



Welcome to [E-XFL.COM](https://www.e-xfl.com)

### Understanding Embedded - FPGAs (Field Programmable Gate Array)

Embedded - FPGAs, or Field Programmable Gate Arrays, are advanced integrated circuits that offer unparalleled flexibility and performance for digital systems. Unlike traditional fixed-function logic devices, FPGAs can be programmed and reprogrammed to execute a wide array of logical operations, enabling customized functionality tailored to specific applications. This reprogrammability allows developers to iterate designs quickly and implement complex functions without the need for custom hardware.

### Applications of Embedded - FPGAs

The versatility of Embedded - FPGAs makes them indispensable in numerous fields. In telecommunications.

#### Details

Product Status	Obsolete
Number of LABs/CLBs	576
Number of Logic Elements/Cells	1368
Total RAM Bits	18432
Number of I/O	160
Number of Gates	13000
Voltage - Supply	3V ~ 3.6V
Mounting Type	Surface Mount
Operating Temperature	-40°C ~ 100°C (TJ)
Package / Case	208-BFQFP
Supplier Device Package	208-PQFP (28x28)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/xilinx/xc4013xl-2pq208i">https://www.e-xfl.com/product-detail/xilinx/xc4013xl-2pq208i</a>

## **XC4000E and XC4000X Series Compared to the XC4000**

For readers already familiar with the XC4000 family of Xilinx Field Programmable Gate Arrays, the major new features in the XC4000 Series devices are listed in this section. The biggest advantages of XC4000E and XC4000X devices are significantly increased system speed, greater capacity, and new architectural features, particularly Select-RAM memory. The XC4000X devices also offer many new routing features, including special high-speed clock buffers that can be used to capture input data with minimal delay.

Any XC4000E device is pinout- and bitstream-compatible with the corresponding XC4000 device. An existing XC4000 bitstream can be used to program an XC4000E device. However, since the XC4000E includes many new features, an XC4000E bitstream cannot be loaded into an XC4000 device.

XC4000X Series devices are not bitstream-compatible with equivalent array size devices in the XC4000 or XC4000E families. However, equivalent array size devices, such as the XC4025, XC4025E, XC4028EX, and XC4028XL, are pinout-compatible.

### **Improvements in XC4000E and XC4000X**

#### **Increased System Speed**

XC4000E and XC4000X devices can run at synchronous system clock rates of up to 80 MHz, and internal performance can exceed 150 MHz. This increase in performance over the previous families stems from improvements in both device processing and system architecture. XC4000 Series devices use a sub-micron multi-layer metal process. In addition, many architectural improvements have been made, as described below.

The XC4000XL family is a high performance 3.3V family based on 0.35 $\mu$  SRAM technology and supports system speeds to 80 MHz.

#### **PCI Compliance**

XC4000 Series -2 and faster speed grades are fully PCI compliant. XC4000E and XC4000X devices can be used to implement a one-chip PCI solution.

#### **Carry Logic**

The speed of the carry logic chain has increased dramatically. Some parameters, such as the delay on the carry chain through a single CLB ( $T_{BYP}$ ), have improved by as

much as 50% from XC4000 values. See [“Fast Carry Logic” on page 18](#) for more information.

#### **Select-RAM Memory: Edge-Triggered, Synchronous RAM Modes**

The RAM in any CLB can be configured for synchronous, edge-triggered, write operation. The read operation is not affected by this change to an edge-triggered write.

#### **Dual-Port RAM**

A separate option converts the 16x2 RAM in any CLB into a 16x1 dual-port RAM with simultaneous Read/Write.

The function generators in each CLB can be configured as either level-sensitive (asynchronous) single-port RAM, edge-triggered (synchronous) single-port RAM, edge-triggered (synchronous) dual-port RAM, or as combinatorial logic.

#### **Configurable RAM Content**

The RAM content can now be loaded at configuration time, so that the RAM starts up with user-defined data.

#### **H Function Generator**

In current XC4000 Series devices, the H function generator is more versatile than in the original XC4000. Its inputs can come not only from the F and G function generators but also from up to three of the four control input lines. The H function generator can thus be totally or partially independent of the other two function generators, increasing the maximum capacity of the device.

#### **IOB Clock Enable**

The two flip-flops in each IOB have a common clock enable input, which through configuration can be activated individually for the input or output flip-flop or both. This clock enable operates exactly like the EC pin on the XC4000 CLB. This new feature makes the IOBs more versatile, and avoids the need for clock gating.

#### **Output Drivers**

The output pull-up structure defaults to a TTL-like totem-pole. This driver is an n-channel pull-up transistor, pulling to a voltage one transistor threshold below  $V_{cc}$ , just like the XC4000 family outputs. Alternatively, XC4000 Series devices can be globally configured with CMOS outputs, with p-channel pull-up transistors pulling to  $V_{cc}$ . Also, the configurable pull-up resistor in the XC4000 Series is a p-channel transistor that pulls to  $V_{cc}$ , whereas in the original XC4000 family it is an n-channel transistor that pulls to a voltage one transistor threshold below  $V_{cc}$ .

**Table 1: XC4000E and XC4000X Series Field Programmable Gate Arrays**

Device	Logic Cells	Max Logic Gates (No RAM)	Max. RAM Bits (No Logic)	Typical Gate Range (Logic and RAM)*	CLB Matrix	Total CLBs	Number of Flip-Flops	Max. User I/O
XC4002XL	152	1,600	2,048	1,000 - 3,000	8 x 8	64	256	64
XC4003E	238	3,000	3,200	2,000 - 5,000	10 x 10	100	360	80
XC4005E/XL	466	5,000	6,272	3,000 - 9,000	14 x 14	196	616	112
XC4006E	608	6,000	8,192	4,000 - 12,000	16 x 16	256	768	128
XC4008E	770	8,000	10,368	6,000 - 15,000	18 x 18	324	936	144
XC4010E/XL	950	10,000	12,800	7,000 - 20,000	20 x 20	400	1,120	160
XC4013E/XL	1368	13,000	18,432	10,000 - 30,000	24 x 24	576	1,536	192
XC4020E/XL	1862	20,000	25,088	13,000 - 40,000	28 x 28	784	2,016	224
XC4025E	2432	25,000	32,768	15,000 - 45,000	32 x 32	1,024	2,560	256
XC4028EX/XL	2432	28,000	32,768	18,000 - 50,000	32 x 32	1,024	2,560	256
XC4036EX/XL	3078	36,000	41,472	22,000 - 65,000	36 x 36	1,296	3,168	288
XC4044XL	3800	44,000	51,200	27,000 - 80,000	40 x 40	1,600	3,840	320
XC4052XL	4598	52,000	61,952	33,000 - 100,000	44 x 44	1,936	4,576	352
XC4062XL	5472	62,000	73,728	40,000 - 130,000	48 x 48	2,304	5,376	384
XC4085XL	7448	85,000	100,352	55,000 - 180,000	56 x 56	3,136	7,168	448

\* Max values of Typical Gate Range include 20-30% of CLBs used as RAM.

**Note:** All functionality in low-voltage families is the same as in the corresponding 5-Volt family, except where numerical references are made to timing or power.

## Description

XC4000 Series devices are implemented with a regular, flexible, programmable architecture of Configurable Logic Blocks (CLBs), interconnected by a powerful hierarchy of versatile routing resources, and surrounded by a perimeter of programmable Input/Output Blocks (IOBs). They have generous routing resources to accommodate the most complex interconnect patterns.

The devices are customized by loading configuration data into internal memory cells. The FPGA can either actively read its configuration data from an external serial or byte-parallel PROM (master modes), or the configuration data can be written into the FPGA from an external device (slave and peripheral modes).

XC4000 Series FPGAs are supported by powerful and sophisticated software, covering every aspect of design from schematic or behavioral entry, floor planning, simulation, automatic block placement and routing of interconnects, to the creation, downloading, and readback of the configuration bit stream.

Because Xilinx FPGAs can be reprogrammed an unlimited number of times, they can be used in innovative designs

where hardware is changed dynamically, or where hardware must be adapted to different user applications. FPGAs are ideal for shortening design and development cycles, and also offer a cost-effective solution for production rates well beyond 5,000 systems per month.

## Taking Advantage of Re-configuration

FPGA devices can be re-configured to change logic function while resident in the system. This capability gives the system designer a new degree of freedom not available with any other type of logic.

Hardware can be changed as easily as software. Design updates or modifications are easy, and can be made to products already in the field. An FPGA can even be re-configured dynamically to perform different functions at different times.

Re-configurable logic can be used to implement system self-diagnostics, create systems capable of being re-configured for different environments or operations, or implement multi-purpose hardware for a given application. As an added benefit, using re-configurable FPGA devices simplifies hardware design and debugging and shortens product time-to-market.

### ***Input Thresholds***

The input thresholds of 5V devices can be globally configured for either TTL (1.2 V threshold) or CMOS (2.5 V threshold), just like XC2000 and XC3000 inputs. The two global adjustments of input threshold and output level are independent of each other. The XC4000XL family has an input threshold of 1.6V, compatible with both 3.3V CMOS and TTL levels.

### ***Global Signal Access to Logic***

There is additional access from global clocks to the F and G function generator inputs.

### ***Configuration Pin Pull-Up Resistors***

During configuration, these pins have weak pull-up resistors. For the most popular configuration mode, Slave Serial, the mode pins can thus be left unconnected. The three mode inputs can be individually configured with or without weak pull-up or pull-down resistors. A pull-down resistor value of 4.7 k $\Omega$  is recommended.

The three mode inputs can be individually configured with or without weak pull-up or pull-down resistors after configuration.

The PROGRAM input pin has a permanent weak pull-up.

### ***Soft Start-up***

Like the XC3000A, XC4000 Series devices have "Soft Start-up." When the configuration process is finished and the device starts up, the first activation of the outputs is automatically slew-rate limited. This feature avoids potential ground bounce when all outputs are turned on simultaneously. Immediately after start-up, the slew rate of the individual outputs is, as in the XC4000 family, determined by the individual configuration option.

### ***XC4000 and XC4000A Compatibility***

Existing XC4000 bitstreams can be used to configure an XC4000E device. XC4000A bitstreams must be recompiled for use with the XC4000E due to improved routing resources, although the devices are pin-for-pin compatible.

## **Additional Improvements in XC4000X Only**

### ***Increased Routing***

New interconnect in the XC4000X includes twenty-two additional vertical lines in each column of CLBs and twelve new horizontal lines in each row of CLBs. The twelve "Quad Lines" in each CLB row and column include optional repowering buffers for maximum speed. Additional high-performance routing near the IOBs enhances pin flexibility.

