AMD Xilinx - XC4020XL-3PQ240C Datasheet

Welcome to E-XFL.COM

Understanding Embedded - FPGAs (Field Programmable Gate Array)

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

Applications of Embedded - FPGAs

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

Details	
Product Status	Obsolete
Number of LABs/CLBs	784
Number of Logic Elements/Cells	1862
Total RAM Bits	25088
Number of I/O	192
Number of Gates	20000
Voltage - Supply	3V ~ 3.6V
Mounting Type	Surface Mount
Operating Temperature	0°C ~ 85°C (TJ)
Package / Case	240-BFQFP
Supplier Device Package	240-PQFP (32x32)
Purchase URL	https://www.e-xfl.com/product-detail/xilinx/xc4020xl-3pq240c

Detailed Functional Description

XC4000 Series devices achieve high speed through advanced semiconductor technology and improved architecture. The XC4000E and XC4000X support system clock rates of up to 80 MHz and internal performance in excess of 150 MHz. Compared to older Xilinx FPGA families, XC4000 Series devices are more powerful. They offer on-chip edge-triggered and dual-port RAM, clock enables on I/O flip-flops, and wide-input decoders. They are more versatile in many applications, especially those involving RAM. Design cycles are faster due to a combination of increased routing resources and more sophisticated software.

Basic Building Blocks

Xilinx user-programmable gate arrays include two major configurable elements: configurable logic blocks (CLBs) and input/output blocks (IOBs).

- CLBs provide the functional elements for constructing the user's logic.
- IOBs provide the interface between the package pins and internal signal lines.

Three other types of circuits are also available:

- 3-State buffers (TBUFs) driving horizontal longlines are associated with each CLB.
- Wide edge decoders are available around the periphery of each device.
- An on-chip oscillator is provided.

Programmable interconnect resources provide routing paths to connect the inputs and outputs of these configurable elements to the appropriate networks.

The functionality of each circuit block is customized during configuration by programming internal static memory cells. The values stored in these memory cells determine the logic functions and interconnections implemented in the FPGA. Each of these available circuits is described in this section.

Configurable Logic Blocks (CLBs)

Configurable Logic Blocks implement most of the logic in an FPGA. The principal CLB elements are shown in Figure 1. Two 4-input function generators (F and G) offer unrestricted versatility. Most combinatorial logic functions need four or fewer inputs. However, a third function generator (H) is provided. The H function generator has three inputs. Either zero, one, or two of these inputs can be the outputs of F and G; the other input(s) are from outside the CLB. The CLB can, therefore, implement certain functions of up to nine variables, like parity check or expandable-identity comparison of two sets of four inputs.

Each CLB contains two storage elements that can be used to store the function generator outputs. However, the storage elements and function generators can also be used independently. These storage elements can be configured as flip-flops in both XC4000E and XC4000X devices; in the XC4000X they can optionally be configured as latches. DIN can be used as a direct input to either of the two storage elements. H1 can drive the other through the H function generator. Function generator outputs can also drive two outputs independent of the storage element outputs. This versatility increases logic capacity and simplifies routing.

Thirteen CLB inputs and four CLB outputs provide access to the function generators and storage elements. These inputs and outputs connect to the programmable interconnect resources outside the block.

Function Generators

Four independent inputs are provided to each of two function generators (F1 - F4 and G1 - G4). These function generators, with outputs labeled F' and G', are each capable of implementing any arbitrarily defined Boolean function of four inputs. The function generators are implemented as memory look-up tables. The propagation delay is therefore independent of the function implemented.

A third function generator, labeled H', can implement any Boolean function of its three inputs. Two of these inputs can optionally be the F' and G' functional generator outputs. Alternatively, one or both of these inputs can come from outside the CLB (H2, H0). The third input must come from outside the block (H1).

Signals from the function generators can exit the CLB on two outputs. F' or H' can be connected to the X output. G' or H' can be connected to the Y output.

A CLB can be used to implement any of the following functions:

- any function of up to four variables, plus any second function of up to four unrelated variables, plus any third function of up to three unrelated variables[1]
- any single function of five variables
- any function of four variables together with some functions of six variables
- some functions of up to nine variables.

Implementing wide functions in a single block reduces both the number of blocks required and the delay in the signal path, achieving both increased capacity and speed.

The versatility of the CLB function generators significantly improves system speed. In addition, the design-software tools can deal with each function generator independently. This flexibility improves cell usage.

1. When three separate functions are generated, one of the function outputs must be captured in a flip-flop internal to the CLB. Only two unregistered function generator outputs are available from the CLB.

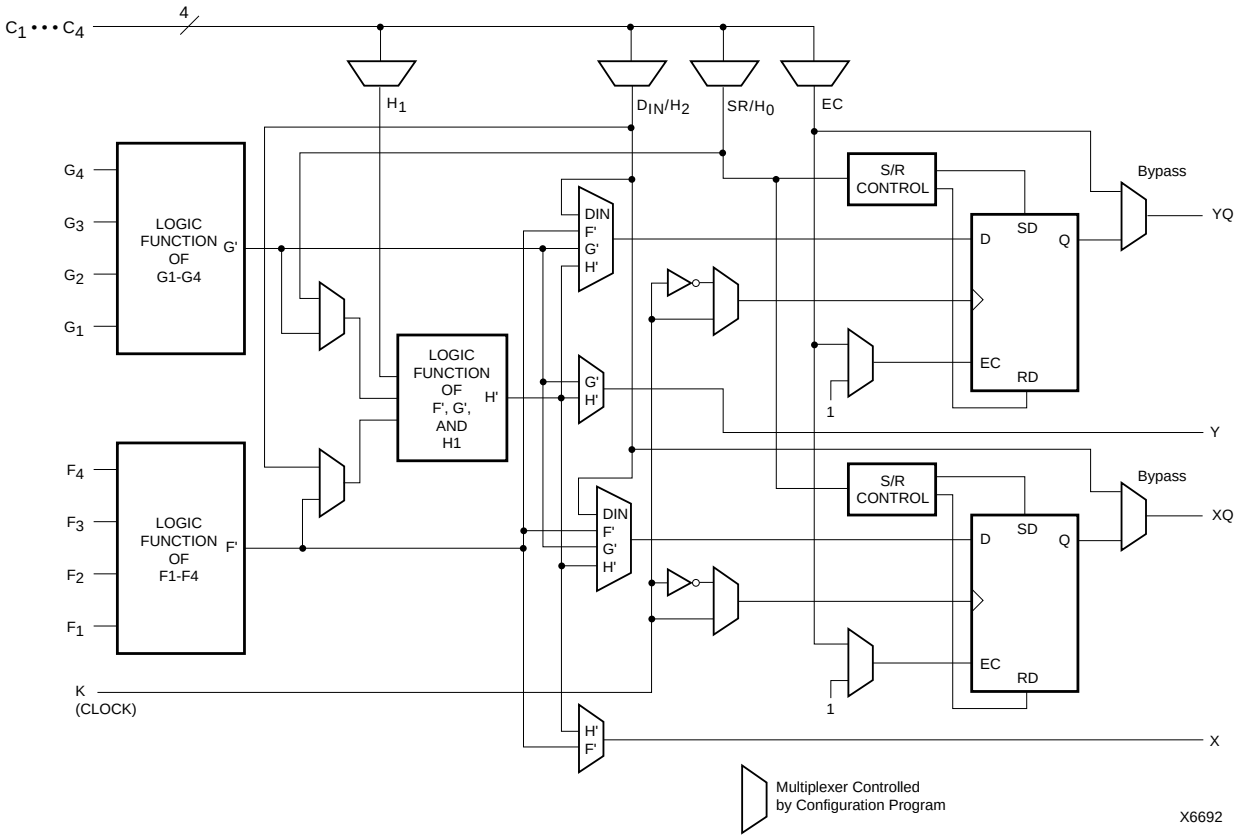


Figure 1: Simplified Block Diagram of XC4000 Series CLB (RAM and Carry Logic functions not shown)

Flip-Flops

The CLB can pass the combinatorial output(s) to the interconnect network, but can also store the combinatorial results or other incoming data in one or two flip-flops, and connect their outputs to the interconnect network as well.

The two edge-triggered D-type flip-flops have common clock (K) and clock enable (EC) inputs. Either or both clock inputs can also be permanently enabled. Storage element functionality is described in Table 2.

Latches (XC4000X only)

The CLB storage elements can also be configured as latches. The two latches have common clock (K) and clock enable (EC) inputs. Storage element functionality is described in Table 2.

Clock Input

Each flip-flop can be triggered on either the rising or falling clock edge. The clock pin is shared by both storage elements. However, the clock is individually invertible for each storage element. Any inverter placed on the clock input is automatically absorbed into the CLB.

Clock Enable

The clock enable signal (EC) is active High. The EC pin is shared by both storage elements. If left unconnected for either, the clock enable for that storage element defaults to the active state. EC is not invertible within the CLB.

Table 2: CLB Storage Element Functionality (active rising edge is shown)

Mode	K	EC	SR	D	Q
Power-Up or GSR	X	X	X	X	SR
Flip-Flop	X	X	1	X	SR
	__/‾	1*	0*	D	D
	0	X	0*	X	Q
Latch	1	1*	0*	X	Q
	0	1*	0*	D	D
Both	X	0	0*	X	Q

Legend:

X	Don't care
__/‾	Rising edge
SR	Set or Reset value. Reset is default.
0*	Input is Low or unconnected (default value)
1*	Input is High or unconnected (default value)

