



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	196
Number of Logic Elements/Cells	466
Total RAM Bits	6272
Number of I/O	77
Number of Gates	5000
Voltage - Supply	3V ~ 3.6V
Mounting Type	Surface Mount
Operating Temperature	-40°C ~ 100°C (TJ)
Package / Case	100-TQFP
Supplier Device Package	100-VQFP (14x14)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/xilinx/xc4005xl-3vq100i">https://www.e-xfl.com/product-detail/xilinx/xc4005xl-3vq100i</a>



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

### Flip-Flops

The CLB can pass the combinational output(s) to the interconnect network, but can also store the combinational 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
Latch	0	X	0*	X	Q
	1	1*	0*	X	Q
Both	0	1*	0*	D	D
	X	0	0*	X	Q

Legend:

X

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)

### 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.

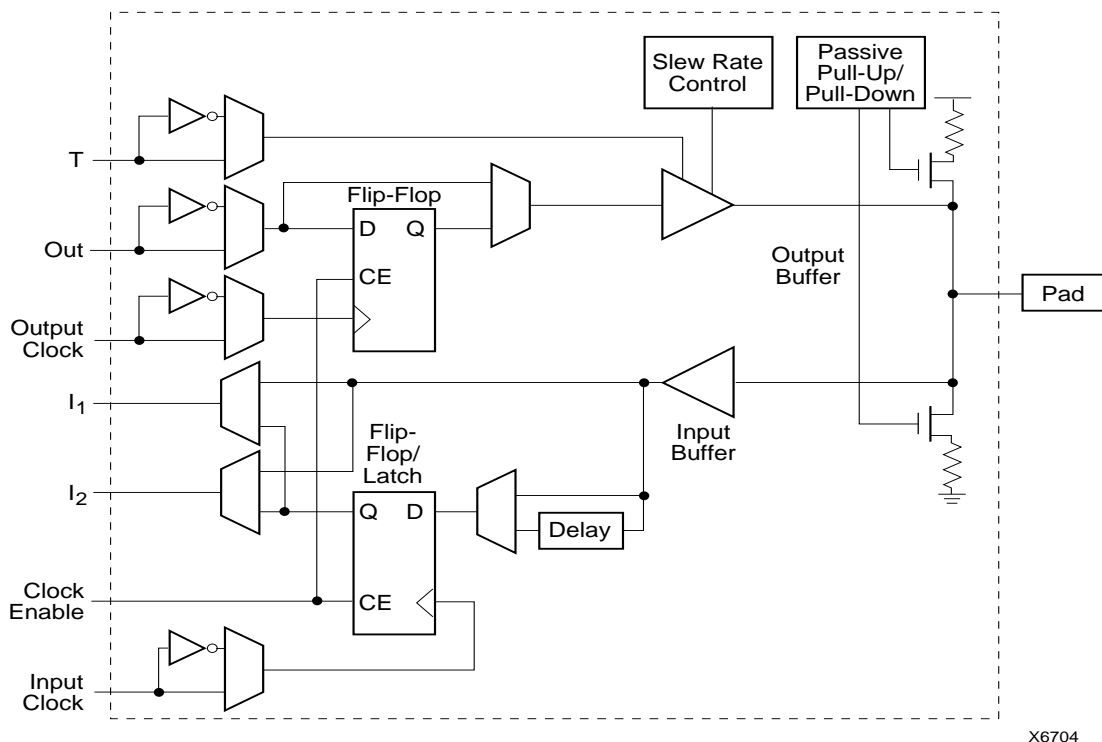
The CLB setup time is specified between the function generator inputs and the clock input K. Therefore, the specified CLB flip-flop setup time includes the delay through the function generator.

### Using Function Generators as RAM

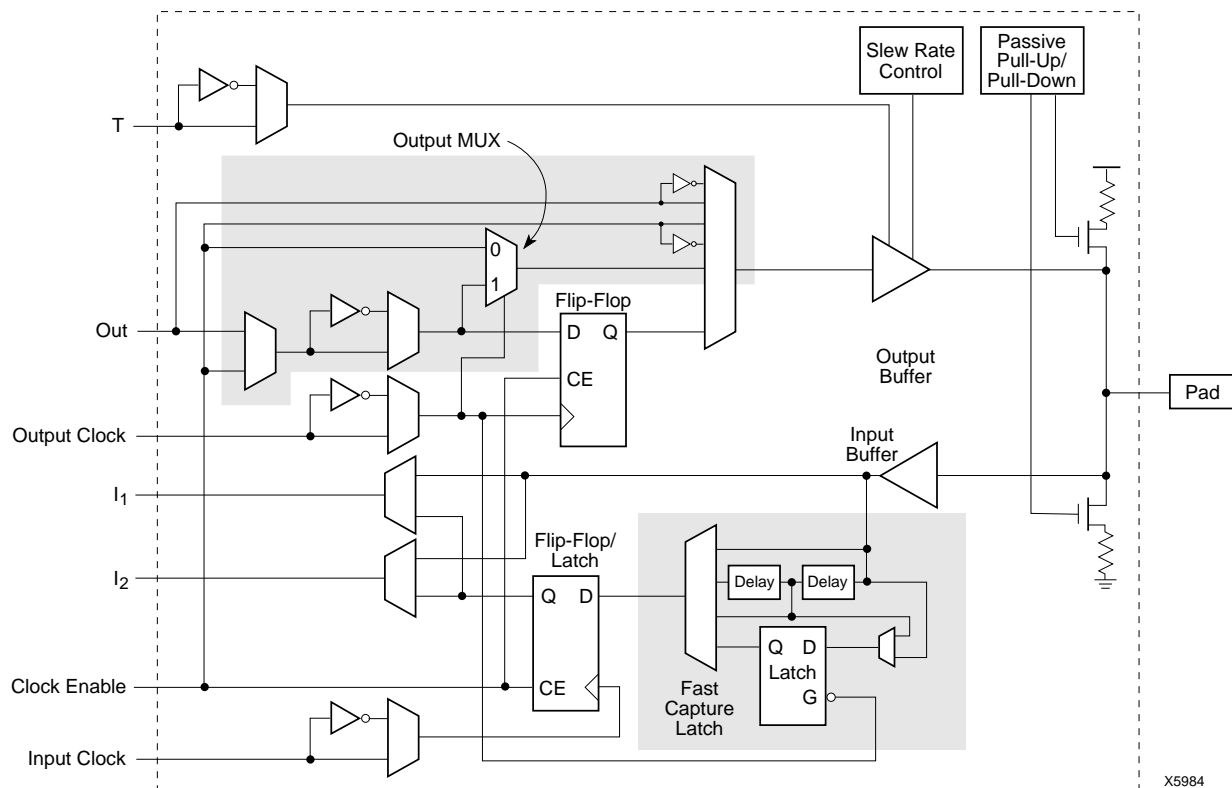
Optional modes for each CLB make the memory look-up tables in the F' and G' function generators usable as an array of Read/Write memory cells. Available modes are level-sensitive (similar to the XC4000/A/H families), edge-triggered, and dual-port edge-triggered. Depending on the selected mode, a single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