### ***Faster Input and Output***

A fast, dedicated early clock sourced by global clock buffers is available for the IOBs. To ensure synchronization with the regular global clocks, a Fast Capture latch driven by the early clock is available. The input data can be initially loaded into the Fast Capture latch with the early clock, then transferred to the input flip-flop or latch with the low-skew global clock. A programmable delay on the input can be used to avoid hold-time requirements. See "IOB Input Signals" on page 20 for more information.

### ***Latch Capability in CLBs***

Storage elements in the XC4000X CLB can be configured as either flip-flops or latches. This capability makes the FPGA highly synthesis-compatible.

### ***IOB Output MUX From Output Clock***

A multiplexer in the IOB allows the output clock to select either the output data or the IOB clock enable as the output to the pad. Thus, two different data signals can share a single output pad, effectively doubling the number of device outputs without requiring a larger, more expensive package. This multiplexer can also be configured as an AND-gate to implement a very fast pin-to-pin path. See "IOB Output Signals" on page 23 for more information.

### ***Additional Address Bits***

Larger devices require more bits of configuration data. A daisy chain of several large XC4000X devices may require a PROM that cannot be addressed by the eighteen address bits supported in the XC4000E. The XC4000X Series therefore extends the addressing in Master Parallel configuration mode to 22 bits.

### Set/Reset

An asynchronous storage element input (SR) can be configured as either set or reset. This configuration option determines the state in which each flip-flop becomes operational after configuration. It also determines the effect of a Global Set/Reset pulse during normal operation, and the effect of a pulse on the SR pin of the CLB. All three set/reset functions for any single flip-flop are controlled by the same configuration data bit.

The set/reset state can be independently specified for each flip-flop. This input can also be independently disabled for either flip-flop.

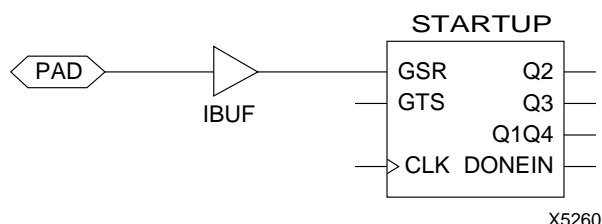
The set/reset state is specified by using the INIT attribute, or by placing the appropriate set or reset flip-flop library symbol.

SR is active High. It is not invertible within the CLB.

### Global Set/Reset

A separate Global Set/Reset line (not shown in Figure 1) sets or clears each storage element during power-up, re-configuration, or when a dedicated Reset net is driven active. This global net (GSR) does not compete with other routing resources; it uses a dedicated distribution network.

Each flip-flop is configured as either globally set or reset in the same way that the local set/reset (SR) is specified. Therefore, if a flip-flop is set by SR, it is also set by GSR. Similarly, a reset flip-flop is reset by both SR and GSR.



**Figure 2: Schematic Symbols for Global Set/Reset**

GSR can be driven from any user-programmable pin as a global reset input. To use this global net, place an input pad and input buffer in the schematic or HDL code, driving the GSR pin of the STARTUP symbol. (See Figure 2.) A specific pin location can be assigned to this input using a LOC attribute or property, just as with any other user-programmable pad. An inverter can optionally be inserted after the input buffer to invert the sense of the Global Set/Reset signal.

Alternatively, GSR can be driven from any internal node.

### Data Inputs and Outputs

The source of a storage element data input is programmable. It is driven by any of the functions F', G', and H', or by the Direct In (DIN) block input. The flip-flops or latches drive the XQ and YQ CLB outputs.

Two fast feed-through paths are available, as shown in Figure 1. A two-to-one multiplexer on each of the XQ and YQ outputs selects between a storage element output and any of the control inputs. This bypass is sometimes used by the automated router to repower internal signals.

### Control Signals

Multiplexers in the CLB map the four control inputs (C1 - C4 in Figure 1) into the four internal control signals (H1, DIN/H2, SR/H0, and EC). Any of these inputs can drive any of the four internal control signals.

When the logic function is enabled, the four inputs are:

- EC — Enable Clock
- SR/H0 — Asynchronous Set/Reset or H function generator Input 0
- DIN/H2 — Direct In or H function generator Input 2
- H1 — H function generator Input 1.

When the memory function is enabled, the four inputs are:

- EC — Enable Clock
- WE — Write Enable
- D0 — Data Input to F and/or G function generator
- D1 — Data input to G function generator (16x1 and 16x2 modes) or 5th Address bit (32x1 mode).

### Using FPGA Flip-Flops and Latches

The abundance of flip-flops in the XC4000 Series invites pipelined designs. This is a powerful way of increasing performance by breaking the function into smaller subfunctions and executing them in parallel, passing on the results through pipeline flip-flops. This method should be seriously considered wherever throughput is more important than latency.

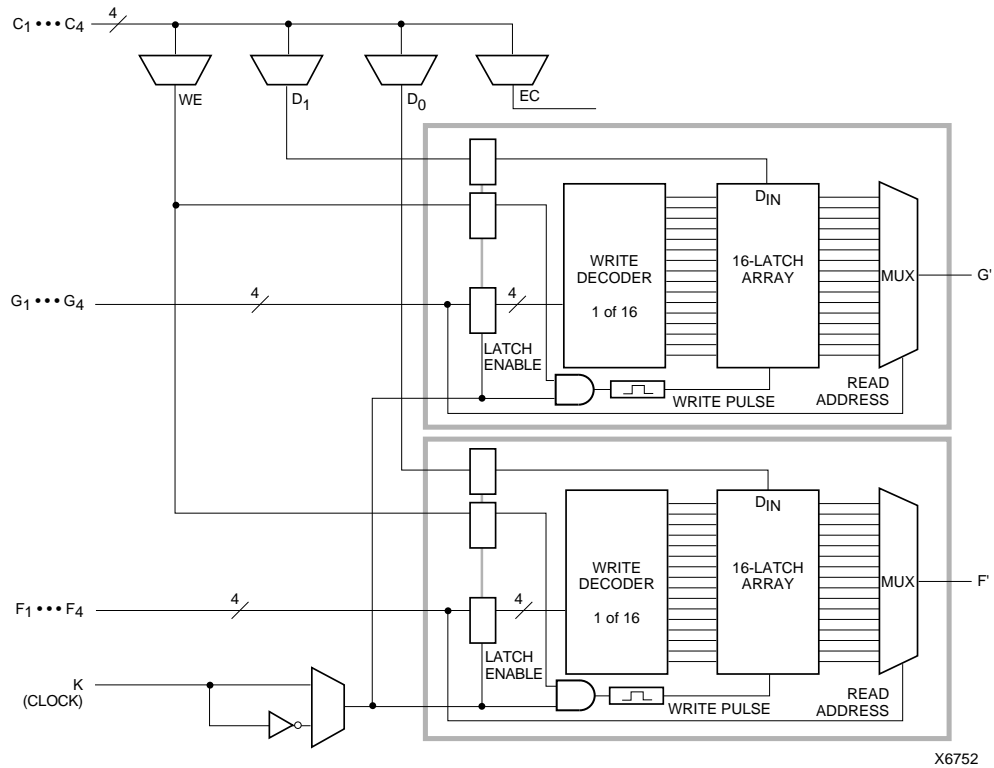
To include a CLB flip-flop, place the appropriate library symbol. For example, FDCE is a D-type flip-flop with clock enable and asynchronous clear. The corresponding latch symbol (for the XC4000X only) is called LDCE.

In XC4000 Series devices, the flip flops can be used as registers or shift registers without blocking the function generators from performing a different, perhaps unrelated task. This ability increases the functional capacity of the devices.

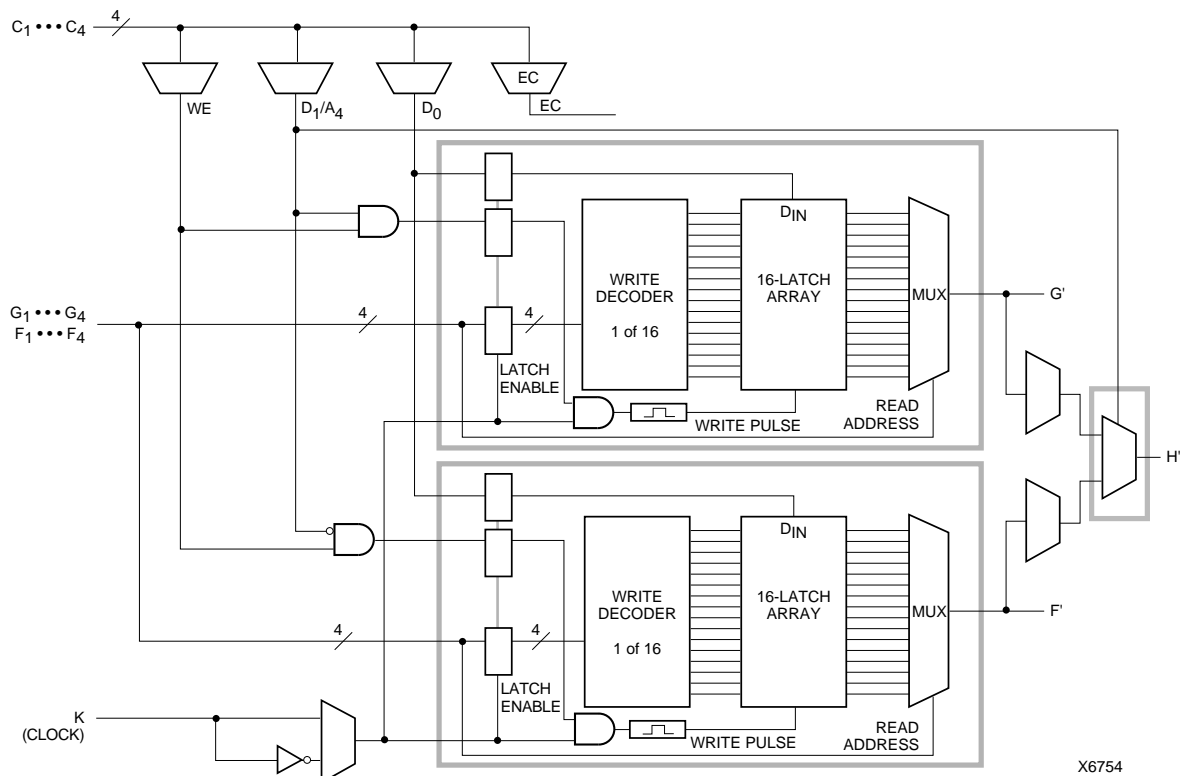
The CLB setup time is specified between the function generator inputs and the clock input K. Therefore, the specified CLB flip-flop setup time includes the delay through the function generator.