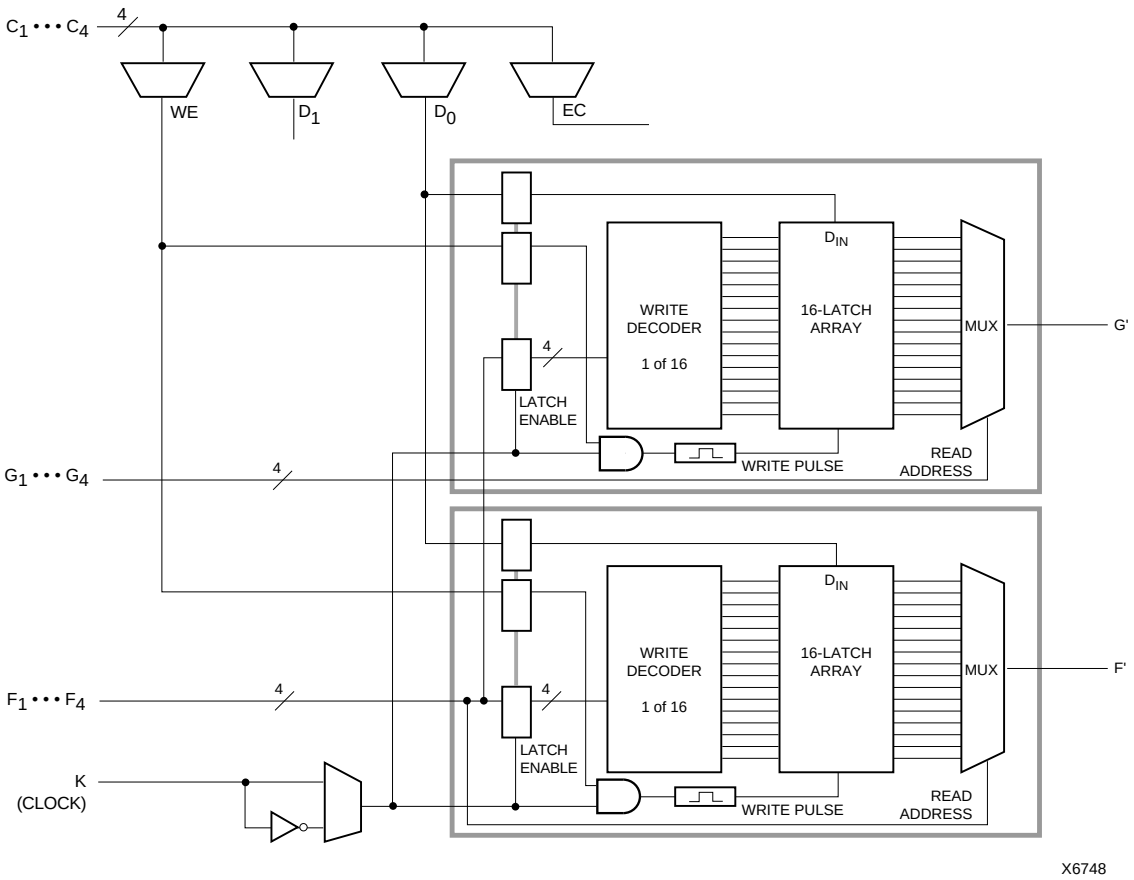


Figure 7: 16x1 Edge-Triggered Dual-Port RAM

Figure 8 shows the write timing for level-sensitive, single-port RAM.

The relationships between CLB pins and RAM inputs and outputs for single-port level-sensitive mode are shown in Table 7.

Figure 9 and Figure 10 show block diagrams of a CLB configured as 16x2 and 32x1 level-sensitive, single-port RAM.

Initializing RAM at Configuration

Both RAM and ROM implementations of the XC4000 Series devices are initialized during configuration. The initial contents are defined via an INIT attribute or property attached to the RAM or ROM symbol, as described in the schematic library guide. If not defined, all RAM contents are initialized to all zeros, by default.

RAM initialization occurs only during configuration. The RAM content is not affected by Global Set/Reset.

Table 7: Single-Port Level-Sensitive RAM Signals

RAM Signal	CLB Pin	Function
D	D0 or D1	Data In
A[3:0]	F1-F4 or G1-G4	Address
WE	WE	Write Enable
O	F' or G'	Data Out

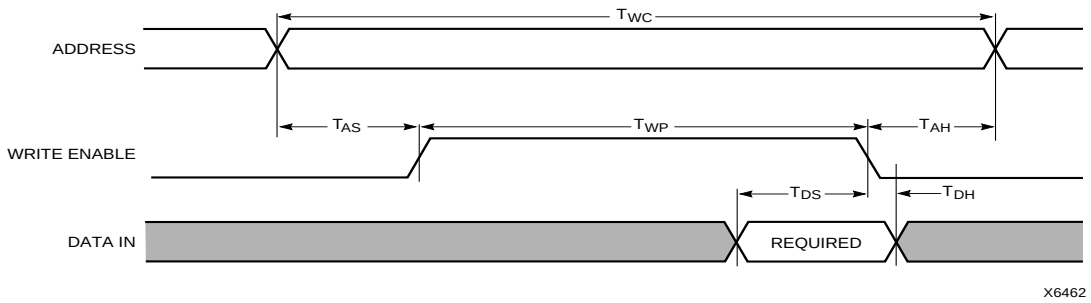


Figure 8: Level-Sensitive RAM Write Timing

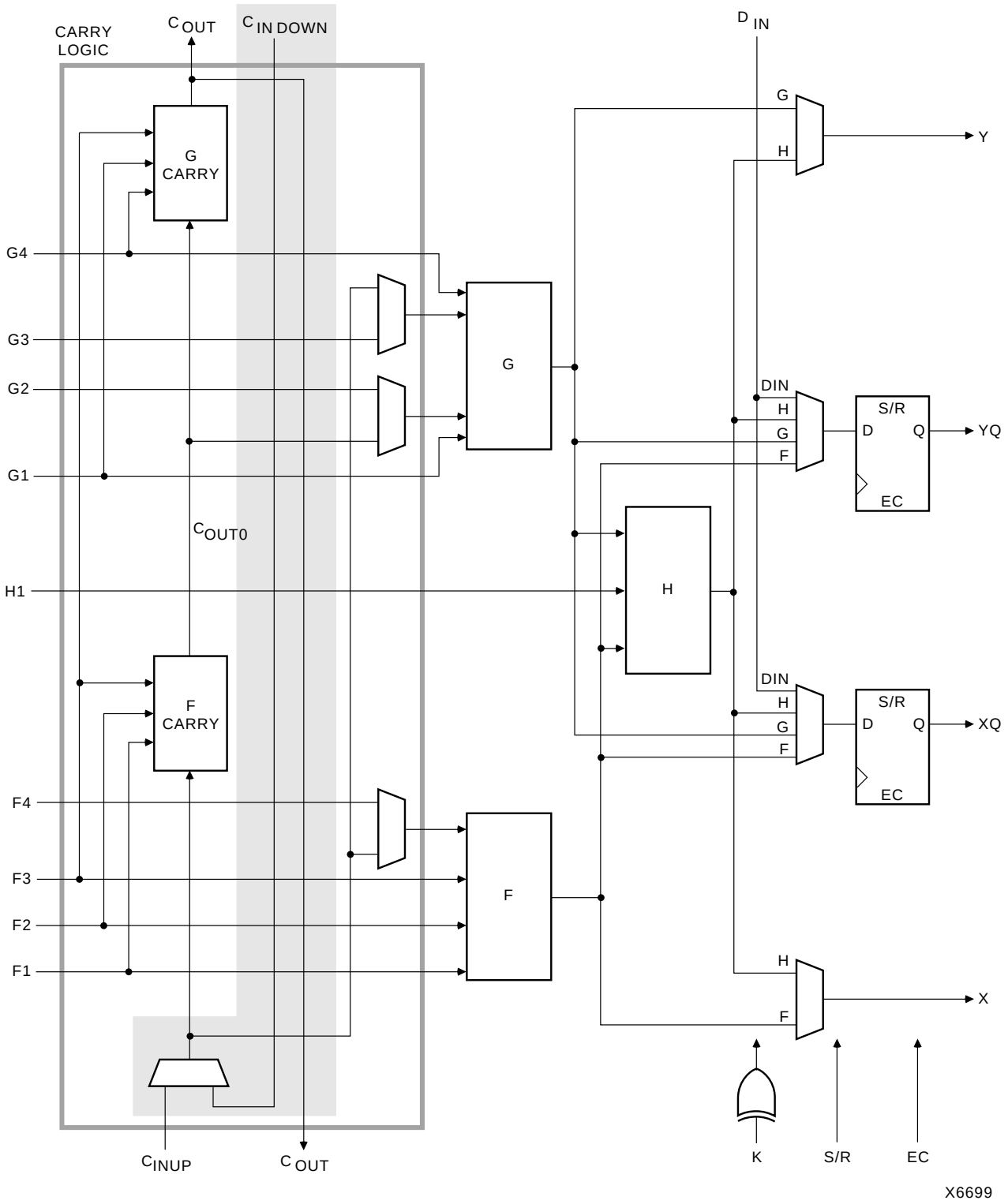


Figure 13: Fast Carry Logic in XC4000E CLB (shaded area not present in XC4000X)

