DLL also operates as a clock mirror. By driving the output from a DLL off-chip and then back on again, the DLL can be used to deskew a board level clock among multiple Virtex devices.

In order to guarantee that the system clock is operating correctly prior to the FPGA starting up after configuration, the DLL can delay the completion of the configuration process until after it has achieved lock.

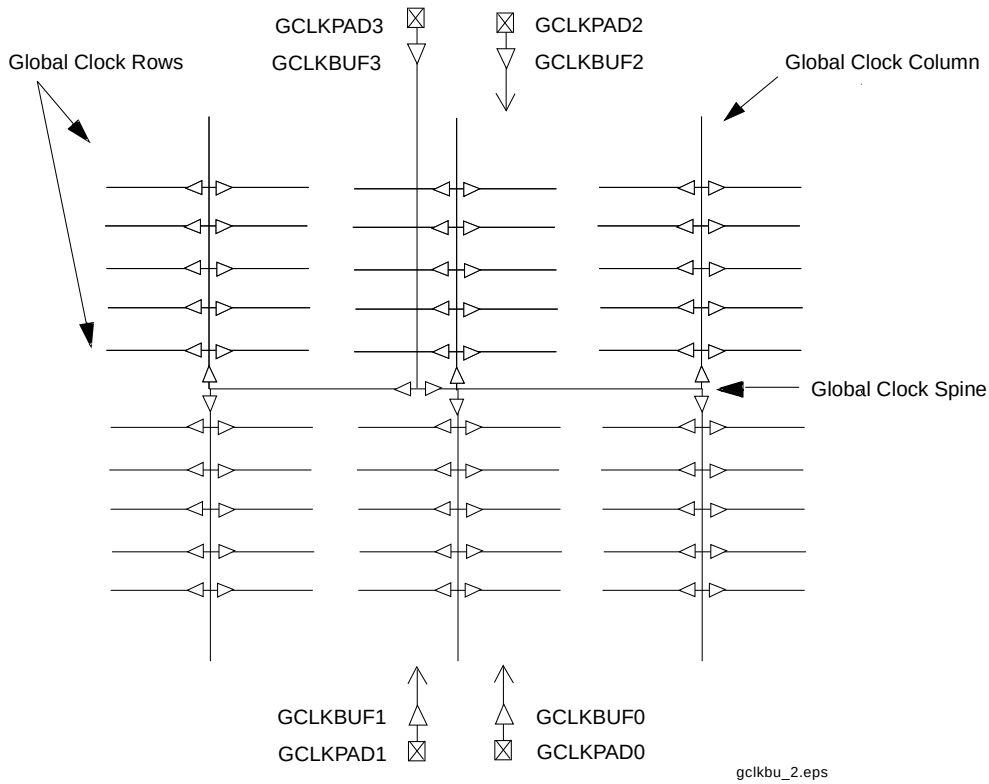


Figure 9: Global Clock Distribution Network

Boundary Scan

Virtex devices support all the mandatory boundary-scan instructions specified in the IEEE standard 1149.1. A Test Access Port (TAP) and registers are provided that implement the EXTEST, INTEST, SAMPLE/PRELOAD, BYPASS, IDCODE, USERCODE, and HIGHZ instructions. The TAP also supports two internal scan chains and configuration/readback of the device.

The TAP uses dedicated package pins that always operate using LVTTL. For TDO to operate using LVTTL, the V_{CCO} for Bank 2 should be 3.3 V. Otherwise, TDO switches rail-to-rail between ground and V_{CCO}.

Boundary-scan operation is independent of individual IOB configurations, and unaffected by package type. All IOBs, including un-bonded ones, are treated as independent 3-state bidirectional pins in a single scan chain. Retention of the bidirectional test capability after configuration facilitates the testing of external interconnections.

Table 7 lists the boundary-scan instructions supported in Virtex FPGAs. Internal signals can be captured during EXTEST by connecting them to un-bonded or unused IOBs. They may also be connected to the unused outputs of IOBs defined as unidirectional input pins.

Before the device is configured, all instructions except USER1 and USER2 are available. After configuration, all instructions are available. During configuration, it is recommended that those operations using the boundary-scan register (SAMPLE/PRELOAD, INTEST, EXTEST) not be performed.

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 7.

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.

Table 7: 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	Tri-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

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 decodes 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).

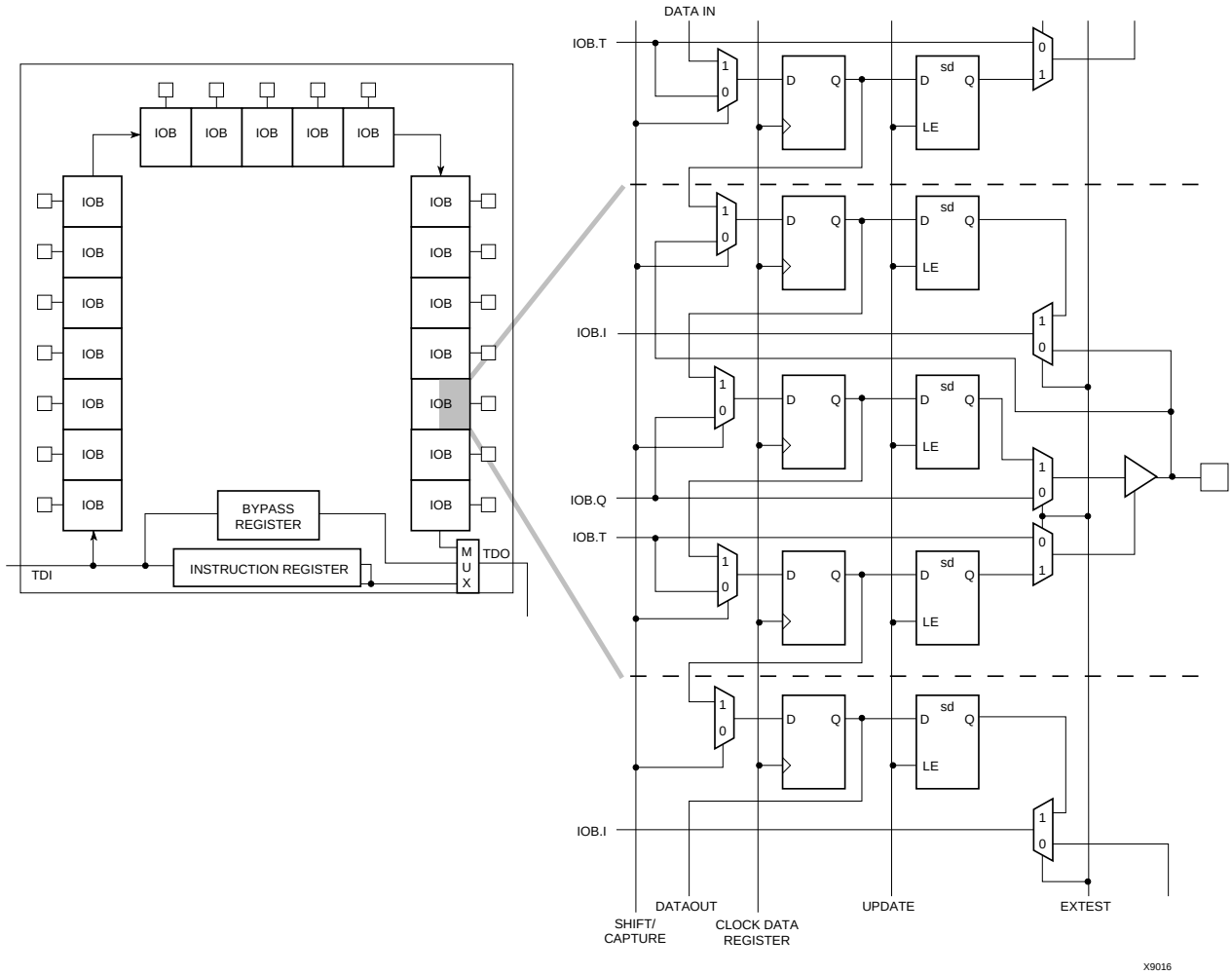


Figure 10: Virtex Series Boundary Scan Logic

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.

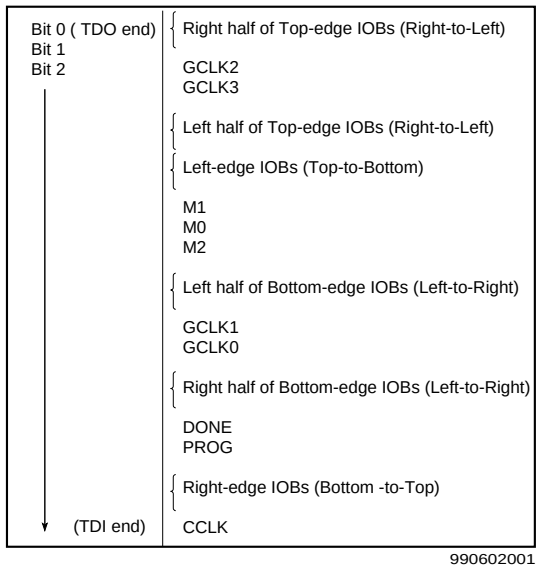


Figure 11: Boundary Scan Bit Sequence

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:fffa: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 8: 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 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 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 netlist 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 floorplanning.

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 netlist 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.

Configuration

Virtex devices are configured by loading configuration data into the internal configuration memory. Some of the pins used for this are dedicated configuration pins, while others may be re-used as general purpose inputs and outputs once configuration is complete.

The dedicated pins are the mode pins (M2, M1, M0), the configuration clock pin (CCLK), the INIT pin, the DONE pin and the boundary-scan pins (TDI, TDO, TMS, TCK). Depending on the configuration mode chosen, CCLK may be an output generated by the FPGA, or may be generated externally, and provided to the FPGA as an input.

Note that some configuration pins that may act as outputs. For correct operation, these pins may require a V_{CCO} of 3.3 V to permit LVTTL operation. All the pins affected fall in banks 2 or 3.

For a more detailed description than that given below, see the Supplementary Description on Configuration and Readback.

Configuration Modes

Virtex supports the following four configuration modes.

- Slave-serial mode
- Master-serial mode
- SelectMAP mode
- Boundary-scan mode

The Configuration mode pins (M2, M1, M0) select among these configuration modes with the option in each case of having the IOB pins either pulled up or left floating prior to configuration. The selection codes are listed in Table 9.

Configuration through the boundary-scan port is always available, independent of the mode selection. Selecting the boundary-scan mode simply turns off the other modes. The three mode pins have internal pull-up resistors, and default to a logic High if left unconnected.

Table 9: Configuration Codes

Configuration Mode	M2	M1	M0	CCLK Direction	Data Width	Serial D_{out}	Configuration Pull-ups
Master-serial mode	0	0	0	Out	1	Yes	No
Boundary-scan mode	1	0	1	N/A	1	No	No
SelectMAP mode	1	1	0	In	8	No	No
Slave-serial mode	1	1	1	In	1	Yes	No
Master-serial mode	1	0	0	Out	1	Yes	Yes
Boundary-scan mode	0	0	1	N/A	1	No	Yes
SelectMAP mode	0	1	0	In	8	No	Yes
Slave-serial mode	0	1	1	In	1	Yes	Yes

Slave Serial Mode

In slave serial mode, the FPGA receives configuration data in bit-serial form from a serial PROM or other source of serial configuration data. The serial bitstream must be setup at the DIN input pin a short time before each rising edge of an externally generated CCLK.

Multiple FPGAs can be daisy-chained for configuration from a single source. After a particular FPGA has been configured, the data for the next device is routed to the DOUT pin. The data on the DOUT pin changes on the rising edge of CCLK.

The change of DOUT on the rising edge of CCLK differs from previous families, but will not cause a problem for mixed configuration chains. This change was made to improve serial-configuration rates for Virtex only chains.

Figure 12 shows a full master/slave system. A Virtex device in slave serial mode should be connected as shown in the third device from the left

Slave-serial mode is selected by applying <111> or <011> to the mode pins (M2, M1, M0). A weak pull-up on the mode pins makes slave serial the default mode if the pins are left unconnected. Figure 13 shows slave-serial configuration timing.

Table 10 provides more detail about the characteristics shown in Figure 13. Configuration must be delayed until the INIT pins of all daisy-chained FPGAs are High.

Table 10: Master/Slave Serial Mode Programming Switching

	Description	Symbol			Units
CCLK	DIN setup/hold, slave mode	1/2	T_{DCC}/T_{CCD}	5.0/0.0	ns, min
	DIN setup/hold, master mode	1/2	T_{DSCK}/T_{SCKD}	5.0/0.0	ns, min
	DOUT	3	T_{CCO}	12.0	ns, max
	High time	4	T_{CCH}	5.0	ns, min
	Low time	5	T_{CCL}	5.0	ns, min
	Maximum Frequency		F_{CC}	66	MHz, max
	Frequency Tolerance, master mode with respect to nominal			+45% -30%	

.

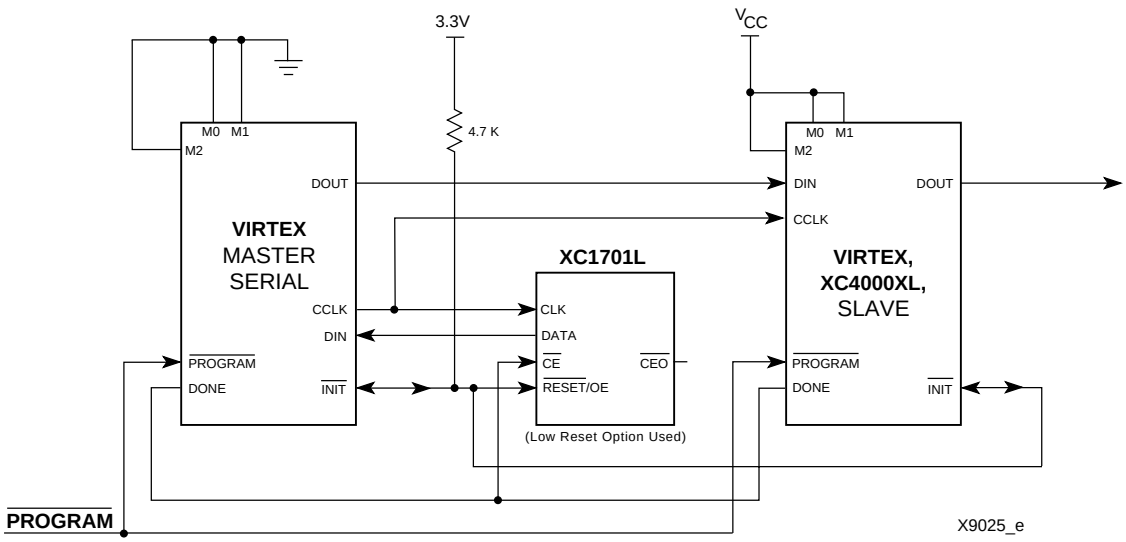


Figure 12: Master/Slave Serial Mode Circuit Diagram

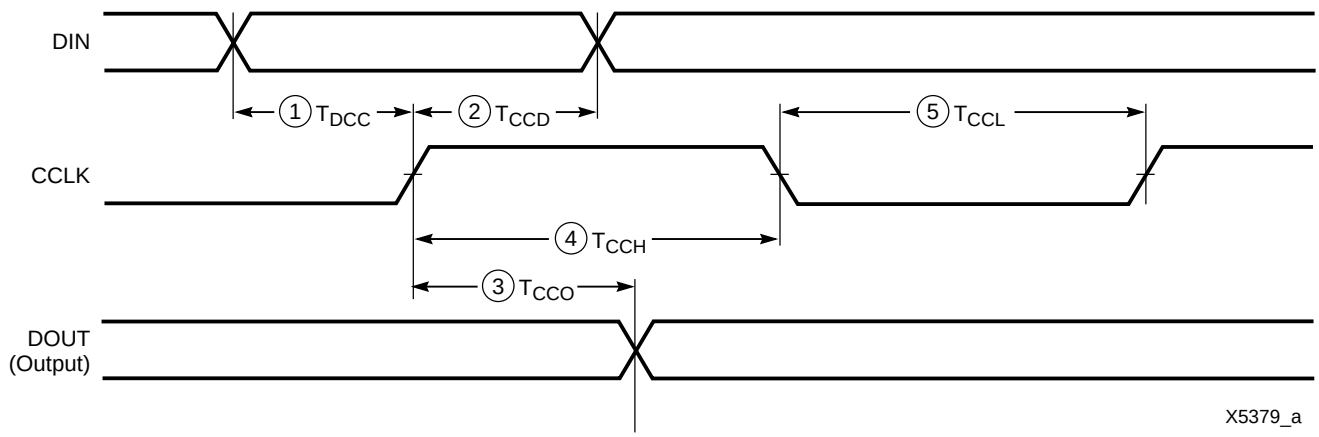


Figure 13: Slave Serial Mode Programming Switching Characteristics

Master Serial Mode

In master serial mode, the CCLK output of the FPGA drives a Xilinx Serial PROM that feeds bit-serial data to the DIN input. The FPGA accepts this data on each rising CCLK edge. After the FPGA has been loaded, the data for the next device in a daisy-chain is presented on the DOUT pin after the rising CCLK edge.

The interface is identical to slave serial except that an internal oscillator is used to generate the configuration clock (CCLK). A wide range of frequencies can be selected for CCLK which always starts at a slow default frequency. Configuration bits then switch CCLK to a higher frequency for the remainder of the configuration. Switching to a lower frequency is prohibited.

The CCLK frequency is set using the ConfigRate option in the bitstream generation software. The maximum CCLK frequency that can be selected is 60 MHz. When selecting a CCLK frequency, ensure that the serial PROM and any daisy-chained FPGAs are fast enough to support the clock rate.

On power-up, the CCLK frequency is 2.5 MHz. This frequency is used until the ConfigRate bits have been loaded when the frequency changes to the selected ConfigRate. Unless a different frequency is specified in the design, the default ConfigRate is 4 MHz.

Figure 12 shows a full master/slave system. In this system, the left-most device operates in master-serial mode. The remaining devices operate in slave-serial mode. The SPROM RESET pin is driven by INIT, and the CE input is driven by DONE. There is the potential for contention on the DONE pin, depending on the start-up sequence options chosen.

The sequence of operations necessary to configure a Virtex FPGA serially appears in Figure 14.

Figure 15 shows the timing of master-serial configuration. Master serial mode is selected by a <000> or <100> on the mode pins (M2, M1, M0). Table 10 shows the timing information for Figure 15.

At power-up, Vcc must rise from 1.0 V to Vcc min in less than 50 ms, otherwise delay configuration by pulling PROGRAM Low until Vcc is valid.

SelectMAP Mode

The SelectMAP mode is the fastest configuration option. Byte-wide data is written into the FPGA with a BUSY flag controlling the flow of data.

An external data source provides a byte stream, CCLK, a Chip Select (CS) signal and a Write signal (WRITE). If BUSY is asserted (High) by the FPGA, the data must be held until BUSY goes Low.

Data can also be read using the SelectMAP mode. If WRITE is not asserted, configuration data is read out of the FPGA as part of a readback operation.

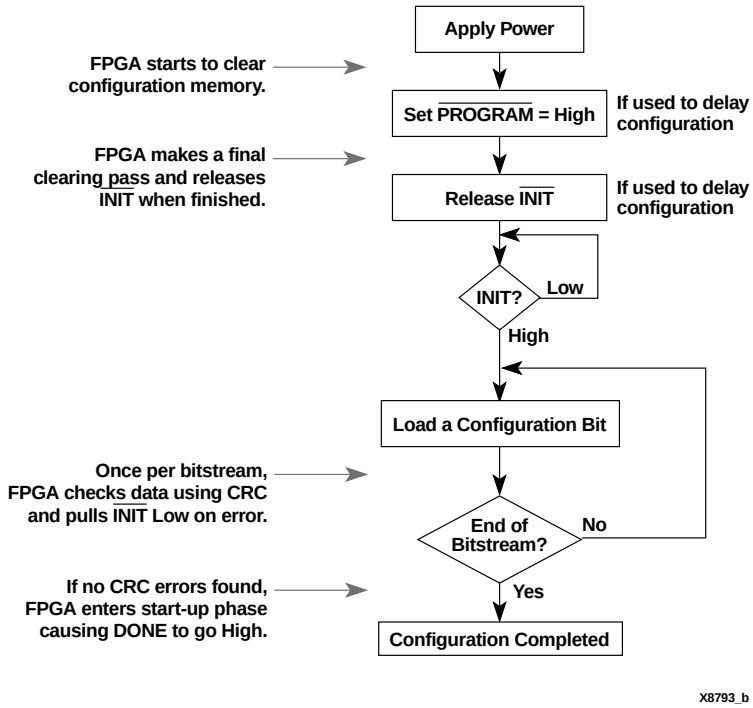


Figure 14: Serial Configuration Flowchart

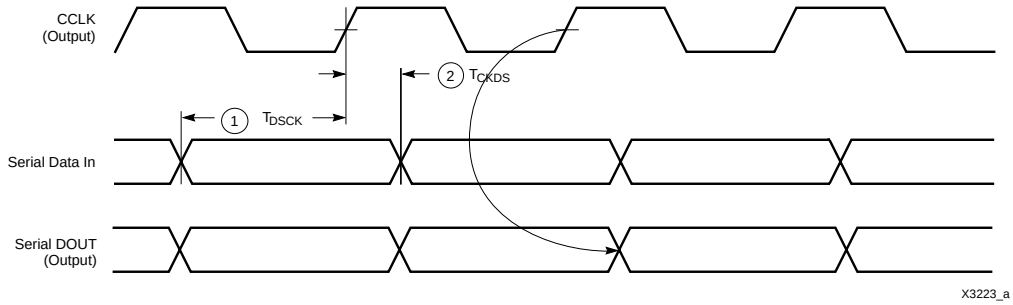


Figure 15: Master Serial Mode Programming Switching Characteristics

After configuration, the pins of the SelectMAP port can be used as additional user I/O. Alternatively, the port may be retained to permit high-speed 8-bit readback.

Retention of the SelectMAP port is selectable on a design-by-design basis when the bitstream is generated. If retention is selected, PROHIBIT constraints are required to prevent the SelectMAP-port pins from being used as user I/O.

Multiple Virtex FPGAs can be configured using the SelectMAP mode, and be made to start-up simultaneously. To configure multiple devices in this way, wire the individual CCLK, Data, $\overline{\text{WRITE}}$, and BUSY pins of all the devices in parallel. The individual devices are loaded separately by asserting the $\overline{\text{CS}}$ pin of each device in turn and writing the appropriate data. See Table 11 for SelectMAP Write Timing Characteristics.

Table 11: SelectMAP Write Timing Characteristics

	Description		Symbol		Units
CCLK	D_{0-7} Setup/Hold	1/2	T_{SMDCC}/T_{SMCCD}	5.0/0.0	ns, min
	$\overline{\text{CS}}$ Setup/Hold	3/4	T_{SMCSCC}/T_{SMCCCS}	7.0/0.0	ns, min
	$\overline{\text{WRITE}}$ Setup/Hold	5/6	T_{SMCCW}/T_{SMWCC}	7.0/0.0	ns, min
	BUSY Propagation Delay	7	T_{SMCKBY}	12.0	ns, max
	Maximum Frequency		F_{CC}	66	MHz, max
	Maximum Frequency with no handshake		F_{CCNH}	50	MHz, max

Write

Write operations send packets of configuration data into the FPGA. The sequence of operations for a multi-cycle write operation is shown below. Note that a configuration packet can be split into many such sequences. The packet does not have to complete within one assertion of $\overline{\text{CS}}$, illustrated in Figure 16.

1. Assert $\overline{\text{WRITE}}$ and $\overline{\text{CS}}$ Low. Note that when $\overline{\text{CS}}$ is asserted on successive CCLKs, $\overline{\text{WRITE}}$ must remain either asserted or deasserted. Otherwise an abort will be initiated, as described below.
2. Drive data onto D[7:0]. Note that to avoid contention, the data source should not be enabled while $\overline{\text{CS}}$ is Low and $\overline{\text{WRITE}}$ is High. Similarly, while $\overline{\text{WRITE}}$ is High, no more that one $\overline{\text{CS}}$ should be asserted.
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 $\overline{\text{WRITE}}$.