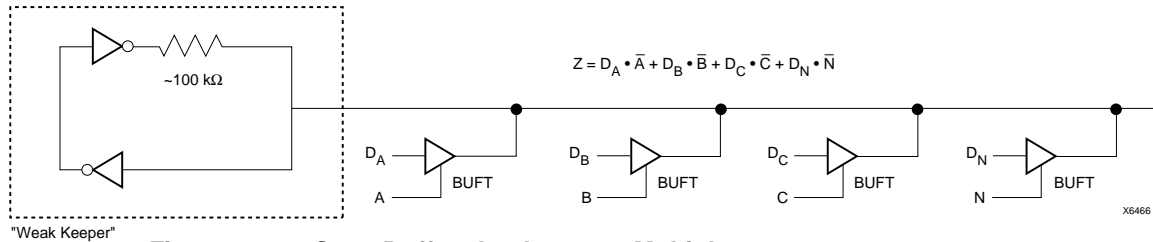
**Figure 13: Fast Carry Logic in XC4000E CLB (shaded area not present in XC4000X)**



**Figure 15: Simplified Block Diagram of XC4000E IOB**



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



**Figure 22: 3-State Buffers Implement a Multiplexer**

## Wide Edge Decoders

Dedicated decoder circuitry boosts the performance of wide decoding functions. When the address or data field is wider than the function generator inputs, FPGAs need multi-level decoding and are thus slower than PALs. XC4000 Series CLBs have nine inputs. Any decoder of up to nine inputs is, therefore, compact and fast. However, there is also a need for much wider decoders, especially for address decoding in large microprocessor systems.

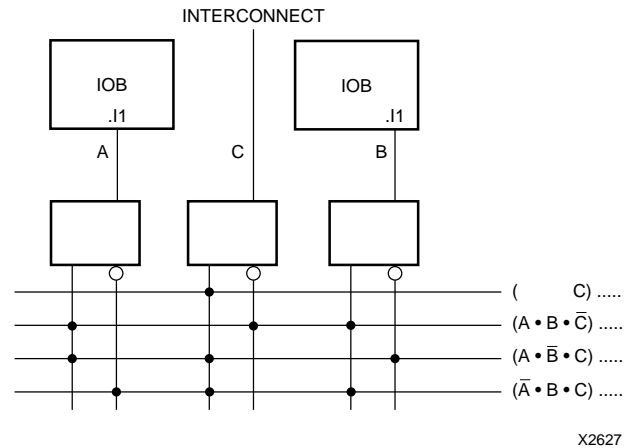
An XC4000 Series FPGA has four programmable decoders located on each edge of the device. The inputs to each decoder are any of the IOB I1 signals on that edge plus one local interconnect per CLB row or column. Each row or column of CLBs provides up to three variables or their complements., as shown in Figure 23. Each decoder generates a High output (resistor pull-up) when the AND condition of the selected inputs, or their complements, is true. This is analogous to a product term in typical PAL devices.

Each of these wired-AND gates is capable of accepting up to 42 inputs on the XC4005E and 72 on the XC4013E. There are up to 96 inputs for each decoder on the XC4028X and 132 on the XC4052X. The decoders may also be split in two when a larger number of narrower decoders are required, for a maximum of 32 decoders per device.

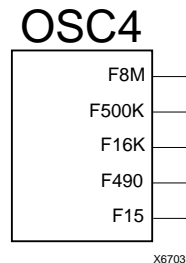
The decoder outputs can drive CLB inputs, so they can be combined with other logic to form a PAL-like AND/OR structure. The decoder outputs can also be routed directly to the chip outputs. For fastest speed, the output should be on the same chip edge as the decoder. Very large PALs can be emulated by ORing the decoder outputs in a CLB. This decoding feature covers what has long been considered a weakness of older FPGAs. Users often resorted to external PALs for simple but fast decoding functions. Now, the dedicated decoders in the XC4000 Series device can implement these functions fast and efficiently.

To use the wide edge decoders, place one or more of the WAND library symbols (WAND1, WAND4, WAND8, WAND16). Attach a DECODE attribute or property to each WAND symbol. Tie the outputs together and attach a PUL-

LUP symbol. Location attributes or properties such as L (left edge) or TR (right half of top edge) should also be used to ensure the correct placement of the decoder inputs.



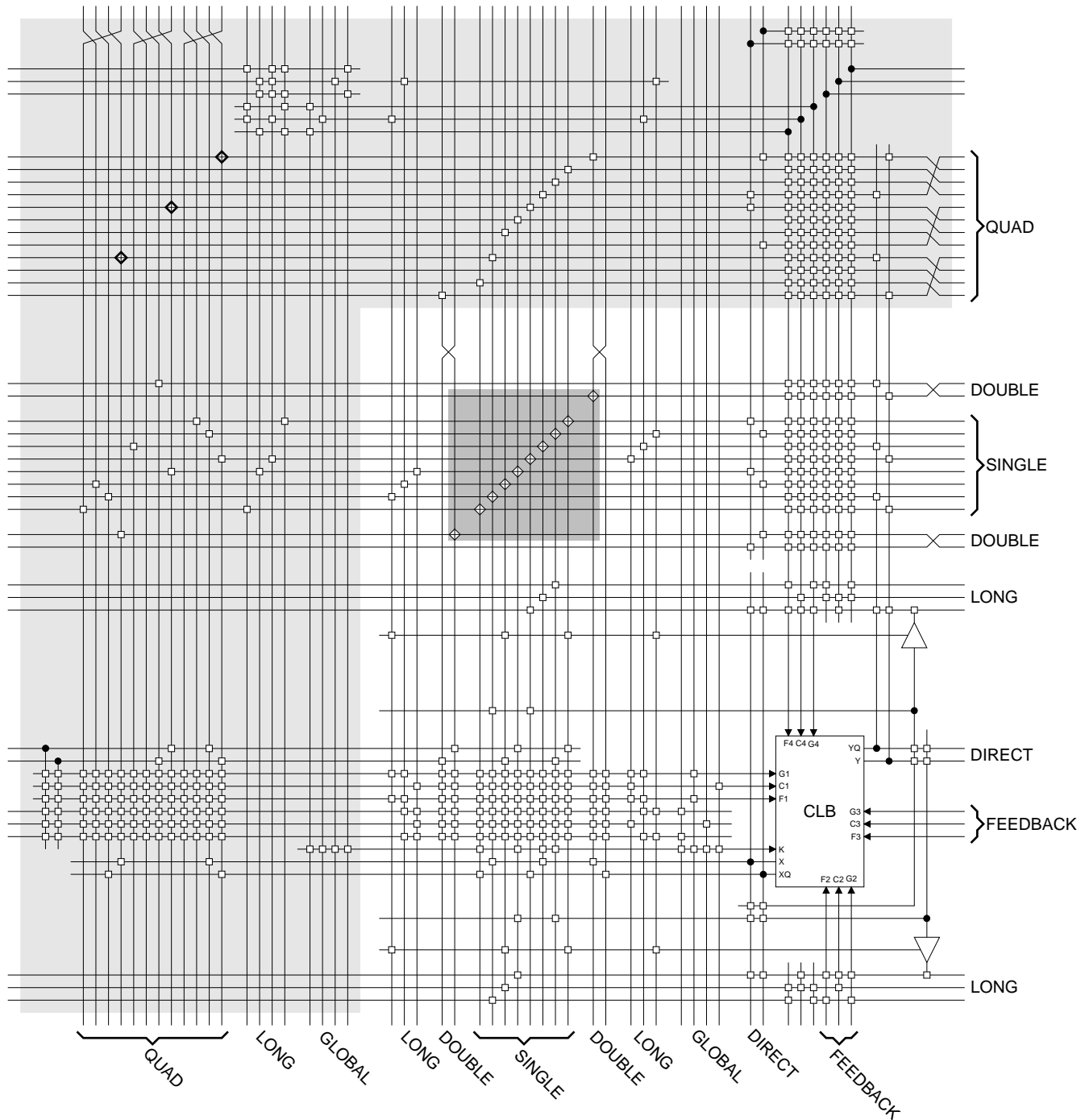
**Figure 23: XC4000 Series Edge Decoding Example**



