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.

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 1.

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.

Figure 1: Virtex Architecture Overview

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 can 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.
Virtex™ 2.5 V
Field Programmable Gate Arrays

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 ZBTAM 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 re-programmability
  - Four programming modes
- 0.22 μm 5-layer metal process
- 100% factory tested

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

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

Table 1: Supported Select I/O Standards

<table>
<thead>
<tr>
<th>I/O Standard</th>
<th>Input Reference Voltage ($V_{REF}$)</th>
<th>Output Source Voltage ($V_{CCO}$)</th>
<th>Board Termination Voltage ($V_{TT}$)</th>
<th>5 V Tolerant</th>
</tr>
</thead>
<tbody>
<tr>
<td>LVTTL 2 – 24 mA</td>
<td>N/A</td>
<td>3.3</td>
<td>N/A</td>
<td>Yes</td>
</tr>
<tr>
<td>LVCMOS2</td>
<td>N/A</td>
<td>2.5</td>
<td>N/A</td>
<td>Yes</td>
</tr>
<tr>
<td>PCI, 5 V</td>
<td>N/A</td>
<td>3.3</td>
<td>N/A</td>
<td>Yes</td>
</tr>
<tr>
<td>PCI, 3.3 V</td>
<td>N/A</td>
<td>3.3</td>
<td>N/A</td>
<td>No</td>
</tr>
<tr>
<td>GTL</td>
<td>0.8</td>
<td>N/A</td>
<td>1.2</td>
<td>No</td>
</tr>
<tr>
<td>GTL+</td>
<td>1.0</td>
<td>N/A</td>
<td>1.5</td>
<td>No</td>
</tr>
<tr>
<td>HSTL Class I</td>
<td>0.75</td>
<td>1.5</td>
<td>0.75</td>
<td>No</td>
</tr>
<tr>
<td>HSTL Class III</td>
<td>0.9</td>
<td>1.5</td>
<td>1.5</td>
<td>No</td>
</tr>
<tr>
<td>HSTL Class IV</td>
<td>0.9</td>
<td>1.5</td>
<td>1.5</td>
<td>No</td>
</tr>
<tr>
<td>SSTL3 Class I &amp; II</td>
<td>1.5</td>
<td>3.3</td>
<td>1.5</td>
<td>No</td>
</tr>
<tr>
<td>SSTL2 Class I &amp; II</td>
<td>1.25</td>
<td>2.5</td>
<td>1.25</td>
<td>No</td>
</tr>
<tr>
<td>CTT</td>
<td>1.5</td>
<td>3.3</td>
<td>1.5</td>
<td>No</td>
</tr>
<tr>
<td>AGP</td>
<td>1.32</td>
<td>3.3</td>
<td>N/A</td>
<td>No</td>
</tr>
</tbody>
</table>
more I/O pins convert to \(V_{\text{REF}}\) pins. Since these are always a superset of the \(V_{\text{REF}}\) pins used for smaller devices, it is possible to design a PCB that permits migration to a larger device if necessary. All the \(V_{\text{REF}}\) pins for the largest device anticipated must be connected to the \(V_{\text{REF}}\) voltage, and not used for I/O.

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

In TQ144 and PQ/HQ240 packages, all \(V_{\text{CCO}}\) pins are bonded together internally, and consequently the same \(V_{\text{CCO}}\) voltage must be connected to all of them. In the CS144 package, bank pairs that share a side are interconnected internally, permitting four choices for \(V_{\text{CCO}}\). In both cases, the \(V_{\text{REF}}\) pins remain internally connected as eight banks, and can 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 can be configured to operate asynchronously. All of the control signals are independently invertible, and are shared by the two flip-flops within the slice.

![Figure 4: 2-Slice Virtex CLB](slice_b.eps)
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, page 7. 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.

Table 3 shows the amount of Block SelectRAM memory that is available in each Virtex device.

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

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

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.