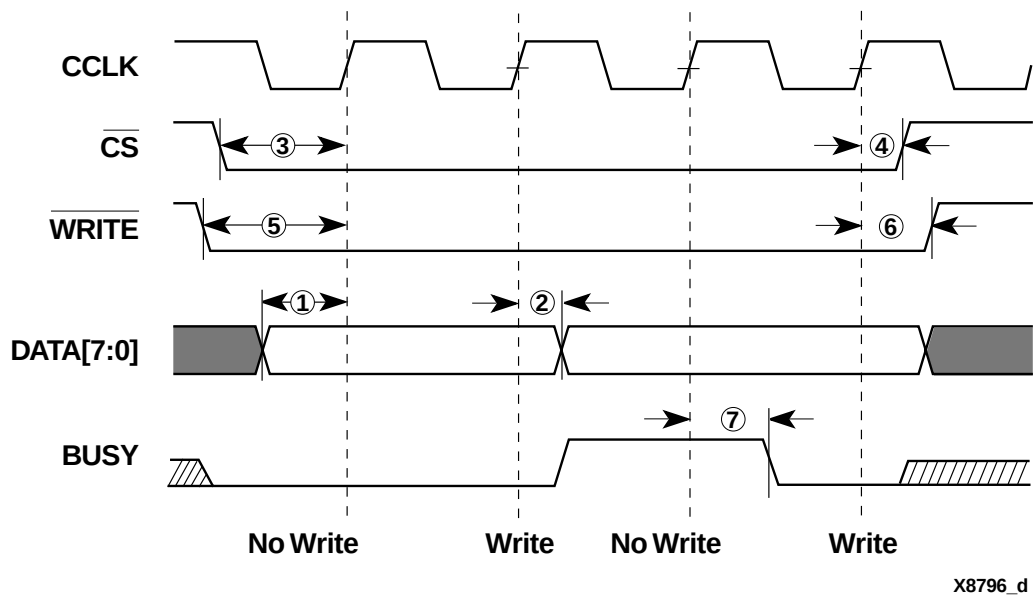


Figure 16: Write Operations

A flowchart for the write operation appears in Figure 17. Note that if CCLK is slower than f_{CCNH}, the FPGA will never assert BUSY, In this case, the above handshake is unnecessary, and data can simply be entered into the FPGA every CCLK cycle.

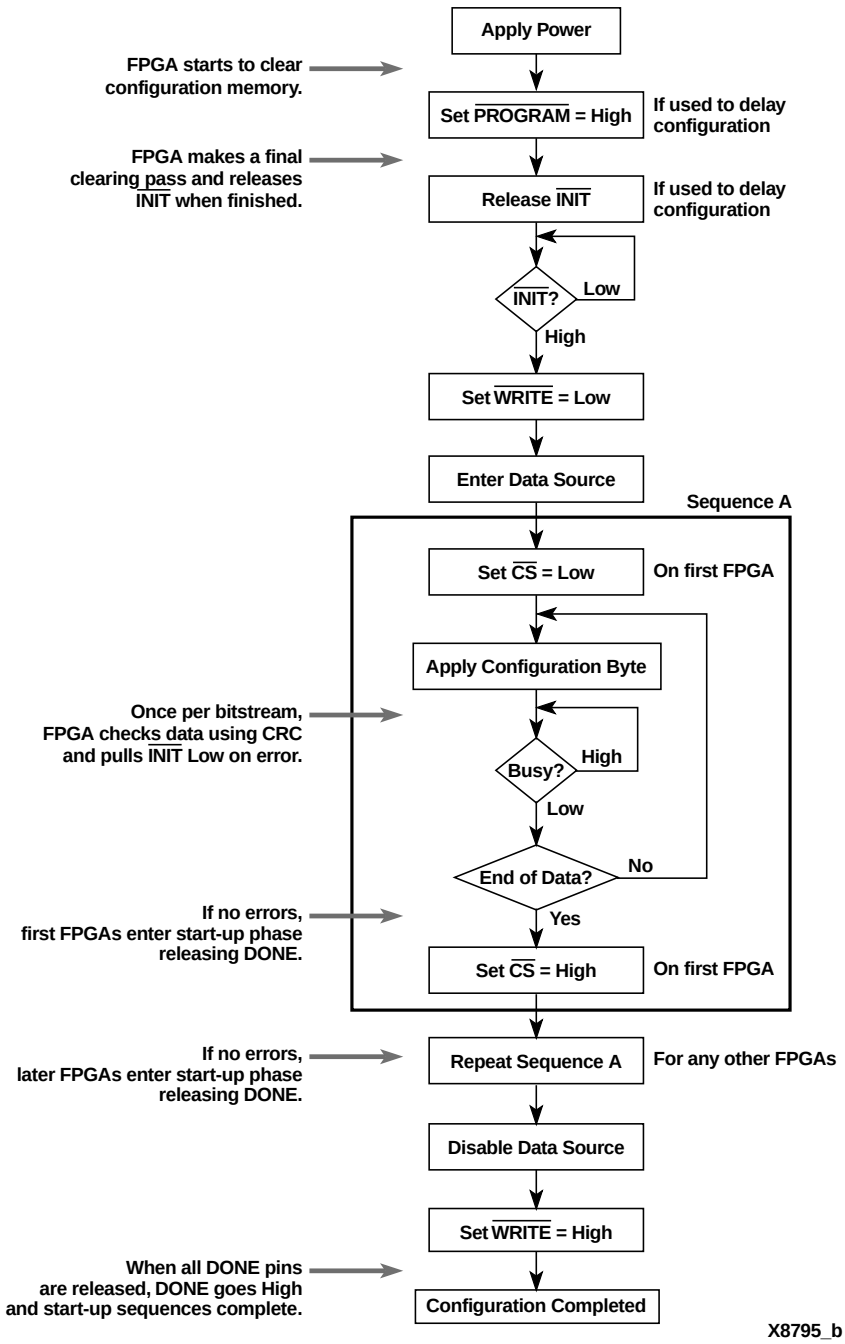


Figure 17: SelectMAP Flowchart for Write Operation

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

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

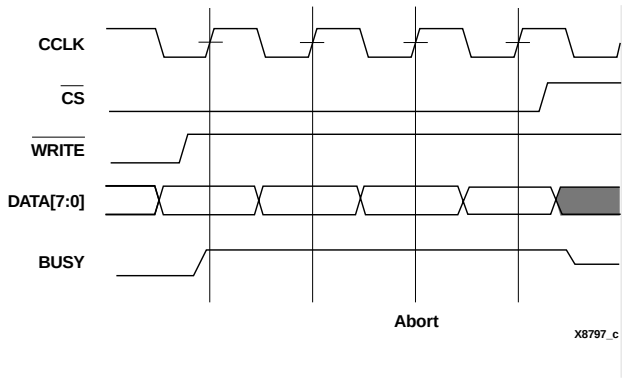


Figure 18: SelectMAP Write Abort Waveforms

Boundary-Scan Mode

In the boundary-scan mode, no non-dedicated pins are required, configuration being done entirely through the IEEE 1149.1 Test Access Port.

Configuration through the TAP uses the CFG_IN instruction. This instruction allows data input on TDI to be converted into data packets for the internal configuration bus.

The following steps are required to configure the FPGA through the boundary-scan port (when using TCK as a start-up clock).

1. Load the CFG_IN instruction into the boundary-scan instruction register (IR)
2. Enter the Shift-DR (SDR) state
3. Shift a configuration bitstream into TDI
4. Return to Run-Test-Idle (RTI)
5. Load the JSTART instruction into IR
6. Enter the SDR state
7. Clock TCK through the startup sequence
8. Return to RTI

Configuration and readback via the TAP is always available. The boundary-scan mode is selected by a <101> or 001> on the mode pins (M2, M1, M0).

Configuration Sequence

The configuration of Virtex devices is a three-phase process. First, the configuration memory is cleared. Next, configuration data is loaded into the memory, and finally, the logic is activated by a start-up process.

Configuration is automatically initiated on power-up unless it is delayed by the user, as described below. The configuration process may also be initiated by asserting PROGRAM. The end of the memory-clearing phase is signalled by INIT going High, and the completion of the entire process is signalled by DONE going High.

The power-up timing of configuration signals is shown in Figure 19. The corresponding timing characteristics are listed in Table 12.

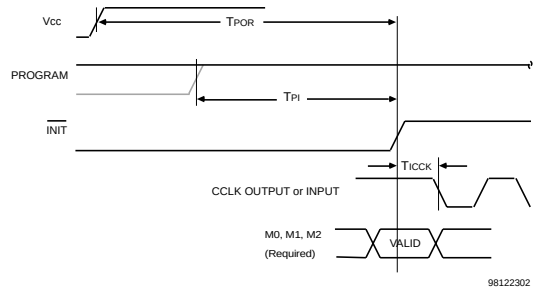


Figure 19: Power-up Timing Configuration Signals

Table 12: Power-up Timing Characteristics

Description	Symbol	Value	Units
Power-on Reset	T_{POR}	2.0	ms, max
Program Latency	T_{PL}	100.0	µs, max
CCLK (output) Delay	T_{ICCK}	0.5	µs, min
		4.0	µs, max
Program Pulse Width	$T_{PROGRAM}$	300	ns, min

Delaying Configuration

Configuration of the FPGA can be delayed by holding the PROGRAM pin Low until the system is ready for the device to configure. During the memory clearance phase, the configuration sequences continuously cycles through the configuration memory clearing all addresses. This activity continues until the completion of one full address cycle after the PROGRAM pin goes High. Thus, configuration is delayed by extending the memory clearance phase.

Alternatively, INIT can be held Low using an open-drain driver. An open-drain is required since INIT is a bidirectional open-drain pin that is held Low by the FPGA while the configuration memory is being cleared. Extending the time that the pin is Low causes the configuration sequencer to act as if the configuration memory is still being cleared. Thus, configuration is delayed by preventing entry into the phase where data is loaded.

Start-Up Sequence

The default Start-up sequence is that one CCLK cycle after DONE goes High, the global tri-state signal (GTS) is released. This permits device outputs to turn on as necessary.

One CCLK cycle later, the Global Set/Reset (GSR) and Global Write Enable (GWE) signals are released. This per-

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

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

Data Stream Format

Virtex devices are configured by sequentially loading frames of data. Table 13 lists the total number of bits required to configure each device. For more detailed information, see application note XAPP151 “Virtex Configuration Architecture Advanced Users Guide”.

Table 13: Virtex Bit-stream Lengths

Device	# of Configuration Bits
XCV50	559,232
XCV100	781,248
XCV150	1,041,128
XCV200	1,335,872
XCV300	1,751,840
XCV400	2,546,080
XCV600	3,608,000
XCV800	4,715,684
XCV1000	6,127,776

Readback

The configuration data stored in the Virtex configuration memory can be readback for verification. Along with the configuration data it is possible to readback the contents all flip-flops/latches, LUTRAMs, 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”.

Virtex DC Characteristics

Definition of Terms

Data sheets may be designated as Advance or Preliminary. The status of specifications in these data sheets is as follows:

Advance: Initial estimates based on simulation and/or extrapolation from other speed grades, devices, or families. Values are subject to change. Use as estimates, not for production.

Preliminary: Based on preliminary characterization. Further changes are not expected.

Unmarked: Data sheets not identified as either Advance or Preliminary are to be considered final.

All specifications are representative of worst-case supply voltage and junction temperature conditions. The parameters included are common to popular designs and typical applications. Contact the factory for design considerations requiring more detailed information.

All specifications are subject to change without notice.

Absolute Maximum Ratings

Symbol	Description			Units
V_{CCINT}	Supply voltage relative to GND		-0.5 to 3.0	V
V_{CCO}	Supply voltage relative to GND		-0.5 to 4.0	V
V_{REF}	Input Reference Voltage		-0.5 to 3.6	V
V_{IN}	Input voltage relative to GND	Using V_{REF}	-0.5 to 3.6	V
		Internal threshold	-0.5 to 5.5	V
V_{TS}	Voltage applied to 3-state output		-0.5 to 5.5	V
V_{CC}	Longest Supply Voltage Rise Time from 1V-2.375V		50	ms
T_{STG}	Storage temperature (ambient)		-65 to +150	°C
T_{SOL}	Maximum soldering temp. (10s @ 1/16 in. = 1.5 mm)		+260	°C
T_J	Junction temperature	Ceramic packages	+125	°C
		Plastic packages	+125	°C

Notes: Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those listed under Operating Conditions is not implied. Exposure to Absolute Maximum Ratings conditions for extended periods of time may affect device reliability.

Power supplies may turn on in any order.

For protracted periods (e.g., longer than a day), V_{IN} should not exceed V_{CCO} by more that 3.6 V.

Recommended Operating Conditions

Symbol	Description		Min	Max	Units
V_{CCINT}	Supply voltage relative to GND, T_J = 0 °C to +85°C	Commercial	2.5 - 5%	2.5 + 5%	V
	Supply voltage relative to GND, T_J = -40°C to +100°C	Industrial	2.5 - 5%	2.5 + 5%	V
V_{CCO}	Supply voltage relative to GND, T_J = 0 °C to +85°C	Commercial	1.2	3.6	V
	Supply voltage relative to GND, T_J = -40°C to +100°C	Industrial	1.2	3.6	V
T_{IN}	Input signal transition time			250	ns

Notes: Correct operation is guaranteed with a minimum V_{CCINT} of 2.25 V (Nominal V_{CCINT} -10%). Below the minimum value stated above, all delay parameters increase by 3% for each 50-mV reduction in V_{CCINT} below the specified range.

At junction temperatures above those listed as Operating Conditions, all delay parameters increase by 0.35% per °C.

Input and output measurement threshold is ~50% of V_{CC}.

DC Characteristics Over Recommended Operating Conditions

Symbol	Description		Device	Min	Max	Units
V_{DRINT}	Data Retention V_{CCINT} Voltage (below which configuration data may be lost)		All	2.0		V
V_{DRIO}	Data Retention V_{CCO} Voltage (below which configuration data may be lost)		All	1.2		V
I_{CCINTQ}	Quiescent V_{CCINT} supply current (Note 1)		XCV50			
			XCV100			
			XCV150			
			XCV200			
			XCV300			
			XCV400			
			XCV600			
			XCV800			
			XCV1000			
I_{CCOQ}	Quiescent V_{CCO} supply current (Note 1)		XCV50			
			XCV100			
			XCV150			
			XCV200			
			XCV300			
			XCV400			
			XCV600			
			XCV800			
			XCV1000			
I_{REF}	V_{REF} current per V_{REF} pin		All		20	µA
I_L	Input or output leakage current		All	-10	+10	µA
C_{IN}	Input capacitance (sample tested)	BGA, PQ, HQ, packages	All		8	pF
I_{RPU}	Pad pull-up (when selected) @ V_{in} = 0 V, V_{CCO} = 3.3 V (sample tested)		All	Note 2	0.25	mA
I_{RPD}	Pad pull-down (when selected) @ V_{in} = 3.6 V (sample tested)			Note 2	0.15	mA

Note 1: With no output current loads, no active input pull-up resistors, all I/O pins Tri-stated and floating.
Note 2: Internal pull-up and pull-down resistors guarantee valid logic levels at unconnected input pins. These pull-up and pull-down resistors do not guarantee valid logic levels when input pins are connected to other circuits.

DC Input and Output levels

Values for V_{IL} and V_{IH} are recommended input voltages. Values for I_{OL} and I_{OH} are guaranteed output currents over the recommended operating conditions at the V_{OL} and V_{OH} test points. Only selected standards are tested. These are chosen to ensure that all standards meet their specifications. The selected standards are tested at minimum V_{CCO} with the respective V_{OL} and V_{OH} voltage levels shown. Other standards are sample tested.

Input/Output Standard	V_{IL}		V_{IH}		V_{OL}	V_{OH}	I_{OL}	I_{OH}
	V, min	V, max	V, min	V, max	V, Max	V, Min	mA	mA
LVTTL (Note 1)	- 0.5	0.8	2.0	5.5	0.4	2.4	24	- 24
LVCMOS2	- 0.5	0.7	1.7	5.5	0.4	1.9	12	-12
PCI, 3.3 V	- 0.5	44% V_{CCINT}	60% V_{CCINT}	V_{CCO} + 0.5	10% V_{CCO}	90% V_{CCO}	Note 2	Note 2
PCI, 5.0 V	- 0.5	0.8	2.0	5.5	0.55	2.4	Note 2	Note 2
GTL	- 0.5	V_{REF} - 0.05	V_{REF} + 0.05	3.6	0.4	n/a	40	n/a
GTL+	- 0.5	V_{REF} - 0.1	V_{REF} + 0.1	3.6	0.6	n/a	36	n/a
HSTL I	- 0.5	V_{REF} - 0.1	V_{REF} + 0.1	3.6	0.4	V_{CCO} - 0.4	8	-8
HSTL III	- 0.5	V_{REF} - 0.1	V_{REF} + 0.1	3.6	0.4	V_{CCO} - 0.4	24	-8
HSTL IV	- 0.5	V_{REF} - 0.1	V_{REF} + 0.1	3.6	0.4	V_{CCO} - 0.4	48	-8
SSTL3 I	- 0.5	V_{REF} - 0.2	V_{REF} + 0.2	3.6	V_{REF} - 0.6	V_{REF} + 0.6	8	-8
SSTL3 II	- 0.5	V_{REF} - 0.2	V_{REF} + 0.2	3.6	V_{REF} - 0.8	V_{REF} + 0.8	16	-16
SSTL2 I	- 0.5	V_{REF} - 0.2	V_{REF} + 0.2	3.6	V_{REF} - 0.50	V_{REF} + 0.50	7.6	-7.6
SSTL2 II	- 0.5	V_{REF} - 0.2	V_{REF} + 0.2	3.6	V_{REF} - 0.50	V_{REF} + 0.50	15.2	-15.2
CTT	- 0.5	V_{REF} - 0.2	V_{REF} + 0.2	3.6	V_{REF} - 0.4	V_{REF} + 0.4	8	-8
AGP	- 0.5	V_{REF} - 0.2	V_{REF} + 0.2	3.6	10% V_{CCO}	90% V_{CCO}	Note 2	Note 2

Note 1: V_{OL} and V_{OH} for lower drive currents are sample tested.
Note 2: Tested according to the relevant specifications.

Virtex Switching Characteristics

Testing of switching parameters is modeled after testing methods specified by MIL-M-38510/605. 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 netlist. All timing parameters assume worst-case operating conditions (supply voltage and junction temperature). Values apply to all Virtex devices unless otherwise noted.

IOB Input Switching Characteristics

Input delays associated with the pad are specified for LVTTL levels. For other standards, adjust the delays with the values shown in"IOB Input Switching Characteristics Standard Adjustments" on page 27.

			Speed Grade			
Description	**Device**	**Symbol**	**-6**	**-5**	**-4**	**Units**
Propagation Delays						
Pad to output, no delay	All	T_{IOPI}	0.8	0.9	1.0	ns, max
Pad to I output, with delay	XCV50	T_{IOPID}	1.5	1.7	1.9	ns, max
	XCV100		1.5	1.7	1.9	ns, max
	XCV150		1.5	1.7	1.9	ns, max
	XCV200		1.5	1.7	1.9	ns, max
	XCV300		1.5	1.7	1.9	ns, max
	XCV400		1.8	2.0	2.3	ns, max
	XCV600		1.8	2.0	2.3	ns, max
	XCV700		2.2	2.4	2.7	ns, max
	XCV1000		2.2	2.4	2.7	ns, max
Pad to output IQ via transparent latch, no delay	All	T_{IOPLI}	1.6	1.8	2.0	ns, max
Pad to output IQ via transparent latch, with delay	XCV50	T_{IOPLID}	3.8	4.1	4.8	ns, max
	XCV100		3.8	4.1	4.8	ns, max
	XCV150		4.0	4.3	5.0	ns, max
	XCV200		4.0	4.4	5.0	ns, max
	XCV300		4.0	4.4	5.1	ns, max
	XCV400		4.2	4.6	5.4	ns, max
	XCV600		4.2	4.7	5.5	ns, max
	XCV700		4.4	4.8	5.7	ns, max
	XCV1000		4.5	5.1	5.9	ns, max
Sequential Delays						
Clock CLK to output IQ	All	T_{IOCKIQ}	0.7	0.7	0.8	ns, max
Setup and Hold Times with respect to Clock CLK		Setup Time/Hold Time				
Pad, no delay	All	T_{IOPICK}/T_{IOICKP}	1.6 / 0.0	1.8 / 0.0	2.0 / 0.0	ns, min
Pad, with delay	All	$T_{IOPICKD}/T_{IOICKPD}$	3.9 / 0.0	4.4 / 0.0	5.0 / 0.0	ns, min
ICE input	All	$T_{IOICECK}/T_{IOCKICE}$	0.8 / 0.0	0.9 / 0.0	1.0 / 0.0	ns, min
SR input (IFF, synchronous)	All	$T_{IOSRCKI}/T_{IOCKISR}$	1.0 / 0.0	1.1 / 0.0	1.3 / 0.0	ns, min
Set/Reset Delays						
SR input to IQ (asynchronous)	All	T_{IOSRIQ}	1.4	1.6	1.8	ns, max
GSR to output IQ	All	T_{GSRQ}	9.7	10.9	12.5	ns, max

IOB Input Switching Characteristics Standard Adjustments

			Speed Grade			
Description	**Symbol**	**Standard**	**-6**	**-5**	**-4**	**Units**
Data Input Delay Adjustments						
Standard-specific data input delay adjustments	$T_{IPOFFSET}$	LVTTL	0.0	0.0	0.0	ns
		LVCMOS2	0.0	-0.1	-0.1	ns
		PCI, 33 MHz, 3.3 V	-0.1	-0.2	-0.2	ns
		PCI, 33 MHz, 5.0 V	0.3	0.3	0.4	ns
		PCI, 66 MHz, 3.3 V	-0.1	-0.2	-0.2	ns
		GTL	0.2	0.3	0.3	ns
		GTL+	0.2	0.2	0.2	ns
		HSTL	0.1	0.1	0.1	ns
		SSTL3	0.0	-0.1	-0.1	ns
		SSTL2	0.0	-0.1	-0.1	ns
		CTT	0.1	0.1	0.1	ns
		AGP	0.0	-0.1	-0.1	ns

IOB Output Switching Characteristics

Output delays terminating at a pad are specified for LVTTL with 12 mA drive and fast slew rate. For other standards, adjust the delays with the values shown in "IOB Output Switching Characteristics Standard Adjustments" on page 29 (T_{OPSL}).

		Speed Grade			
Description	**Symbol**	**-6**	**-5**	**-4**	**Units**
Propagation Delays					
O input to Pad	T_{IOOP}	3.3	3.7	4.2	ns, max
O input to Pad via transparent latch	T_{IOOLP}	3.7	4.2	4.8	ns, max
3-State Delays					
T input to Pad high-impedance (Note 1)	T_{IOTHZ}	2.3	2.8	3.0	ns, max
T input to valid data on Pad	T_{IOTON}	3.4	3.8	4.4	ns, max
T input to Pad high-impedance via transparent latch (Note 1)	$T_{IOTLPHZ}$	2.8	3.1	3.5	ns, max
T input to valid data on Pad via transparent latch	$T_{IOTLPON}$	3.8	4.3	4.9	ns, max
GTS to Pad high impedance (Note 1)	T_{GTS}	4.9	5.5	6.3	ns, max
Sequential Delays					
Clock CLK to Pad	T_{IOCKP}	3.3	3.7	4.2	ns, max
Clock CLK to Pad high-impedance (synchronous) (Note 1)	T_{IOCKHZ}	2.7	3.0	3.4	ns, max
Clock CLK to valid data on Pad (synchronous)	T_{IOCKON}	3.7	4.2	4.8	ns, max
Setup Times before Clock CLK					
O input	T_{IOOCK}	1.1	1.2	1.3	ns, min
OCE input	$T_{IOOCECK}$	0.8	0.9	1.0	ns, min
SR input (OFF)	$T_{IOSRCKO}$	1.1	1.2	1.4	ns, min
3-State Setup Times, TCE input	$T_{IOTCECK}$	0.9	0.9	1.1	ns, min
3-State Setup Times, T input	T_{IOTCK}	0.7	0.8	0.9	ns, min
3-State Setup Times, SR input (TFF)	$T_{IOSRCKT}$	1.0	1.1	1.3	ns, min
Hold Times after Clock CLK					
All Hold Times		0.0	0.0	0.0	ns, min
Set/Reset Delays					
SR input to Pad (asynchronous)	T_{IOSRP}	4.1	4.6	5.3	ns, max
SR input to Pad high-impedance (asynchronous) (Note 1)	T_{IOSRHZ}	3.5	3.9	4.4	ns, max
SR input to valid data on Pad (asynchronous)	T_{IOSRON}	4.5	5.1	5.8	ns, max
GSR to Pad	T_{GSRQ}	9.7	10.9	12.5	ns, max

Note 1: Tri-state turn-off delays should not be adjusted.

IOB Output Switching Characteristics Standard Adjustments

Output delays terminating at a pad are specified for LVTTL with 12 mA drive and fast slew rate. For other standards, adjust the delays by the values shown.