**Figure 24: XC4000 Series Oscillator Symbol**

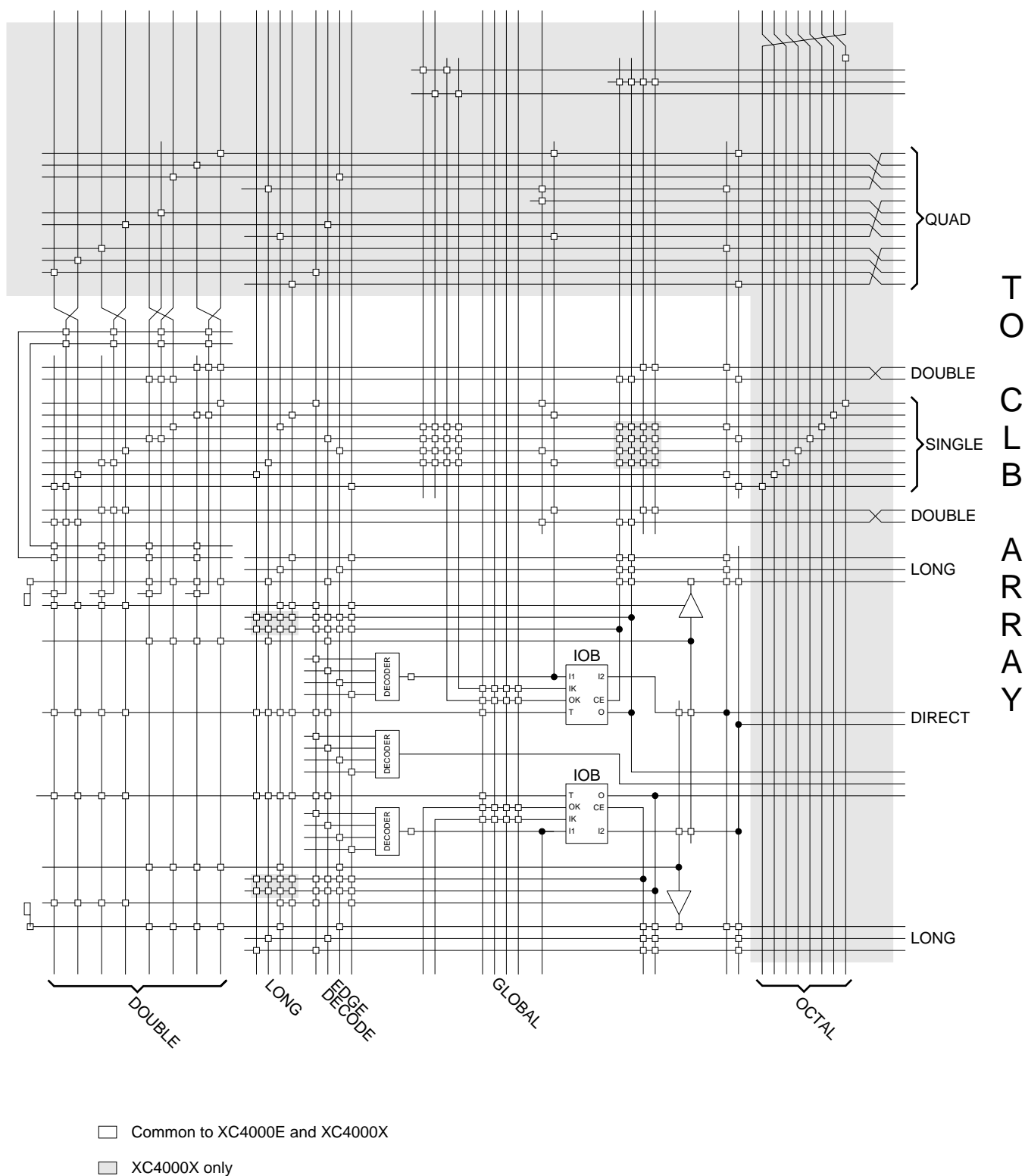
## On-Chip Oscillator

XC4000 Series devices include an internal oscillator. This oscillator is used to clock the power-on time-out, for configuration memory clearing, and as the source of CCLK in Master configuration modes. The oscillator runs at a nominal 8 MHz frequency that varies with process, Vcc, and temperature. The output frequency falls between 4 and 10 MHz.