### Using Function Generators as RAM

Optional modes for each CLB make the memory look-up tables in the F' and G' function generators usable as an array of Read/Write memory cells. Available modes are level-sensitive (similar to the XC4000/A/H families), edge-triggered, and dual-port edge-triggered. Depending on the selected mode, a single CLB can be configured as either a 16x2, 32x1, or 16x1 bit array.



**Figure 4: 16x2 (or 16x1) Edge-Triggered Single-Port RAM**



**Figure 5: 32x1 Edge-Triggered Single-Port RAM (F and G addresses are identical)**



### Fast Carry Logic

Each CLB F and G function generator contains dedicated arithmetic logic for the fast generation of carry and borrow signals. This extra output is passed on to the function generator in the adjacent CLB. The carry chain is independent of normal routing resources.

Dedicated fast carry logic greatly increases the efficiency and performance of adders, subtractors, accumulators, comparators and counters. It also opens the door to many new applications involving arithmetic operation, where the previous generations of FPGAs were not fast enough or too inefficient. High-speed address offset calculations in micro-processor or graphics systems, and high-speed addition in digital signal processing are two typical applications.

The two 4-input function generators can be configured as a 2-bit adder with built-in hidden carry that can be expanded to any length. This dedicated carry circuitry is so fast and efficient that conventional speed-up methods like carry generate/propagate are meaningless even at the 16-bit level, and of marginal benefit at the 32-bit level.

This fast carry logic is one of the more significant features of the XC4000 Series, speeding up arithmetic and counting into the 70 MHz range.

The carry chain in XC4000E devices can run either up or down. At the top and bottom of the columns where there are no CLBs above or below, the carry is propagated to the right. (See Figure 11.) In order to improve speed in the high-capacity XC4000X devices, which can potentially have very long carry chains, the carry chain travels upward only, as shown in Figure 12. Additionally, standard interconnect can be used to route a carry signal in the downward direction.

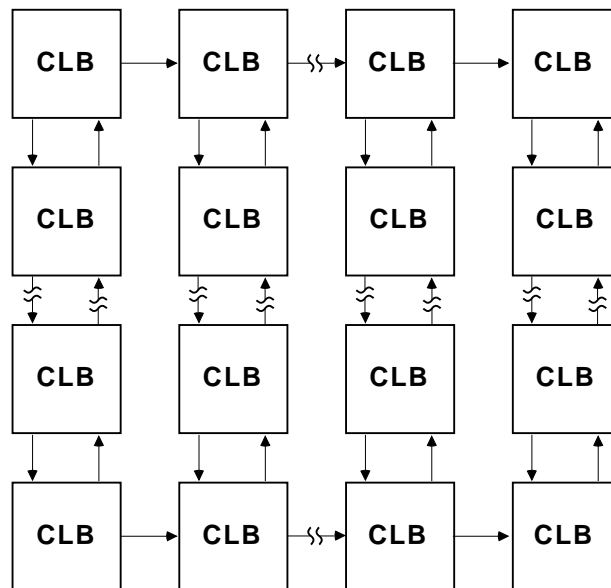
Figure 13 on page 19 shows an XC4000E CLB with dedicated fast carry logic. The carry logic in the XC4000X is similar, except that COUT exits at the top only, and the signal CINDOWN does not exist. As shown in Figure 13, the carry logic shares operand and control inputs with the function generators. The carry outputs connect to the function generators, where they are combined with the operands to form the sums.

Figure 14 on page 20 shows the details of the carry logic for the XC4000E. This diagram shows the contents of the box labeled "CARRY LOGIC" in Figure 13. The XC4000X carry logic is very similar, but a multiplexer on the pass-through carry chain has been eliminated to reduce delay. Additionally, in the XC4000X the multiplexer on the G4 path has a memory-programmable 0 input, which permits G4 to directly connect to COUT. G4 thus becomes an additional high-speed initialization path for carry-in.

The dedicated carry logic is discussed in detail in Xilinx document XAPP 013: "Using the Dedicated Carry Logic in

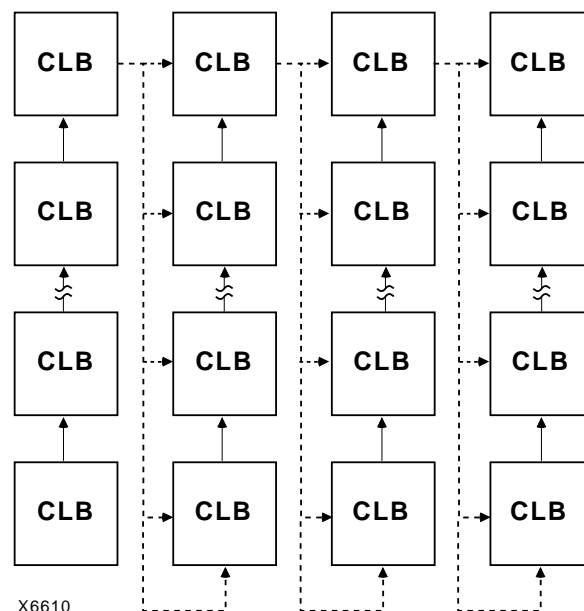
XC4000." This discussion also applies to XC4000E devices, and to XC4000X devices when the minor logic changes are taken into account.

The fast carry logic can be accessed by placing special library symbols, or by using Xilinx Relationally Placed Macros (RPMs) that already include these symbols.



X6687

**Figure 11: Available XC4000E Carry Propagation Paths**



X6610

**Figure 12: Available XC4000X Carry Propagation Paths (dotted lines use general interconnect)**

The oscillator output is optionally available after configuration. Any two of four resynchronized taps of a built-in divider are also available. These taps are at the fourth, ninth, fourteenth and nineteenth bits of the divider. Therefore, if the primary oscillator output is running at the nominal 8 MHz, the user has access to an 8 MHz clock, plus any two of 500 kHz, 16kHz, 490Hz and 15Hz (up to 10% lower for low-voltage devices). These frequencies can vary by as much as -50% or +25%.

These signals can be accessed by placing the OSC4 library element in a schematic or in HDL code (see [Figure 24](#)).

The oscillator is automatically disabled after configuration if the OSC4 symbol is not used in the design.

## Programmable Interconnect

All internal connections are composed of metal segments with programmable switching points and switching matrices to implement the desired routing. A structured, hierarchical matrix of routing resources is provided to achieve efficient automated routing.

The XC4000E and XC4000X share a basic interconnect structure. XC4000X devices, however, have additional routing not available in the XC4000E. The extra routing resources allow high utilization in high-capacity devices. All XC4000X-specific routing resources are clearly identified throughout this section. Any resources not identified as XC4000X-specific are present in all XC4000 Series devices.

This section describes the varied routing resources available in XC4000 Series devices. The implementation software automatically assigns the appropriate resources based on the density and timing requirements of the design.

## Interconnect Overview

There are several types of interconnect.

- CLB routing is associated with each row and column of the CLB array.
- IOB routing forms a ring (called a VersaRing) around the outside of the CLB array. It connects the I/O with the internal logic blocks.

- Global routing consists of dedicated networks primarily designed to distribute clocks throughout the device with minimum delay and skew. Global routing can also be used for other high-fanout signals.

Five interconnect types are distinguished by the relative length of their segments: single-length lines, double-length lines, quad and octal lines (XC4000X only), and longlines. In the XC4000X, direct connects allow fast data flow between adjacent CLBs, and between IOBs and CLBs.

Extra routing is included in the IOB pad ring. The XC4000X also includes a ring of octal interconnect lines near the IOBs to improve pin-swapping and routing to locked pins.

XC4000E/X devices include two types of global buffers. These global buffers have different properties, and are intended for different purposes. They are discussed in detail later in this section.

## CLB Routing Connections

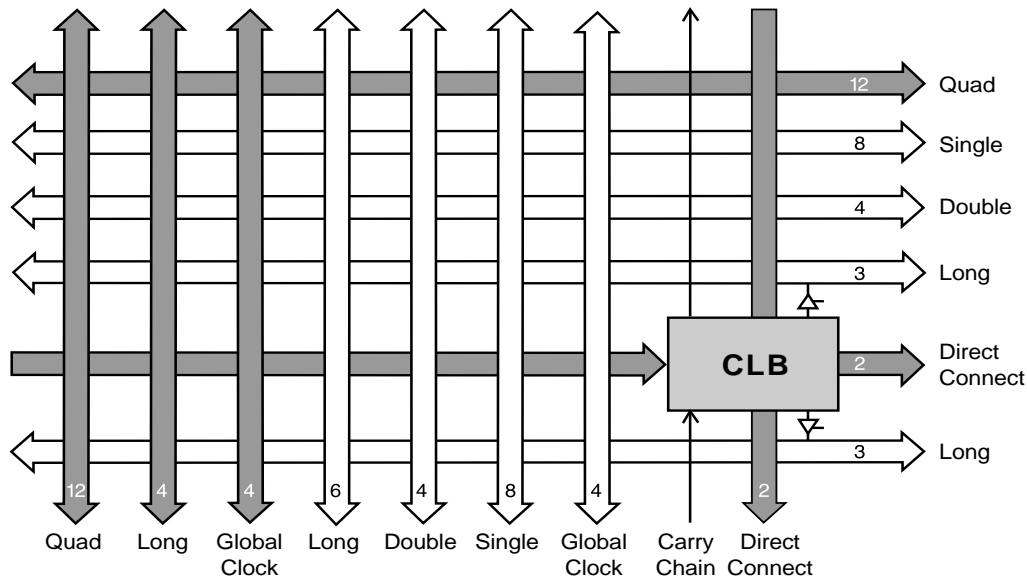
A high-level diagram of the routing resources associated with one CLB is shown in [Figure 25](#). The shaded arrows represent routing present only in XC4000X devices.

[Table 14](#) shows how much routing of each type is available in XC4000E and XC4000X CLB arrays. Clearly, very large designs, or designs with a great deal of interconnect, will route more easily in the XC4000X. Smaller XC4000E designs, typically requiring significantly less interconnect, do not require the additional routing.

[Figure 27 on page 30](#) is a detailed diagram of both the XC4000E and the XC4000X CLB, with associated routing. The shaded square is the programmable switch matrix, present in both the XC4000E and the XC4000X. The L-shaped shaded area is present only in XC4000X devices. As shown in the figure, the XC4000X block is essentially an XC4000E block with additional routing.