			Speed Grade			
Description	**Symbol**	**Standard**	**-6**	**-5**	**-4**	**Units**
Output Delay Adjustments						
Standard-specific adjustments for output delays terminating at pads (based on standard capacitive load, Csl)	$T_{OPADJUST}$	LVTTL, Slow,2 mA	15.2	17.1	19.6	ns
		4 mA	7.7	8.7	10.0	ns
		6 mA	5.0	5.6	6.5	ns
		8 mA	4.2	4.8	5.5	ns
		12 mA	3.0	3.4	3.9	ns
		16 mA	2.2	2.5	2.9	ns
		24 mA	2.1	2.4	2.8	ns
		LVTTL, Fast,2 mA	13.5	15.2	17.4	ns
		4 mA	5.5	6.2	7.1	ns
		6 mA	3.2	3.6	4.2	ns
		8 mA	1.2	1.4	1.6	ns
		12 mA	0.0	0.0	0.0	ns
		16 mA	-0.1	-0.1	-0.1	ns
		24 mA	-0.5	-0.5	-0.6	ns
		LVCMOS2	0.1	0.1	0.2	ns
		PCI, 33 MHz, 3.3 V	2.3	2.6	3.0	ns
		PCI, 33 MHz, 5.0 V	2.8	3.2	3.6	ns
		PCI, 66 MHz, 3.3 V	-0.4	-0.4	-0.5	ns
		GTL	0.5	0.6	0.7	ns
		GTL+	0.8	0.9	1.0	ns
		HSTL I	-0.5	-0.5	-0.6	ns
		HSTL III	-0.9	-1.0	-1.1	ns
		HSTL IV	-1.0	-1.1	-1.2	ns
		SSTL3 I	-0.5	-0.5	-0.6	ns
		SSTL3 II	-1.0	-1.1	-1.2	ns
		SSTL2 I	-0.5	-0.5	-0.6	ns
		SSTL2 II	-0.9	-1.0	-1.1	ns
		CTT	-0.6	-0.6	-0.7	ns
		AGP	-0.9	-1.0	-1.1	ns

Calculation of T_{ioop} as a Function of Capacitance

The values for T_{ioop} were based on the standard capacitive load (Csl) for each IO standard as listed in Table 14.

For other capacitive loads, use the formulas below to calculate the corresponding T_{ioop}.

$$T_{ioop} = T_{ioop} + T_{opadjust} + (C_{load} - C_{sl}) * fl$$

Where:

$T_{opadjust}$ is reported above in the Output Delay Adjustment section.
C_{load} is the capacitive load for the design.

Table 14: Constants for Use in Calculation of T_{ioop}

Standard	Csl (pF)	fl (ns/pF)
LVCMOS2	35	0.041
LVTTL Fast Slew Rate, 2mA drive	35	0.41
LVTTL Fast Slew Rate, 4mA drive	35	0.20
LVTTL Fast Slew Rate, 6mA drive	35	0.13
LVTTL Fast Slew Rate, 8mA drive	35	0.079
LVTTL Fast Slew Rate, 12mA drive	35	0.048
LVTTL Fast Slew Rate, 16mA drive	35	0.043
LVTTL Fast Slew Rate, 24mA drive	35	0.033
LVTTL Slow Slew Rate, 2mA drive	35	0.41
LVTTL Slow Slew Rate, 4mA drive	35	0.20
LVTTL Slow Slew Rate, 6mA drive	35	0.14
LVTTL Slow Slew Rate, 8mA drive	35	0.086
LVTTL Slow Slew Rate, 12mA drive	35	0.058
LVTTL Slow Slew Rate, 16mA drive	35	0.053
LVTTL Slow Slew Rate, 24mA drive	35	0.44
HSTL Class I	20	0.022
HSTL Class III	20	0.016
HSTL Class IV	20	0.014
SSTL2 Class I	30	0.028
SSTL2 Class II	30	0.016
SSTL3 Class 1	30	0.029
SSTL3 Class II	30	0.016
CTT	20	0.035
AGP	10	0.037
GTL	0	0.014
GTL+	0	0.017
PCI 33MHz 5V	50	0.050
PCI 33MHZ 3.3 V	10	0.050
PCI 66 MHz 3.3 V	10	0.033

Clock Distribution Guidelines

			Speed Grade			
Description	**Device**	**Symbol**	**-6**	**-5**	**-4**	**Units**
Global Clock Skew						
Global Clock Skew between IOB Flip-flops	XCV50	$T_{GSKEWIOB}$	0.10	0.12	0.14	ns, max
	XCV100		0.12	0.13	0.15	ns, max
	XCV150		0.12	0.13	0.15	ns, max
	XCV200		0.13	0.14	0.16	ns, max
	XCV300		0.14	0.16	0.18	ns, max
	XCV400		0.13	0.13	0.14	ns, max
	XCV600		0.14	0.15	0.17	ns, max
	XCV800		0.16	0.17	0.20	ns, max
	XCV1000		0.20	0.23	0.25	ns, max

Note: These clock-skew delays are provided for guidance only. They reflect the delays encountered in a typical design under worst-case conditions. Precise values for a particular design are provided by the timing analyzer.

Clock Distribution Switching Characteristics

		Speed Grade			
Description	**Symbol**	**-6**	**-5**	**-4**	**Units**
GCLK IOB and Buffer					
Global Clock PAD to output.	T_{GPIO}	0.7	0.8	0.9	ns, max
IN input to OUT output	T_{GIO}	0.7	0.8	0.9	ns, max

CLB Switching Characteristics

Delays originating at F/G inputs vary slightly according to the input used. The values listed below are worst-case. Precise values are provided by the timing analyzer.

		Speed Grade			
Description	**Symbol**	**-6**	**-5**	**-4**	**Units**
Combinatorial Delays					
4-input function: F/G inputs to X/Y outputs	T_{ILO}	0.6	0.7	0.8	ns, max
5-input function: F/G inputs to F5 output	T_{IF5}	1.0	1.1	1.2	ns, max
5-input function: F/G inputs to X output	T_{IF5X}	1.0	1.2	1.3	ns, max
6-input function: F/G inputs to Y output via F6 MUX	T_{IF6Y}	1.2	1.4	1.6	ns, max
6-input function: F5IN input to Y output	T_{F5INY}	0.4	0.5	0.6	ns, max
Incremental delay routing through transparent latch to XQ/YQ outputs	T_{IFNCTL}	0.4	0.5	0.6	ns, max
BY input to YB output	T_{BYYB}	0.5	0.6	0.7	ns, max
Sequential Delays					
FF Clock CLK to XQ/YQ outputs	T_{CKO}	1.1	1.3	1.4	ns, max
Latch Clock CLK to XQ/YQ outputs	T_{CKLO}	0.7	0.7	0.9	ns, max
Setup Times before Clock CLK					
4-input function: F/G Inputs	T_{ICK}	1.0	1.1	1.2	ns, min
5-input function: F/G inputs	T_{IF5CK}	1.4	1.6	1.8	ns, min
6-input function: F5IN input	T_{F5INCK}	0.8	0.9	1.0	ns, min
6-input function: F/G inputs via F6 MUX	T_{IF6CK}	1.6	1.8	2.0	ns, min
BX/BY inputs	T_{DICK}	1.6	1.8	2.0	ns, min
CE input	T_{CECK}	0.8	0.9	1.0	ns, min
SR/BY inputs (synchronous)	T_{RCK}	1.3	1.5	1.7	ns, min
Hold Times after Clock CLK					
All Hold Times		0.0	0.0	0.0	ns, min
Clock CLK					
Minimum Pulse Width, High	T_{CH}	1.5	1.7	2.0	ns, min
Minimum Pulse Width, Low	T_{CL}	1.5	1.7	2.0	ns, min
Set/Reset					
Minimum Pulse Width, SR/BY inputs	T_{RPW}	2.9	3.4	3.9	ns, min
Delay from SR/BY inputs to XQ/YQ outputs (asynchronous)	T_{RQ}	1.6	1.9	2.2	ns, max
Delay from GSR to XQ/YQ outputs	T_{GSRQ}	10.0	11.0	13.0	ns, max

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.

		Speed Grade			
Description	**Symbol**	**-6**	**-5**	**-4**	**Units**
Combinatorial Delays					
F operand inputs to X via XOR	T_{OPX}	0.8	0.9	1.0	ns, max
F operand input to XB output	T_{OPXB}	1.2	1.4	1.6	ns, max
F operand input to Y via XOR	T_{OPY}	1.6	1.9	2.2	ns, max
F operand input to YB output	T_{OPYB}	1.3	1.5	1.7	ns, max
F operand input to COUT output	T_{OPCYF}	1.3	1.5	1.7	ns, max
G operand inputs to Y via XOR	T_{OPGY}	1.0	1.1	1.3	ns, max
G operand input to YB output	T_{OPGYB}	1.4	1.6	1.9	ns, max
G operand input to COUT output	T_{OPCYG}	1.4	1.6	1.8	ns, max
BX initialization input to COUT	T_{BXCY}	0.8	0.9	1.0	ns, max
CIN input to X output via XOR	T_{CINX}	0.5	0.5	0.6	ns, max
CIN input to XB	T_{CINXB}	0.1	0.1	0.1	ns, max
CIN input to Y via XOR	T_{CINY}	0.5	0.6	0.7	ns, max
CIN input to YB	T_{CINYB}	0.2	0.2	0.2	ns, max
CIN input to COUT output	T_{BYP}	0.1	0.2	0.2	ns, max
Multiplier Operation					
F1/2 operand inputs to XB output via AND	T_{FANDXB}	0.4	0.5	0.6	ns, max
F1/2 operand inputs to YB output via AND	T_{FANDYB}	0.5	0.6	0.6	ns, max
F1/2 operand inputs to COUT output via AND	T_{FANDCY}	0.5	0.5	0.6	ns, max
G1/2 operand inputs to YB output via AND	T_{GANDYB}	0.4	0.4	0.5	ns, max
G1/2 operand inputs to COUT output via AND	T_{GANDCY}	0.4	0.4	0.5	ns, max
Setup Times before Clock CLK					
CIN input to FFX	T_{CCKX}	0.8	0.9	1.1	ns, min
CIN input to FFY	T_{CCKY}	0.9	1.0	1.1	ns, min
Setup Time Adjustment					ns
Hold Times after Clock CLK					
All Hold Times		0.0	0.0	0.0	ns, min

CLB SelectRAM Switching Characteristics

Description	Symbol	Speed Grade			Units
		-6	-5	-4	
Sequential Delays					
Clock CLK to X/Y outputs (WE active)	T_{SHCKO}	2.3	2.6	3.0	ns, max
Shift-Register Mode Clock CLK to X/Y outputs		2.3	2.6	3.0	ns, max
Setup Times before Clock CLK					
F/G address inputs	T_{AS}/T_{AH}	0.6	0.7	0.8	ns, min
BX/BY data inputs (DIN)	T_{DS}/T_{DH}	1.0	1.2	1.3	ns, min
CE input (WE)	T_{WS}/T_{WH}	0.6	0.6	0.7	ns, min
Shift-Register Mode					
BX/BY data inputs (DIN)	T_{SHDICK}	0.7	0.8	0.9	ns, min
CE input (WS)	T_{SHCECK}	0.8	0.9	1.0	ns, min
Hold Times after Clock CLK					
All Hold Times		0.0	0.0	0.0	ns, min
Clock CLK					
Minimum Pulse Width, High	T_{WPH}	2.4	2.7	3.1	ns, min
Minimum Pulse Width, Low	T_{WPL}	2.4	2.7	3.1	ns, min
Minimum clock period to meet address write cycle time	T_{WC}	5.8	6.7	7.7	ns, min
Shift-Register Mode					
Minimum Pulse Width, High	T_{SRPH}	2.4	2.7	3.1	ns, min
Minimum Pulse Width, Low	T_{SRPL}	2.4	2.7	3.1	ns, min

BLOCKRAM Switching Characteristics

Description	Symbol	Speed Grade			Units
		-6	-5	-4	
Sequential Delays					
Clock CLK to DOUT output	T_{BCKO}	3.3	3.8	4.4	ns, max
Setup Times before Clock CLK					
ADDR inputs	T_{BACk}	1.2	1.4	1.6	ns, min
DIN inputs	T_{BDCK}	1.2	1.4	1.6	ns, min
EN input	T_{BECK}	2.7	3.1	3.6	ns, min
RST input	T_{BRCK}	2.5	2.9	3.3	ns, min
WEN input	T_{BWCK}	2.4	2.8	3.2	ns, min
Hold Times after Clock CLK					
All Hold Times		0.0	0.0	0.0	ns, min
Clock CLK					
Minimum Pulse Width, High	T_{BPWH}	1.5	1.7	2.0	ns, min
Minimum Pulse Width, Low	T_{BPWL}	1.5	1.7	2.0	ns, min
CLKA -> CLKB setup time for different ports	T_{BCCS}	3.0	3.5	4.0	ns, min

TBUF Switching Characteristics

Description	Symbol	Speed Grade			Units
		-6	-5	-4	
Combinatorial Delays					
IN input to OUT output	T_{IO}	0.2	0.2	0.2	ns, max
TRI input to OUT output high-impedance	T_{OFF}	0.2	0.2	0.2	ns, max
TRI input to valid data on OUT output	T_{ON}	0.2	0.2	0.2	ns, max

JTAG Test Access Port Switching Characteristics

Description	Symbol	Speed Grade			Units
		-6	-5	-4	
TMS and TDI Setup times before TCK	T_{TAPTK}	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

Virtex Pin-to-Pin Output Parameter Guidelines

Testing of switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% functionally tested. Listed below are representative values for typical pin locations and normal clock loading. Values are expressed in nanoseconds unless otherwise noted.

Global Clock Input to Output Delay for LVTTL, 12 mA, Fast Slew Rate, *with* DLL

		Speed Grade	All	-6	-5	-4	Units
Description	**Symbol**	**Device**	**Min**	**Max**	**Max**	**Max**	
LVTTL Global Clock Input to Output Delay using Output Flip-flop, 12 mA, Fast Slew Rate, *with* DLL. For data *output* with different standards, adjust the delays with the values shown in Output Delay Adjustments ($T_{OPADJUST}$).		XCV50		3.5	3.8	4.3	ns
		XCV100		3.5	3.8	4.3	ns
		XCV150		3.5	3.8	4.3	ns
		XCV200		3.5	3.8	4.3	ns
		XCV300		3.5	3.8	4.3	ns
		XCV400		3.5	3.8	4.3	ns
		XCV600		3.5	3.8	4.3	ns
		XCV800		3.5	3.8	4.3	ns
		XCV1000		3.5	3.8	4.3	ns

Notes: Listed above are representative values where one global clock input drives one vertical clock line in each accessible column, and where all accessible IOB and CLB flip-flops are clocked by the global clock net.
Output timing is measured at 50% V_{CC} threshold with 35 pF external capacitive load. For different loads, see Table 14.
DLL output jitter is already included in the timing calculation.

Global Clock Input to Output Delay for LVTTL, 12 mA, Fast Slew Rate, *without* DLL

		Speed Grade	All	-6	-5	-4	Units
Description	**Symbol**	**Device**	**Min**	**Max**	**Max**	**Max**	
LVTTL Global Clock Input to Output Delay using Output Flip-flop, 12 mA, Fast Slew Rate, *without* DLL. For data *output* with different standards, adjust the delays with the values shown in Output Delay Adjustments ($T_{OPADJUST}$).		XCV50		5.0	5.6	6.4	ns
		XCV100		5.0	5.6	6.4	ns
		XCV150		5.1	5.7	6.5	ns
		XCV200		5.1	5.7	6.5	ns
		XCV300		5.1	5.7	6.6	ns
		XCV400		5.2	5.8	6.7	ns
		XCV600		5.3	5.9	6.7	ns
		XCV800		5.3	6.0	6.9	ns
		XCV1000		5.4	6.1	7.0	ns

Notes: Listed above are representative values where one global clock input drives one vertical clock line in each accessible column, and where all accessible IOB and CLB flip-flops are clocked by the global clock net.
Output timing is measured at 50% V_{CC} threshold with 35 pF external capacitive load. For different loads, see Table 14.

Virtex Pin-to-Pin Input Parameter Guidelines

Testing of switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% functionally tested. Listed below are representative values for typical pin locations and normal clock loading. Values are expressed in nanoseconds unless otherwise noted

Global Clock Set-Up and Hold for LVTTL Standard, *with* DLL

Description	Symbol	Speed Grade / Device	-6 Min	-5 Min	-4 Min	Units
Input Setup and Hold Time Relative to Global Clock Input Signal for LVTTL Standard. For data input with different standards, adjust the setup time delay by the values shown in Input Delay Adjustments ($T_{IPOFFSET}$).						
Full Delay Global Clock and IFF, with DLL		XCV50	1.7 / -0.4	1.8 / -0.4	2.1 / -0.4	ns
		XCV100	1.7 / -0.4	1.9 / -0.4	2.1 / -0.4	ns
		XCV150	1.7 / -0.4	1.9 / -0.4	2.1 / -0.4	ns
		XCV200	1.7 / -0.4	1.9 / -0.4	2.1 / -0.4	ns
		XCV300	1.7 / -0.4	1.9 / -0.4	2.2 / -0.4	ns
		XCV400	1.7 / -0.4	1.9 / -0.4	2.2 / -0.4	ns
		XCV600	1.7 / -0.4	1.9 / -0.4	2.2 / -0.4	ns
		XCV800	1.8 / -0.4	1.9 / -0.4	2.2 / -0.4	ns
		XCV1000	1.8 / -0.4	1.9 / -0.4	2.2 / -0.4	ns

IFF = Input Flip-Flop or Latch

Note: Setup 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.
DLL output jitter is already included in the timing calculation.

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

Description	Symbol	Speed Grade / Device	-6 Min	-5 Min	-4 Min	Units
Input Setup and Hold Time Relative to Global Clock Input Signal for LVTTL Standard. For data input with different standards, adjust the setup time delay by the values shown in Input Delay Adjustments ($T_{IPOFFSET}$).						
Full Delay Global Clock and IFF, without DLL		XCV50	2.3 / 0.0	2.6 / 0.0	2.9 / 0.0	ns
		XCV100	2.3 / 0.0	2.6 / 0.0	3.0 / 0.0	ns
		XCV150	2.4 / 0.0	2.7 / 0.0	3.1 / 0.0	ns
		XCV200	2.5 / 0.0	2.8 / 0.0	3.2 / 0.0	ns
		XCV300	2.5 / 0.0	2.8 / 0.0	3.1 / 0.0	ns
		XCV400	2.6 / 0.0	2.9 / 0.0	3.3 / 0.0	ns
		XCV600	2.6 / 0.0	2.9 / 0.0	3.3 / 0.0	ns
		XCV800	2.7 / 0.0	3.1 / 0.0	3.5 / 0.0	ns
		XCV1000	2.8 / 0.0	3.2 / 0.0	3.6 / 0.0	ns

IFF = Input Flip-Flop or Latch

Note: Setup 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.

DLL Timing Parameters

Switching parameters testing is modeled after testing methods specified by MIL-M-38510/605; all devices are 100 percent functionally tested. Because of the difficulty in directly measuring many internal timing parameters, those parameters are derived from benchmark timing patterns. The following guidelines reflect worst-case values across the recommended operating conditions[1].

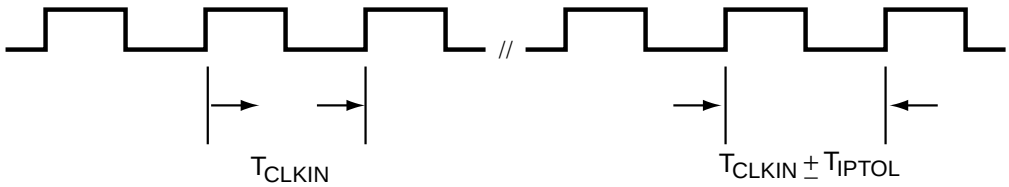
	Speed Grade	-4		-5		-6		
Description	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Units**
Input Clock Frequency (CLKDLLHF)	$F_{CLKINHF}$	60	180	60	180	60	200	MHz
Input Clock Frequency (CLKDLL)	$F_{CLKINLF}$	25	90	25	90	25	100	MHz
Input Clock Pulse Width (CLKDLLHF)	$T_{DLLPWHF}$	2.4	-	2.4	-	2.0	-	ns
Input Clock Pulse Width (CLKDLL)	$T_{DLLPWLF}$	3.0	-	3.0		2.5	-	ns

Note: All specifications correspond to Commercial Operating Temperatures (0°C to +100°C).

			CLKDLLHF		CLKDLL		
Description	**Symbol**		**Min**	**Max**	**Min**	**Max**	**Units**
Input Clock Period Tolerance	T_{IPTOL}		-	1.0	-	1.0	ns
Time Required for DLL to Acquire Lock	T_{LOCK}						
	F_{CLKIN}	> 60 MHz	-	20	-	20	µs
	F_{CLKIN}	50 - 60 MHz	-	-	-	25	µs
	F_{CLKIN}	40 - 50 MHz	-	-	-	50	µs
	F_{CLKIN}	30 - 40 MHz	-	-	-	90	µs
	F_{CLKIN}	25 - 30 MHz	-	-	-	120	µs
DLL Output Skew (Between any DLL output)	T_{SKEW}		-	±150	-	± 150	ps
DLL Output Jitter Long Term	T_{OJIT}		-	±100	-	± 100	ps
DLL Output Jitter Cycle to Cycle	T_{OJITCC}			± 60		± 60	ps

Note 1: All specifications correspond to Commercial Operating Temperatures (0°C to +100°C).

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



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

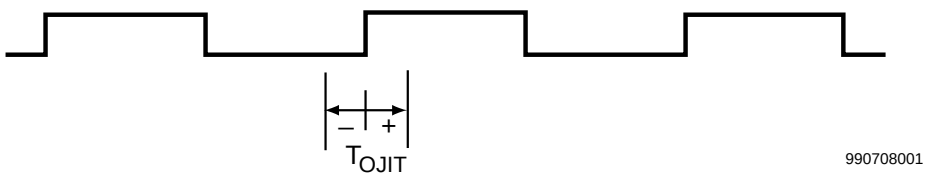


Figure 20: Frequency Tolerance and Clock Jitter

Virtex Pin Definitions

Table 15: Special Purpose Pins

Pin Name	Dedicated Pin	Direction	Description
GCK0, GCK1, GCK2, GCK3	Yes	Input	Clock input pins that connect to Global Clock Buffers. These pins become user inputs when not needed for clocks.
M0, M1, M2	Yes	Input	Mode pins are used to specify the configuration mode.
CCLK	Yes	Input or Output	The configuration Clock I/O pin: it is an input for SelectMAP and slave-serial modes, and output in master-serial mode
$\overline{\text{PROGRAM}}$	Yes	Input	Initiates a configuration sequence when asserted Low.
DONE	Yes	Bidirectional	Indicates that configuration loading is complete, and that the start-up sequence is in progress. The output may be open drain.
$\overline{\text{INIT}}$	No	Bidirectional (Open-drain)	When Low, indicates that the configuration memory is being cleared. The pin becomes a user I/O after configuration.
BUSY/ DOUT	No	Output	In SelectMAP mode, BUSY controls the rate at which configuration data is loaded. The pin becomes a user I/O after configuration unless the SelectMAP port is retained. In bit-serial modes, DOUT provides preamble and configuration data to downstream devices in a daisy-chain. The pin becomes a user I/O after configuration.
D0/DIN, D1, D2, D3, D4, D5, D6, D7	No	Input or Output	In SelectMAP mode, D0-7 are configuration data pins. These pins become user I/Os after configuration unless the SelectMAP port is retained. In bit-serial modes, DIN is the single data input. This pin becomes a user I/O after configuration.
$\overline{\text{WRITE}}$	No	Input	In SelectMAP mode, the active-low Write Enable signal. The pin becomes a user I/O after configuration unless the SelectMAP port is retained.
$\overline{\text{CS}}$	No	Input	In SelectMAP mode, the active-low Chip Select signal. The pin becomes a user I/O after configuration unless the SelectMAP port is retained.
TDI, TDO, TMS, TCK	Yes	Mixed	Boundary-scan Test-Access-Port pins, as defined in IEEE 1149.1.
DXN, DXP	Yes	N/A	Temperature-sensing diode pins. (Anode: DXP, cathode: DXN)
V_{CCINT}	Yes	Input	Power-supply pins for the internal core logic.
V_{CCO}	Yes	Input	Power-supply pins for the output drivers (subject to banking rules)
V_{REF}	No	Input	Input threshold voltage pins. Become user I/Os when an external threshold voltage is not needed (subject to banking rules).
GND	Yes	Input	Ground

Virtex Pin Outs

Pin-Out Tables

See the Xilinx WebLINX web site (http://www.xilinx.com/partinfo/databook.htm) for updates or additional pin-out information. For convenience, Table 16, Table 17 and Table 18 list the locations of special-purpose and power-supply pins. Pins not listed are user I/Os.