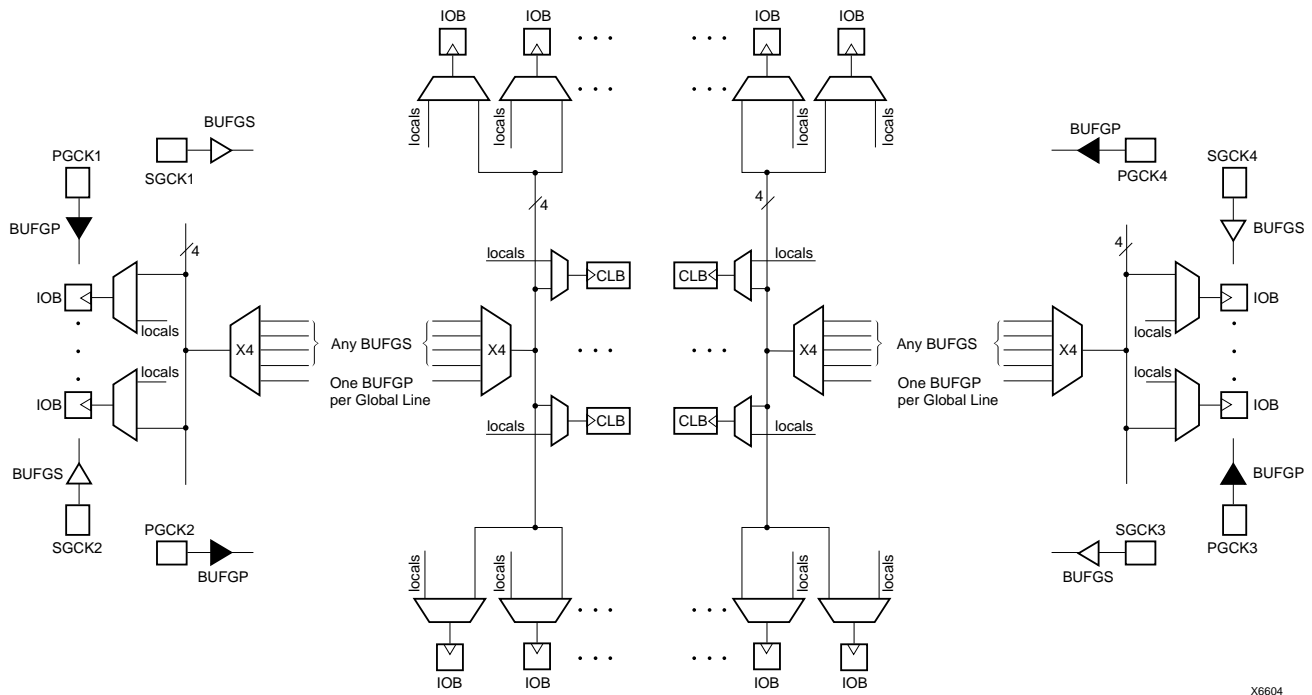
- Common to XC4000E and XC4000X
- XC4000X only
- Programmable Switch Matrix

**Figure 27: Detail of Programmable Interconnect Associated with XC4000 Series CLB**



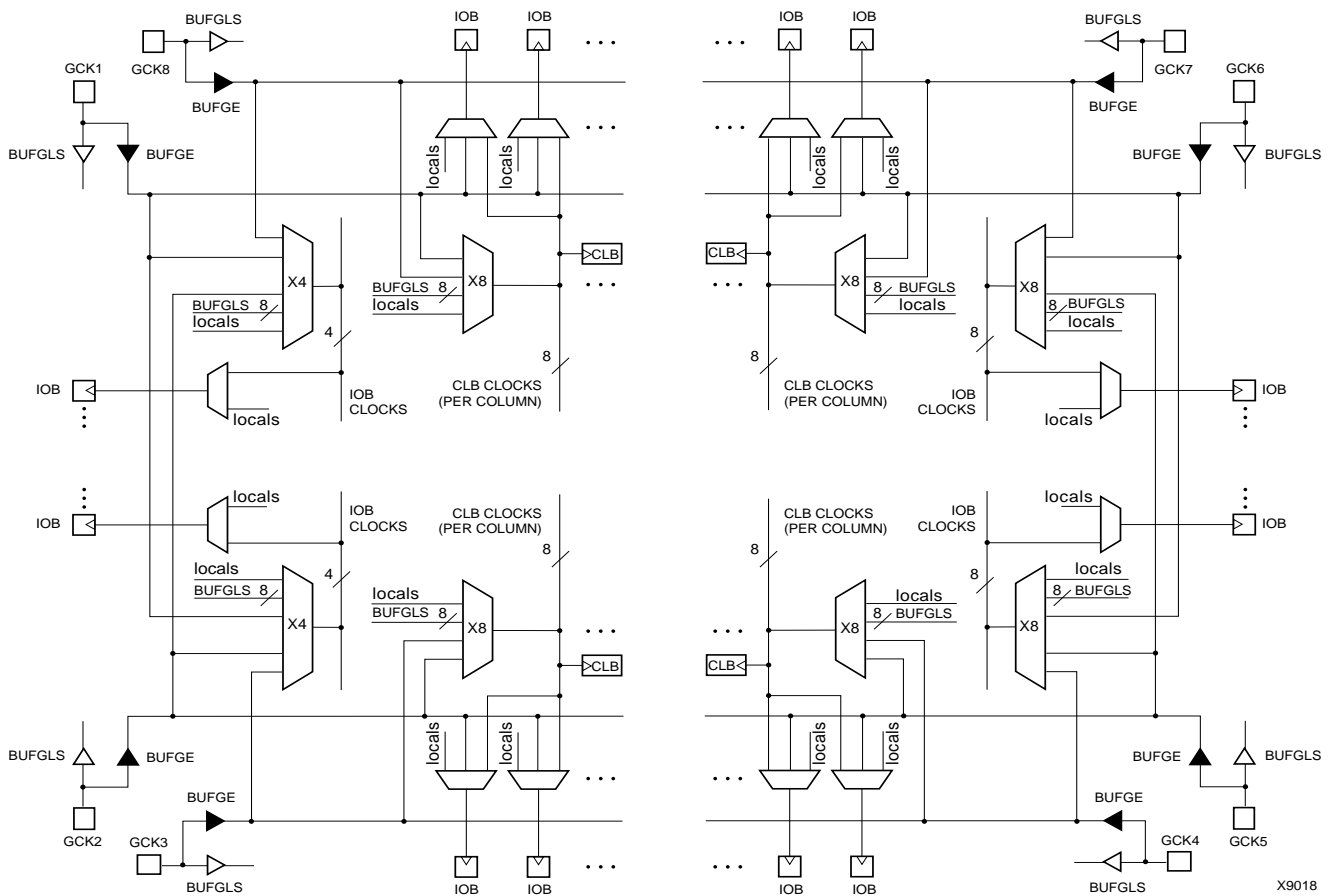
**Figure 33: Detail of Programmable Interconnect Associated with XC4000 Series IOB (Left Edge)**





X6604

**Figure 34: XC4000E Global Net Distribution**



X9018

**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 (BUFGSL)
- 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.