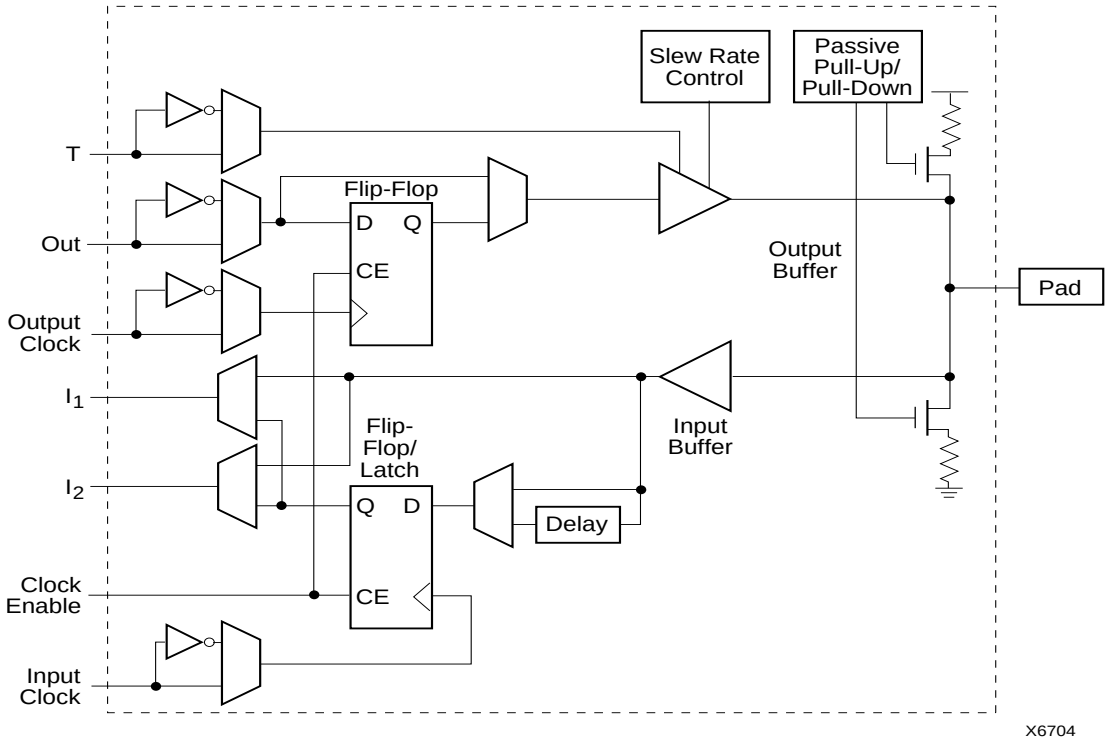


Figure 15: Simplified Block Diagram of XC4000E IOB

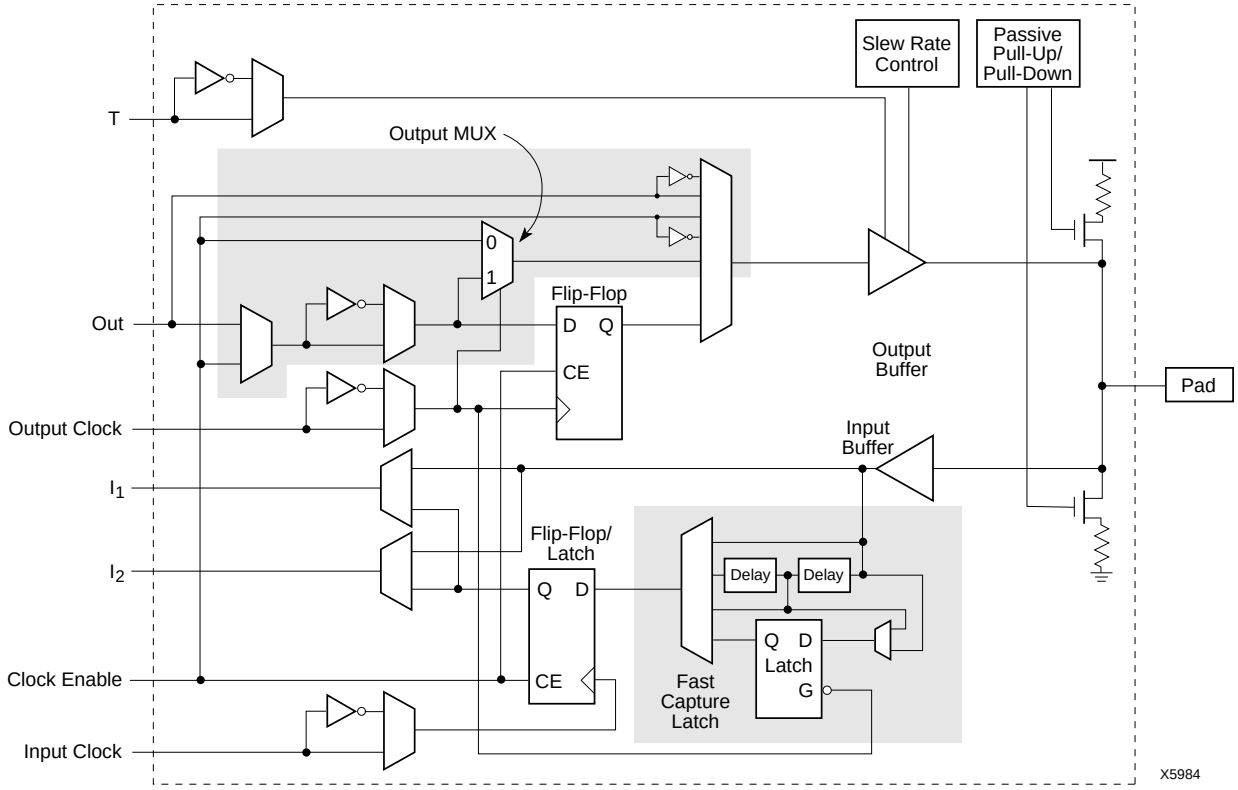


Figure 16: Simplified Block Diagram of XC4000X IOB (shaded areas indicate differences from XC4000E)

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.

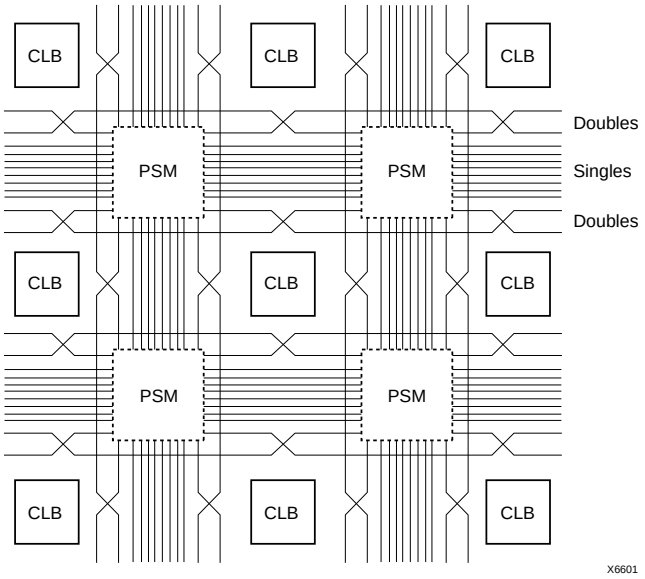


Figure 28: Single- and Double-Length Lines, with Programmable Switch Matrices (PSMs)

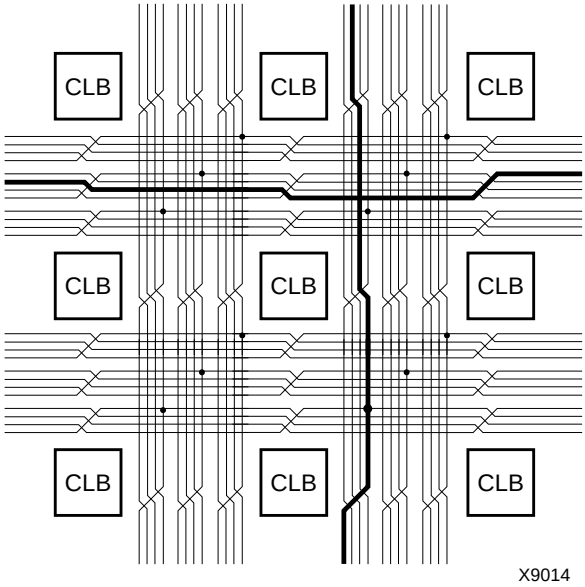


Figure 29: Quad Lines (XC4000X only)

Double-Length Lines

The double-length lines consist of a grid of metal segments, each twice as long as the single-length lines: they run past two CLBs before entering a switch matrix. Double-length lines are grouped in pairs with the switch matrices staggered, so that each line goes through a switch matrix at every other row or column of CLBs (see Figure 28).

There are four vertical and four horizontal double-length lines associated with each CLB. These lines provide faster signal routing over intermediate distances, while retaining routing flexibility. Double-length lines are connected by way of the programmable switch matrices. Routing connectivity is shown in Figure 27.

Quad Lines (XC4000X only)

XC4000X devices also include twelve vertical and twelve horizontal quad lines per CLB row and column. Quad lines are four times as long as the single-length lines. They are interconnected via buffered switch matrices (shown as diamonds in Figure 27 on page 30). Quad lines run past four CLBs before entering a buffered switch matrix. They are grouped in fours, with the buffered switch matrices staggered, so that each line goes through a buffered switch matrix at every fourth CLB location in that row or column. (See Figure 29.)

The buffered switch matrixes have four pins, one on each edge. All of the pins are bidirectional. Any pin can drive any or all of the other pins.

Each buffered switch matrix contains one buffer and six pass transistors. It resembles the programmable switch matrix shown in Figure 26, with the addition of a programmable buffer. There can be up to two independent inputs and up to two independent outputs. Only one of the independent inputs can be buffered.

The place and route software automatically uses the timing requirements of the design to determine whether or not a quad line signal should be buffered. A heavily loaded signal is typically buffered, while a lightly loaded one is not. One scenario is to alternate buffers and pass transistors. This allows both vertical and horizontal quad lines to be buffered at alternating buffered switch matrices.

Due to the buffered switch matrices, quad lines are very fast. They provide the fastest available method of routing heavily loaded signals for long distances across the device.

Longlines

Longlines form a grid of metal interconnect segments that run the entire length or width of the array. Longlines are intended for high fan-out, time-critical signal nets, or nets that are distributed over long distances. In XC4000X devices, quad lines are preferred for critical nets, because the buffered switch matrices make them faster for high fan-out nets.

Two horizontal longlines per CLB can be driven by 3-state or open-drain drivers (TBUFs). They can therefore implement unidirectional or bidirectional buses, wide multiplexers, or wired-AND functions. (See "Three-State Buffers" on page 26 for more details.)

Each horizontal longline driven by TBUFs has either two (XC4000E) or eight (XC4000X) pull-up resistors. To activate these resistors, attach a PULLUP symbol to the long-line net. The software automatically activates the appropriate number of pull-ups. There is also a weak keeper at each end of these two horizontal longlines. This

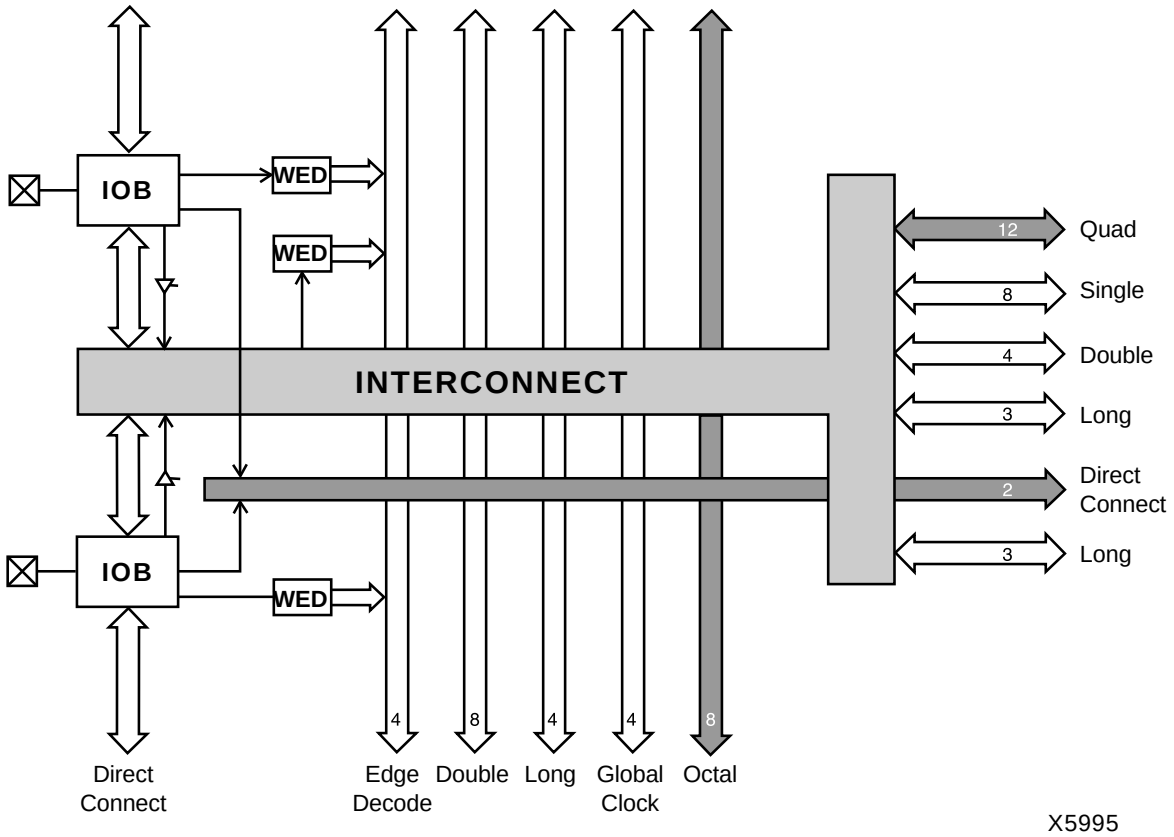


Figure 31: High-Level Routing Diagram of XC4000 Series VersaRing (Left Edge)
WED = Wide Edge Decoder, IOB = I/O Block (shaded arrows indicate XC4000X only)