Table 16: Virtex Pin-out Tables (Chip-scale and QFP Packages)

Pin Name	Device	CS144	TQ144	PQ/HQ240
GCK0	All	K7	90	92
GCK1	All	M7	93	89
GCK2	All	A7	19	210
GCK3	All	A6	16	213
M0	All	M1	110	60
M1	All	L2	112	58
M2	All	N2	108	62
CCLK	All	B13	38	179
$\overline{\text{PROGRAM}}$	All	L12	72	122
DONE	All	M12	74	120
$\overline{\text{INIT}}$	All	L13	71	123
BUSY/DOUT	All	C11	39	178
D0/DIN	All	C12	40	177
D1	All	E10	45	167
D2	All	E12	47	163
D3	All	F11	51	156
D4	All	H12	59	145
D5	All	J13	63	138
D6	All	J11	65	134
D7	All	K10	70	124
$\overline{\text{WRITE}}$	All	C10	32	185
$\overline{\text{CS}}$	All	D10	33	184
TDI	All	A11	34	183
TDO	All	A12	36	181
TMS	All	B1	143	2

Table 16: Virtex Pin-out Tables (Chip-scale and QFP Packages) (Continued)

Pin Name	Device	CS144	TQ144	PQ/HQ240
TCK	All	C3	2	239
V_{CCINT}	All	A9, B6, C5, G3, G12, M5, M9, N6	10, 15, 25, 57, 84, 94, 99, 126	16, 32, 43, 77, 88, 104, 137, 148, 164, 198, 214, 225
V_{CCO}	All	Banks 0 and 1: A2, A13, D7 Banks 2 and 3: B12, G11, M13 Banks 4 and 5: N1, N7 N13 Banks 6 and 7: B2, G2, M2	Unbanked: 1, 17, 37, 55, 73, 92, 109, 128	Unbanked: 15, 30, 44, 61, 76, 90, 105, 121, 136, 150, 165, 180, 197, 212, 226, 240
V_{REF}, Bank 0 (V_{REF} 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_{REF} pins are general I/O.	XCV50	C4, D6	5, 13	218, 232
	XCV100/150	... + B4	... + 7	... + 229
	XCV200/300			... + 236
	XCV400			... + 215
	XCV600			... + 230
	XCV800			... + 222
V_{REF}, Bank 1 (V_{REF} 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_{REF} pins are general I/O.	XCV50	A10, B8	22, 30	191, 205
	XCV100/150	... + D9	... + 28	... + 194
	XCV200/300			... + 187
	XCV400			... + 208
	XCV600			... + 193
	XCV800			... + 201

Table 16: Virtex Pin-out Tables (Chip-scale and QFP Packages) (Continued)

Pin Name	Device	CS144	TQ144	PQ/HQ240
V_{REF}, Bank 2 (V_{REF} 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_{REF} pins are general I/O.	XCV50	D11, F10	42, 50	157, 171
	XCV100/150	... + D13	... + 44	... + 168
	XCV200/300			... + 175
	XCV400			... + 154
	XCV600			... + 169
	XCV800			... + 161
V_{REF}, Bank 3 (V_{REF} 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_{REF} pins are general I/O.	XCV50	H11, K12	60, 68	130, 144
	XCV100/150	... + J10	... + 66	... + 133
	XCV200/300			... + 126
	XCV400			... + 147
	XCV600			... + 132
	XCV800			... + 140
V_{REF}, Bank 4 (V_{REF} 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_{REF} pins are general I/O.	XCV50	L8, L10	79, 87	97, 111
	XCV100/150	... + N10	... + 81	... + 108
	XCV200/300			... + 115
	XCV400			... + 94
	XCV600			... + 109
	XCV800			... + 101
V_{REF}, Bank 5 (V_{REF} 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_{REF} pins are general I/O.	XCV50	L4, L6	96, 104	70, 84
	XCV100/150	... + N4	... + 102	... + 73
	XCV200/300			... + 66
	XCV400			... + 87
	XCV600			... + 72
	XCV800			... +80

Table 16: Virtex Pin-out Tables (Chip-scale and QFP Packages) (Continued)

Pin Name	Device	CS144	TQ144	PQ/HQ240
V_{REF}, Bank 6 (V_{REF} 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_{REF} pins are general I/O.	XCV50	H2, K1	116, 123	36, 50
	XCV100/150	... + J3	... + 118	... + 47
	XCV200/300			... + 54
	XCV400			... + 33
	XCV600			... + 48
	XCV800			... + 40
V_{REF}, Bank 7 (V_{REF} 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_{REF} pins are general I/O.	XCV50	D4, E1	133, 140	9, 23
	XCV100/150	... + D2	... + 138	... + 12
	XCV200/300			... + 5
	XCV400			... + 26
	XCV600			... + 11
	XCV800			... + 19
GND	All	A1, B9, B11, C7, D5, E4, E11, F1, G10, J1, J12, L3, L5, L7, L9, N12	9, 18, 26, 35, 46, 54, 64, 75, 83, 91, 100, 111, 120, 129, 136, 144,	1, 8, 14, 22, 29, 37, 45, 51, 59, 69, 75, 83, 91, 98, 106, 112, 119, 129, 135, 143, 151, 158, 166, 172, 182, 190, 196, 204, 211, 219, 227, 233

Table 17: Virtex Pin-out Tables (BGA)

Pin Name	Device	BG256	BG352	BG432	BG560
GCK0	All	Y11	AE13	AL16	AL17
GCK1	All	Y10	AF14	AK16	AJ17
GCK2	All	A10	B14	A16	D17
GCK3	All	B10	D14	D17	A17
M0	All	Y1	AD24	AH28	AJ29
M1	All	U3	AB23	AH29	AK30
M2	All	W2	AC23	AJ28	AN32
CCLK	All	B19	C3	D4	C4
$\overline{\text{PROGRAM}}$	All	Y20	AC4	AH3	AM1
DONE	All	W19	AD3	AH4	AJ5
$\overline{\text{INIT}}$	All	U18	AD2	AJ2	AH5
BUSY/DOUT	All	D18	E4	D3	D4
D0/DIN	All	C19	D3	C2	E4
D1	All	E20	G1	K4	K3
D2	All	G19	J3	K2	L4
D3	All	J19	M3	P4	P3
D4	All	M19	R3	V4	W4
D5	All	P19	U4	AB1	AB5
D6	All	T20	V3	AB3	AC4
D7	All	V19	AC3	AG4	AJ4
$\overline{\text{WRITE}}$	All	A19	D5	B4	D6
$\overline{\text{CS}}$	All	B18	C4	D5	A2
TDI	All	C17	B3	B3	D5
TDO	All	A20	D4	C4	E6
TMS	All	D3	D23	D29	B33
TCK	All	A1	C24	D28	E29
DXN	All	W3	AD23	AH27	AK29
DXP	All	V4	AE24	AK29	AJ28

Table 17: Virtex Pin-out Tables (BGA) (Continued)

Pin Name	Device	BG256	BG352	BG432	BG560
V_{CCINT} (V_{CCINT} pins are listed incrementally. Connect all pins listed for both the required device and all smaller devices listed in the same package.)	XCV50/100/150/200	C10, D6, D15, F4, F17, L3, L18, R4, R17, U6, U15, V10	A20, C14, D10, J24, K4, P2, P25, V24, W2, AC10, AE14, AE19		
	XCV300		... + B16, D12, L1, L25, R23, T1, AF11, AF16	A10, A17, B23, C14, C19, K3, K29, N2, N29, T1, T29, W2, W31, AB2, AB30, AJ10, AJ16, AK13, AK19, AK22	
	XCV400/600			... + B26, C7, F1, F30, AE29, AF1, AH8, AH24	A21, B14, B18, B28, C24, E9, E12, F2, H30, J1, K32, N1, N33, U5, U30, Y2, Y31, AD2, AD32, AG3, AG31, AK8, AK11, AK17, AK20, AL14, AL27, AN25
	XCV800/1000				... + B12, C22, M3, N29, AB2, AB32, AJ13, AL22,
V_{CCO}, Bank 0	All	D7, D8	A17, B25, D19	A21, C29, D21	A22, A26, A30, B19, B32
V_{CCO}, Bank 1	All	D13, D14	A10, D7, D13	A1, A11, D11	A10, A16, B13, C3, E5
V_{CCO}, Bank 2	All	G17, H17	B2, H4, K1	C3, L1, L4	B2, D1, H1, M1, R2
V_{CCO}, Bank 3	All	N17, P17	P4, U1, Y4	AA1, AA4, AJ3	V1, AA2, AD1, AK1, AL2

Table 17: Virtex Pin-out Tables (BGA) (Continued)

Pin Name	Device	BG256	BG352	BG432	BG560
V_{CCO}, Bank 4	All	U13, U14	AC8, AE2, AF10	AH11, AL1, AL11	AM2, AM15, AN4, AN8, AN12
V_{CCO}, Bank 5	All	U7, U8	AC14, AC20, AF17	AH21, AJ29, AL21	AL31, AM21, AN18, AN24, AN30
V_{CCO}, Bank 6	All	N4, P4	U26, W23, AE25	AA28, AA31, AL31	W32, AB33, AF33, AK33, AM32
V_{CCO}, Bank 7	All	G4, H4	G23, K26, N23	A31, L28, L31	C32, D33, K33, N32, T33
V_{REF}, 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_{REF} pins are general I/O.	XCV50	A8, B4			
	XCV100/150	... + A4	A16,C19, C21		
	XCV200/300		... + D21	B19, D22, D24, D26	
	XCV400		... + B15	... + C18	A19, D20, D26, E23, E27
	XCV600			... + C24	... + E24
	XCV800			... + B21	... + E21
	XCV1000				... + D29
V_{REF}, Bank 1 (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_{REF} pins are general I/O.	XCV50	A17, B12			
	XCV100/150	... + B15	B6, C9, C12		
	XCV200/300		... + D6	A13, B7, C6, C10	
	XCV400		... + C13	... + B15	A6, D7, D11, D16, E15
	XCV600			... + D10	... + D10
	XCV800			... + B12	... + D13
	XCV1000				... + E7

Table 17: Virtex Pin-out Tables (BGA) (Continued)

Pin Name	Device	BG256	BG352	BG432	BG560
V_{REF}, Bank 2 (V_{REF} 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_{REF} pins are general I/O.	XCV50	C20, J18			
	XCV100/150	... + F19	E2, H2, M4		
	XCV200/300		... + D2	E2, G3, J2, N1	
	XCV400		... + M1	... + R3	G5, H4, L5, P4, R1
	XCV600			... + H1	... + K5
	XCV800			... + M3	... + N5
	XCV1000				... + B3
V_{REF}, Bank 3 (V_{REF} 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_{REF} pins are general I/O.	XCV50	M18, V20			
	XCV100/150	... + R19	R4, V4, Y3		
	XCV200/300		... + AC2	V2, AB4, AD4, AF3	
	XCV400		. + R1	... + U2	V4, W5, AD3, AE5, AK2
	XCV600			... + AC3	... + AF1
	XCV800			... + Y3	... + AA4
	XCV1000				... + AH4
V_{REF}, Bank 4 (V_{REF} 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_{REF} pins are general I/O.	XCV50	V12, Y18			
	XCV100/150	... + W15	AC12, AE5, AE8,		
	XCV200/300		... + AE4	AJ7, AL4, AL8, AL13	
	XCV400		... + AF12	... + AK15	AL7, AL10, AL16, AM4, AM14
	XCV600			... + AK8	... + AL9
	XCV800			... + AJ12	... + AK13
	XCV1000				... + AN3

Table 17: Virtex Pin-out Tables (BGA) (Continued)

Pin Name	Device	BG256	BG352	BG432	BG560
V_{REF}, Bank 5 (V_{REF} 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_{REF} pins are general I/O.	XCV50	V9, Y3			
	XCV100/150	... + W6	AC15, AC18, AD20		
	XCV200/300		... + AE23	AJ18, AJ25, AK23, AK27	
	XCV400		... + AF15	... + AJ17	AJ18, AJ25, AL20, AL24, AL29
	XCV600			... + AL24	... + AM26
	XCV800			... + AH19	... + AN23
	XCV1000				... + AK28
V_{REF}, Bank 6 (V_{REF} 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_{REF} pins are general I/O.	XCV50	M2, R3			
	XCV100/150	...+ T1	R24, Y26, AA25,		
	XCV200/300		... + AD26	V28, AB28, AE30, AF28	
	XCV400		... + P24	... + U28	V29, Y32, AD31, AE29, AK32
	XCV600			... + AC28	... + AE31
	XCV800			... + Y30	... + AA30
	XCV1000				... + AH30
V_{REF}, Bank 7 (V_{REF} 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_{REF} pins are general I/O.	XCV50	G3, H1			
	XCV100/150	... + D1	D26, G26, L26		
	XCV200/300		... + E24	F28, F31, J30, N30	
	XCV400		... + M25	... + R31	E31, G31, K31, P31, T31
	XCV600			... + J28	... + H32
	XCV800			... + M28	... + L33
	XCV1000				... + D31

Table 17: Virtex Pin-out Tables (BGA) (Continued)

Pin Name	Device	BG256	BG352	BG432	BG560
GND	All	C3, C18, D4, D5, D9, D10, D11, D12, D16, D17. E4, E17, J4, J9, J10, J11, J12, J17, K4, K9, K10, K11, K12, K17, L4, L9, L10, L11, L12, L17, M4, M9, M10, M11, M12, M17, T4, T17, U4, U5, U9, U10, U11, U12, U16, U17, V3, V18	A1, A2, A5, A8, A14, A19, A22, A25, A26, B1, B26, E1, E26, H1, H26, N1, P26, W1, W26, AB1, AB26, AE1, AE26, AF1, AF2, AF5, AF8, AF13, AF19, AF22, AF25, AF26	A2, A3, A7, A9, A14, A18, A23, A25, A29, A30, B1, B2, B30, B31, C1, C31, D16, G1, G31, J1, J31, P1, P31, T4, T28, V1, V31, AC1, AC31, AE1, AE31, AH16, AJ1, AJ31, AK1, AK2, AK30, AK31, AL2, AL3, AL7, AL9 AL14, AL18 AL23, AL25, AL29, AL30	A1, A7, A12, A14, A18, A20, A24, A29, A32, A33, B1, B6, B9, B15, B23, B27, B31, C2, E1, F32, G2, G33, J32, K1, L2, M33, P1, P33, R32, T1, V33, W2, Y1, Y33, AB1, AC32, AD33, AE2, AG1, AG32, AH2, AJ33, AL32, AM3, AM7, AM11, AM19, AM25, AM28, AM33, AN1, AN2, AN5, AN10, AN14, AN16, AN20, AN22, AN27, AN33
No Connect					C31, AC2, AK4, AL3

Table 18. Virtex Pin-out 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
$\overline{\text{PROGRAM}}$	All	P15	W20	AA22	AT5
DONE	All	R14	Y19	AB21	AU5
$\overline{\text{INIT}}$	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
$\overline{\text{WRITE}}$	All	C13	A20	C22	B4
$\overline{\text{CS}}$	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 18. Virtex Pin-out Tables (Fine-pitch BGA) (Continued)

Pin Name	Device	FG256	FG456	FG676	FG680
V_{CCINT}	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_{CCO}, Bank 0	All	E8, F8	F7, F8, F9, F10 G10, G11	H9, H10, H11, H12, J12, J13	E26, E27, E29, E30, E33, E34
V_{CCO}, Bank 1	All	E9, F9	F13, F14, F15, F16, G12, G13	H15, H16, H17, H18, J14, J15	E6, E7, E10, E11, E13, E14
V_{CCO}, Bank 2	All	H11, H12	G17, H17, J17, K16, K17, L16	J19, K19, L19, M18, M19, N18	F5, G5, K5, L5, N5, P5
V_{CCO}, Bank 3	All	J11, J12	M16, N16, N17, P17, R17, T17	P18, R18, R19, T19, U19, V19	AF5, AG5, AN5, AK5, AJ5, AP5
V_{CCO}, Bank 4	All	L9. M9	T12, T13, U13, U14, U15, U16,	V14, V15, W15, W16, W17, W18	AR6, AR7, AR10, AR11, AR13, AR14
V_{CCO}, Bank 5	All	L8, M8	T10, T11, U7, U8, U9, U10	V12, V13, W9,W10, W11, W12	AR26, AR27, AR29, AR30, AR33, AR34
V_{CCO}, Bank 6	All	J5, J6	M7, N6, N7, P6, R6, T6	P9, R8, R9, T8, U8, V8	AF35, AG35, AJ35, AK35, AN35, AP35
V_{CCO}, Bank 7	All	H5, H6	G6, H6, J6, K6, K7, L7	J8, K8, L8, M8, M9, N9	F35, G35, K35, L35, N35, P35

Table 18. Virtex Pin-out Tables (Fine-pitch BGA) (Continued)

Pin Name	Device	FG256	FG456	FG676	FG680
V_{REF}, 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_{REF} pins are general I/O.	XCV50	B4, B7			
	XCV100/150	... + C6	A9, C6, E8		
	XCV200/300	... + A3	... + B4		
	XCV400			A12, C11, D6, E8, G10	
	XCV600			...+B7	A33, B28, B30, C23, C24, D33
	XCV800			...+B10	...+A26
	XCV1000				...+D34
V_{REF}, Bank 1 (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_{REF} pins are general I/O.	XCV50	B9, C11			
	XCV100/150	... + E11	A18, B13, E14		
	XCV200/300	... + A14	... + A19		
	XCV400			A14, C20, C21, D15, G16	
	XCV600			...+B19	B6, B8, B18, D11, D13, D17
	XCV800			...+A17	...+B14
	XCV1000				...+B5
V_{REF}, Bank 2 (V_{REF} 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_{REF} pins are general I/O.	XCV50	F13, H13			
	XCV100/150	... + F14	F21, H18, K21		
	XCV200/300	... + E13	... + D22		
	XCV400			F24, H23, K20, M23, M26	
	XCV600			...+G26	G1, H4, J1, L2, V5, W3
	XCV800			...+K25	...+N1
	XCV1000				...+D2

Table 18. Virtex Pin-out Tables (Fine-pitch BGA) (Continued)

Pin Name	Device	FG256	FG456	FG676	FG680
V_{REF}, Bank 3 (V_{REF} 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_{REF} pins are general I/O.	XCV50	K16, L14			
	XCV100/150	... + L13	N21, R19, U21		
	XCV200/300	... + M13	... + U20		
	XCV400			R23, R25, U21, W22, W23	
	XCV600			...+W26	AC1, AJ2, AK3, AL4, AR1, Y1
	XCV800			...+U25	...+AF3
	XCV1000				...+AP4
V_{REF}, Bank 4 (V_{REF} 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_{REF} pins are general I/O.	XCV50	P9, T12			
	XCV100/150	... + T11	AA13, AB16, AB19		
	XCV200/300	... + R13	... + AB20		
	XCV400			AC15, AD18, AD21, AD22, AF15	
	XCV600			...AF20	AT19, AU7, AU17, AV8, AV10, AW11
	XCV800			...AF17	...+AV14
	XCV1000				...+AU6
V_{REF}, Bank 5 (V_{REF} 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_{REF} pins are general I/O.	XCV50	T4, P8			
	XCV100/150	... + R5	W8, Y10, AA5		
	XCV200/300	... + T2	... + Y6		
	XCV400			AA10, AB8, AB12, AC7, AF12	
	XCV600			...+AF8	AT27, AU29, AU31, AV35, AW21, AW23
	XCV800			...+AE10	...+AT25
	XCV1000				...+AV36

Table 18. Virtex Pin-out Tables (Fine-pitch BGA) (Continued)

Pin Name	Device	FG256	FG456	FG676	FG680
V_{REF}, Bank 6 (V_{REF} 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_{REF} pins are general I/O.	XCV50	J3, N1			
	XCV100/150	... + M1	N2, R4, T3		
	XCV200/300	... + N2	... + Y1		
	XCV400			AB3, R1, R4, U6, V5	
	XCV600			...+Y1	AB35, AD37, AH39, AK39, AM39, AN36
	XCV800			...+U2	...+AE39
	XCV1000				...+AT39
V_{REF}, Bank 7 (V_{REF} 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_{REF} pins are general I/O.	XCV50	C1, H3			
	XCV100/150	... + D1	E2, H4, K3		
	XCV200/300	... + B1	... + D2		
	XCV400			F4, G4, K6, M2, M5	
	XCV600			...+H1	E38, G38, L36, N36, U36, U38
	XCV800			...+K1	...+N38
	XCV1000				...+F36

Table 18. Virtex Pin-out Tables (Fine-pitch BGA) (Continued)

Pin Name	Device	FG256	FG456	FG676	FG680
GND	All	A1, A16, B2, B15, F6, F7, F10, F11, G6, G7, G8, G9, G10, G11, H7, H8, H9, H10, J7, J8, J9, J10, K6, K7, K8, K9, K10, K11, L6, L7, L10, L11, R2, R15, T1, T16	A1, A22, B2, B21, C3, C20, J9, J10, J11, J12, J13, J14, K9, K10, K11, K12, K13, K14, L9, L10, L11, L12, L13, L14, M9, M10, M11, M12, M13, M14, N9, N10, N11, N12, N13, N14, P9, P10, P11, P12, P13, P14, Y3, Y20, AA2, AA21, AB1, AB22	A1, A26, B2, B9, B14, B18, B25, C3, C24, D4, D23, E5, E22, J2, J25, K10, K11, K12, K13, K14, K15, K16, K17, L10, L11, L12, L13, L14, L15, L16, L17, M10, M11, M12, M13, M14, M15, M16, M17, N2, N10, N11, N12, N13, N14, N15, N16, N17, P10, P11, P12, P13, P14, P15, P16, P17, P25, R10, R11, R12, R13, R14, R15, R16, R17, T10, T11, T12, T13, T14, T15, T16, T17, U10, U11, U12, U13, U14, U15, U16, U17, V2, V25, AB5, AB22, AC4, AC23, AD3, AD24, AE2, AE9, AE13, AE18, AE25, AF1, AF26	A1, A2, A3, A37, A38, A39, AA5, AA35, AH4, AH5, AH35, AH36, AR5, AR12, AR19, AR20, AR21, AR28, AR35, AT4, AT12, AT20, AT28, AT36, AU1, AU3, AU20, AU37, AU39, AV1, AV2, AV38, AV39, AW1, AW2, AW3, AW37, AW38, AW39, B1, B2, B38, B39, C1, C3, C20, C37, C39, D4, D12, D20, D28, D36, E5, E12, E19, E20, E21, E28, E35, M4, M5, M35, M36, W5, W35, Y3, Y4, Y5, Y35, Y36, Y37

Table 18. Virtex Pin-out Tables (Fine-pitch BGA) (Continued)

Pin Name	Device	FG256	FG456	FG676	FG680
No Connect (No-connect pins are listed incrementally. All pins listed for both the required device and all larger devices listed in the same package are no connects.)	XCV800			A2, A3, A15, A25, B1, B6, B11, B16, B21, B24, B26, C1, C2, C25, C26, F2, F6, F21, F25, L2, L25, N25, P2, T2, T25, AA2, AA6, AA21, AA25, AD1, AD2, AD25, AE1, AE3, AE6, AE11, AE14, AE16, AE21, AE24, AE26, AF2, AF24, AF25	
	XCV600			...	
	XCV400			...+ A9, A10, A13, A16, A24, AC1, AC25, AE12, AE15, AF3, AF10, AF11, AF13, AF14, AF16, AF18, AF23, B4, B12, B13, B15, B17, D1, D25, H26, J1, K26, L1, M1, M25, N1, N26, P1, P26, R2, R26, T1, T26, U26, V1	
	XCV300		D4, D19, W4, W19		
	XCV200		... + A2, A6, A12, B11, B16, C2, D1, D18, E17, E19, G2, G22, L2, L19, M2, M21, R3, R20, U3, U18, Y22, AA1, AA3, AA11, AA16, AB7, AB12, AB21,	...	