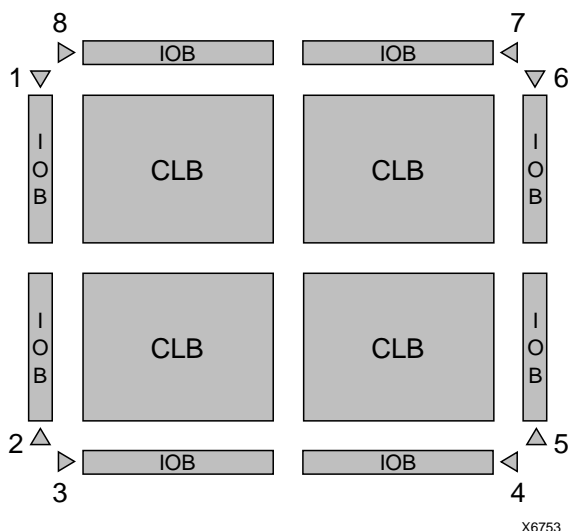
### Global Low-Skew Buffers

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 BUFGSL 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 BUFGSL 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.



**Figure 36: Any BUFGLS (GCK1 - GCK8) Can Drive Any or All Clock Inputs on the Device**

### Global Early Buffers

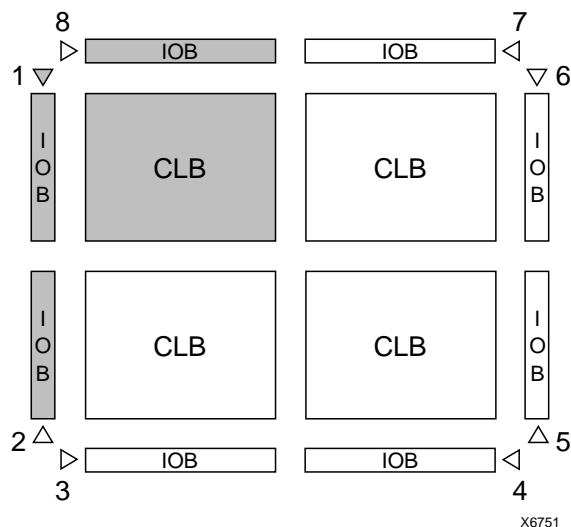
Each corner of the XC4000X device has two Global Early buffers. The primary purpose of the Global Early buffers is to provide an earlier clock access than the potentially heavily-loaded Global Low-Skew buffers. A clock source applied to both buffers will result in the Global Early clock edge occurring several nanoseconds earlier than the Global Low-Skew buffer clock edge, due to the lighter loading.

Global Early buffers also facilitate the fast capture of device inputs, using the Fast Capture latches described in **"IOB Input Signals"** on page 20. For Fast Capture, take a single clock signal, and route it through both a Global Early buffer and a Global Low-Skew buffer. (The two buffers share an input pad.) Use the Global Early buffer to clock the Fast Capture latch, and the Global Low-Skew buffer to clock the normal input flip-flop or latch, as shown in **Figure 17** on page 23.

The Global Early buffers can also be used to provide a fast Clock-to-Out on device output pins. However, an early clock in the output flip-flop IOB must be taken into consideration when calculating the internal clock speed for the design.

The Global Early buffers at the left and right edges of the chip have slightly different capabilities than the ones at the top and bottom. Refer to **Figure 37**, **Figure 38**, and **Figure 35** on page 36 while reading the following explanation.

Each Global Early buffer can access the eight vertical Global lines for all CLBs in the quadrant. Therefore, only one-fourth of the CLB clock pins can be accessed. This restriction is in large part responsible for the faster speed of the buffers, relative to the Global Low-Skew buffers.

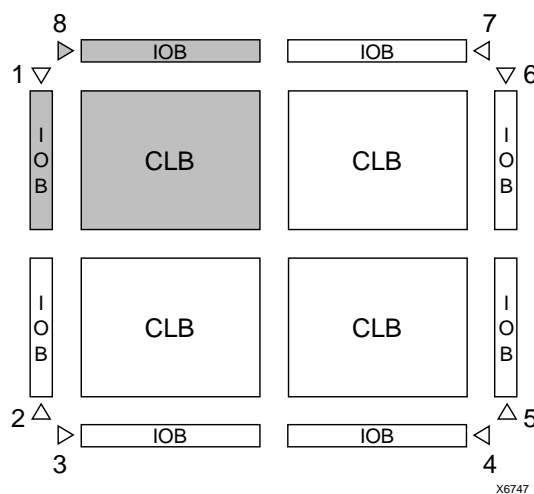


**Figure 37: Left and Right BUFGEs Can Drive Any or All Clock Inputs in Same Quadrant or Edge (GCK1 is shown. GCK2, GCK5 and GCK6 are similar.)**

The left-side Global Early buffers can each drive two of the four vertical lines accessing the IOBs on the entire left edge of the device. The right-side Global Early buffers can each drive two of the eight vertical lines accessing the IOBs on the entire right edge of the device. (See **Figure 37**.)

Each left and right Global Early buffer can also drive half of the IOBs along either the top or bottom edge of the device, using a dedicated line that can only be accessed through the Global Early buffers.

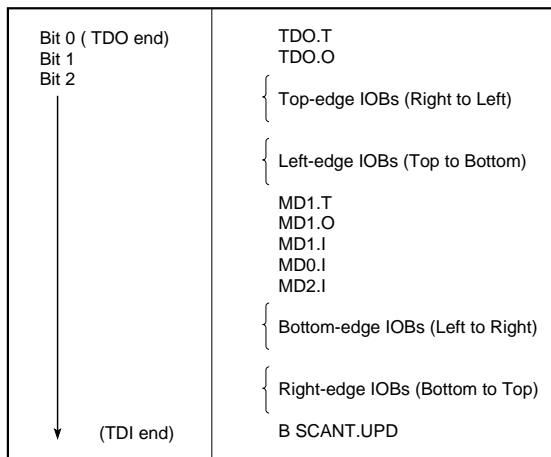
The top and bottom Global Early buffers can drive half of the IOBs along either the left or right edge of the device, as shown in **Figure 38**. They can only access the top and bottom IOBs via the CLB global lines.



**Figure 38: Top and Bottom BUFGEs Can Drive Any or All Clock Inputs in Same Quadrant (GCK8 is shown. GCK3, GCK4 and GCK7 are similar.)**

**Table 17: Boundary Scan Instructions**

Instruction	I1	I2	I0	Test Selected	TDO Source	I/O Data Source
0	0	0	0	EXTEST	DR	DR
0	0	1	1	SAMPLE/PR ELOAD	DR	Pin/Logic
0	1	0	0	USER 1	BSCAN. TDO1	User Logic
0	1	1	1	USER 2	BSCAN. TDO2	User Logic
1	0	0	0	READBACK	Readback Data	Pin/Logic
1	0	1	1	CONFIGURE	DOUT	Disabled
1	1	0	0	Reserved	—	—
1	1	1	1	BYPASS	Bypass Register	—



X6075

**Figure 42: Boundary Scan Bit Sequence**

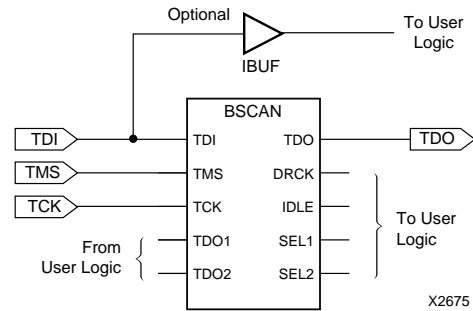
## Avoiding Inadvertent Boundary Scan

If TMS or TCK is used as user I/O, care must be taken to ensure that at least one of these pins is held constant during configuration. In some applications, a situation may occur where TMS or TCK is driven during configuration. This may cause the device to go into boundary scan mode and disrupt the configuration process.