CLB inputs and outputs are distributed on all four sides, providing maximum routing flexibility. In general, the entire architecture is symmetrical and regular. It is well suited to established placement and routing algorithms. Inputs, outputs, and function generators can freely swap positions within a CLB to avoid routing congestion during the placement and routing operation.





x5994

**Figure 25: High-Level Routing Diagram of XC4000 Series CLB (shaded arrows indicate XC4000X only)**

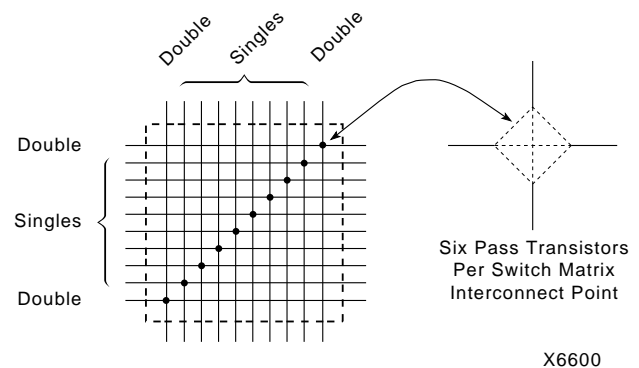
**Table 14: Routing per CLB in XC4000 Series Devices**

	XC4000E		XC4000X	
	Vertical	Horizontal	Vertical	Horizontal
Singles	8	8	8	8
Doubles	4	4	4	4
Quads	0	0	12	12
Longlines	6	6	10	6
Direct Connects	0	0	2	2
Globals	4	0	8	0
Carry Logic	2	0	1	0
Total	24	18	45	32

### Programmable Switch Matrices

The horizontal and vertical single- and double-length lines intersect at a box called a programmable switch matrix (PSM). Each switch matrix consists of programmable pass transistors used to establish connections between the lines (see Figure 26).

For example, a single-length signal entering on the right side of the switch matrix can be routed to a single-length line on the top, left, or bottom sides, or any combination thereof, if multiple branches are required. Similarly, a double-length signal can be routed to a double-length line on any or all of the other three edges of the programmable switch matrix.



X6600

**Figure 26: Programmable Switch Matrix (PSM)**

### Single-Length Lines

Single-length lines provide the greatest interconnect flexibility and offer fast routing between adjacent blocks. There are eight vertical and eight horizontal single-length lines associated with each CLB. These lines connect the switching matrices that are located in every row and a column of CLBs.

Single-length lines are connected by way of the programmable switch matrices, as shown in Figure 28. Routing connectivity is shown in Figure 27.

Single-length lines incur a delay whenever they go through a switching matrix. Therefore, they are not suitable for routing signals for long distances. They are normally used to conduct signals within a localized area and to provide the branching for nets with fanout greater than one.

circuit prevents undefined floating levels. However, it is overridden by any driver, even a pull-up resistor.

Each XC4000E longline has a programmable splitter switch at its center, as does each XC4000X longline driven by TBUFs. This switch can separate the line into two independent routing channels, each running half the width or height of the array.

Each XC4000X longline not driven by TBUFs has a buffered programmable splitter switch at the 1/4, 1/2, and 3/4 points of the array. Due to the buffering, XC4000X longline performance does not deteriorate with the larger array sizes. If the longline is split, the resulting partial longlines are independent.

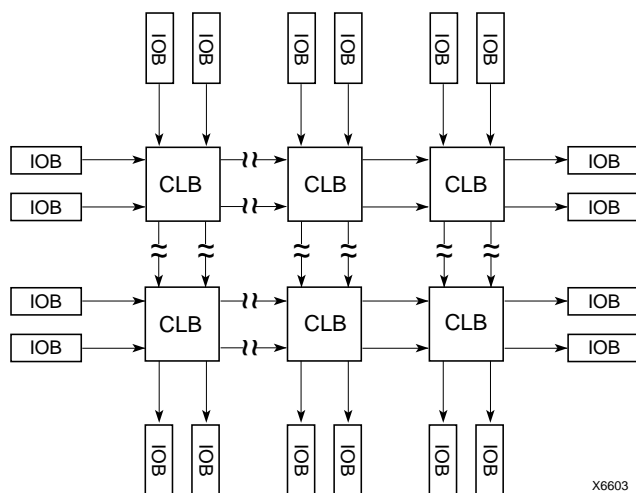
Routing connectivity of the longlines is shown in [Figure 27 on page 30](#).

### **Direct Interconnect (XC4000X only)**

The XC4000X offers two direct, efficient and fast connections between adjacent CLBs. These nets facilitate a data flow from the left to the right side of the device, or from the top to the bottom, as shown in [Figure 30](#). Signals routed on the direct interconnect exhibit minimum interconnect propagation delay and use no general routing resources.

The direct interconnect is also present between CLBs and adjacent IOBs. Each IOB on the left and top device edges has a direct path to the nearest CLB. Each CLB on the right and bottom edges of the array has a direct path to the nearest two IOBs, since there are two IOBs for each row or column of CLBs.

The place and route software uses direct interconnect whenever possible, to maximize routing resources and minimize interconnect delays.



**Figure 30: XC4000X Direct Interconnect**

### **I/O Routing**

XC4000 Series devices have additional routing around the IOB ring. This routing is called a VersaRing. The VersaRing facilitates pin-swapping and redesign without affecting board layout. Included are eight double-length lines spanning two CLBs (four IOBs), and four longlines. Global lines and Wide Edge Decoder lines are provided. XC4000X devices also include eight octal lines.

A high-level diagram of the VersaRing is shown in [Figure 31](#). The shaded arrows represent routing present only in XC4000X devices.

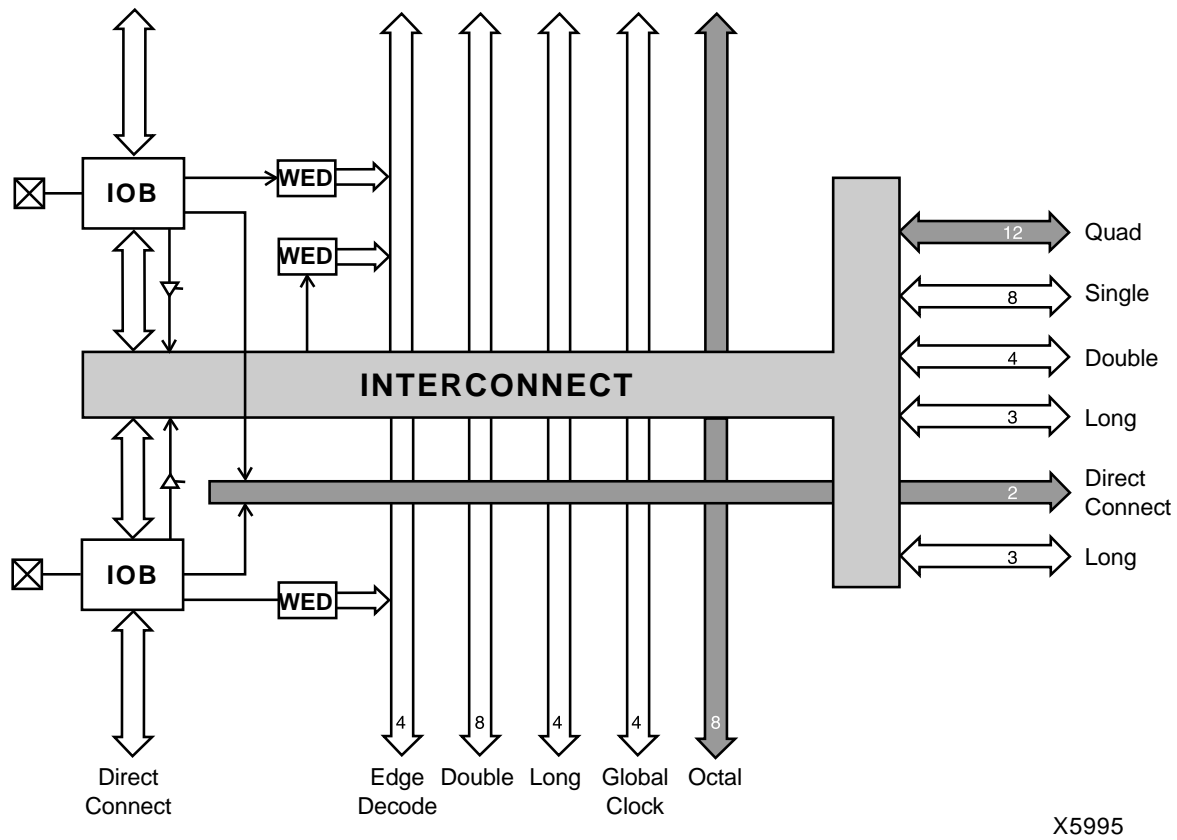
[Figure 33 on page 34](#) is a detailed diagram of the XC4000E and XC4000X VersaRing. The area shown includes two IOBs. There are two IOBs per CLB row or column, therefore this diagram corresponds to the CLB routing diagram shown in [Figure 27 on page 30](#). The shaded areas represent routing and routing connections present only in XC4000X devices.

### **Octal I/O Routing (XC4000X only)**

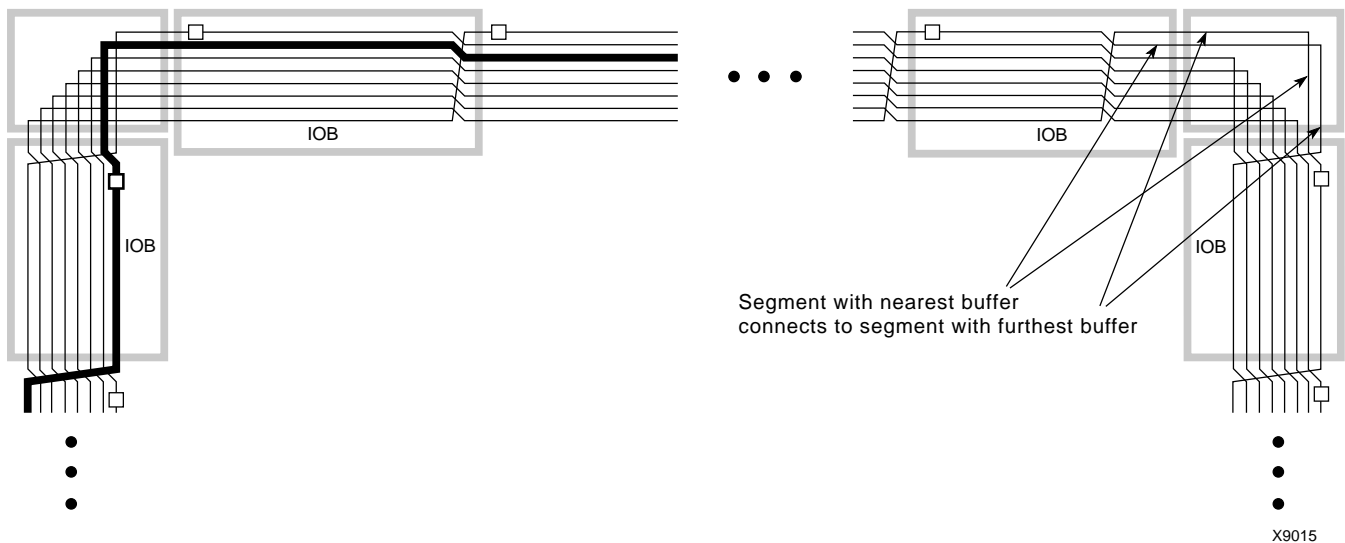
Between the XC4000X CLB array and the pad ring, eight interconnect tracks provide for versatility in pin assignment and fixed pinout flexibility. (See [Figure 32 on page 33](#).)