Table 18. **Virtex Pin-out Tables (Fine-pitch BGA) (Continued)**

Pin Name	Device	FG256	FG456	FG676	FG680
	XCV150		... + A13. A14, C8, C9, E13, F11, H21, J1, J4 K2, K18, K19, M17, N1, P1, P5, P22, R22, W13, W15, AA9, AA10, AB8, AB14		

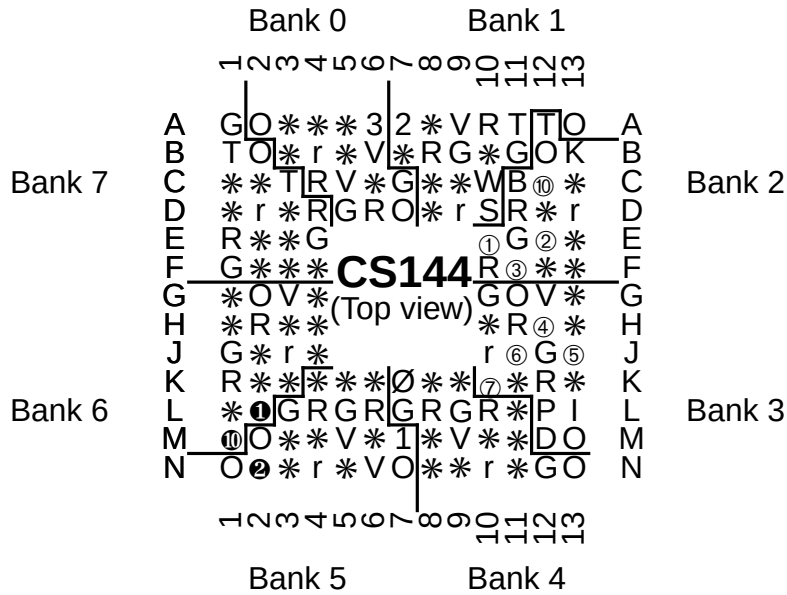
Pin-Out Diagrams

The following diagrams, page 59 through page 69, illustrate the locations of special-purpose pins on Virtex FPGAs. Table 19 lists the symbols used in these diagrams. The diagrams also show I/O-bank boundaries.

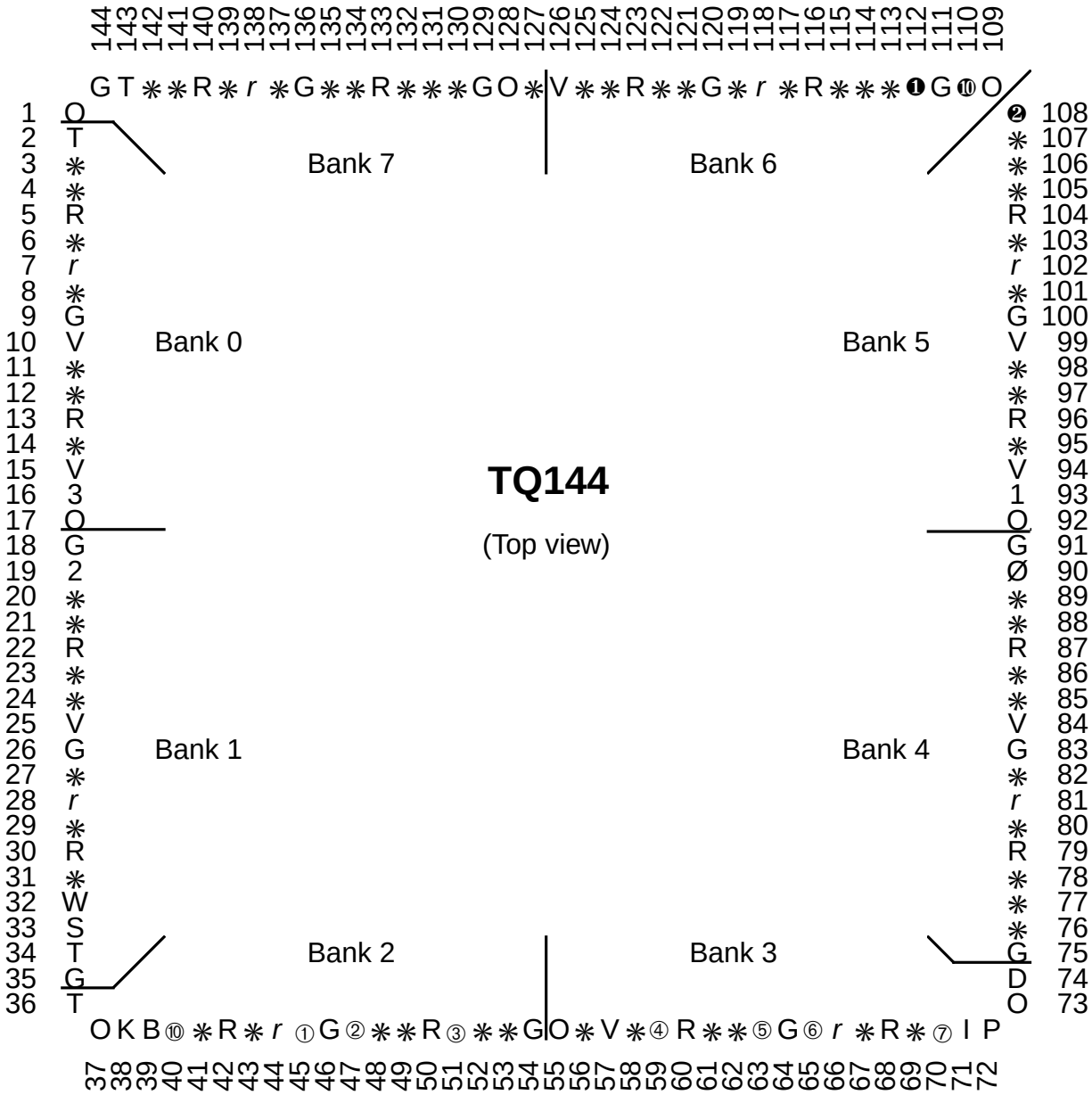
Table 19: Pin-out Diagram Symbols

Symbol	Pin Function
✳	General I/O
❄	Device-dependent general I/O, n/c on smaller devices
V	V_{CCINT}
v	Device-dependent V_{CCINT}, n/c on smaller devices
O	V_{CCO}
R	V_{REF}
r	Device-dependent V_{REF}, remains I/O on smaller devices
G	Ground
Ø, 1, 2, 3	Global Clocks
⓾, ❶, ❷	M0, M1, M2
⑩, ①, ②, ③, ④, ⑤, ⑥, ⑦	D0/DIN, D1, D2, D3, D4, D5, D6, D7
B	DOUT/BUSY
D	DONE
P	$\overline{\text{PROGRAM}}$
I	$\overline{\text{INIT}}$
K	CCLK
W	$\overline{\text{WRITE}}$
S	$\overline{\text{CS}}$
T	Boundary-scan Test Access Port
+	Temperature diode, anode
–	Temperature diode, cathode
n	No connect