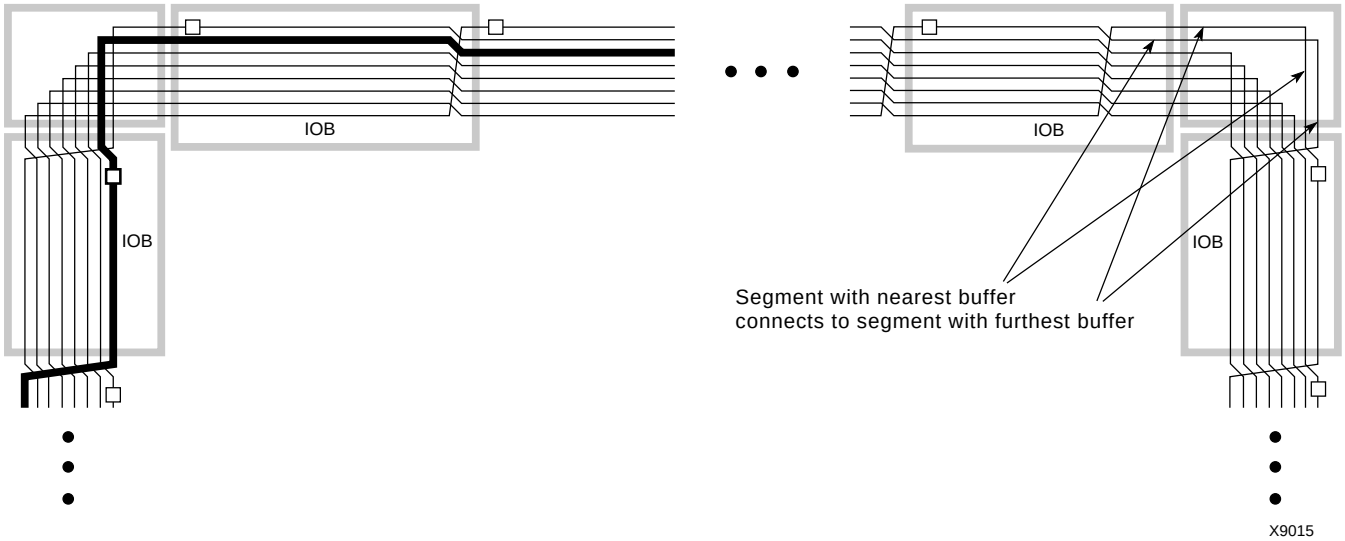
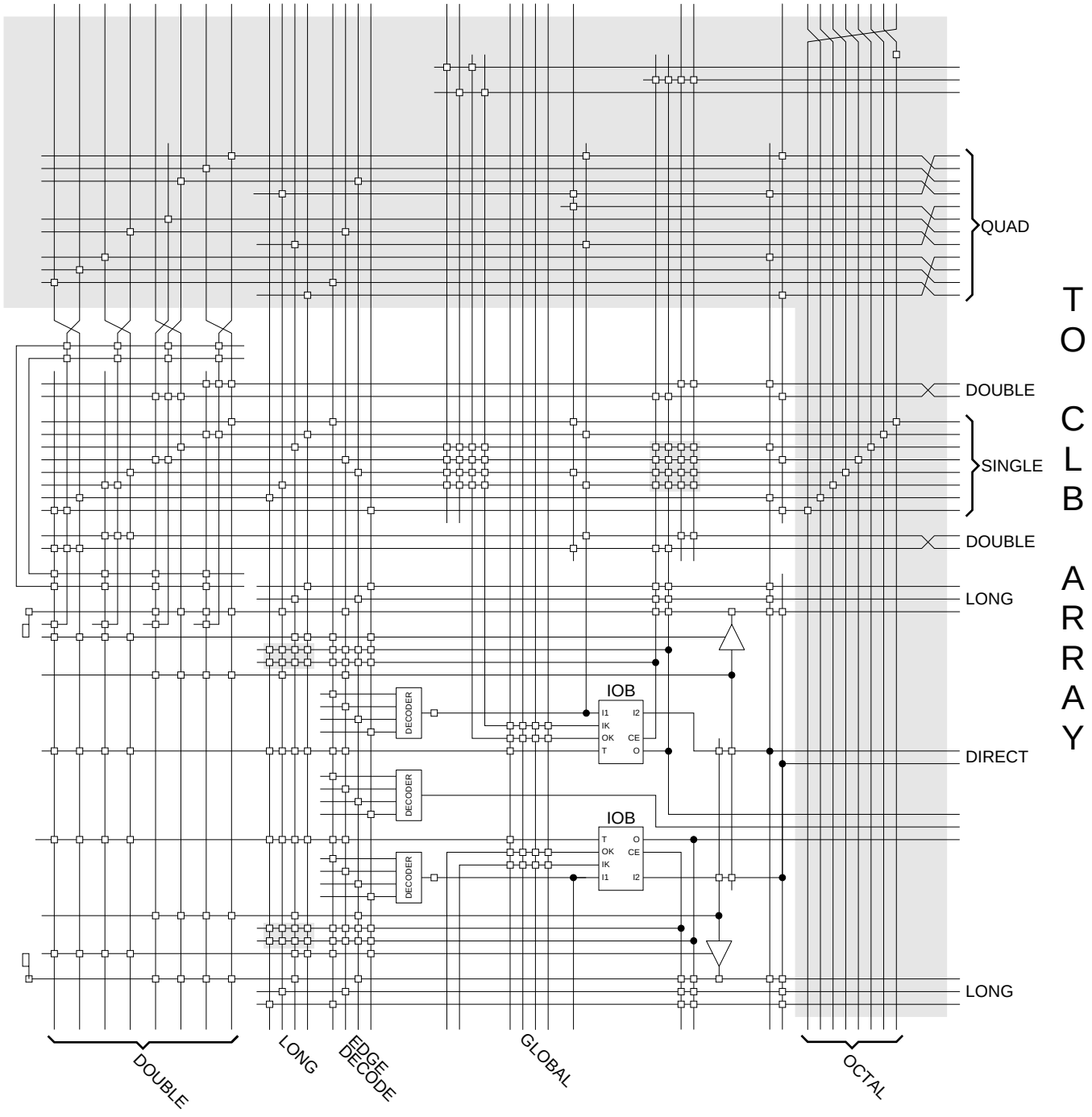


Figure 32: XC4000X Octal I/O Routing



Common to XC4000E and XC4000X

XC4000X only

Figure 33: Detail of Programmable Interconnect Associated with XC4000 Series IOB (Left Edge)