These routing tracks are called octals, because they can be broken every eight CLBs (sixteen IOBs) by a programmable buffer that also functions as a splitter switch. The buffers are staggered, so each line goes through a buffer at every eighth CLB location around the device edge.

The octal lines bend around the corners of the device. The lines cross at the corners in such a way that the segment most recently buffered before the turn has the farthest distance to travel before the next buffer, as shown in [Figure 32](#).



**Figure 31: High-Level Routing Diagram of XC4000 Series VersaRing (Left Edge)**  
WED = Wide Edge Decoder, IOB = I/O Block (shaded arrows indicate XC4000X only)



**Figure 32: XC4000X Octal I/O Routing**

The top and bottom Global Early buffers are about 1 ns slower clock to out than the left and right Global Early buffers.

The Global Early buffers can be driven by either semi-dedicated pads or internal logic. They share pads with the Global Low-Skew buffers, so a single net can drive both global buffers, as described above.

To use a Global Early buffer, place a BUFGE element in a schematic or in HDL code. If desired, attach a LOC attribute or property to direct placement to the designated location. For example, attach a LOC=T attribute or property to direct that a BUFGE be placed in one of the two Global Early buffers on the top edge of the device, or a LOC=TR to indicate the Global Early buffer on the top edge of the device, on the right.

## Power Distribution

Power for the FPGA is distributed through a grid to achieve high noise immunity and isolation between logic and I/O. Inside the FPGA, a dedicated Vcc and Ground ring surrounding the logic array provides power to the I/O drivers, as shown in [Figure 39](#). An independent matrix of Vcc and Ground lines supplies the interior logic of the device.

This power distribution grid provides a stable supply and ground for all internal logic, providing the external package power pins are all connected and appropriately de-coupled. Typically, a 0.1  $\mu$ F capacitor connected between each Vcc pin and the board's Ground plane will provide adequate de-coupling.

Output buffers capable of driving/sinking the specified 12 mA loads under specified worst-case conditions may be capable of driving/sinking up to 10 times as much current under best case conditions.

Noise can be reduced by minimizing external load capacitance and reducing simultaneous output transitions in the same direction. It may also be beneficial to locate heavily loaded output buffers near the Ground pads. The I/O Block output buffers have a slew-rate limited mode (default) which should be used where output rise and fall times are not speed-critical.

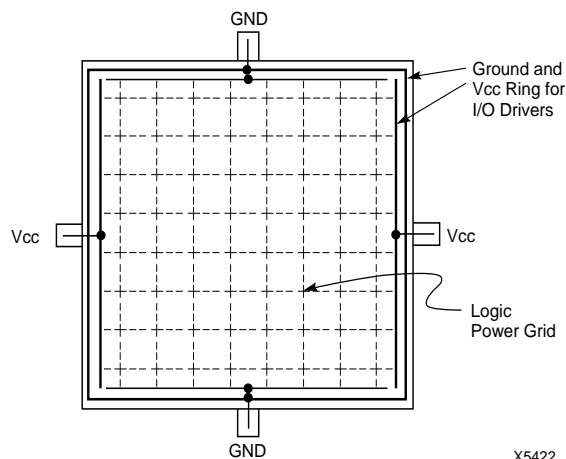


Figure 39: XC4000 Series Power Distribution

## Pin Descriptions

There are three types of pins in the XC4000 Series devices:

- Permanently dedicated pins
- User I/O pins that can have special functions
- Unrestricted user-programmable I/O pins.

Before and during configuration, all outputs not used for the configuration process are 3-stated with a 50 k $\Omega$  - 100 k $\Omega$  pull-up resistor.

After configuration, if an IOB is unused it is configured as an input with a 50 k $\Omega$  - 100 k $\Omega$  pull-up resistor.

XC4000 Series devices have no dedicated Reset input. Any user I/O can be configured to drive the Global Set/Reset net, GSR. See ["Global Set/Reset" on page 11](#) for more information on GSR.

XC4000 Series devices have no Powerdown control input, as the XC3000 and XC2000 families do. The XC3000/XC2000 Powerdown control also 3-stated all of the device

I/O pins. For XC4000 Series devices, use the global 3-state net, GTS, instead. This net 3-states all outputs, but does not place the device in low-power mode. See ["IOB Output Signals" on page 23](#) for more information on GTS.

Device pins for XC4000 Series devices are described in [Table 16](#). Pin functions during configuration for each of the seven configuration modes are summarized in [Table 22 on page 58](#), in the "Configuration Timing" section.

**Table 16: Pin Descriptions (Continued)**

Pin Name	I/O During Config.	I/O After Config.	Pin Description
$\overline{CS0}$ , CS1, $\overline{WS}$ , $\overline{RS}$	I	I/O	These four inputs are used in Asynchronous Peripheral mode. The chip is selected when $\overline{CS0}$ is Low and CS1 is High. While the chip is selected, a Low on Write Strobe ( $\overline{WS}$ ) loads the data present on the D0 - D7 inputs into the internal data buffer. A Low on Read Strobe ( $\overline{RS}$ ) changes D7 into a status output — High if Ready, Low if Busy — and drives D0 - D6 High. In Express mode, CS1 is used as a serial-enable signal for daisy-chaining. $\overline{WS}$ and $\overline{RS}$ should be mutually exclusive, but if both are Low simultaneously, the Write Strobe overrides. After configuration, these are user-programmable I/O pins.
A0 - A17	O	I/O	During Master Parallel configuration, these 18 output pins address the configuration EPROM. After configuration, they are user-programmable I/O pins.
A18 - A21 (XC4003XL to XC4085XL)	O	I/O	During Master Parallel configuration with an XC4000X master, these 4 output pins add 4 more bits to address the configuration EPROM. After configuration, they are user-programmable I/O pins. (See Master Parallel Configuration section for additional details.)
D0 - D7	I	I/O	During Master Parallel and Peripheral configuration, these eight input pins receive configuration data. After configuration, they are user-programmable I/O pins.
DIN	I	I/O	During Slave Serial or Master Serial configuration, DIN is the serial configuration data input receiving data on the rising edge of CCLK. During Parallel configuration, DIN is the D0 input. After configuration, DIN is a user-programmable I/O pin.
DOUT	O	I/O	During configuration in any mode but Express mode, DOUT is the serial configuration data output that can drive the DIN of daisy-chained slave FPGAs. DOUT data changes on the falling edge of CCLK, one-and-a-half CCLK periods after it was received at the DIN input. In Express mode for XC4000E and XC4000X only, DOUT is the status output that can drive the CS1 of daisy-chained FPGAs, to enable and disable downstream devices. After configuration, DOUT is a user-programmable I/O pin.
<b>Unrestricted User-Programmable I/O Pins</b>			
I/O	Weak Pull-up	I/O	These pins can be configured to be input and/or output after configuration is completed. Before configuration is completed, these pins have an internal high-value pull-up resistor (25 k $\Omega$ - 100 k $\Omega$ ) that defines the logic level as High.

## Boundary Scan

The 'bed of nails' has been the traditional method of testing electronic assemblies. This approach has become less appropriate, due to closer pin spacing and more sophisticated assembly methods like surface-mount technology and multi-layer boards. The IEEE Boundary Scan Standard 1149.1 was developed to facilitate board-level testing of electronic assemblies. Design and test engineers can imbed a standard test logic structure in their device to achieve high fault coverage for I/O and internal logic. This structure is easily implemented with a four-pin interface on any boundary scan-compatible IC. IEEE 1149.1-compatible devices may be serial daisy-chained together, connected in parallel, or a combination of the two.

The XC4000 Series implements IEEE 1149.1-compatible BYPASS, PRELOAD/SAMPLE and EXTEST boundary scan instructions. When the boundary scan configuration option is selected, three normal user I/O pins become dedicated inputs for these functions. Another user output pin becomes the dedicated boundary scan output. The details

of how to enable this circuitry are covered later in this section.

By exercising these input signals, the user can serially load commands and data into these devices to control the driving of their outputs and to examine their inputs. This method is an improvement over bed-of-nails testing. It avoids the need to over-drive device outputs, and it reduces the user interface to four pins. An optional fifth pin, a reset for the control logic, is described in the standard but is not implemented in Xilinx devices.

The dedicated on-chip logic implementing the IEEE 1149.1 functions includes a 16-state machine, an instruction register and a number of data registers. The functional details can be found in the IEEE 1149.1 specification and are also discussed in the Xilinx application note XAPP 017: "*Boundary Scan in XC4000 Devices*."

Figure 40 on page 43 shows a simplified block diagram of the XC4000E Input/Output Block with boundary scan implemented. XC4000X boundary scan logic is identical.



## Configuration Modes

XC4000E devices have six configuration modes. XC4000X devices have the same six modes, plus an additional configuration mode. These modes are selected by a 3-bit input code applied to the M2, M1, and M0 inputs. There are three self-loading Master modes, two Peripheral modes, and a Serial Slave mode, which is used primarily for daisy-chained devices. The coding for mode selection is shown in [Table 18](#).