To prevent activation of boundary scan during configuration, do either of the following:

- TMS: Tie High to put the Test Access Port controller in a benign RESET state
- TCK: Tie High or Low—don't toggle this clock input.

For more information regarding boundary scan, refer to the Xilinx Application Note XAPP 017.001, "Boundary Scan in XC4000E Devices."



**Figure 43: Boundary Scan Schematic Example**

## Configuration

Configuration is the process of loading design-specific programming data into one or more FPGAs to define the functional operation of the internal blocks and their interconnections. This is somewhat like loading the command registers of a programmable peripheral chip. XC4000 Series devices use several hundred bits of configuration data per CLB and its associated interconnects. Each configuration bit defines the state of a static memory cell that controls either a function look-up table bit, a multiplexer input, or an interconnect pass transistor. The XACTstep development system translates the design into a netlist file. It automatically partitions, places and routes the logic and generates the configuration data in PROM format.

## Special Purpose Pins

Three configuration mode pins (M2, M1, M0) are sampled prior to configuration to determine the configuration mode. After configuration, these pins can be used as auxiliary connections. M2 and M0 can be used as inputs, and M1 can be used as an output. The XACTstep development system does not use these resources unless they are explicitly specified in the design entry. This is done by placing a special pad symbol called MD2, MD1, or MD0 instead of the input or output pad symbol.

In XC4000 Series devices, the mode pins have weak pull-up resistors during configuration. With all three mode pins High, Slave Serial mode is selected, which is the most popular configuration mode. Therefore, for the most common configuration mode, the mode pins can be left unconnected. (Note, however, that the internal pull-up resistor value can be as high as 100 kΩ.) After configuration, these pins can individually have weak pull-up or pull-down resistors, as specified in the design. A pull-down resistor value of 4.7 kΩ is recommended.

These pins are located in the lower left chip corner and are near the readback nets. This location allows convenient routing if compatibility with the XC2000 and XC3000 family conventions of M0/RT, M1/RD is desired.

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.

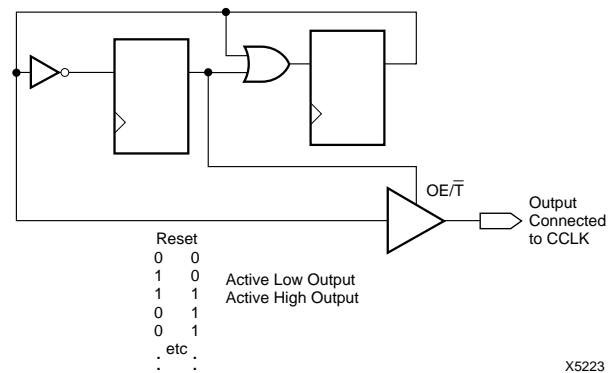
The user has some control over the relative timing of these events and can, therefore, make sure that they occur at the proper time and the finish point F is reached. Timing is controlled using options in the bitstream generation software.

### 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**



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)
I/Os	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:
- 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
  - 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)
I/Os	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:
- 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.
  - 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.

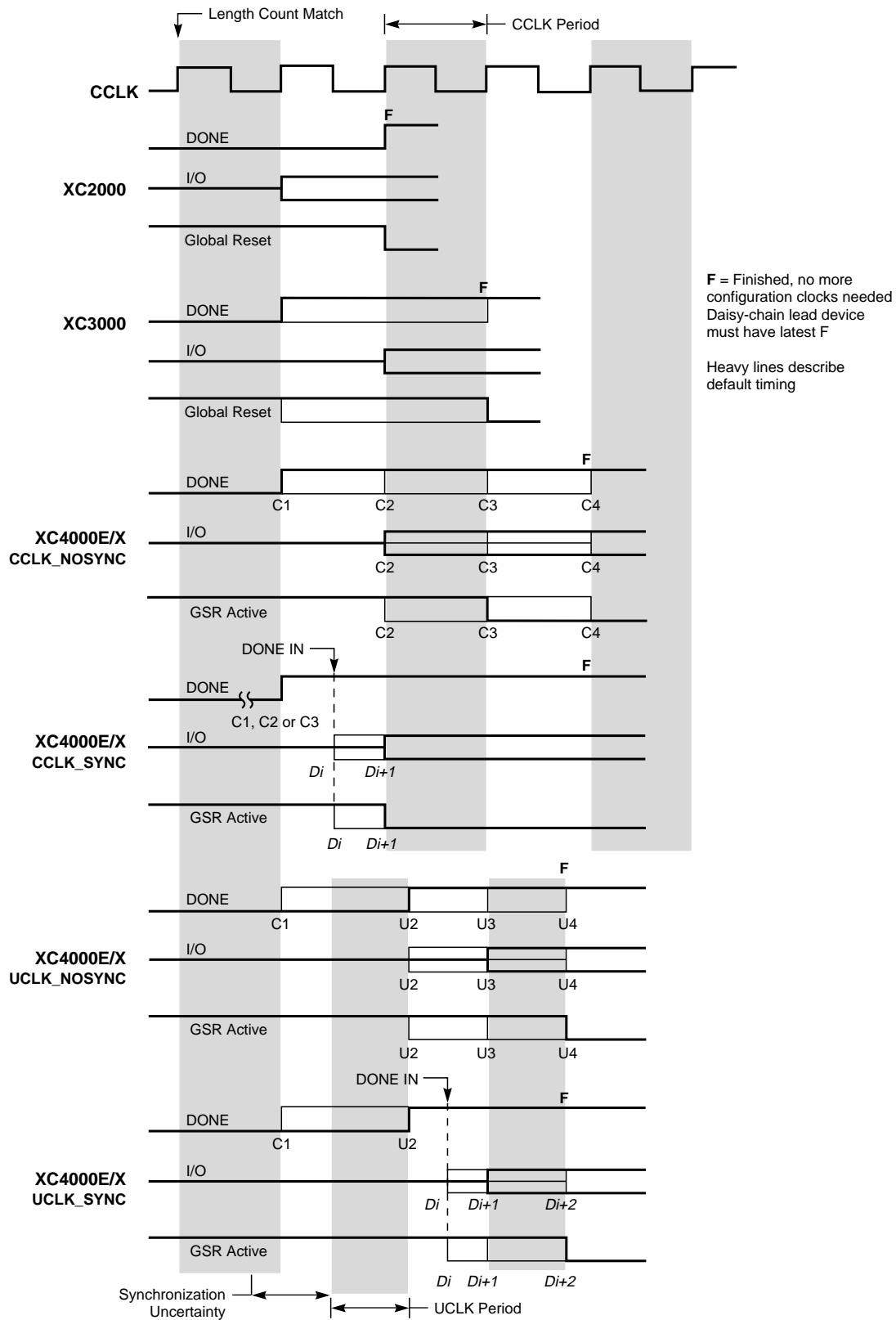
## Cyclic Redundancy Check (CRC) for Configuration and Readback