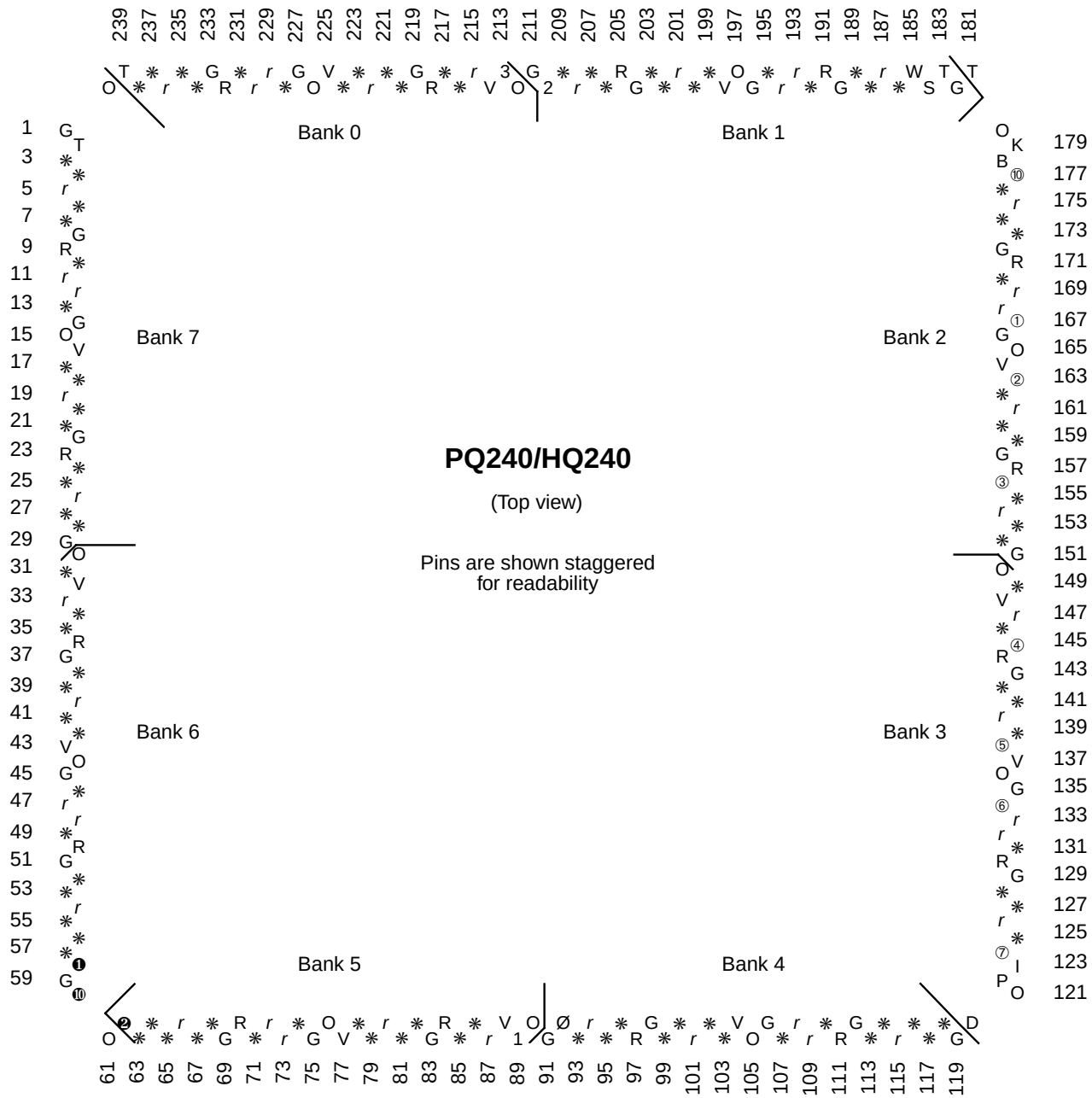
CS144 Pin-out Diagram



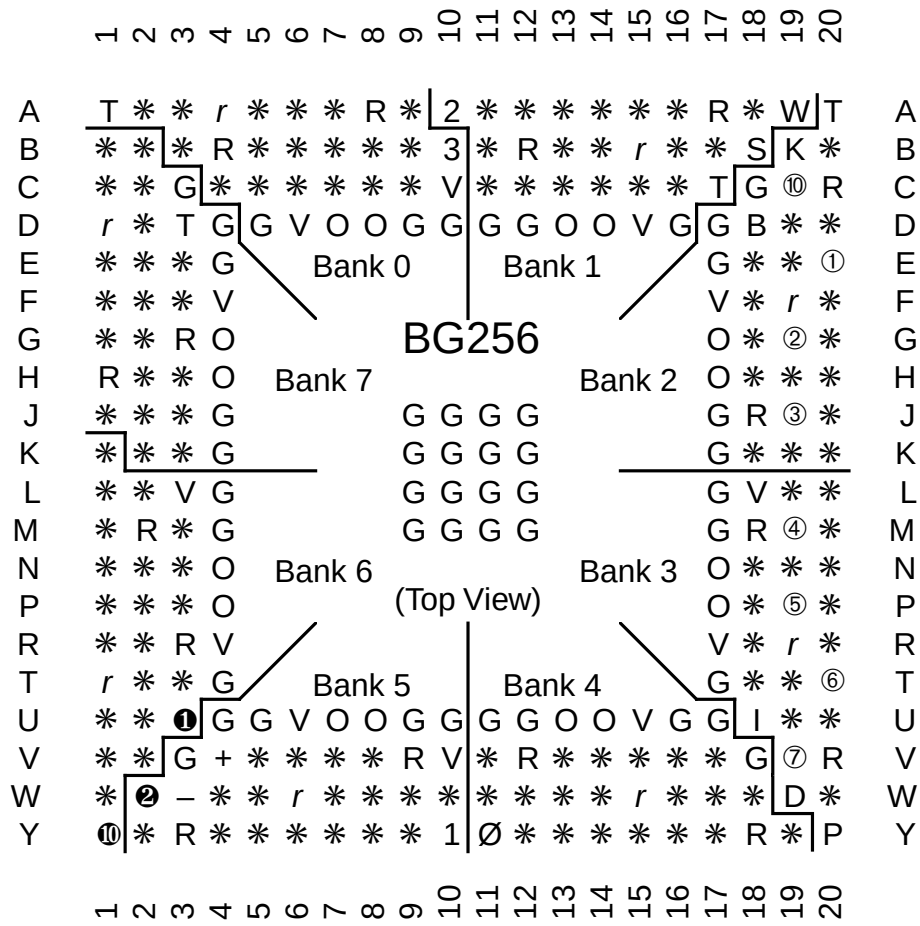
TQ144 Pin-out Diagram



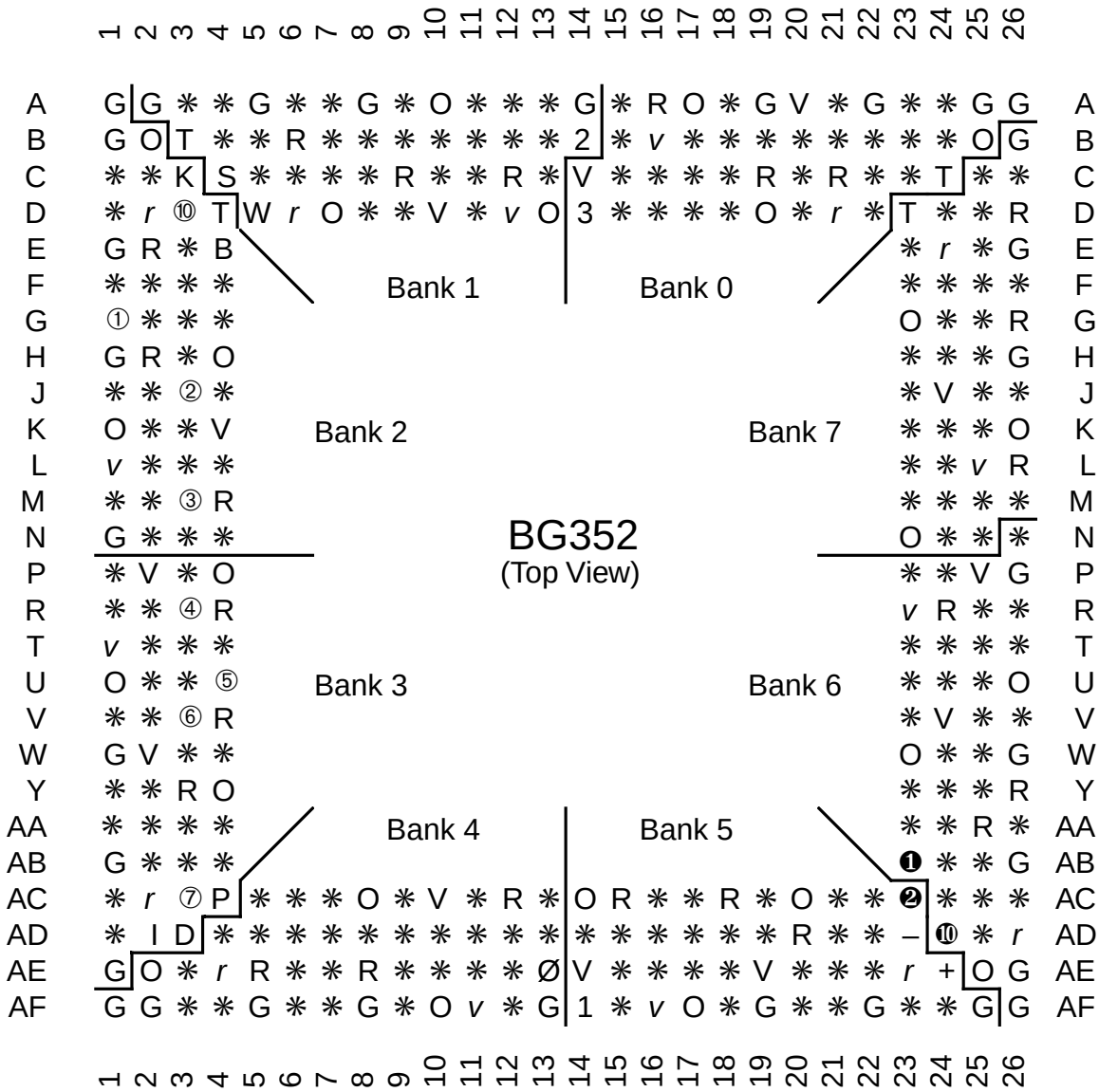
PQ240/HQ240 Pin-out Diagram



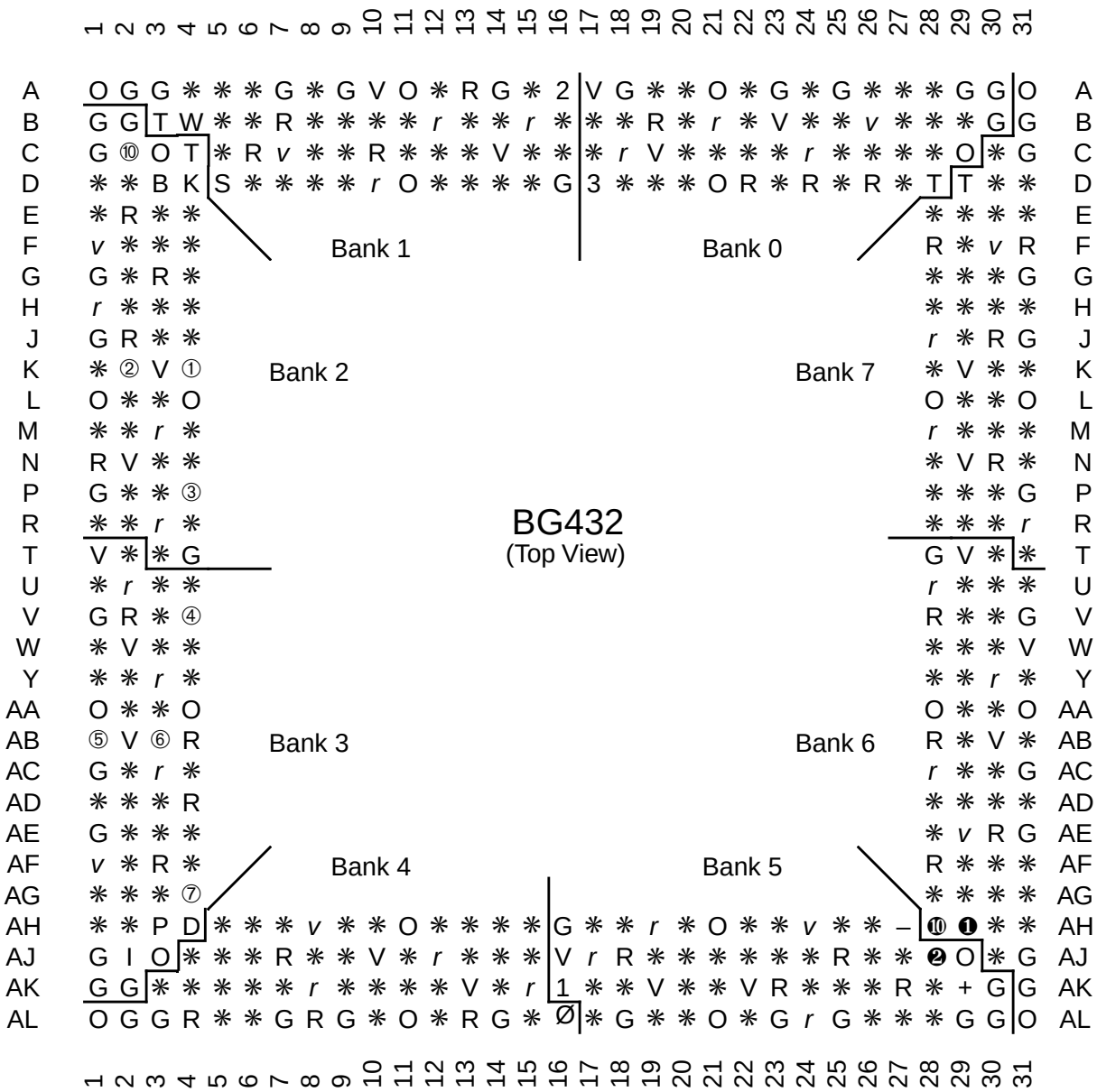
BG256 Pin-out Diagram



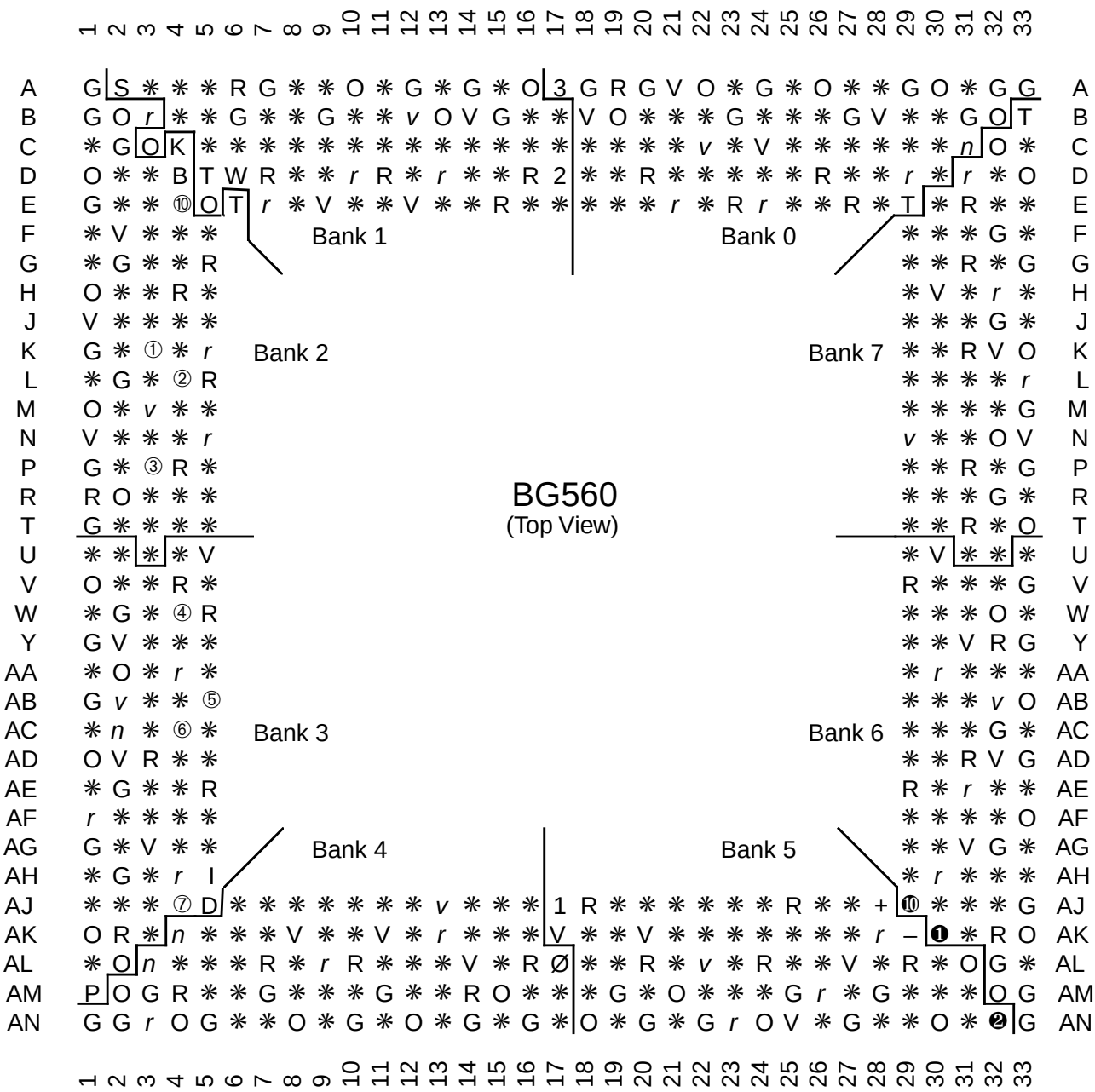
BG352 Pin-out Diagram



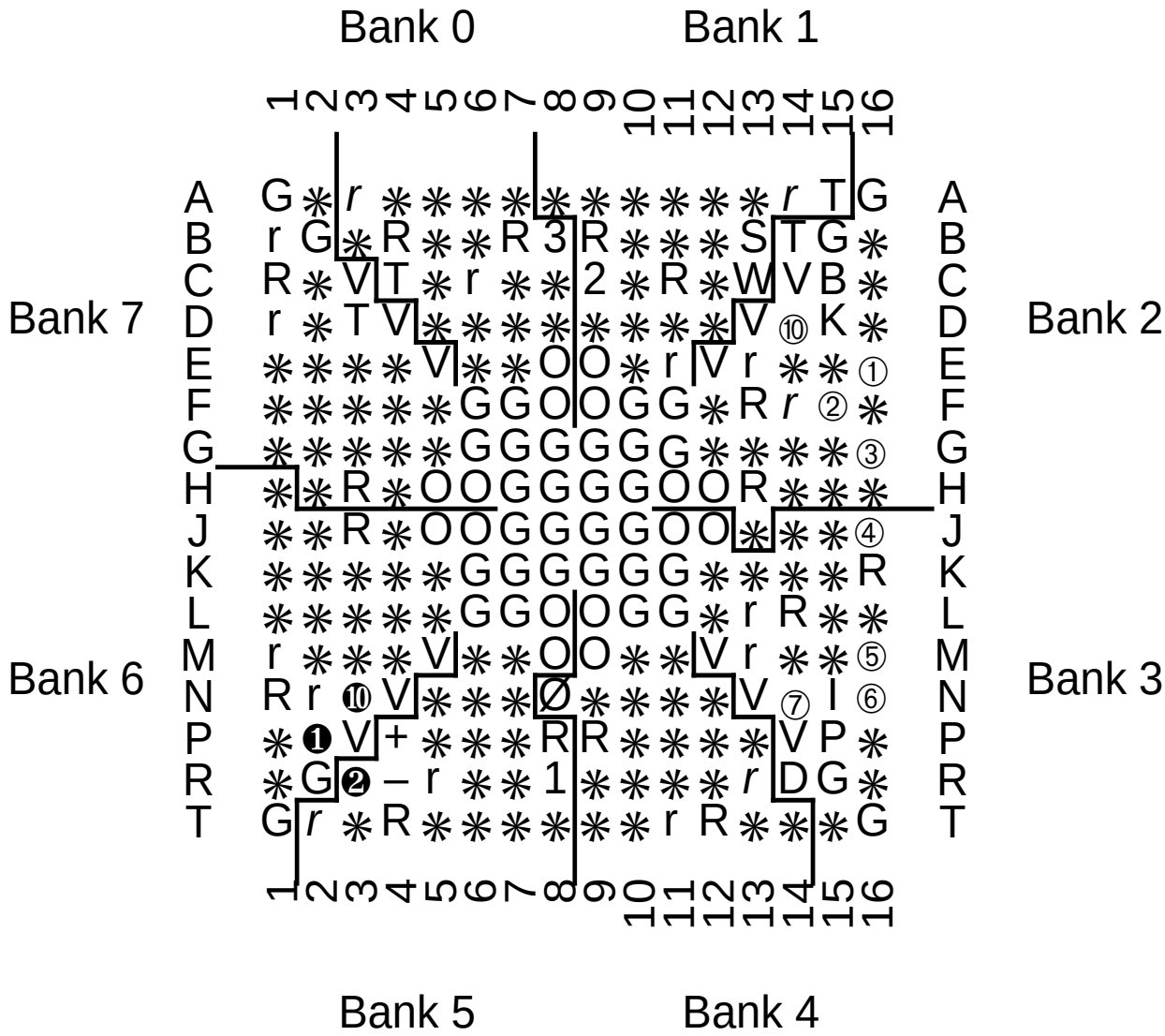
BG432 Pin-out Diagram



BG560 Pin-out Diagram



FG256 Pin-out Diagram

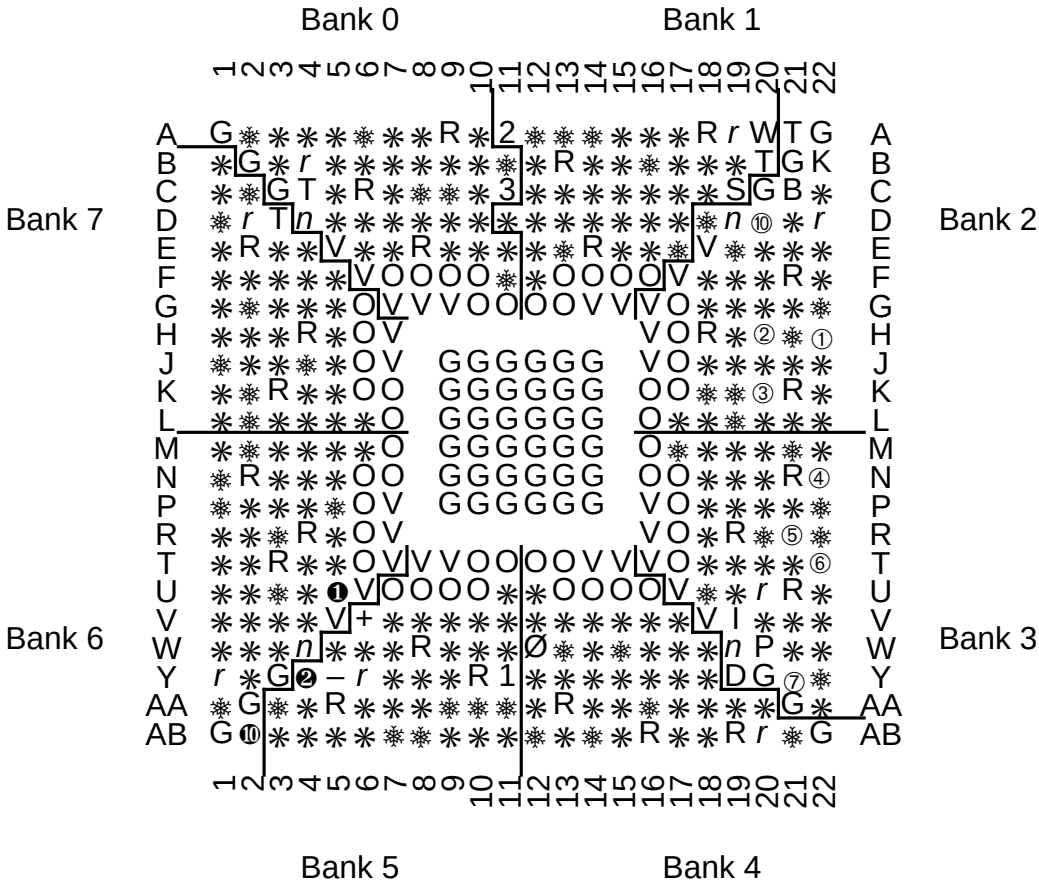


FG256

(Top view)

Note: Packages FG256, FG456, and FG676 are layout compatible.

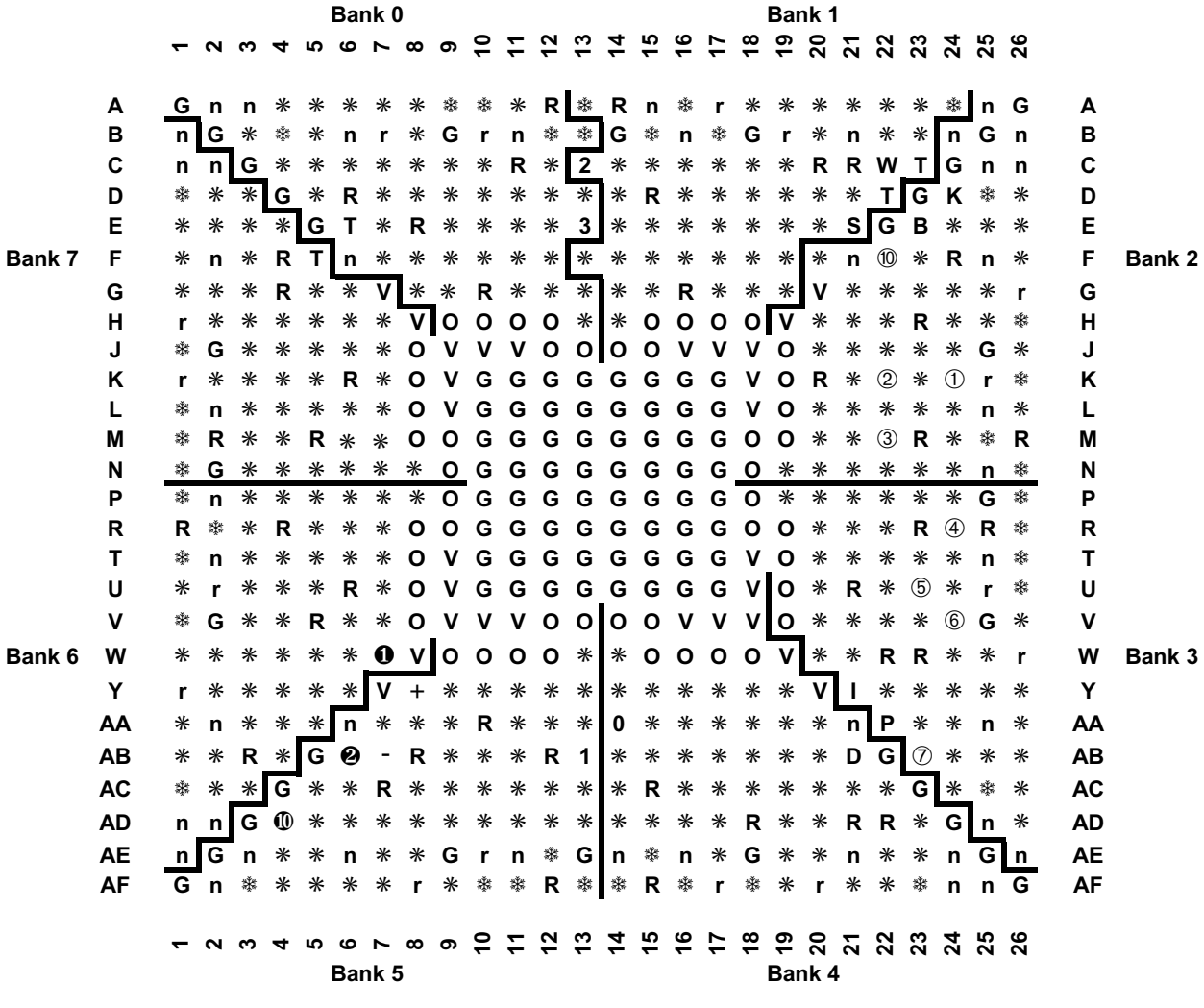
FG456 Pin-out Diagram



FG456
(Top view)

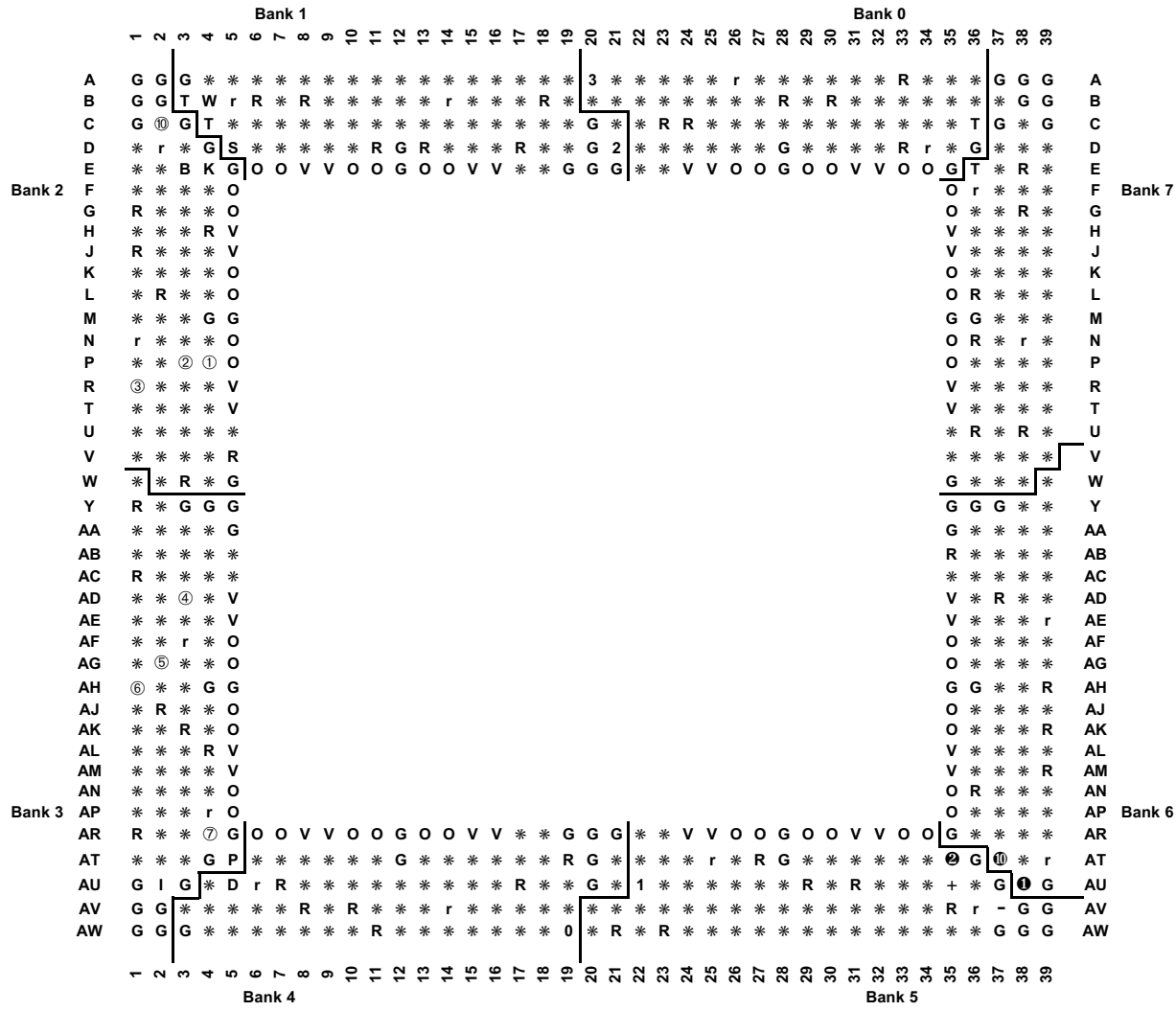
Note: Packages FG256, FG456, and FG676 are layout compatible.

FG676 Pinout Diagram



Note: Packages FG256, FG456, and FG676 are layout compatible.

FG680 Pin-out Diagram



fg680_12

Virtex Device/Package Combinations and Maximum I/O

Package	Maximum User I/O (excluding dedicated clock pins)								
	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							
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

Example: XCV300 -6 PQ 240 C

- Device Type
- Speed Grade
  - -4
  - -5
  - -6
- Temperature Range
  - C = Commercial (T_J = 0°C to +85°C)
  - I = Industrial (T_J = -40°C to +100°C)
- Number of Pins
- Package Type
  - BG = Ball Grid Array
  - FG = Fine-pitch Ball Grid Array
  - PQ = Plastic Quad Flat Pack
  - HQ = High Heat Dissipation QFP
  - TQ = Thin Quad Flat Pack
  - CS = Chip-scale Package

Revision Control

Version	Description
1.0 (11/98)	Initial document release.
1.2 (1/99)	Updated package drawings and specs.
1.3 (2/99)	Update of package drawings, updated specifications.
1.4 (5/99)	Addition of package drawings and specifications.
1.5 (5/99)	Replaced FG 676 & FG680 package drawings.
1.6 (7/99)	Changed Boundary Scan Information and changed Figure 11, Boundary Scan Bit Sequence. Updated IOB Input & Output delays. Added Capacitance info for different IO 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 LVTTL, 12 mA, Fast Slew Rate. Added IOB Input Switching Characteristics Standard Adjustments.

XILINX®
The Programmable Logic Company℠

Headquarters

Xilinx, Inc.
2100 Logic Drive
San Jose, CA 95124
U.S.A.

Tel: 1 (800) 255-7778
or 1 (408) 559-7778
Fax: 1 (408) 559-7114

Net: hotline@xilinx.com
Web: http://www.xilinx.com

North America

Irvine, California
Tel: (714) 727-0780

Englewood, Colorado
Tel: (303) 220-7541

Sunnyvale, California
Tel: (408) 245-9850

Schaumburg, Illinois
Tel: (847) 605-1972

Nashua, New Hampshire
Tel: (603) 891-1098

Raleigh, North Carolina
Tel: (919) 846-3922

West Chester, Pennsylvania
Tel: (610) 430-3300

Dallas, Texas
Tel: (972) 960-1043

Europe

Xilinx Sarl
Jouy en Josas, France
Tel: (33) 1-34-63-01-01
Net: frhelp@xilinx.com

Xilinx GmbH
München, Germany
Tel: (49) 89-93088-0
Net: dlhelp@xilinx.com

Xilinx, Ltd.
Byfleet, United Kingdom
Tel: (44) 1-932-349401
Net: ukhelp@xilinx.com

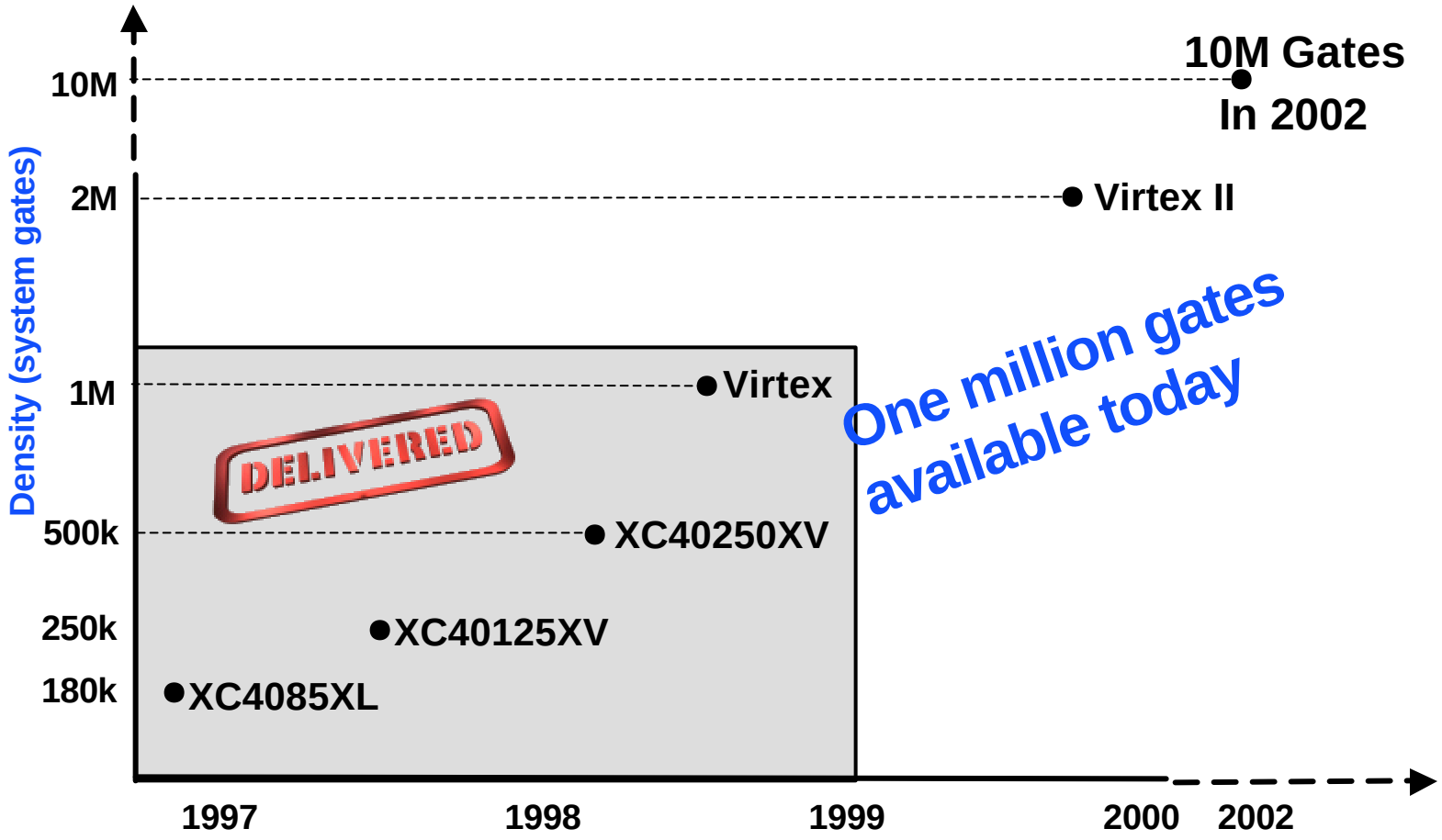
Japan

Xilinx, K.K.
Tokyo, Japan
Tel: (81) 3-3297-9191
Net: jhotline@xilinx.com

Asia Pacific

Xilinx Asia Pacific
Hong Kong
Tel: (852) 2424-5200
Net: hongkong@xilinx.com

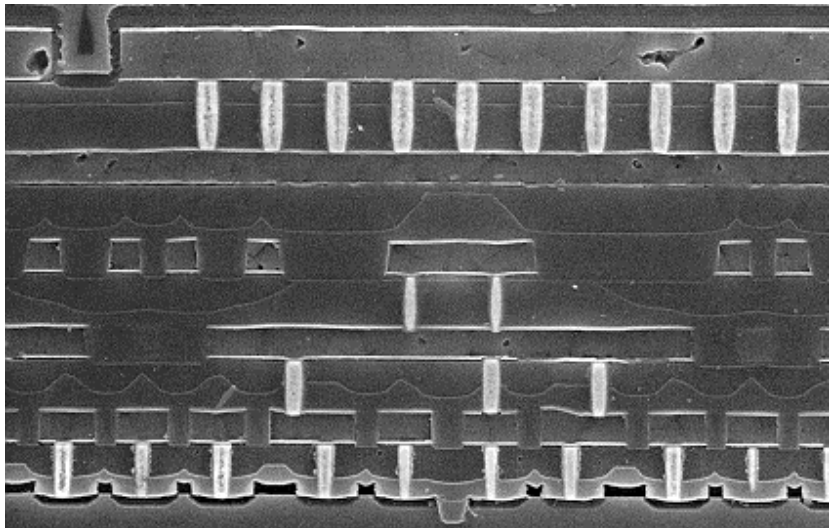
Virtex FPGAs Shipping Now



Continuing to Expand the FPGA Space

XILINX®

Process Leadership Delivers New Records



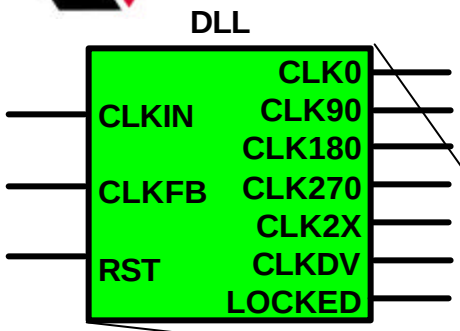
- **0.22 micron process**
  - One Million Gates
    - Four times nearest competitor
  - 27,648 Logic Cells
    - More than double nearest competitor
  - 75 million transistors
    - More than 10 times complexity of Pentium II
- **Five Metal Layers**
  - Massive Segmented Routing
    - predictably fast design

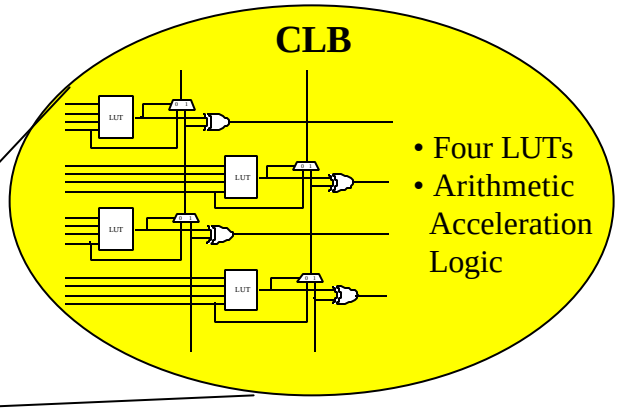
Virtex Architecture Designed to Solve System Level Design Problems

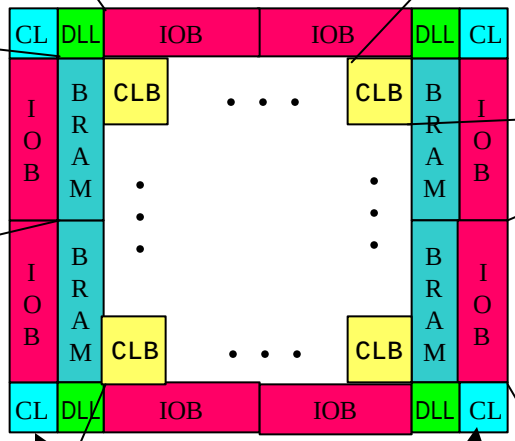
Extensive customer interviews exposed design challenges