![Figure 6: Dual-Port Block SelectRAM](xcv_ds_006)

![Table 4: Block SelectRAM Port Aspect Ratios](Table 4: Block SelectRAM Port Aspect Ratios)

<table>
<thead>
<tr>
<th>Width</th>
<th>Depth</th>
<th>ADDR Bus</th>
<th>Data Bus</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4096</td>
<td>ADDR&lt;11:0&gt;</td>
<td>DATA&lt;0&gt;</td>
</tr>
<tr>
<td>2</td>
<td>2048</td>
<td>ADDR&lt;10:0&gt;</td>
<td>DATA&lt;1:0&gt;</td>
</tr>
<tr>
<td>4</td>
<td>1024</td>
<td>ADDR&lt;9:0&gt;</td>
<td>DATA&lt;3:0&gt;</td>
</tr>
<tr>
<td>8</td>
<td>512</td>
<td>ADDR&lt;8:0&gt;</td>
<td>DATA&lt;7:0&gt;</td>
</tr>
<tr>
<td>16</td>
<td>256</td>
<td>ADDR&lt;7:0&gt;</td>
<td>DATA&lt;15:0&gt;</td>
</tr>
</tbody>
</table>

![Figure 7: Virtex Local Routing](x8794b)
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.
- 72 buffered Hex lines route GRM signals to another GRMs six-blocks away in each one of the four directions. Organized in a staggered pattern, Hex lines can be driven only at their endpoints. Hex-line signals can be accessed either at the endpoints or at the midpoint (three blocks from the source). One third of the Hex lines are bidirectional, while the remaining ones are uni-directional.
- 12 Longlines are buffered, bidirectional wires that distribute signals across the device quickly and efficiently. Vertical Longlines span the full height of the device, and horizontal ones span the full width of the device.

I/O Routing

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

Dedicated Routing

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

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

Global Routing

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

- The primary global routing resources are four dedicated global nets with dedicated input pins that are designed to distribute high-fanout clock signals with minimal skew. Each global clock net can drive all CLB, IOB, and block RAM clock pins. The primary global nets can only be driven by global buffers. There are four global buffers, one for each global net.
- The secondary local clock routing resources consist of 24 backbone lines, 12 across the top of the chip and 12 across bottom. From these lines, up to 12 unique signals per column can be distributed via the 12 longlines in the column. These secondary resources are more flexible than the primary resources since they are not restricted to routing only to clock pins.

Clock Distribution

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

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

See [DLL Timing Parameters, page 20 of Module 3](#), for frequency range information.

**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 VCCO for Bank 2 should be 3.3 V. Otherwise, TDO switches rail-to-rail between ground and VCCO.

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 5 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 can 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.
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.

![Simplified DLL Macro Symbol BUFGDLL](image1)

**Figure 3: Simplified DLL Macro Symbol BUFGDLL**

```
CLKDLL

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLKN</td>
<td>CLK0</td>
<td>CLK90</td>
</tr>
<tr>
<td>CLKB</td>
<td>CLK180</td>
<td>CLK270</td>
</tr>
<tr>
<td></td>
<td>CLK2X</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CLKDV</td>
<td>RST</td>
</tr>
<tr>
<td></td>
<td>LOCKED</td>
<td></td>
</tr>
</tbody>
</table>
```

**Figure 4: Standard DLL Symbol CLKDLL**
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.

<table>
<thead>
<tr>
<th>Phase (degrees)</th>
<th>% Period Shift</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>90</td>
<td>25%</td>
</tr>
<tr>
<td>180</td>
<td>50%</td>
</tr>
<tr>
<td>270</td>
<td>75%</td>
</tr>
</tbody>
</table>

Figure 7: DLL 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).
In addition to the test instructions outlined above, the boundary-scan circuitry can be used to configure the FPGA, and also to read back the configuration data.

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

### Instruction Set

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

### Data Registers

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

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

The FPGA supports up to two additional internal scan chains that can be specified using the BSCAN macro. The macro provides two user pins (SEL1 and SEL2) which are 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).

### Bit Sequence

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

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

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

---

**Figure 10:** Virtex Series Boundary Scan Logic