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 con-

figuration 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



X9024

**Figure 47: Start-up Timing**

### **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.



**Table 22: Pin Functions During Configuration**

CONFIGURATION MODE <M2:M1:M0>						USER OPERATION
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>	
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
LDC (LOW)	LDC (LOW)	LDC (LOW)	LDC (LOW)	LDC (LOW)	LDC (LOW)	I/O
INIT	INIT	INIT	INIT	INIT	INIT	I/O
DONE	DONE	DONE	DONE	DONE	DONE	DONE
PROGRAM (I)	PROGRAM (I)	PROGRAM (I)	PROGRAM (I)	PROGRAM (I)	PROGRAM (I)	PROGRAM
CCLK (I)	CCLK (O)	CCLK (I)	CCLK (O)	CCLK (O)	CCLK (O)	CCLK (I)
		RDY/BUSY (O)	RDY/BUSY (O)	RCLK (O)	RCLK (O)	I/O
			RS (I)			I/O
			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)
			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

## Master Serial Mode

In Master Serial mode, the CCLK output of the lead FPGA drives a Xilinx Serial PROM that feeds the FPGA DIN input. Each rising edge of the CCLK output increments the Serial PROM internal address counter. The next data bit is put on the SPROM data output, connected to the FPGA DIN pin. The lead FPGA accepts this data on the subsequent rising CCLK edge.

The lead FPGA then presents the preamble data—and all data that overflows the lead device—on its DOUT pin. There is an internal pipeline 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.

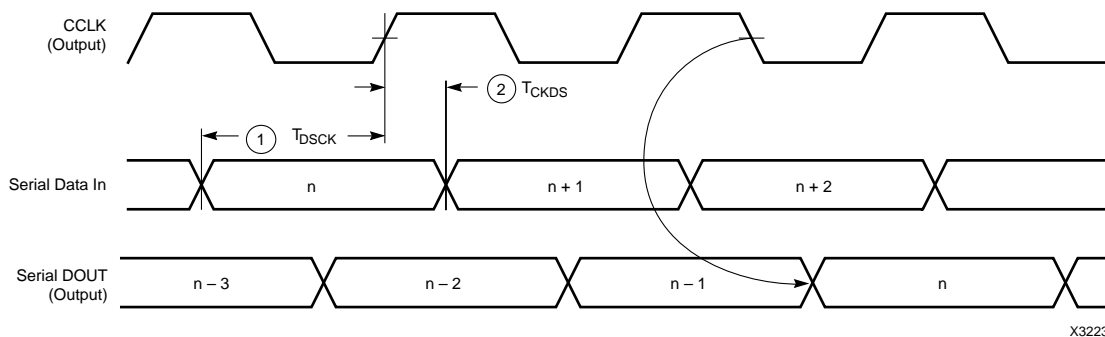
In the bitstream generation software, the user can specify Fast ConfigRate, which, starting several bits into the first frame, increases the CCLK frequency by a factor of eight.

For actual timing values please refer to “**Configuration Switching Characteristics**” on page 68. Be sure that the serial PROM and slaves are fast enough to support this data rate. XC2000, XC3000/A, and XC3100A devices do not support the Fast ConfigRate option.

The SPROM CE input can be driven from either  $\overline{\text{LDC}}$  or DONE. Using  $\overline{\text{LDC}}$  avoids potential contention on the DIN pin, if this pin is configured as user-I/O, but  $\overline{\text{LDC}}$  is then restricted to be a permanently High user output after configuration. Using DONE can also avoid contention on DIN, provided the early DONE option is invoked.

Figure 51 on page 60 shows a full master/slave system. The leftmost device is in Master Serial mode.

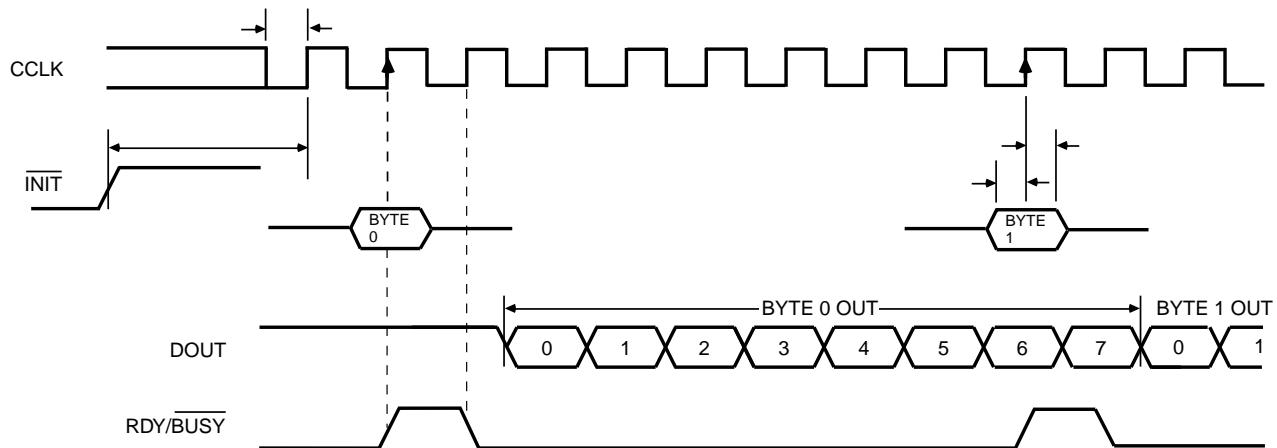
Master Serial mode is selected by a <000> on the mode pins (M2, M1, M0).



	Description	Symbol	Min	Max	Units
CCLK	DIN setup	1 $T_{\text{DSCK}}$	20		ns
	DIN hold	2 $T_{\text{CKDS}}$	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 PROGRAM Low until Vcc is valid.  
2. Master Serial mode timing is based on testing in slave mode.