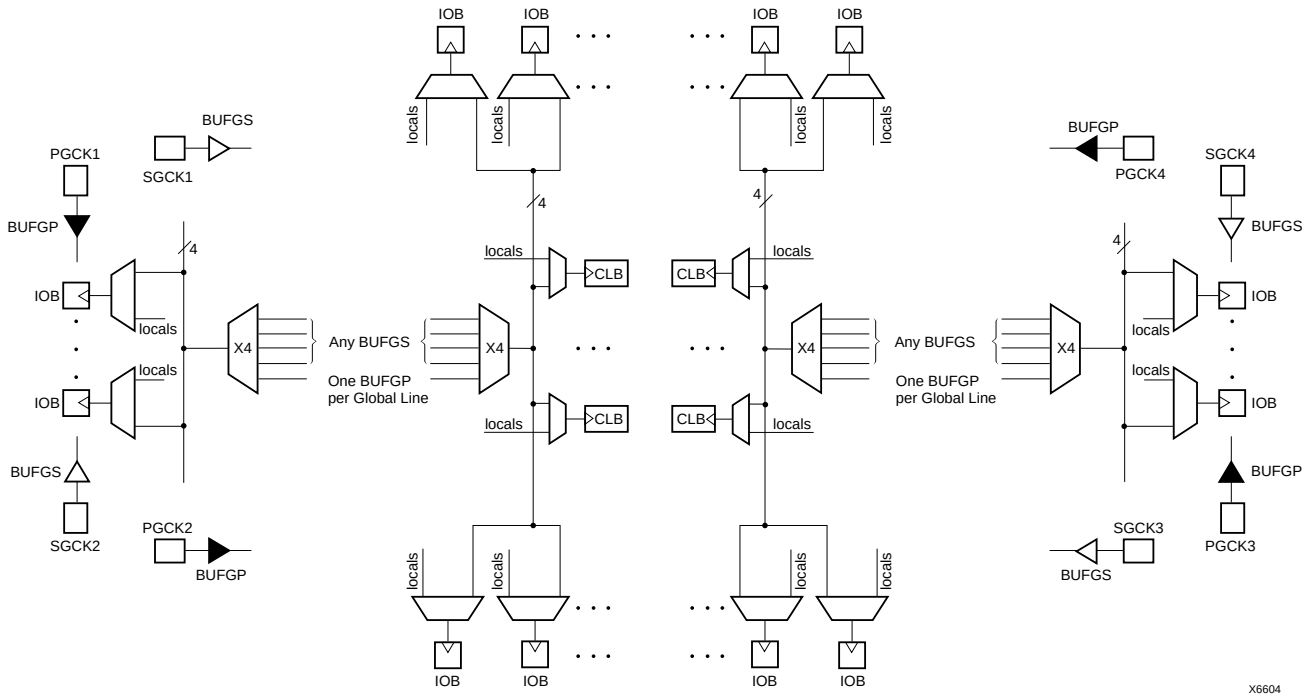


Figure 34: XC4000E Global Net Distribution

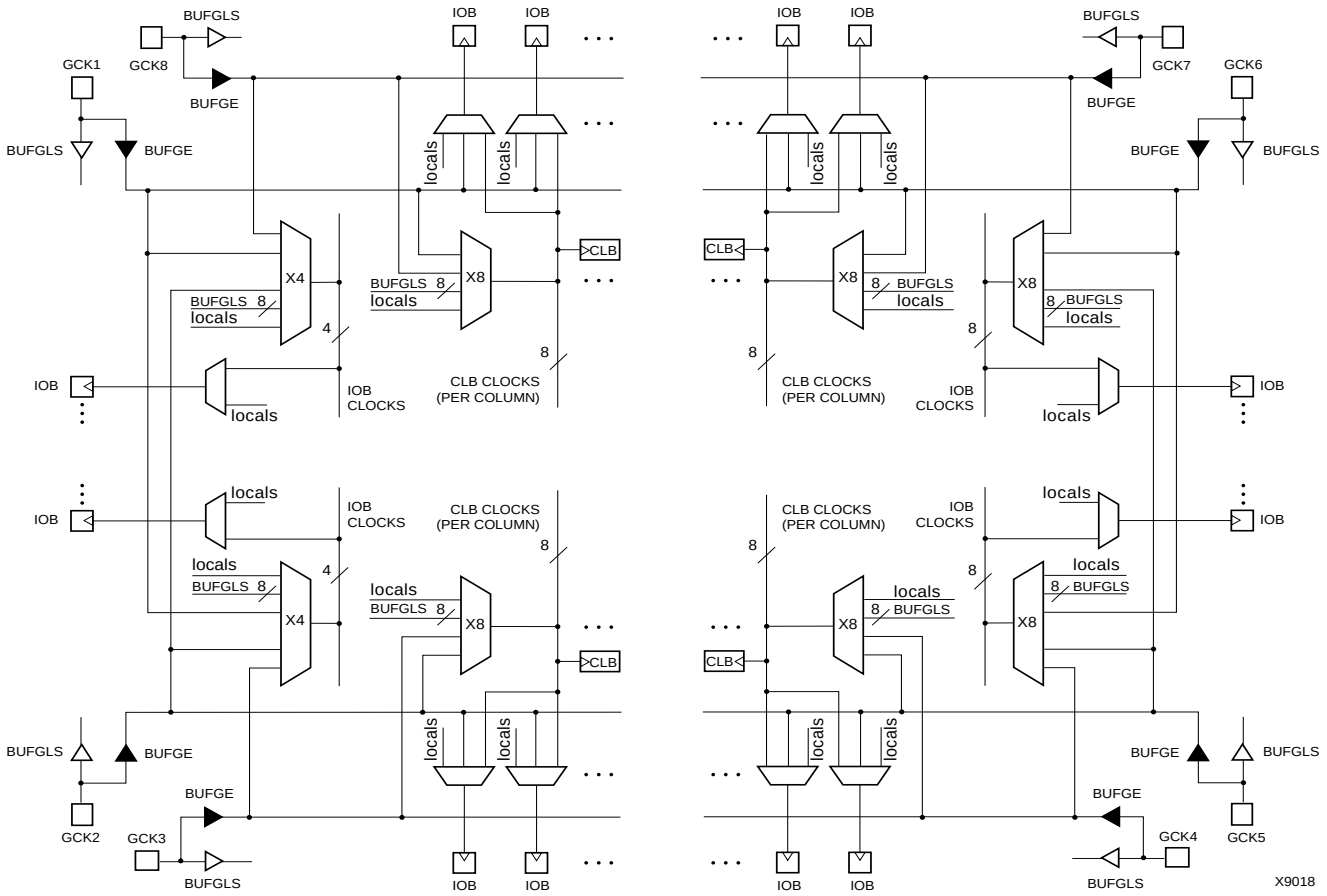


Figure 35: XC4000X Global Net Distribution

Global Nets and Buffers (XC4000X only)

Eight vertical longlines in each CLB column are driven by special global buffers. These longlines are in addition to the vertical longlines used for standard interconnect. The global lines are broken in the center of the array, to allow faster distribution and to minimize skew across the whole array. Each half-column global line has its own buffered multiplexer, as shown in Figure 35. The top and bottom global lines cannot be connected across the center of the device, as this connection might introduce unacceptable skew. The top and bottom halves of the global lines must be separately driven — although they can be driven by the same global buffer.

The eight global lines in each CLB column can be driven by either of two types of global buffers. They can also be driven by internal logic, because they can be accessed by single, double, and quad lines at the top, bottom, half, and quarter points. Consequently, the number of different clocks that can be used simultaneously in an XC4000X device is very large.

There are four global lines feeding the IOBs at the left edge of the device. IOBs along the right edge have eight global lines. There is a single global line along the top and bottom edges with access to the IOBs. All IOB global lines are broken at the center. They cannot be connected across the center of the device, as this connection might introduce unacceptable skew.

IOB global lines can be driven from two types of global buffers, or from local interconnect. Alternatively, top and bottom IOBs can be clocked from the global lines in the adjacent CLB column.

Two different types of clock buffers are available in the XC4000X:

- Global Low-Skew Buffers (BUFGLS)
- Global Early Buffers (BUFGE)

Global Low-Skew Buffers are the standard clock buffers. They should be used for most internal clocking, whenever a large portion of the device must be driven.

Global Early Buffers are designed to provide a faster clock access, but CLB access is limited to one-fourth of the device. They also facilitate a faster I/O interface.

Figure 35 is a conceptual diagram of the global net structure in the XC4000X.

Global Early buffers and Global Low-Skew buffers share a single pad. Therefore, the same IPAD symbol can drive one buffer of each type, in parallel. This configuration is particularly useful when using the Fast Capture latches, as described in "IOB Input Signals" on page 20. Paired Global Early and Global Low-Skew buffers share a common input; they cannot be driven by two different signals.

Choosing an XC4000X Clock Buffer

The clocking structure of the XC4000X provides a large variety of features. However, it can be simple to use, without understanding all the details. The software automatically handles clocks, along with all other routing, when the appropriate clock buffer is placed in the design. In fact, if a buffer symbol called BUFG is placed, rather than a specific type of buffer, the software even chooses the buffer most appropriate for the design. The detailed information in this section is provided for those users who want a finer level of control over their designs.

If fine control is desired, use the following summary and Table 15 on page 35 to choose an appropriate clock buffer.

- The simplest thing to do is to use a Global Low-Skew buffer.
- If a faster clock path is needed, try a BUFG. The software will first try to use a Global Low-Skew Buffer. If timing requirements are not met, a faster buffer will automatically be used.
- If a single quadrant of the chip is sufficient for the clocked logic, and the timing requires a faster clock than the Global Low-Skew buffer, use a Global Early buffer.

Global Low-Skew Buffers

Each corner of the XC4000X device has two Global Low-Skew buffers. Any of the eight Global Low-Skew buffers can drive any of the eight vertical Global lines in a column of CLBs. In addition, any of the buffers can drive any of the four vertical lines accessing the IOBs on the left edge of the device, and any of the eight vertical lines accessing the IOBs on the right edge of the device. (See Figure 36 on page 38.)

IOBs at the top and bottom edges of the device are accessed through the vertical Global lines in the CLB array, as in the XC4000E. Any Global Low-Skew buffer can, therefore, access every IOB and CLB in the device.

The Global Low-Skew buffers can be driven by either semi-dedicated pads or internal logic.

To use a Global Low-Skew buffer, instantiate a BUFGLS 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 BUFGLS be placed in one of the two Global Low-Skew buffers on the top edge of the device, or a LOC=TR to indicate the Global Low-Skew buffer on the top edge of the device, on the right.

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 initiated 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.

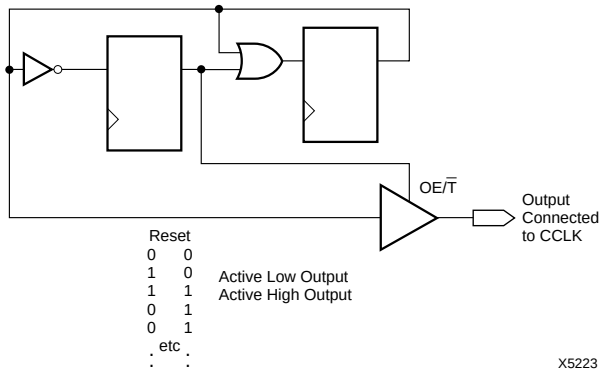


Figure 44: CCLK Generation for XC3000 Master Driving an XC4000 Series Slave

Table 20: XC4000E Program Data

Device	XC4003E	XC4005E	XC4006E	XC4008E	XC4010E	XC4013E	XC4020E	XC4025E
Max Logic Gates	3,000	5,000	6,000	8,000	10,000	13,000	20,000	25,000
CLBs (Row x Col.)	100 (10 x 10)	196 (14 x 14)	256 (16 x 16)	324 (18 x 18)	400 (20 x 20)	576 (24 x 24)	784 (28 x 28)	1,024 (32 x 32)
IOBs	80	112	128	144	160	192	224	256
Flip-Flops	360	616	768	936	1,120	1,536	2,016	2,560
Bits per Frame	126	166	186	206	226	266	306	346
Frames	428	572	644	716	788	932	1,076	1,220
Program Data	53,936	94,960	119,792	147,504	178,096	247,920	329,264	422,128
PROM Size (bits)	53,984	95,008	119,840	147,552	178,144	247,968	329,312	422,176

Notes: 1. Bits per Frame = (10 x number of rows) + 7 for the top + 13 for the bottom + 1 + 1 start bit + 4 error check bits
Number of Frames = (36 x number of columns) + 26 for the left edge + 41 for the right edge + 1
Program Data = (Bits per Frame x Number of Frames) + 8 postamble bits
PROM Size = Program Data + 40 (header) + 8
2. The user can add more "one" bits as leading dummy bits in the header, or, if CRC = off, as trailing dummy bits at the end of any frame, following the four error check bits. However, the Length Count value **must** be adjusted for all such extra "one" bits, even for extra leading ones at the beginning of the header.