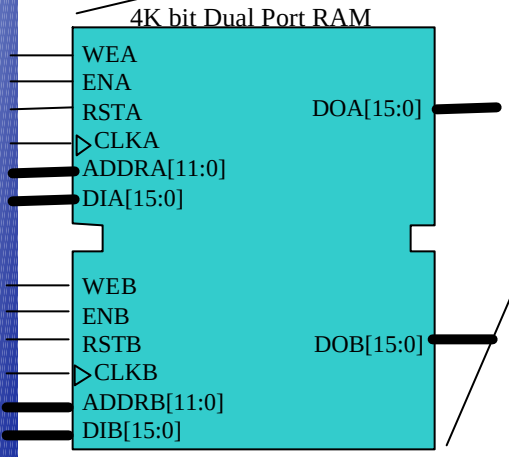
- System Timing
  - Chip to chip performance limits system speeds
  - Clock skew is the number one killer of system speed
- System Memory
  - Memory bandwidth is always key
  - Memory requirements vary in size and depth
- System Interfaces
  - Process technology leads to mixed voltage systems
  - High performance, low power signal standards emerging
- System Integration
  - Intellectual property is critical for high density design
  - Intellectual property must drop in easily

Virtex Technology







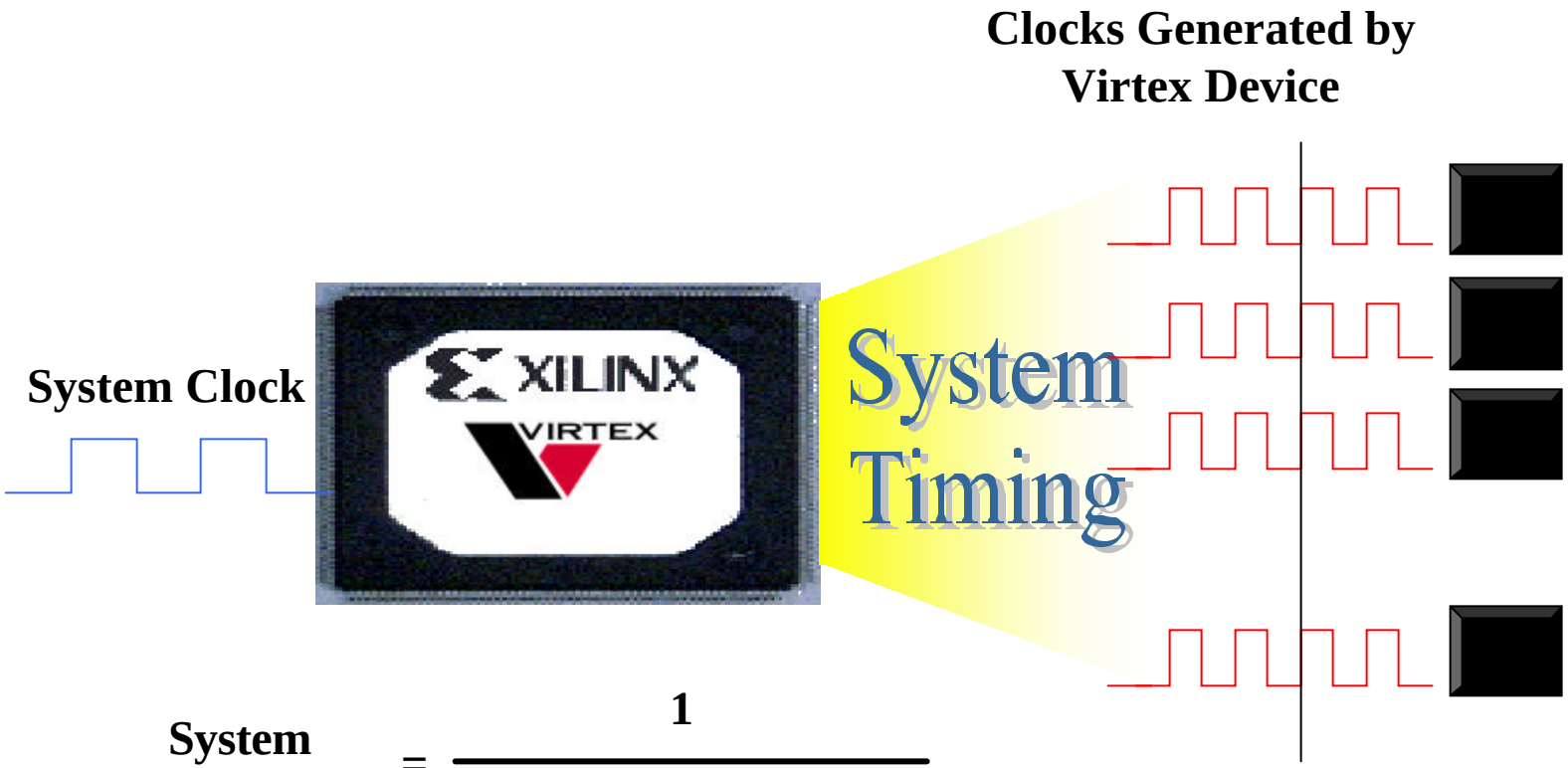


I/O Standard	Application
LVTTL	General Purpose
LVCMOS2	
PCI 33MHz 3.3V	
PCI 33MHz 5.0V	PCI
PCI 66MHz 3.3V	
GTL	Back-Plane
GTL+	
HSTL-I	
HSTL-III	Hitachi SRAM
SSTL3-I	
SSTL3-II	SDRAM
SSTL2-I	
CTT	Memory
AGP	Graphics

Thermal Management

SelectMAP
Advanced Configuration Technology

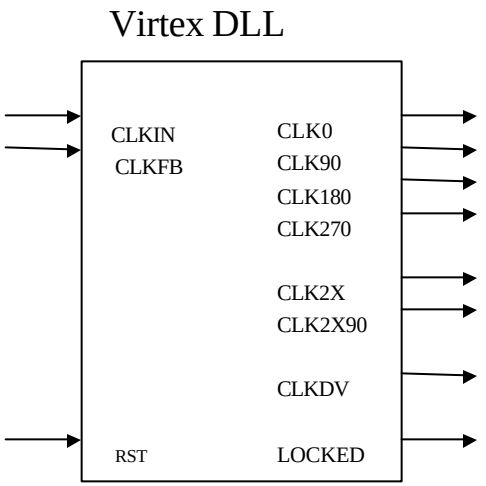
Pumping up System Performance



$$\text{System Performance} = \frac{1}{\text{Clock to Ouput} + \text{Input Setup} + \text{Clock Skew}}$$

Virtex Delay Locked Loop (DLL)

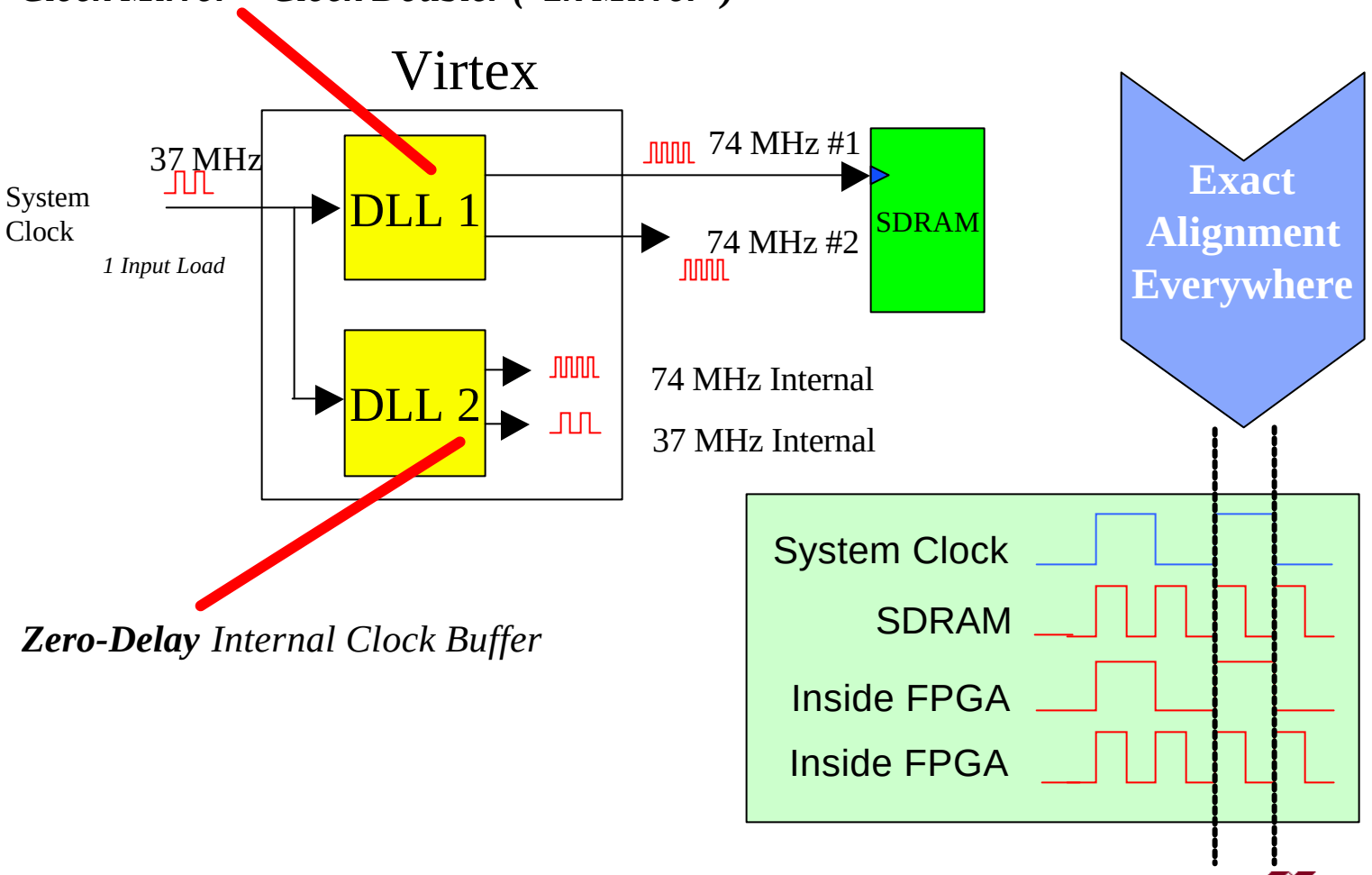
- Remove skew of multiple system clocks
  - 4 DLLs on every Virtex FPGA
  - Multiple outputs from each DLL
- Drive System Performance over 160 MHz
  - 2.5 ns Setup Time on all device
  - 0ns Hold Time on all devices
  - ***3.5 ns Clock-to-Out on all devices***
  - ***Less than 500ps skew across FPGA***
- Make clocks you need
  - Multiply clock frequency by 2
  - Divide clock frequency
  - Shift clock phase
- Generate clocks to other devices
  - Use multiple DLLs to remove internal and external clock skew



DLL Customer Design Example

HDTV Broadcast System

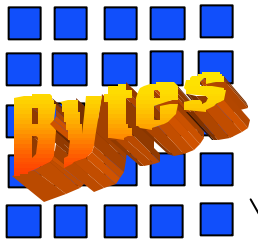
Clock Mirror + Clock Doubler ("2x Mirror")



***Zero-Delay** Internal Clock Buffer*

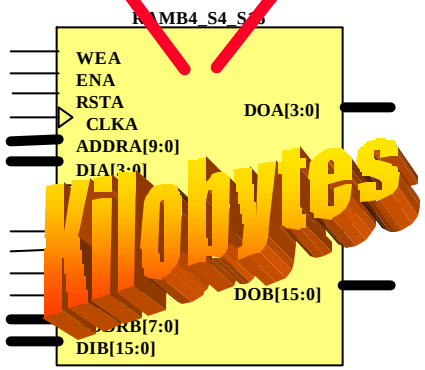
Building a Hierarchical Memory System

200 MHz Distributed SelectRAM+ Memory



System Memory

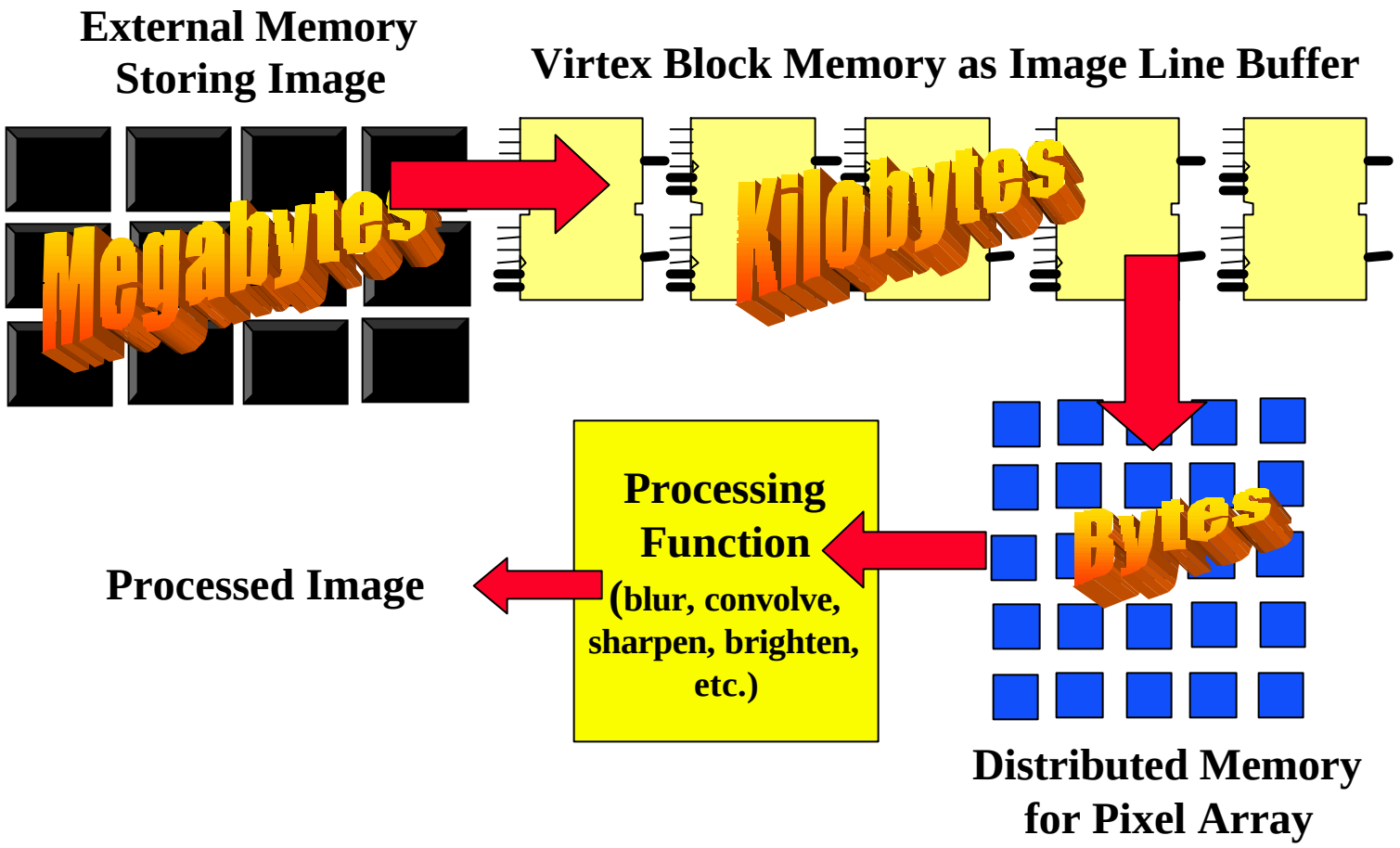
166 MHz Access to External Memory



200 MHz Block SelectRAM+ Memory

Memory Hierarchy Design Example

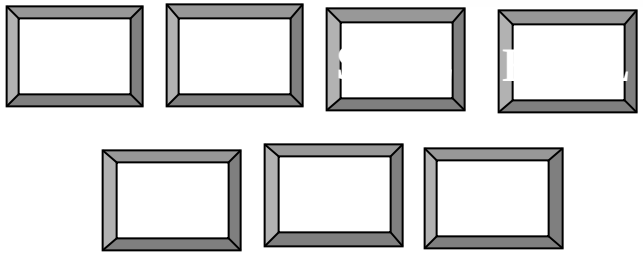
Photoshop Accelerator



Interface to any Device for Deep Sub-micron Era

Connect directly to external signals of varied voltages and swing points

System Interfaces



GTL GTL+ AGP

Backplanes

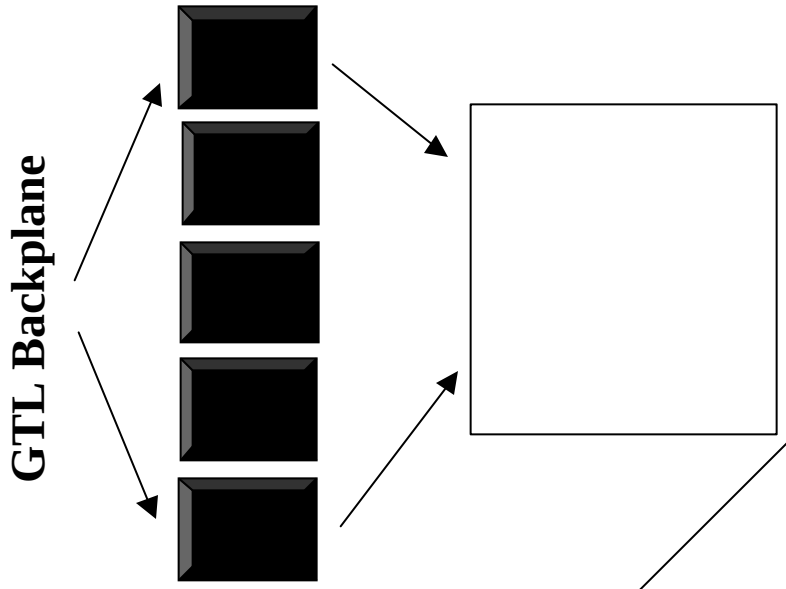
External Devices

Directly Interface to Virtex FPGA

Standard	Voh	Swing	Application
LVTTL	3.3	na	General purpose
LVCMOS2	2.5	na	General purpose
PCI 33MHz 3.3V	3.3	na	PCI
PCI 33MHz 5.0V	3.3	na	PCI
PCI 66MHz 3.3V	3.3	na	PCI
GTL	na	0.80	Backplane
GTL+	na	1.00	Backplane
HSTL-I	1.5	0.75	High Speed SRAM
HSTL-III	1.5	0.90	High Speed SRAM
HSTL-IV	1.5	0.75	High Speed SRAM
SSTL3-I	3.3	0.90	Synchronous DRAM
STTL3-II	3.3	1.50	Synchronous DRAM
SSTL2-I,II	2.5	1.10	Synchronous DRAM
AGP	3.3	1.32	Graphics
CTT	3.3	1.5	High Speed Memory

- ***Every I/O supports every standard***
- ***Device support multiple standards simultaneously***

SelectI/O Technology Cost Savings
Design Example

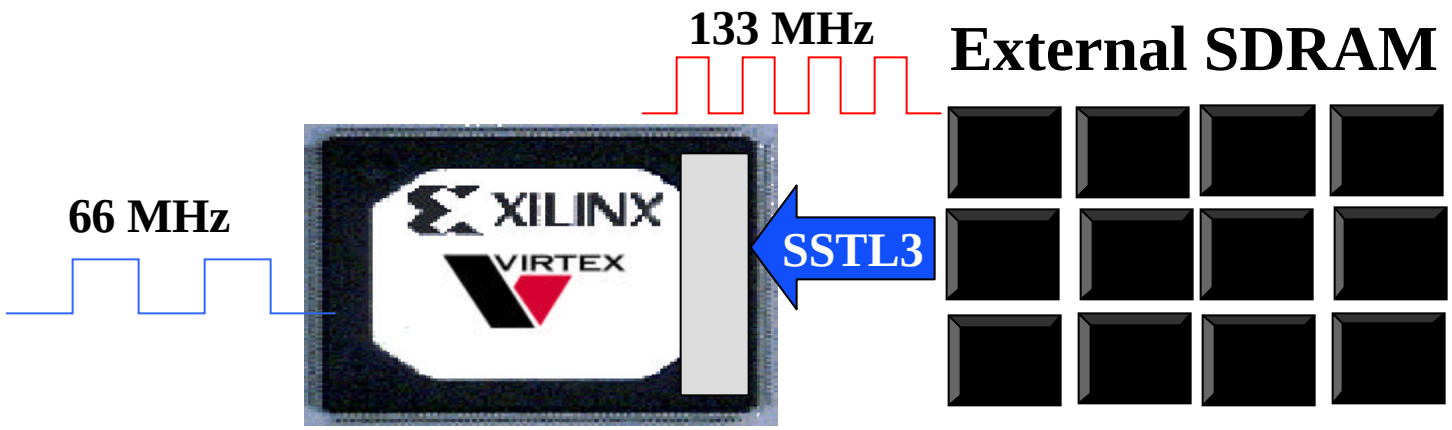


GTL to LVTTL Buffers

$200

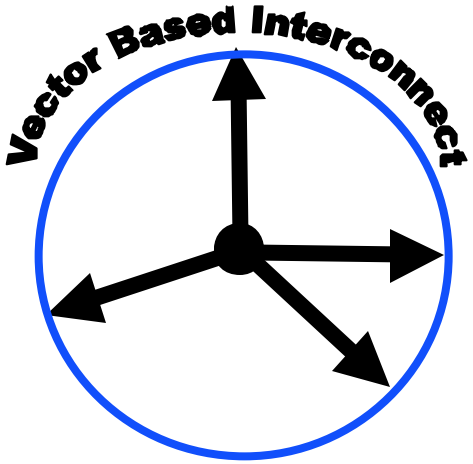
Buffers included

Combining System Level Features for High Speed Cache Memory



- Virtex DLL for 133 MHz clock synchronization of FPGA and SDRAM
- SSTL3 Interface to 133 MHz Fast SDRAM
- Virtex Block Memory for System Cache

Drop in Intellectual Property with Predictably Fast Performance



Delays are predictable as the straight line distance from source to destination

Core Friendly Architecture

System Integration

Available Virtex Cores

Building Blocks

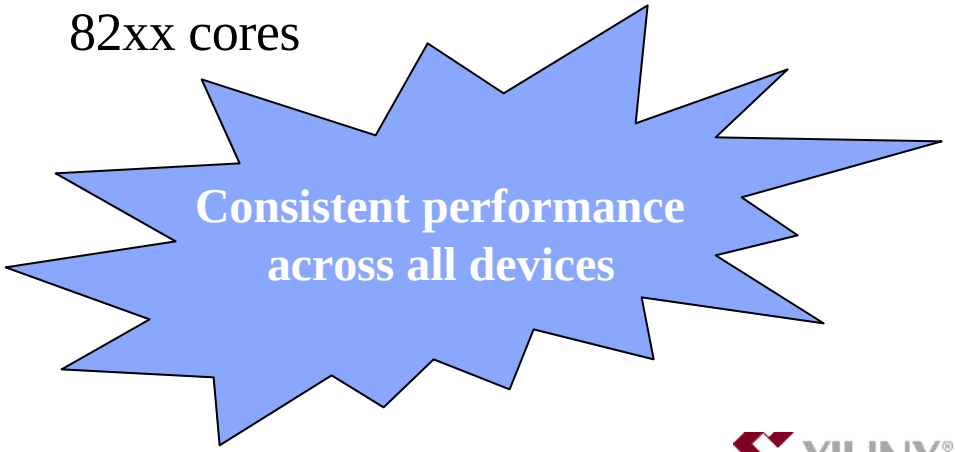
- Multiplexers
- Counters
- Registers
- Adders/Subtractors
- Accumulators
- Shift Registers
- Comparators
- Complementors

Memory

- Asynchronous FIFO
- Synchronous FIFO
- DMA controller
- Dual Port Block RAM
- Single Port Block RAM

Complex Cores

- Filters
- Reed Solomon - Encoder
- Reed Solomon - Decoder
- HDLC
- 622 MBPS SONET
- JPEG Encoder
- Video Streaming Core
- SDRAM Controller
- UART
- 82xx cores

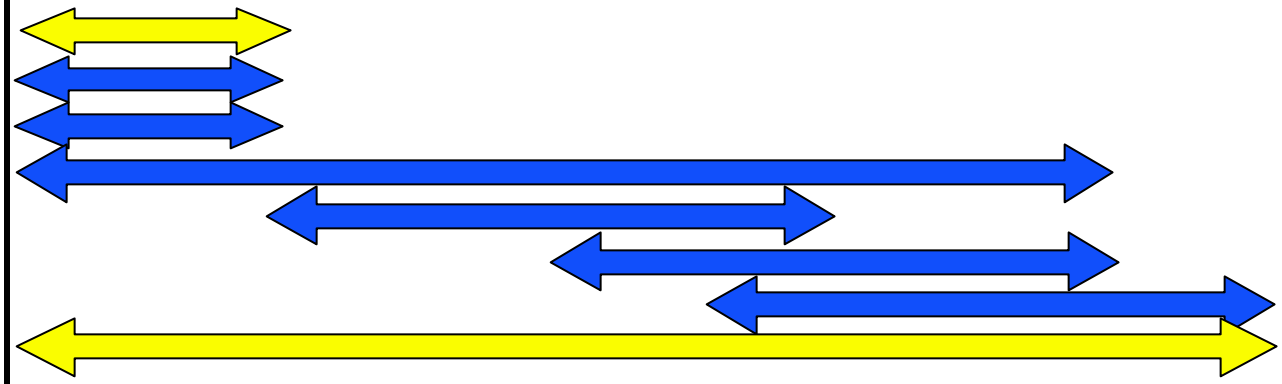


Virtex FPGA Family

Available now

Available now

Device	XCV50	XCV100	XCV150	XCV200	XCV300	XCV400	XCV600	XCV800	XCV1000
System Gates	50,000	100,000	150,000	200,000	**300,000**	400,000	600,000	800,000	**1,000,000**
Logic Cells	1728	2700	3888	5292	**6912**	10800	15552	21168	**27648**
Block RAM Bits	32,768	40,960	49,152	57,344	**65,536**	81,920	98,304	114,688	**131,072**
Packages									
0.8mm CSP									
VQ100									
TQ144									
PQ/HQ240									
BG352									
BG432									
BG560									
1.0mm BGA									



Industry Leading Price Points

Device	Volume Price
XCV50	$9.50
XCV300	$50.00
XCV1000	$350.00

End of 1999 pricing based on 100,000 units

Summary

- Virtex FPGAs shipping now and customers are completing designs today with it
- A powerful, unmatched combination of a robust feature set and expert technology to solve designers' challenges throughout the system—at industry leading price points
- Virtex FPGAs enable increased programmable content in existing and new applications

Introduction

The new Xilinx Virtex series, now shipping, fundamentally redefines programmable logic by expanding the traditional capabilities of field programmable gate arrays (FPGAs) to include a powerful set of features that address system level problems for high performance designs. The Virtex series has numerous built-in features to solve designers' challenges throughout the system: broad capability for chip-to-chip communications through support for new I/O standards, clock signal synchronization on the FPGA and on the board, and a memory hierarchy to manage fast access to RAM on and off chip.