**Table 18: Configuration Modes**

Mode	M2	M1	M0	CCLK	Data
Master Serial	0	0	0	output	Bit-Serial
Slave Serial	1	1	1	input	Bit-Serial
Master Parallel Up	1	0	0	output	Byte-Wide, increment from 00000
Master Parallel Down	1	1	0	output	Byte-Wide, decrement from 3FFFF
Peripheral Synchronous*	0	1	1	input	Byte-Wide
Peripheral Asynchronous	1	0	1	output	Byte-Wide
Reserved	0	1	0	—	—
Reserved	0	0	1	—	—

\* Can be considered byte-wide Slave Parallel

A detailed description of each configuration mode, with timing information, is included later in this data sheet. During configuration, some of the I/O pins are used temporarily for the configuration process. All pins used during configuration are shown in [Table 22 on page 58](#).

### Master Modes

The three Master modes use an internal oscillator to generate a Configuration Clock (CCLK) for driving potential slave devices. They also generate address and timing for external PROM(s) containing the configuration data.

Master Parallel (Up or Down) modes generate the CCLK signal and PROM addresses and receive byte parallel data. The data is internally serialized into the FPGA data-frame format. The up and down selection generates starting addresses at either zero or 3FFFF (3FFFFFF when 22 address lines are used), for compatibility with different microprocessor addressing conventions. The Master Serial mode generates CCLK and receives the configuration data in serial form from a Xilinx serial-configuration PROM.

CCLK speed is selectable as either 1 MHz (default) or 8 MHz. Configuration always starts at the default slow frequency, then can switch to the higher frequency during the first frame. Frequency tolerance is -50% to +25%.

### Additional Address lines in XC4000 devices

The XC4000X devices have additional address lines (A18-A21) allowing the additional address space required to daisy-chain several large devices.

The extra address lines are programmable in XC4000EX devices. By default these address lines are not activated. In the default mode, the devices are compatible with existing XC4000 and XC4000E products. If desired, the extra address lines can be used by specifying the address lines option in bitgen as 22 (bitgen -g AddressLines:22). The lines (A18-A21) are driven when a master device detects, via the bitstream, that it should be using all 22 address lines. Because these pins will initially be pulled high by internal pull-ups, designers using Master Parallel Up mode should use external pull down resistors on pins A18-A21. If Master Parallel Down mode is used external resistors are not necessary.

All 22 address lines are always active in Master Parallel modes with XC4000XL devices. The additional address lines behave identically to the lower order address lines. If the Address Lines option in bitgen is set to 18, it will be ignored by the XC4000XL device.

The additional address lines (A18-A21) are not available in the PC84 package.

### Peripheral Modes

The two Peripheral modes accept byte-wide data from a bus. A RDY/BUSY status is available as a handshake signal. In Asynchronous Peripheral mode, the internal oscillator generates a CCLK burst signal that serializes the byte-wide data. CCLK can also drive slave devices. In the synchronous mode, an externally supplied clock input to CCLK serializes the data.

### Slave Serial Mode

In Slave Serial mode, the FPGA receives serial configuration data on the rising edge of CCLK and, after loading its configuration, passes additional data out, resynchronized on the next falling edge of CCLK.

Multiple slave devices with identical configurations can be wired with parallel DIN inputs. In this way, multiple devices can be configured simultaneously.

### Serial Daisy Chain

Multiple devices with different configurations can be connected together in a "daisy chain," and a single combined bitstream used to configure the chain of slave devices.

To configure a daisy chain of devices, wire the CCLK pins of all devices in parallel, as shown in [Figure 51 on page 60](#). Connect the DOUT of each device to the DIN of the next. The lead or master FPGA and following slaves each passes resynchronized configuration data coming from a single source. The header data, including the length count,

is passed through and is captured by each FPGA when it recognizes the 0010 preamble. Following the length-count data, each FPGA outputs a High on DOUT until it has received its required number of data frames.

After an FPGA has received its configuration data, it passes on any additional frame start bits and configuration data on DOUT. When the total number of configuration clocks applied after memory initialization equals the value of the 24-bit length count, the FPGAs begin the start-up sequence and become operational together. FPGA I/O are normally released two CCLK cycles after the last configuration bit is received. **Figure 47 on page 53** shows the start-up timing for an XC4000 Series device.

The daisy-chained bitstream is not simply a concatenation of the individual bitstreams. The PROM file formatter must be used to combine the bitstreams for a daisy-chained configuration.

### Multi-Family Daisy Chain

All Xilinx FPGAs of the XC2000, XC3000, and XC4000 Series use a compatible bitstream format and can, therefore, be connected in a daisy chain in an arbitrary sequence. There is, however, one limitation. The lead device must belong to the highest family in the chain. If the chain contains XC4000 Series devices, the master normally cannot be an XC2000 or XC3000 device.

The reason for this rule is shown in **Figure 47 on page 53**. Since all devices in the chain store the same length count value and generate or receive one common sequence of CCLK pulses, they all recognize length-count match on the same CCLK edge, as indicated on the left edge of **Figure 47**. The master device then generates additional CCLK pulses until it reaches its finish point F. The different families generate or require different numbers of additional CCLK pulses until they reach F. Not reaching F means that the device does not really finish its configuration, although DONE may have gone High, the outputs became active, and the internal reset was released. For the XC4000 Series device, not reaching F means that readback cannot be ini-

tiated and most boundary scan instructions cannot be used.

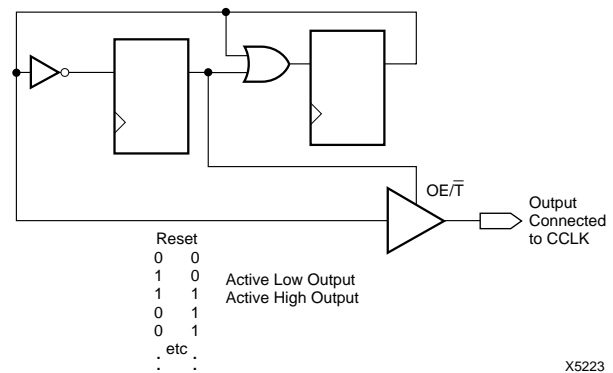
The user has some control over the relative timing of these events and can, therefore, make sure that they occur at the proper time and the finish point F is reached. Timing is controlled using options in the bitstream generation software.

### XC3000 Master with an XC4000 Series Slave

Some designers want to use an inexpensive lead device in peripheral mode and have the more precious I/O pins of the XC4000 Series devices all available for user I/O. **Figure 44** provides a solution for that case.

This solution requires one CLB, one IOB and pin, and an internal oscillator with a frequency of up to 5 MHz as a clock source. The XC3000 master device must be configured with late Internal Reset, which is the default option.

One CLB and one IOB in the lead XC3000-family device are used to generate the additional CCLK pulse required by the XC4000 Series devices. When the lead device removes the internal RESET signal, the 2-bit shift register responds to its clock input and generates an active Low output signal for the duration of the subsequent clock period. An external connection between this output and CCLK thus creates the extra CCLK pulse.



**Figure 44: CCLK Generation for XC3000 Master Driving an XC4000 Series Slave**

## Setting CCLK Frequency

For Master modes, CCLK can be generated in either of two frequencies. In the default slow mode, the frequency ranges from 0.5 MHz to 1.25 MHz for XC4000E and XC4000EX devices and from 0.6 MHz to 1.8 MHz for XC4000XL devices. In fast CCLK mode, the frequency ranges from 4 MHz to 10 MHz for XC4000E/EX devices and from 5 MHz to 15 MHz for XC4000XL devices. The frequency is selected by an option when running the bitstream generation software. If an XC4000 Series Master is driving an XC3000- or XC2000-family slave, slow CCLK mode must be used. In addition, an XC4000XL device driving a XC4000E or XC4000EX should use slow mode. Slow mode is the default.

**Table 19: XC4000 Series Data Stream Formats**

Data Type	All Other Modes (D0...)
Fill Byte	11111111b
Preamble Code	0010b
Length Count	COUNT(23:0)
Fill Bits	1111b
Start Field	0b
Data Frame	DATA(n-1:0)
CRC or Constant Field Check	xxxx (CRC) or 0110b
Extend Write Cycle	—
Postamble	01111111b
Start-Up Bytes	xxh
Legend:	
Not shaded	Once per bitstream
Light	Once per data frame
Dark	Once per device

## Data Stream Format

The data stream (“bitstream”) format is identical for all configuration modes.

The data stream formats are shown in [Table 19](#). Bit-serial data is read from left to right, and byte-parallel data is effectively assembled from this serial bitstream, with the first bit in each byte assigned to D0.

The configuration data stream begins with a string of eight ones, a preamble code, followed by a 24-bit length count and a separator field of ones. This header is followed by the actual configuration data in frames. The length and number of frames depends on the device type (see [Table 20](#) and [Table 21](#)). Each frame begins with a start field and ends with an error check. A postamble code is required to signal the end of data for a single device. In all cases, additional start-up bytes of data are required to provide four clocks for the startup sequence at the end of configuration. Long daisy chains require additional startup bytes to shift the last data through the chain. All startup bytes are don't-cares; these bytes are not included in bitstreams created by the Xilinx software.

A selection of CRC or non-CRC error checking is allowed by the bitstream generation software. The non-CRC error checking tests for a designated end-of-frame field for each frame. For CRC error checking, the software calculates a running CRC and inserts a unique four-bit partial check at the end of each frame. The 11-bit CRC check of the last frame of an FPGA includes the last seven data bits.

Detection of an error results in the suspension of data loading and the pulling down of the  $\overline{\text{INIT}}$  pin. In Master modes, CCLK and address signals continue to operate externally. The user must detect  $\overline{\text{INIT}}$  and initialize a new configuration by pulsing the  $\overline{\text{PROGRAM}}$  pin Low or cycling Vcc.

Low. During this time delay, or as long as the  $\overline{\text{PROGRAM}}$  input is asserted, the configuration logic is held in a Configuration Memory Clear state. The configuration-memory frames are consecutively initialized, using the internal oscillator.

At the end of each complete pass through the frame addressing, the power-on time-out delay circuitry and the level of the  $\overline{\text{PROGRAM}}$  pin are tested. If neither is asserted, the logic initiates one additional clearing of the configuration frames and then tests the  $\overline{\text{INIT}}$  input.

### Initialization

During initialization and configuration, user pins  $\text{HDC}$ ,  $\overline{\text{LDC}}$ ,  $\overline{\text{INIT}}$  and  $\text{DONE}$  provide status outputs for the system interface. The outputs  $\overline{\text{LDC}}$ ,  $\overline{\text{INIT}}$  and  $\text{DONE}$  are held Low and  $\text{HDC}$  is held High starting at the initial application of power.