Table 21: XC4000EX/XL Program Data

Device	XC4002XL	XC4005	XC4010	XC4013	XC4020	XC4028	XC4036	XC4044	XC4052	XC4062	XC4085
Max Logic Gates	2,000	5,000	10,000	13,000	20,000	28,000	36,000	44,000	52,000	62,000	85,000
CLBs (Row x Column)	64 (8 x 8)	196 (14 x 14)	400 (20 x 20)	576 (24 x 24)	784 (28 x 28)	1,024 (32 x 32)	1,296 (36 x 36)	1,600 (40 x 40)	1,936 (44 x 44)	2,304 (48 x 48)	3,136 (56 x 56)
IOBs	64	112	160	192	224	256	288	320	352	384	448
Flip-Flops	256	616	1,120	1,536	2,016	2,560	3,168	3,840	4,576	5,376	7,168
Bits per Frame	133	205	277	325	373	421	469	517	565	613	709
Frames	459	741	1,023	1,211	1,399	1,587	1,775	1,963	2,151	2,339	2,715
Program Data	61,052	151,910	283,376	393,580	521,832	668,124	832,480	1,014,876	1,215,320	1,433,804	1,924,940
PROM Size (bits)	61,104	151,960	283,424	393,632	521,880	668,172	832,528	1,014,924	1,215,368	1,433,852	1,924,992

Notes: 1. Bits per frame = (13 x number of rows) + 9 for the top + 17 for the bottom + 8 + 1 start bit + 4 error check bits.
Frames = (47 x number of columns) + 27 for the left edge + 52 for the right edge + 4.
Program data = (bits per frame x number of frames) + 5 postamble bits.
PROM size = (program data + 40 header bits + 8 start bits) rounded up to the nearest byte.
2. The user can add more "one" bits as leading dummy bits in the header, or, if CRC = off, as trailing dummy bits at the end of any frame, following the four error check bits. However, the Length Count value must be adjusted for all such extra "one" bits, even for extra leading "ones" at the beginning of the header.

Cyclic Redundancy Check (CRC) for Configuration and Readback

The Cyclic Redundancy Check is a method of error detection in data transmission applications. Generally, the transmitting system performs a calculation on the serial bitstream. The result of this calculation is tagged onto the data stream as additional check bits. The receiving system performs an identical calculation on the bitstream and compares the result with the received checksum.

Each data frame of the configuration bitstream has four error bits at the end, as shown in Table 19. If a frame data error is detected during the loading of the FPGA, the configuration process with a potentially corrupted bitstream is terminated. The FPGA pulls the $\overline{\text{INIT}}$ pin Low and goes into a Wait state.

During Readback, 11 bits of the 16-bit checksum are added to the end of the Readback data stream. The checksum is computed using the CRC-16 CCITT polynomial, as shown in Figure 45. The checksum consists of the 11 most significant bits of the 16-bit code. A change in the checksum indicates a change in the Readback bitstream. A comparison to a previous checksum is meaningful only if the readback data is independent of the current device state. CLB outputs should not be included (Read Capture option not

used), and if RAM is present, the RAM content must be unchanged.

Statistically, one error out of 2048 might go undetected.

Configuration Sequence

There are four major steps in the XC4000 Series power-up configuration sequence.

- Configuration Memory Clear
- Initialization
- Configuration
- Start-Up

The full process is illustrated in Figure 46.

Configuration Memory Clear

When power is first applied or is reapplied to an FPGA, an internal circuit forces initialization of the configuration logic. When Vcc reaches an operational level, and the circuit passes the write and read test of a sample pair of configuration bits, a time delay is started. This time delay is nominally 16 ms, and up to 10% longer in the low-voltage devices. The delay is four times as long when in Master Modes (M0 Low), to allow ample time for all slaves to reach a stable Vcc. When all $\overline{\text{INIT}}$ pins are tied together, as recommended, the longest delay takes precedence. Therefore, devices with different time delays can easily be mixed and matched in a daisy chain.

This delay is applied only on power-up. It is not applied when re-configuring an FPGA by pulsing the $\overline{\text{PROGRAM}}$ pin

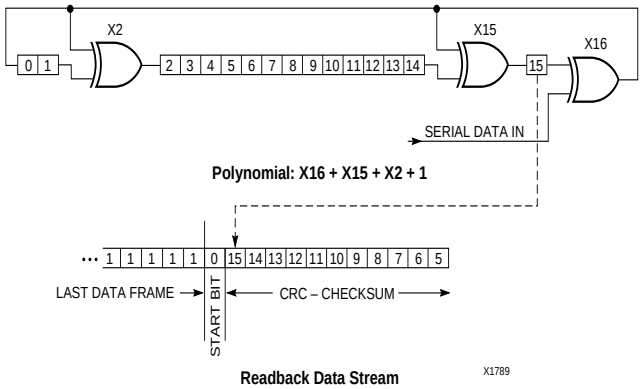


Figure 45: Circuit for Generating CRC-16

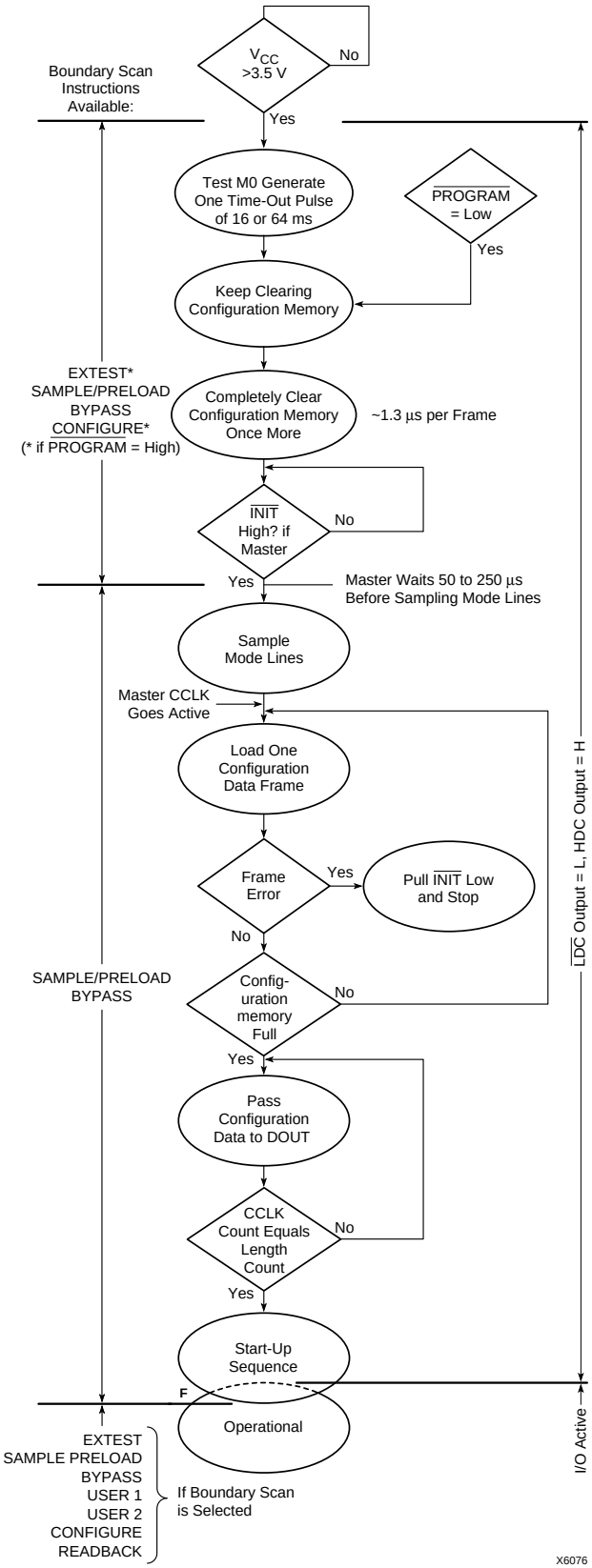


Figure 46: Power-up Configuration Sequence

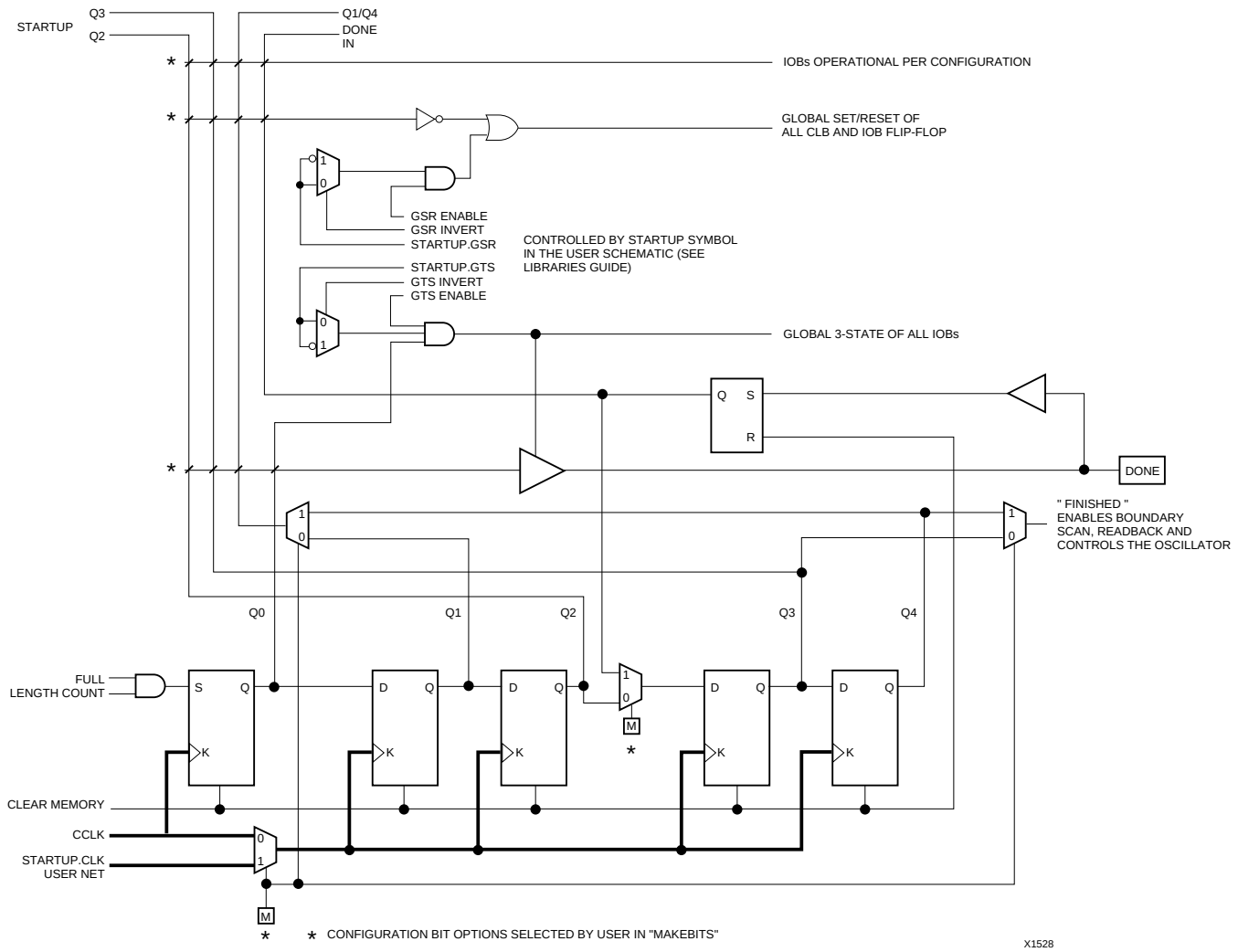


Figure 48: Start-up Logic

Readback

The user can read back the content of configuration memory and the level of certain internal nodes without interfering with the normal operation of the device.