**Figure 53: Master Serial Mode Programming Switching Characteristics**

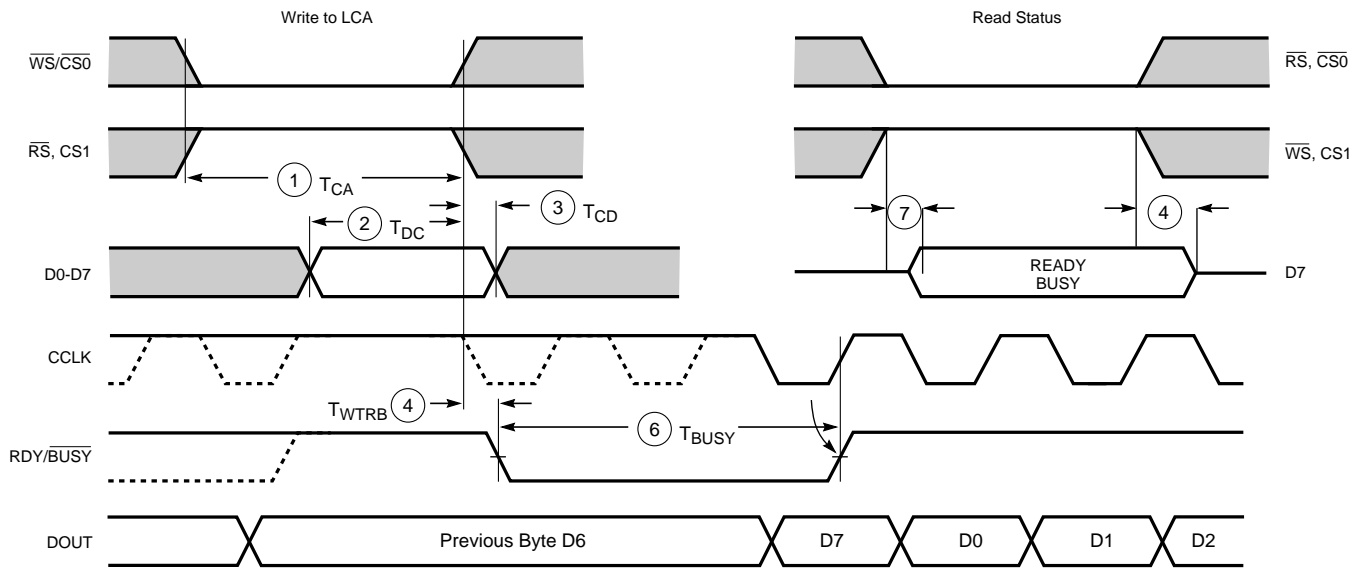


X6096

	Description	Symbol	Min	Max	Units
CCLK	INIT (High) setup time	$T_{IC}$	5		$\mu s$
	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 INIT goes High. Subsequent data bytes are clocked in on every eighth consecutive rising edge of CCLK.
  2. The RDY/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/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**



X6097

	Description	Symbol	Min	Max	Units
Write	Effective Write time (CS0, WS=Low; RS, CS1=High)	1 $T_{CA}$	100		ns
	DIN setup time	2 $T_{DC}$	60		ns
	DIN hold time	3 $T_{CD}$	0		ns
RDY	RDY/BUSY delay after end of Write or Read	4 $T_{WTRB}$		60	ns
	RDY/BUSY active after beginning of Read	7		60	ns
	RDY/BUSY Low output (Note 4)	6 $T_{BUSY}$	2	9	CCLK periods

- Notes:
1. Configuration must be delayed until the  $\overline{INIT}$  pins of all daisy-chained FPGAs are High.
  2. The time from the end of  $\overline{WS}$  to CCLK cycle for the new byte of data depends on the completion of previous byte processing and the phase of the internal timing generator for CCLK.
  3. CCLK and DOUT timing is tested in slave mode.
  4.  $T_{BUSY}$  indicates that the double-buffered parallel-to-serial converter is not yet ready to receive new data. The shortest  $T_{BUSY}$  occurs when a byte is loaded into an empty parallel-to-serial converter. The longest  $T_{BUSY}$  occurs when a new word is loaded into the input register before the second-level buffer has started shifting out data.

This timing diagram shows very relaxed requirements. Data need not be held beyond the rising edge of  $\overline{WS}$ . RDY/BUSY will go active within 60 ns after the end of  $\overline{WS}$ . A new write may be asserted immediately after RDY/BUSY goes Low, but write may not be terminated until RDY/BUSY has been High for one CCLK period.

**Figure 59: Asynchronous Peripheral Mode Programming Switching Characteristics**

## XC4000 Series Electrical Characteristics and Device-Specific Pinout Table

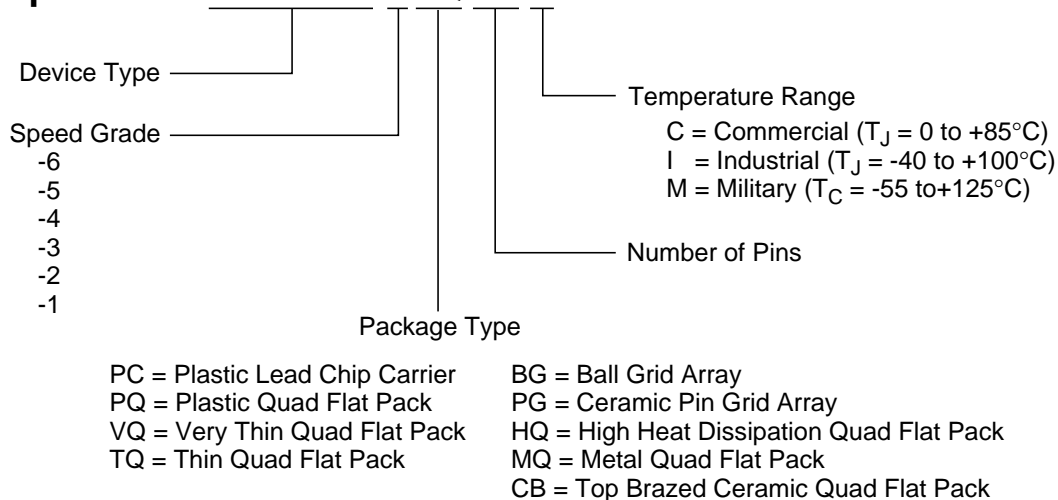
For the latest Electrical Characteristics and package/pinout information for each XC4000 Family, see the Xilinx web site at

[http://www.xilinx.com/xlnx/xweb/xil\\_publications\\_index.jsp](http://www.xilinx.com/xlnx/xweb/xil_publications_index.jsp)

## Ordering Information

### Example:

# XC4013E-3HQ240C



X9020

## Revision Control

Version	Description
3/30/98 (1.5)	Updated XC4000XL timing and added XC4002XL
1/29/99 (1.5)	Updated pin diagrams
5/14/99 (1.6)	Replaced Electrical Specification and pinout pages for E, EX, and XL families with separate updates and added URL link for electrical specifications/pinouts for Web users