The open drain  $\overline{\text{INIT}}$  pin is released after the final initialization pass through the frame addresses. There is a deliberate delay of 50 to 250  $\mu\text{s}$  (up to 10% longer for low-voltage devices) before a Master-mode device recognizes an inactive  $\overline{\text{INIT}}$ . Two internal clocks after the  $\overline{\text{INIT}}$  pin is recognized as High, the FPGA samples the three mode lines to determine the configuration mode. The appropriate interface lines become active and the configuration preamble and data can be loaded. Configuration

The 0010 preamble code indicates that the following 24 bits represent the length count. The length count is the total number of configuration clocks needed to load the complete configuration data. (Four additional configuration clocks are required to complete the configuration process, as discussed below.) After the preamble and the length count have been passed through to all devices in the daisy chain,  $\text{DOUT}$  is held High to prevent frame start bits from reaching any daisy-chained devices.

A specific configuration bit, early in the first frame of a master device, controls the configuration-clock rate and can increase it by a factor of eight. Therefore, if a fast configuration clock is selected by the bitstream, the slower clock rate is used until this configuration bit is detected.

Each frame has a start field followed by the frame-configuration data bits and a frame error field. If a frame data error is detected, the FPGA halts loading, and signals the error by pulling the open-drain  $\overline{\text{INIT}}$  pin Low. After all configuration frames have been loaded into an FPGA,  $\text{DOUT}$  again follows the input data so that the remaining data is passed on to the next device.

### Delaying Configuration After Power-Up

There are two methods of delaying configuration after power-up: put a logic Low on the  $\overline{\text{PROGRAM}}$  input, or pull the bidirectional  $\overline{\text{INIT}}$  pin Low, using an open-collector (open-drain) driver. (See [Figure 46 on page 50](#).)

A Low on the  $\overline{\text{PROGRAM}}$  input is the more radical approach, and is recommended when the power-supply

rise time is excessive or poorly defined. As long as  $\overline{\text{PROGRAM}}$  is Low, the FPGA keeps clearing its configuration memory. When  $\overline{\text{PROGRAM}}$  goes High, the configuration memory is cleared one more time, followed by the beginning of configuration, provided the  $\overline{\text{INIT}}$  input is not externally held Low. Note that a Low on the  $\overline{\text{PROGRAM}}$  input automatically forces a Low on the  $\overline{\text{INIT}}$  output. The XC4000 Series  $\overline{\text{PROGRAM}}$  pin has a permanent weak pull-up.

Using an open-collector or open-drain driver to hold  $\overline{\text{INIT}}$  Low before the beginning of configuration causes the FPGA to wait after completing the configuration memory clear operation. When  $\overline{\text{INIT}}$  is no longer held Low externally, the device determines its configuration mode by capturing its mode pins, and is ready to start the configuration process. A master device waits up to an additional 250  $\mu\text{s}$  to make sure that any slaves in the optional daisy chain have seen that  $\overline{\text{INIT}}$  is High.

### Start-Up

Start-up is the transition from the configuration process to the intended user operation. This transition involves a change from one clock source to another, and a change from interfacing parallel or serial configuration data where most outputs are 3-stated, to normal operation with I/O pins active in the user-system. Start-up must make sure that the user-logic 'wakes up' gracefully, that the outputs become active without causing contention with the configuration signals, and that the internal flip-flops are released from the global Reset or Set at the right time.

[Figure 47](#) describes start-up timing for the three Xilinx families in detail. The configuration modes can use any of the four timing sequences.

To access the internal start-up signals, place the  $\text{STARTUP}$  library symbol.

### Start-up Timing

Different FPGA families have different start-up sequences.

The XC2000 family goes through a fixed sequence.  $\text{DONE}$  goes High and the internal global Reset is de-activated one CCLK period after the I/O become active.

The XC3000A family offers some flexibility.  $\text{DONE}$  can be programmed to go High one CCLK period before or after the I/O become active. Independent of  $\text{DONE}$ , the internal global Reset is de-activated one CCLK period before or after the I/O become active.

The XC4000 Series offers additional flexibility. The three events —  $\text{DONE}$  going High, the internal Set/Reset being de-activated, and the user I/O going active — can all occur in any arbitrary sequence. Each of them can occur one CCLK period before or after, or simultaneous with, any of the others. This relative timing is selected by means of software options in the bitstream generation software.



### **Start-up from a User Clock (STARTUP.CLK)**

When, instead of CCLK, a user-supplied start-up clock is selected, Q1 is used to bridge the unknown phase relationship between CCLK and the user clock. This arbitration causes an unavoidable one-cycle uncertainty in the timing of the rest of the start-up sequence.

### **DONE Goes High to Signal End of Configuration**

XC4000 Series devices read the expected length count from the bitstream and store it in an internal register. The length count varies according to the number of devices and the composition of the daisy chain. Each device also counts the number of CCLKs during configuration.

Two conditions have to be met in order for the DONE pin to go high:

- the chip's internal memory must be full, and
- the configuration length count must be met, *exactly*.

This is important because the counter that determines when the length count is met begins with the very first CCLK, not the first one after the preamble.

Therefore, if a stray bit is inserted before the preamble, or the data source is not ready at the time of the first CCLK, the internal counter that holds the number of CCLKs will be one ahead of the actual number of data bits read. At the end of configuration, the configuration memory will be full, but the number of bits in the internal counter will not match the expected length count.

As a consequence, a Master mode device will continue to send out CCLKs until the internal counter turns over to zero, and then reaches the correct length count a second time. This will take several seconds [ $2^{24} * \text{CCLK period}$ ] — which is sometimes interpreted as the device not configuring at all.

If it is not possible to have the data ready at the time of the first CCLK, the problem can be avoided by increasing the number in the length count by the appropriate value. The *XACT User Guide* includes detailed information about manually altering the length count.

Note that DONE is an open-drain output and does not go High unless an internal pull-up is activated or an external pull-up is attached. The internal pull-up is activated as the default by the bitstream generation software.

### **Release of User I/O After DONE Goes High**

By default, the user I/O are released one CCLK cycle after the DONE pin goes High. If CCLK is not clocked after DONE goes High, the outputs remain in their initial state — 3-stated, with a 50 k $\Omega$  - 100 k $\Omega$  pull-up. The delay from DONE High to active user I/O is controlled by an option to the bitstream generation software.

### **Release of Global Set/Reset After DONE Goes High**

By default, Global Set/Reset (GSR) is released two CCLK cycles after the DONE pin goes High. If CCLK is not clocked twice after DONE goes High, all flip-flops are held in their initial set or reset state. The delay from DONE High to GSR inactive is controlled by an option to the bitstream generation software.

### **Configuration Complete After DONE Goes High**

Three full CCLK cycles are required after the DONE pin goes High, as shown in [Figure 47 on page 53](#). If CCLK is not clocked three times after DONE goes High, readback cannot be initiated and most boundary scan instructions cannot be used.

### **Configuration Through the Boundary Scan Pins**

XC4000 Series devices can be configured through the boundary scan pins. The basic procedure is as follows:

- Power up the FPGA with  $\overline{\text{INIT}}$  held Low (or drive the  $\overline{\text{PROGRAM}}$  pin Low for more than 300 ns followed by a High while holding  $\overline{\text{INIT}}$  Low). Holding  $\overline{\text{INIT}}$  Low allows enough time to issue the CONFIG command to the FPGA. The pin can be used as I/O after configuration if a resistor is used to hold  $\overline{\text{INIT}}$  Low.
- Issue the CONFIG command to the TMS input
- Wait for  $\overline{\text{INIT}}$  to go High
- Sequence the boundary scan Test Access Port to the SHIFT-DR state
- Toggle TCK to clock data into TDI pin.

The user must account for all TCK clock cycles after INIT goes High, as all of these cycles affect the Length Count compare.

For more detailed information, refer to the Xilinx application note XAPP017, “*Boundary Scan in XC4000 Devices*.” This application note also applies to XC4000E and XC4000X devices.



## Synchronous Peripheral Mode

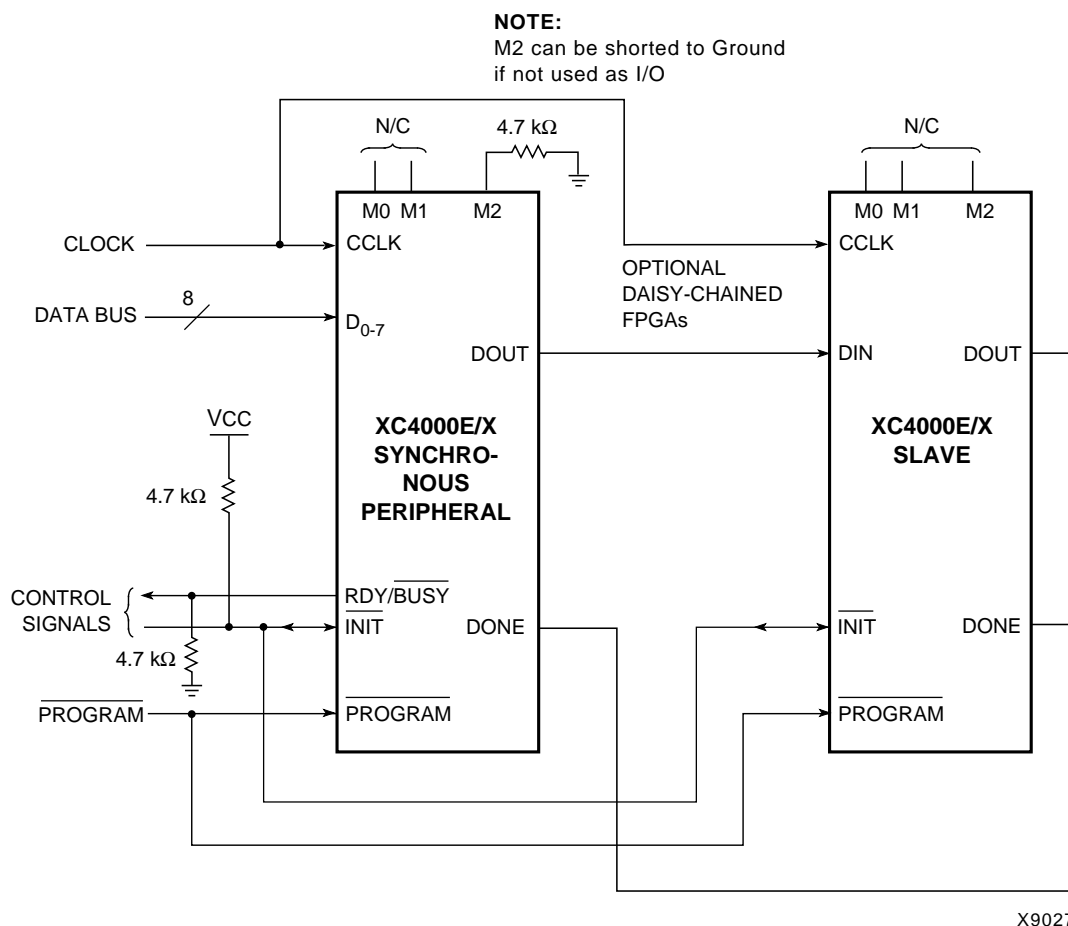
Synchronous Peripheral mode can also be considered Slave Parallel mode. An external signal drives the CCLK input(s) of the FPGA(s). The first byte of parallel configuration data must be available at the Data inputs of the lead FPGA a short setup time before the rising CCLK edge. Subsequent data bytes are clocked in on every eighth consecutive rising CCLK edge.