Readback not only reports the downloaded configuration bits, but can also include the present state of the device, represented by the content of all flip-flops and latches in CLBs and IOBs, as well as the content of function generators used as RAMs.

Note that in XC4000 Series devices, configuration data is *not* inverted with respect to configuration as it is in XC2000 and XC3000 families.

XC4000 Series Readback does not use any dedicated pins, but uses four internal nets (RDBK.TRIG, RDBK.DATA, RDBK.RIP and RDBK.CLK) that can be routed to any IOB. To access the internal Readback signals, place the READBACK library symbol and attach the appropriate pad symbols, as shown in Figure 49.

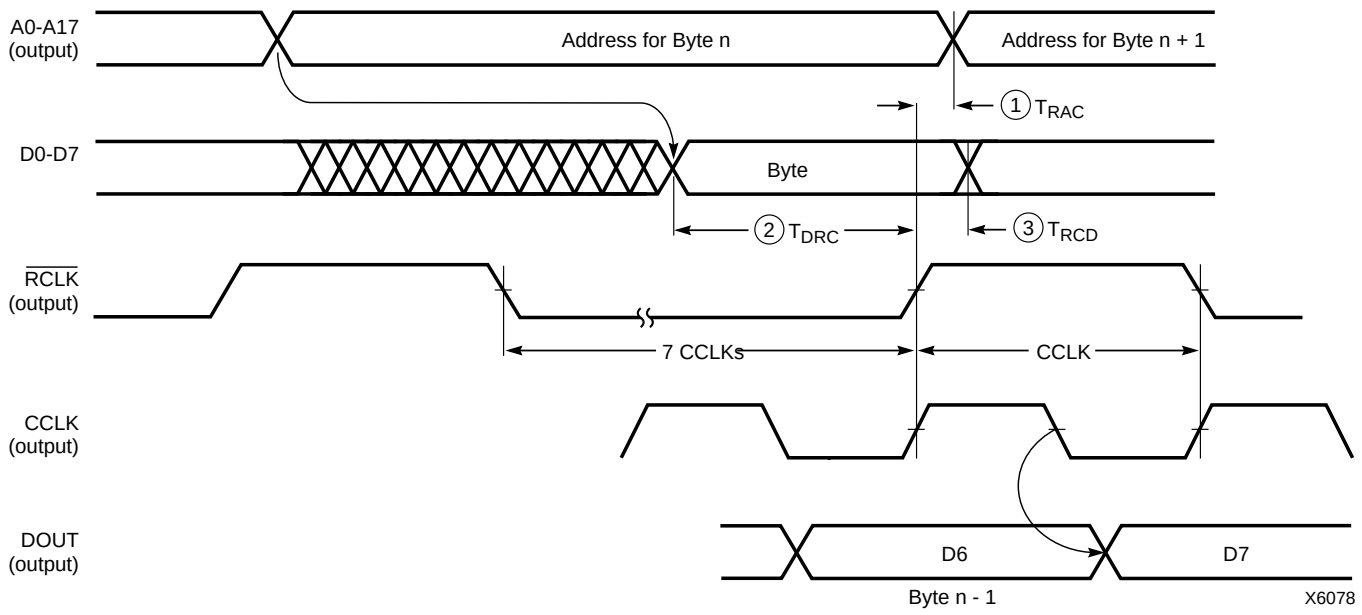
After Readback has been initiated by a High level on RDBK.TRIG after configuration, the RDBK.RIP (Read In Progress) output goes High on the next rising edge of RDBK.CLK. Subsequent rising edges of this clock shift out Readback data on the RDBK.DATA net.

Readback data does not include the preamble, but starts with five dummy bits (all High) followed by the Start bit (Low) of the first frame. The first two data bits of the first frame are always High.

Each frame ends with four error check bits. They are read back as High. The last seven bits of the last frame are also read back as High. An additional Start bit (Low) and an 11-bit Cyclic Redundancy Check (CRC) signature follow, before RDBK.RIP returns Low.

Table 22: Pin Functions During Configuration

CONFIGURATION MODE <M2:M1:M0>						
SLAVE SERIAL <1:1:1>	**MASTER SERIAL <0:0:0>**	**SYNCH. PERIPHERAL <0:1:1>**	**ASYNCH. PERIPHERAL <1:0:1>**	**MASTER PARALLEL DOWN <1:1:0>**	**MASTER PARALLEL UP <1:0:0>**	**USER OPERATION**
M2(HIGH) (I)	M2(LOW) (I)	M2(LOW) (I)	M2(HIGH) (I)	M2(HIGH) (I)	M2(HIGH) (I)	(I)
M1(HIGH) (I)	M1(LOW) (I)	M1(HIGH) (I)	M1(LOW) (I)	M1(HIGH) (I)	M1(LOW) (I)	(O)
M0(HIGH) (I)	M0(LOW) (I)	M0(HIGH) (I)	M0(HIGH) (I)	M0(LOW) (I)	M0(LOW) (I)	(I)
HDC (HIGH)	HDC (HIGH)	HDC (HIGH)	HDC (HIGH)	HDC (HIGH)	HDC (HIGH)	I/O
$\overline{\text{LDC}}$ (LOW)	$\overline{\text{LDC}}$ (LOW)	$\overline{\text{LDC}}$ (LOW)	$\overline{\text{LDC}}$ (LOW)	$\overline{\text{LDC}}$ (LOW)	$\overline{\text{LDC}}$ (LOW)	I/O
$\overline{\text{INIT}}$	$\overline{\text{INIT}}$	$\overline{\text{INIT}}$	$\overline{\text{INIT}}$	$\overline{\text{INIT}}$	$\overline{\text{INIT}}$	I/O
DONE	DONE	DONE	DONE	DONE	DONE	DONE
$\overline{\text{PROGRAM}}$ (I)	$\overline{\text{PROGRAM}}$ (I)	$\overline{\text{PROGRAM}}$ (I)	$\overline{\text{PROGRAM}}$ (I)	$\overline{\text{PROGRAM}}$ (I)	$\overline{\text{PROGRAM}}$ (I)	$\overline{\text{PROGRAM}}$
CCLK (I)	CCLK (O)	CCLK (I)	CCLK (O)	CCLK (O)	CCLK (O)	CCLK (I)
		RDY/$\overline{\text{BUSY}}$ (O)	RDY/$\overline{\text{BUSY}}$ (O)	$\overline{\text{RCLK}}$ (O)	$\overline{\text{RCLK}}$ (O)	I/O
			$\overline{\text{RS}}$ (I)			I/O
			$\overline{\text{CS0}}$ (I)			I/O
		DATA 7 (I)	DATA 7 (I)	DATA 7 (I)	DATA 7 (I)	I/O
		DATA 6 (I)	DATA 6 (I)	DATA 6 (I)	DATA 6 (I)	I/O
		DATA 5 (I)	DATA 5 (I)	DATA 5 (I)	DATA 5 (I)	I/O
		DATA 4 (I)	DATA 4 (I)	DATA 4 (I)	DATA 4 (I)	I/O
		DATA 3 (I)	DATA 3 (I)	DATA 3 (I)	DATA 3 (I)	I/O
		DATA 2 (I)	DATA 2 (I)	DATA 2 (I)	DATA 2 (I)	I/O
		DATA 1 (I)	DATA 1 (I)	DATA 1 (I)	DATA 1 (I)	I/O
DIN (I)	DIN (I)	DATA 0 (I)	DATA 0 (I)	DATA 0 (I)	DATA 0 (I)	I/O
DOUT	DOUT	DOUT	DOUT	DOUT	DOUT	SGCK4-GCK6-I/O
TDI	TDI	TDI	TDI	TDI	TDI	TDI-I/O
TCK	TCK	TCK	TCK	TCK	TCK	TCK-I/O
TMS	TMS	TMS	TMS	TMS	TMS	TMS-I/O
TDO	TDO	TDO	TDO	TDO	TDO	TDO-(O)
			$\overline{\text{WS}}$ (I)	A0	A0	I/O
				A1	A1	PGCK4-GCK7-I/O
			CS1	A2	A2	I/O
				A3	A3	I/O
				A4	A4	I/O
				A5	A5	I/O
				A6	A6	I/O
				A7	A7	I/O
				A8	A8	I/O
				A9	A9	I/O
				A10	A10	I/O
				A11	A11	I/O
				A12	A12	I/O
				A13	A13	I/O
				A14	A14	I/O
				A15	A15	SGCK1-GCK8-I/O
				A16	A16	PGCK1-GCK1-I/O
				A17	A17	I/O
				A18*	A18*	I/O
				A19*	A19*	I/O
				A20*	A20*	I/O
				A21*	A21*	I/O
						ALL OTHERS