With the Virtex series, digital designers for the first time can use an FPGA to perform not only familiar logic functions, but also tasks that were formerly handled at the board level by separate, dedicated parts. The Virtex series eliminates the need for components such as phase lock loops, voltage translation buffers, and memory when on-chip RAM is sufficient. This high level of integration allows designers to reduce overall system power requirements, cut costs, and save board space.

Board-level functions supported by the Virtex series include multiple, fully digital delay locked loops (DLLs) and support for more than a dozen deep submicon signaling standards. With these and other unique features, the Virtex series has created a new industry benchmark for FPGA functionality and performance. The Virtex series has transformed the FPGA from its former role as a "glue logic" device into the industry's first programmable solution that can serve as the board-level center for system design.

Wide Ranging Technology Advances

The Virtex series continues more than a decade of innovation and technical leadership in programmable logic by Xilinx, and it marks the fourth generation of FPGA devices developed by the company since Xilinx introduced the world's first such device in 1984.

The culmination of four years of research and development, Virtex technology represents significant breakthroughs in several fields of programmable logic: semiconductor manufacturing processes, architecture, software design tools, and software cores. Xilinx has applied for more than 20 patents for inventions incorporated in Virtex technology.

Using beta software that has been available since November 1997, customers worldwide have been implementing designs on Virtex FPGAs, and several designs are nearing completion. Full production support for the Virtex series is now in version 1.5 of the Xilinx Foundation™ Series and Alliance ™ Series design and implementation tools, which began shipping earlier this year.

Scalable FPGA Platform

Virtex technology provides the foundation for a scalable platform of mainstream advanced 0.22 micron, five-layer metal FPGA devices operating at system frequencies of up to 160 MHz. Densities for the new Virtex FPGAs range from the Xilinx XCV50™ device, with 50,000 system gates at the low end, to the high-end Xilinx XCV1000 ™ device, the industry's first one-million gate FPGA. Xilinx began shipping the XCV1000 in October 1998.

The Virtex XCV1000 FPGA is the world's largest FPGA device and ranks as one of the most complex standard integrated circuits built to date, featuring approximately 75 million transistors. The Virtex XCV1000 doubles the density of programmable logic previously available to digital designers in the 500,000-gate Xilinx XC40250XV ™ device.

All of the initial nine members of the Virtex series are scheduled to be shipping in production volumes during the first quarter of 1999. The Xilinx Virtex series FPGAs are affordably priced and deliver more gates, higher performance, and unique features at a lower price than other FPGAs. Virtex pricing will start below $10 for 50,000 system gates. Prices are for 100,000-unit orders for delivery in late 1999. Packaging options for the Virtex series will include plastic quad flat packs (PQFP), ball grid arrays (BGA), 1.0-mm fine pitch BGA, and 0.8-mm chip scale package (CSP).

Virtex Device Offerings			
Virtex device	**Logic cells**	**System gates**	**Block RAM bits**
XCV50	1,728	57,906	32,768
XCV100™	2,700	108,904	40,960
XCV150™	3,888	164,674	49,152
XCV200™	5,292	236,666	57,344
XCV300™	6,912	322,970	65,536
XCV400™	10,800	468,252	81,920
XCV600™	15,552	661,111	98,304
XCV800™	21,168	888,439	114,688
XCV1000	27,648	1,124,022	131,072

Vast Routing Resources

The Virtex series features a flexible, regular architecture that comprises an array of configurable logic blocks (CLBs) surrounded by programmable input-output blocks (IOBs), all interconnected with a highly efficient segmented routing structure. Five layers of metal provide a significant increase in routing resources over previous families of FPGAs. This reduces the interconnect delay and allows the Virtex series to accommodate very complex designs while delivering fast and predictable performance.

The Virtex CLB implements logic using four independent four-input lookup tables, four independent set/reset registers, multiplexer logic, and specialized arithmetic logic. The CLB was developed in parallel with the synthesis tools for the Virtex series to guarantee high performance results when used with VHDL and Verilog design methodologies. Complex logic such as 32-bit arithmetic functions, pipelined multiplication, and 64-to-1 multiplexing can be easily described in a high level language and will operate above 100 MHz in any Virtex device.

Foundation for Multi-million Gate Devices

Xilinx Virtex technology will allow the new series of system level FPGAs to migrate easily to the next generation 0.18-micron process that Xilinx is developing with its semiconductor fabrication partners. The 0.18-micron process will enable Xilinx to manufacture a Virtex FPGA device with two million system gates in the near future. High-end Virtex FPGAs are expected to compete with standard-cell application-specific integrated circuits (ASICs), while lower density Virtex devices are positioned to compete with mask-programmed gate array ASICs.

Dramatic Improvement in Design Productivity

For more than a decade, FPGAs have provided digital designers with continuously higher levels of logic density and performance while offering more flexibility and less risk than traditional ASICs. As a result, electronics manufacturers have been able to bring their products to market faster. The new Xilinx Virtex technology preserves this basic value of FPGAs, even for very high-density devices.

To leverage optimal use of the Virtex architecture, Xilinx focused its software efforts on the highest quality of synthesis results and very fast compile times. Implementation tools from Xilinx, and synthesis software from its leading EDA partners, were developed in tandem with the Virtex architecture to ensure high productivity for engineers who design with FPGAs. As a result, designers can expect timing-driven compile times of less than five hours for a one-million gate design—or about 200,000 gates an hour. This dramatic reduction in compile times is attributable to the vast routing resources of Virtex devices and to features such as an abundance of registers, latches, multiplexers, and carry logic.

All features of the Virtex architecture are accessible to designers who work with hardware description languages (HDLs) and who design at the register transfer level (RTL). Xilinx EDA partners such as Aldec, Cadence, Exemplar Logic, OrCAD, Model Technology, Mentor Graphics, Synplicity, Synopsys, and Veribest, all provide support for the Virtex series in their front-end design tools.

DLL for Advanced Clock Synchronization

As system performance requirements exceed 100 MHz, digital designers cannot tolerate long clock-to-output times, input set-up times, or on-chip clock skew. To solve this, the Xilinx Virtex devices offer four independent, fully digital delay locked loop circuits that allow internal and external clock synchronization. This capability removes clock skew from the entire system and delivers an increase in system performance up to 100 percent. Each DLL can lock on to a clock frequency up to 180 MHz and provide a variety of outputs that allow high precision in placing clock edges where required. Common applications for Virtex series DLLs include:

- The *zero-delay internal clock* for locking on to an external clock source, synchronizing clock edge that is external to the Virtex FPGA with the clock edge of a register inside the device. This provides clock-to-output delays of less than five nanoseconds.

- The *clock mirror* for using the Virtex Series DLL to lock on to a system clock and provide multiple synchronized clocks to the rest of the system. This is useful when interfacing to a system such as PCI that specifies a loading limit on the system clock. Combining prudent board design techniques with the Virtex Series DLL clock mirror gives designers the ability to synchronize clock edges system-wide.

- The *clock frequency doubler* for locking on and doubling an input frequency. This feature simplifies board design by allowing lower frequency signals to be routed on the board. For systems with logic on both clock domains, the edges of the 1X and 2X clock are aligned to avoid race conditions and mestability problems.

- The *clock phase synthesis* for generating 90º, 180º, and 270º phase shifts from an input frequency. This allows the system designer to move clock edges in a controlled way to meet specific setup, hold, and clock-to-output requirements.

SelectRAM+ ™ and Memory Hierarchy

For high-density designs, the demand for RAM increases at a greater rate than the demand for logic. The Virtex series is designed to connect with any amount of RAM through the Xilinx SelectRAM+ ä features.

- For bytes of RAM, Virtex FPGAs provide distributed SelectRAM ™. Pioneered in the Xilinx XC4000 ™ family, distributed SelectRAM is the use of the Virtex CLB lookup table resources as 16 x 1 static memory elements.

- For kilobytes of memory required in such applications as ATM buffers, video line buffers, and data-path FIFOs, Xilinx has introduced Virtex block SelectRAM+ memory. Each block of SelectRAM+ memory is a highly configurable, 4096-bit block of dual-ported synchronous SRAM. Each port can be configured to a variety of depths and widths ranging from 4096 x 1 to 256 x 16. Five-nanosecond clock-to-output and 1.2-

nanosecond address setup times allow over 160 MHz burst performance from these elements. Each port can be configured for independent width and depths. A typical application for this flexibility is to minimize device I/O requirements. Data buffers can interface to external logic at a high data rate while operations on the data can be performed on wider busses internally. Combined with the clock doubling capability of the Virtex DLL, implementing such a buffer is an easy task for designers.

- For high bandwidth access to megabytes of fast memory off chip, Virtex devices are capable of interfacing directly with SSTL and HSTL logic levels used by high-speed memory devices, as described in the following section.

Virtex SelectI/O ™

In recent years, a large number of new and specialized signaling standards tailored for specific applications have emerged, each with its own specifications for current, voltage, and termination techniques. The Virtex series addresses this problem by building configurable I/O structures that can interface to many different standards. These include Stub Series Transceiver Logic (SSTL) and High Speed Transceiver Logic (HSTL) to connect with fast 3.3-volt and 2.5-volt memories, and the Advanced Graphics Port (AGP) to interface with the Intel Pentium II processor for graphics applications.

The Virtex series also supports the Gunning Transceiver Logic (GTL and GTL+) I/O standards designed to interface with high performance microprocessors and backplanes. The Virtex SelectI/O feature also supports high-speed busses such as PCI33 and PCI66, which require specific current versus voltage characteristics.

These I/O standards solve performance issues, but some of them require specialized output and input buffers on the board. With the Virtex series, up to eight different voltage and signal standards can be directly connected to a single Virtex device.

Direct interfacing with these I/O standards eliminates the need for external buffers, increases performance, cuts costs, and reduces board space. Virtex SelectI/O supports the following bus standards:

LVTTL	HSTL Class I	SSTL2 Class 1
LVCMOS2	HSTL Class III	SSTL2 Class II
PCI	HSTL Class IV	CTT
GTL	SSTL3 Class I	AGP

GTL+ SSTL3 Class II

Virtex Core Methodology

The Virtex series extends the Xilinx Smart-IP technology first available on the XC4000 family of FPGAs. The vector-based interconnect structure of the Virtex series further extends the suitability of FPGAs to take advantage of the Xilinx high performance, highly predictable, and easy to use core methodology. Based on an abundance of routing resources of various lengths, the vector-based interconnect structure allows more cores with higher performance requirements to be implemented in HDLs than were possible in older architectures. Smart-IP ™ technology allows the core designer to leverage the absolute maximum performance for the fastest cores without giving up predictability or ease of customization. IP core developed with Xilinx Smart-IP technology are unique because they maintain their performance and predictability regardless of the size of the device in which they are implemented, or the number of cores used in a device.

Cores Available for Virtex Series

Building blocks	Complex cores	Memory
Multiplexers	Filters	Asynchronous FIFO
Counters	Reed Solomon - encoder	Synchronous FIFO
Registers	Reed Solomon - decoder	DMA controller
Adders/Subtractors	HDLC	Dual port block RAM
Accumulators	622 Mbits/sec SONET	Single port block RAM
Shift registers	JPEC encoder	
Comparators	Video streaming core	
Complementors	SDRAM controller	
	UART	
	82xx cores	

Summary

With advanced features such as thermal management, SelectI/O, support for multiple types of fast and unlimited memory, multiple digital delay locked loops, and a scalable architecture designed to support

fast software compile times and easy implementation of cores, the new Xilinx Virtex series clearly sets a new standard for mainstream FPGA design for the next century.

For more detailed information on the Xilinx Virtex series, including data sheets and applications notes, visit the product section of the Xilinx website at www.xilinx.com.

###

leading-edge software support at the time of product delivery. We are pleased that Synplicity's robust tools will enable designers to take advantage of this increase in density."

As part of the formal alliance established between Synplicity and Xilinx earlier this year, Synplicity has delivered a powerful and highly integrated solution to Virtex designers. Synplicity's enhanced Virtex mapper in Synplify delivers higher performance by utilizing the advanced interconnect structure in Virtex for placement-based timing optimizations. By providing software support concurrent with Virtex availability, Synplicity reinforces its commitment to provide customers with timely, high-quality support of new programmable logic devices.

"As gate counts continue to rise and designs become more and more complex, it is imperative that chip architectures and EDA tools be developed concurrently," said Andy Haines, vice president of marketing for Synplicity. "Over the years, we have established an excellent relationship with Xilinx and, through frequent technical exchanges, have been able to implement innovative solutions that produce superior quality of results."

Haines continued, "We look forward to continuing this relationship and are committed to maintaining our technology leadership position in FPGA synthesis."

About Synplicity's Solutions

Synplicity's offerings represent a new breed of synthesis for FPGA design engineers. The company's product line includes Synplify, a popular logic synthesis tool known for its quality results and speed; and HDL Analyst, a powerful HDL code analysis and debugging environment.

First introduced in 1995, Synplicity's Synplify synthesis tool represents a new breed of synthesis tools designed independent of existing academic or commercial code and features the company's innovative B.E.S.T.™ algorithms. The tool accepts industry-standard Verilog and VHDL descriptions and produces optimized implementations for programmable devices from many leading vendors. The recent release of Synplify 5.0 contains a unique multi-level timing constraints management system, giving designers, for the first time, the most accurate automated solution combined with the most robust user-controlled features, providing designers flexibility in

the way they use their synthesis tool. Designed to deliver the highest quality of results, Synplify is also extremely fast and easy-to-use. It includes a built-in language-sensitive editor and optional graphical (block diagram) analysis tool that gives direct feedback for fast design debug.

Pricing and Availability

Xilinx's Virtex FPGAs surpass the million-gate mark, run at over 100 MHz, and offer sufficient on-chip functionality to deliver the industry's first fully-programmable, system-level solution. Support for these devices is available now from Synplicity. Pricing for Synplify 5.0 node-locked Windows platform is $12,000, and floating licenses for Windows or workstation platforms are $24,000.

About Synplicity

Founded in 1994, Synplicity, Inc., delivers the benefits of logic synthesis and embedded synthesis technologies to programmable logic designers by developing fast, easy-to-use, affordable tools with extremely high quality of results. Synplicity's electronic design automation (EDA) products support industry-standard design languages (VHDL and Verilog), run on popular platforms (Windows '95, Windows NT and UNIX), and support leading PLD manufacturers. The company is located at 610 Caribbean Drive, Sunnyvale, Calif. 94089. Telephone: 408/548-6000; Fax: 408/548-0050; Email: info@synplicity.com; Web: http://www.synplicity.com.

# # #

About Exemplar Logic

Exemplar Logic, Inc. pioneered applying logic synthesis techniques to the design of FPGA, CPLDs, and ASICs. The company develops and markets EDA software tools, and is the world's number one leading PLD synthesis tool supplier, as reported by Dataquest (April 1997). Exemplar's products are sold and supported worldwide by Exemplar Logic, its OEMs and VARs, Mentor Graphics, and through the FPGA and CPLD component distribution channel.

Exemplar's design environments implement a complete high-level design (HLD) solution for FPGA, CPLD and ASIC design, offering synthesis, simulation and timing analysis for Windows 95, Windows-NT, and UNIX (HP, Sun) platforms in server and non-server environments.

Exemplar Logic is a privately held California corporation with headquarters located 6503 Dumbarton Circle, Fremont, CA 94555. Additional offices located throughout the world: Tokyo, London, Boston, and Denver. For information on Exemplar Logic's products, contact Exemplar Logic by calling 510-789-3333 or by sending eMail to sales@exemplar.com, or visit the Exemplar web site at www.exemplar.com

(Nasdaq: MENT, XLNX)

# # #

Contacts: Public Relations- Beverly Scott, Exemplar Logic, Inc., Manager, Corporate and Marketing Communications, Telephone: 510-789-3318, eMail: scott@exemplar.com. For Vendor Relations – Tom Hill, Exemplar Logic, Inc., Manager of Vendor Relations, Telephone: 503-685-7750 or eMail: hill@exemplar.com

than any other alternative, programmable or ASIC. Because of this, we expect increased programmable content in our customers' applications."

Million-gate device shipping to customers now

The Virtex development transpired over several years. Key to the development was extensive research in which Xilinx engaged customers to find out what they consider to be their most important system design challenges for the next five years. As a result, Virtex FPGAs include pre-engineered system-level solutions addressing customers' needs for push-button, high-performance design with capabilities that exceed today's FPGAs.

"These million gate FPGAs enable our test modules to provide thousands of continuous measurements at telecom speeds up to 2.4 gigabits per second," said Carl Uyehara, vice president of engineering at Adtech, Inc., a leading supplier of broadband test systems. "We especially like the fact that Virtex FPGAs are programmable. This allows us to use a single test module for multiple transmission technologies such as ATM and frame relay, which provides huge cost savings and convenience for our customers plus future enhancement capability."

New applications redefine what is possible with an FPGA

Virtex FPGAs are designed-in as main system components previously addressed by ASICs, specifically standard cell-based devices. Some examples are 66 MHz/64 Bit PCI applications; OC-3, OC-12 and OC-48 telecommunication equipment; next generation (beyond gigabit) network equipment; satellite base stations; high-performance graphics editing machines; massively parallel compute engines; HDTV broadcast systems; and medical imaging machines.

Virtex technology serves as the ideal platform to redefine FPGA implementation

The Virtex technology is the combination of leading-edge process technology, a system-level feature set, and breakthrough software technology. To build a million-gate FPGA, specific design optimization of the technology was necessary. The Virtex architecture represents aggressive use of an advanced 0.22-micron process to pack one million gates and full utilization of five metal layers for an abundance of high performance routing tracks. The architecture is also easily scalable from 50,000 to one million system gates to offer the robust feature set across a wide range of densities in nine devices.

"The critical features of Virtex FPGAs, such as the segmented routing and 0.22 micron feature size, allowed new levels of performance for our high-speed digital designs. The lower voltage and higher performance logic of the Virtex process are unmatched by any other supplier," said Ted Pascaru,

senior staff engineer at Hughes Space and Communications Company, Los Angeles, the leader in satellite communications. "The level of support—from the design development to the hotline assistance—that Xilinx offered with the new family also impressed us."

Still, an abundance of high performance, routable gates is not enough for system level design. Virtex FPGAs incorporate new system-level features including high performance clocking, I/O, and memory functions critical to million-gate designs. While a product backgrounder is available to describe each of these functions more in-depth, brief descriptions are:

- *Multiple digital locked loops (DLL)*: Large clock skew limits performance by taking away valuable nanoseconds from a clock period. Multiple DLLs in Virtex FPGAs allow internal and external clock synchronization—removing clock skew from the entire system and delivering up to a 100 percent increase in system performance, at greater than 160 MHz.
- *SelectI/O technology*: Rapid process technology advancements have proliferated different I/O standards complicating data flow between devices. Virtex SelectI/O technology supports multiple standards simultaneously including LVTTL, LVCMOS2, PCI33, PCI66, GTL/GTL+, SSTL, HSTL, AGP, and CTT. The flexibility of the SelectI/O technology will also support future standards.
- *SelectRAM+ memory hierarchy*: High bandwidth memory is critical in data intensive applications. Virtex SelectRAM+ memory hierarchy provides high bandwidth for memory blocks sizes in bytes (distributed memory), kilobytes (block memory), and megabytes (SSTL3 interface to external DRAM and SRAMs) for 200 MHz access to any amount of external memory.

"Virtex FPGAs have allowed us to implement our next generation digital TV broadcast systems in record time," said John Simmons, project manager, of NDS, a world leader in digital broadcasting solutions. "A key time saver was the availability of multiple DLLs that allowed us to synchronize a 74 MHz clock to more than 30 devices including multiple FPGAs, SDRAMs, and other components. Designing a no-skew clock system from scratch would take months. Xilinx delivered a ready-made solution to us with Virtex FPGAs."

Industry's fastest compile times available in Alliance Series 1.5 software

Software solutions focused on highest quality synthesis results and very fast compile times round out Virtex technology. VHDL and Verilog synthesis from Xilinx and leading EDA vendors support Virtex FPGAs now and deliver new performance levels for FPGAs. Xilinx has also delivered a quantum

leap in productivity by delivering timing-driven place and route tools capable of compiling 200,000 gates per hour for Virtex FPGAs.

As previously announced, the 1.5 release of Xilinx software supports the Virtex series. Virtex designers benefit from the release's key elements: the new Xilinx AKA*speed* technology, a suite of algorithms and algorithmic strategies combined with advanced new feature sets and applications optimized to address the elements of Virtex designs. AKA*speed* technology provides minimum timing delays, voltage and temperature prorating, graphical constraints editor, and enhancements to the existing technology elements such as timing driven implementation, K-paths, advanced timing analysis algorithms, robust constraints language, incremental design capabilities and the industry's most complete intellectual property strategy.

Superior programmable logic capability at no extra cost

All the benefits of Virtex technology are offered at new industry price points for FPGAs. This is enabled by the Xilinx leadership in product development that is based on the industry's most advanced processes. Projected high volume pricing will start below $10 for 50,000 system gates. * In comparison to other FPGAs, Virtex FPGAs deliver more gates, higher performance, and unique features at a lower price. Packaging options for the Virtex family will include plastic quad flat packs (PQFP), ball grid arrays (BGA), 1.0-mm fine pitch BGAs, and 0.8-mm chip scale packages (CSP).

Xilinx is the leading innovator of complete programmable logic solutions, including advanced integrated circuits, software design tools, predefined system functions delivered as cores, and unparalleled field engineering support. Founded in 1984 and headquartered in San Jose, Calif., Xilinx invented field programmable gate arrays (FPGA) and commands more than half of the world market for these devices today. Xilinx solutions enables customers to significantly reduce the time required to develop products for the computer, peripheral, telecommunication, networking, industrial control, instrumentation, high-reliability/military, and consumer markets. For more information, visit the Xilinx website at www.xilinx.com.

—30—

*Prices based on 100,000 unit quantity at the end of 1999 for slowest speed grade, least expensive package offering.