The same CCLK edge that accepts data, also causes the RDY/ $\overline{\text{BUSY}}$  output to go High for one CCLK period. The pin name is a misnomer. In Synchronous Peripheral mode it is really an ACKNOWLEDGE signal. Synchronous operation does not require this response, but it is a meaningful signal for test purposes. Note that RDY/ $\overline{\text{BUSY}}$  is pulled High with a high-impedance pullup prior to  $\overline{\text{INIT}}$  going High.

The lead FPGA serializes the data and presents the preamble data (and all data that overflows the lead device) on its DOUT pin. There is an internal delay of 1.5 CCLK periods, which means that DOUT changes on the falling CCLK edge, and the next FPGA in the daisy chain accepts data on the subsequent rising CCLK edge.

In order to complete the serial shift operation, 10 additional CCLK rising edges are required after the last data byte has been loaded, plus one more CCLK cycle for each daisy-chained device.

Synchronous Peripheral mode is selected by a <011> on the mode pins (M2, M1, M0).



**Figure 56: Synchronous Peripheral Mode Circuit Diagram**

## Asynchronous Peripheral Mode

## Write to FPGA

Asynchronous Peripheral mode uses the trailing edge of the logic AND condition of  $\overline{WS}$  and  $\overline{CS0}$  being Low and  $RS$  and  $CS1$  being High to accept byte-wide data from a micro-processor bus. In the lead FPGA, this data is loaded into a double-buffered UART-like parallel-to-serial converter and is serially shifted into the internal logic.

The lead FPGA presents the preamble data (and all data that overflows the lead device) on its DOUT pin. The RDY/ $\overline{\text{BUSY}}$  output from the lead FPGA acts as a handshake signal to the microprocessor. RDY/ $\overline{\text{BUSY}}$  goes Low when a byte has been received, and goes High again when the byte-wide input buffer has transferred its information into the shift register, and the buffer is ready to receive new data. A new write may be started immediately, as soon as the RDY/ $\overline{\text{BUSY}}$  output has gone Low, acknowledging receipt of the previous data. Write may not be terminated until RDY/ $\overline{\text{BUSY}}$  is High again for one CCLK period. Note that RDY/ $\overline{\text{BUSY}}$  is pulled High with a high-impedance pull-up prior to  $\overline{\text{INIT}}$  going High.

The length of the  $\overline{\text{BUSY}}$  signal depends on the activity in the UART. If the shift register was empty when the new byte was received, the  $\overline{\text{BUSY}}$  signal lasts for only two CCLK periods. If the shift register was still full when the new byte was received, the  $\overline{\text{BUSY}}$  signal can be as long as nine CCLK periods.

Note that after the last byte has been entered, only seven of its bits are shifted out. CCLK remains High with DOUT equal to bit 6 (the next-to-last bit) of the last byte entered.

The READY/ $\overline{\text{BUSY}}$  handshake can be ignored if the delay from any one Write to the end of the next Write is guaranteed to be longer than 10 CCLK periods.

**Status Read**

The logic AND condition of the  $\overline{CS0}$ , CS1 and  $\overline{RS}$  inputs puts the device status on the Data bus.

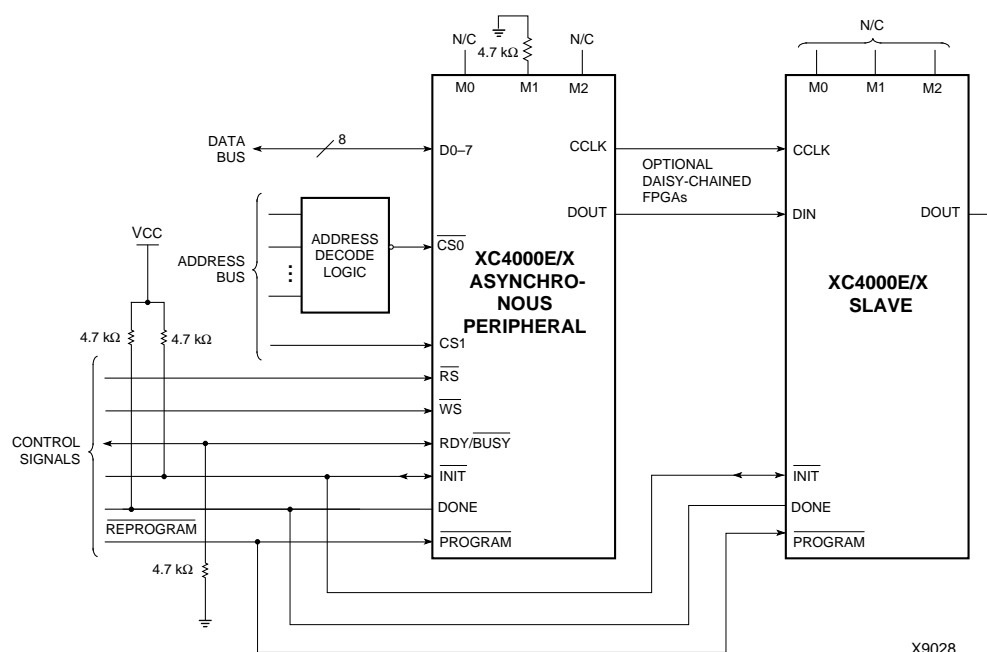
- D7 High indicates Ready
- D7 Low indicates Busy
- D0 through D6 go unconditionally High

It is mandatory that the whole start-up sequence be started and completed by one byte-wide input. Otherwise, the pins used as Write Strobe or Chip Enable might become active outputs and interfere with the final byte transfer. If this transfer does not occur, the start-up sequence is not completed all the way to the finish (point F in [Figure 47 on page 53](#)).

In this case, at worst, the internal reset is not released. At best, Readback and Boundary Scan are inhibited. The length-count value, as generated by the XACT<sup>step</sup> software, ensures that these problems never occur.

Although RDY/ $\overline{\text{BUSY}}$  is brought out as a separate signal, microprocessors can more easily read this information on one of the data lines. For this purpose, D7 represents the RDY/ $\overline{\text{BUSY}}$  status when  $\overline{\text{RS}}$  is Low,  $\overline{\text{WS}}$  is High, and the two chip select lines are both active.

Asynchronous Peripheral mode is selected by a <101> on the mode pins (M2, M1, M0).



**Figure 58: Asynchronous Peripheral Mode Circuit Diagram**

**Table 25: Component Availability Chart for XC4000E FPGAs**

	PINS	TYPE	CODE	84	100	100	120	144	156	160	191	208	208	223	225	240	240	299	304
				Plast. PLCC	Plast. PQFP	Plast. VQFP	Ceram. PGA	Plast. TQFP	Ceram. PGA	Plast. PQFP	Ceram. PGA	High-Perf. QFP	Plast. PQFP	Ceram. PGA	Plast. BGA	High-Perf. QFP	Plast. PQFP	Ceram. PGA	High-Perf. QF
				PC84	PQ100	VQ100	PG120	TQ144	PG156	PQ160	PG191	HQ208	PQ208	PG223	BG225	HQ240	PQ240	PG299	HQ304
XC4003E	-4	C I	C I	C I	C I														
	-3	C I	C I	C I	C I														
	-2	C I	C I	C I	C I														
	-1	C	C	C	C														
XC4005E	-4	C I	C I					C I	C I	C I			C I						
	-3	C I	C I					C I	C I	C I			C I						
	-2	C I	C I					C I	C I	C I			C I						
	-1	C	C					C	C	C			C						
XC4006E	-4	C I						C I	C I	C I			C I						
	-3	C I						C I	C I	C I			C I						
	-2	C I						C I	C I	C I			C I						
	-1	C						C	C	C			C						
XC4008E	-4	C I								C I	C I		C I						
	-3	C I								C I	C I		C I						
	-2	C I								C I	C I		C I						
	-1	C								C	C		C						
XC4010E	-4	C I								C I	C I	C I	C I			C I			
	-3	C I								C I	C I	C I	C I			C I			
	-2	C I								C I	C I	C I	C I			C I			
	-1	C								C	C	C	C			C			
XC4013E	-4									C I		C I	C I	C I	C I	C I	C I		
	-3									C I		C I	C I	C I	C I	C I	C I		
	-2									C I		C I	C I	C I	C I	C I	C I		
	-1									C		C	C	C	C	C	C		
XC4020E	-4											C I		C I		C I			
	-3											C I		C I		C I			
	-2											C I		C I		C I			
	-1											C		C		C			
XC4025E	-4													C I		C I		C I	C I
	-3													C I		C I		C I	C I
	-2													C		C		C	C

1/29/99

C = Commercial  $T_J = 0^\circ$  to  $+85^\circ\text{C}$

I = Industrial  $T_J = -40^\circ\text{C}$  to  $+100^\circ\text{C}$

**Table 26: Component Availability Chart for XC4000EX FPGAs**

	PINS	TYPE	CODE	208	240	299	304	352	411	432
				High-Perf. QFP	High-Perf. QFP	Ceram. PGA	High-Perf. QFP	Plast. BGA	Ceram. PGA	Plast. BGA
				HQ208	HQ240	PG299	HQ304	BG352	PG411	BG432
XC4028EX	-4	C I	C I	C I	C I	C I	C I	C I		
	-3	C I	C I	C I	C I	C I	C I	C I		
	-2	C	C	C	C	C	C	C		
XC4036EX	-4			C I	C I		C I	C I	C I	C I
	-3			C I	C I		C I	C I	C I	C I
	-2			C	C		C	C	C	C

1/29/99

C = Commercial  $T_J = 0^\circ$  to  $+85^\circ\text{C}$

I = Industrial  $T_J = -40^\circ\text{C}$  to  $+100^\circ\text{C}$