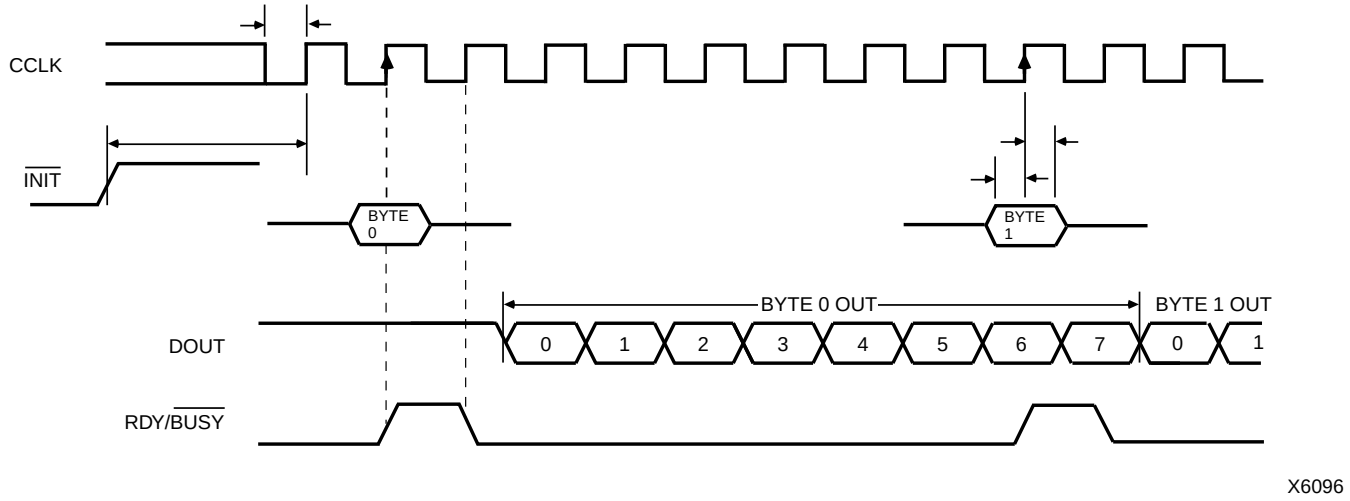


	Description	Symbol		Min	Max	Units
RCLK	Delay to Address valid	1	T_{RAC}	0	200	ns
	Data setup time	2	T_{DRC}	60		ns
	Data hold time	3	T_{RCD}	0		ns

Notes: 1. At power-up, Vcc must rise from 2.0 V to Vcc min in less than 25 ms, otherwise delay configuration by pulling $\overline{\text{PROGRAM}}$ Low until Vcc is valid.
2. The first Data byte is loaded and CCLK starts at the end of the first $\overline{\text{RCLK}}$ active cycle (rising edge).

This timing diagram shows that the EPROM requirements are extremely relaxed. EPROM access time can be longer than 500 ns. EPROM data output has no hold-time requirements.

Figure 55: Master Parallel Mode Programming Switching Characteristics



	Description	Symbol	Min	Max	Units
CCLK	$\overline{INIT}$ (High) setup time	T_{IC}	5		µs
	D0 - D7 setup time	T_{DC}	60		ns
	D0 - D7 hold time	T_{CD}	0		ns
	CCLK High time	T_{CCH}	50		ns
	CCLK Low time	T_{CCL}	60		ns
	CCLK Frequency	F_{CC}		8	MHz

Notes:
1. Peripheral Synchronous mode can be considered Slave Parallel mode. An external CCLK provides timing, clocking in the **first** data byte on the **second** rising edge of CCLK after $\overline{INIT}$ goes High. Subsequent data bytes are clocked in on every eighth consecutive rising edge of CCLK.
2. The RDY/$\overline{BUSY}$ line goes High for one CCLK period after data has been clocked in, although synchronous operation does not require such a response.
3. The pin name RDY/$\overline{BUSY}$ is a misnomer. In Synchronous Peripheral mode this is really an ACKNOWLEDGE signal.
4. Note that data starts to shift out serially on the DOUT pin 0.5 CCLK periods after it was loaded in parallel. Therefore, additional CCLK pulses are clearly required after the last byte has been loaded.

Figure 57: Synchronous Peripheral Mode Programming Switching Characteristics

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 $\overline{RS}$ and CS1 being High to accept byte-wide data from a microprocessor 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/$\overline{BUSY}$ output from the lead FPGA acts as a handshake signal to the microprocessor. RDY/$\overline{BUSY}$ 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/$\overline{BUSY}$ output has gone Low, acknowledging receipt of the previous data. Write may not be terminated until RDY/$\overline{BUSY}$ is High again for one CCLK period. Note that RDY/$\overline{BUSY}$ is pulled High with a high-impedance pull-up prior to $\overline{INIT}$ going High.

The length of the $\overline{BUSY}$ signal depends on the activity in the UART. If the shift register was empty when the new byte was received, the $\overline{BUSY}$ signal lasts for only two CCLK periods. If the shift register was still full when the new byte was received, the $\overline{BUSY}$ 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/$\overline{BUSY}$ 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 $\overline{CS0}$, CS1and $\overline{RS}$ 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*step* software, ensures that these problems never occur.

Although RDY/$\overline{BUSY}$ 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/$\overline{BUSY}$ status when $\overline{RS}$ is Low, $\overline{WS}$ 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).

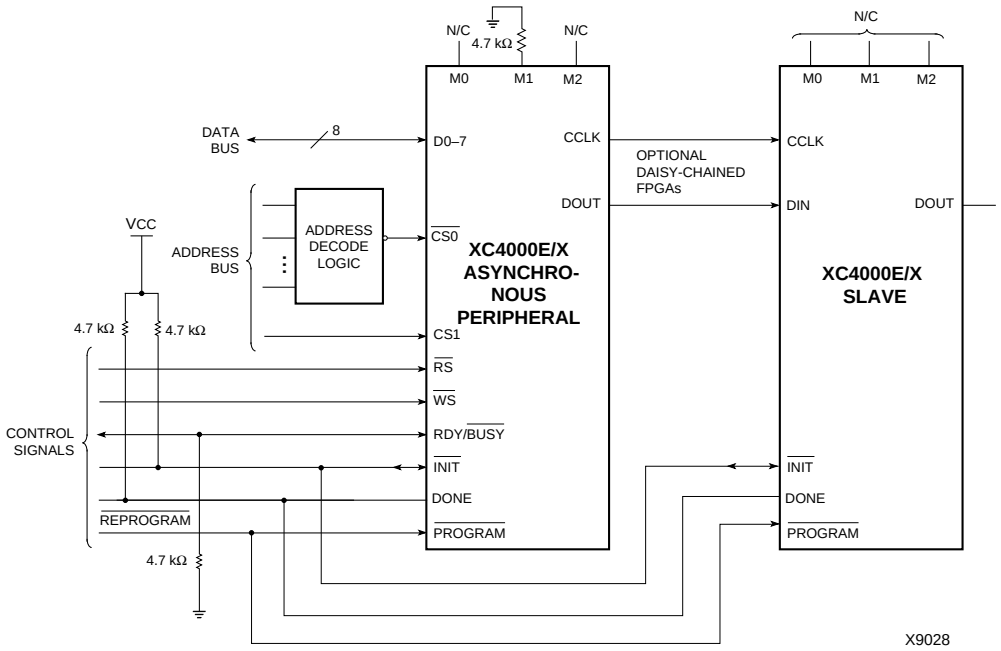
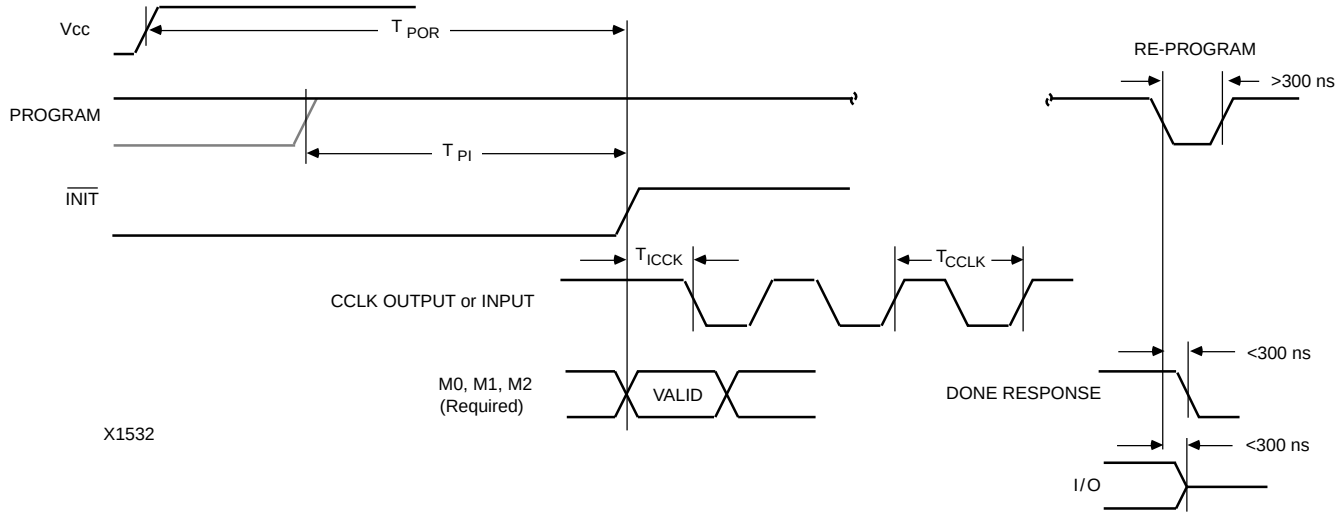


Figure 58: Asynchronous Peripheral Mode Circuit Diagram

Configuration Switching Characteristics



Master Modes (XC4000E/EX)

Description		Symbol	Min	Max	Units
Power-On Reset	M0 = High	T_{POR}	10	40	ms
	M0 = Low	T_{POR}	40	130	ms
Program Latency		T_{PI}	30	200	µs per CLB column
CCLK (output) Delay		T_{ICCK}	40	250	µs
CCLK (output) Period, slow		T_{CCLK}	640	2000	ns
CCLK (output) Period, fast		T_{CCLK}	80	250	ns

Master Modes (XC4000XL)

Description		Symbol	Min	Max	Units
Power-On Reset	M0 = High	T_{POR}	10	40	ms
	M0 = Low	T_{POR}	40	130	ms
Program Latency		T_{PI}	30	200	µs per CLB column
CCLK (output) Delay		T_{ICCK}	40	250	µs
CCLK (output) Period, slow		T_{CCLK}	540	1600	ns
CCLK (output) Period, fast		T_{CCLK}	67	200	ns

Slave and Peripheral Modes (All)

Description	Symbol	Min	Max	Units
Power-On Reset	T_{POR}	10	33	ms
Program Latency	T_{PI}	30	200	µs per CLB column
CCLK (input) Delay (required)	T_{ICCK}	4		µs
CCLK (input) Period (required)	T_{CCLK}	100		ns