



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 | 1024 |
| Number of Logic Elements/Cells | 2432 |
| Total RAM Bits | 32768 |
| Number of I/O | 129 |
| Number of Gates | 28000 |
| Voltage - Supply | 3V ~ 3.6V |
| Mounting Type | Surface Mount |
| Operating Temperature | 0°C ~ 85°C (TJ) |
| Package / Case | 160-BQFP Exposed Pad |
| Supplier Device Package | 160-PQFP (28x28) |
| Purchase URL | https://www.e-xfl.com/product-detail/xilinx/xc4028xl-3hq160c |

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 variables¹
- 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.

tions of the CLB, with the exception of the redefinition of the control signals. In 16x2 and 16x1 modes, the H' function generator can be used to implement Boolean functions of F', G', and D1, and the D flip-flops can latch the F', G', H', or D0 signals.

Single-Port Edge-Triggered Mode

Edge-triggered (synchronous) RAM simplifies timing requirements. XC4000 Series edge-triggered RAM timing operates like writing to a data register. Data and address are presented. The register is enabled for writing by a logic High on the write enable input, WE. Then a rising or falling clock edge loads the data into the register, as shown in Figure 3.

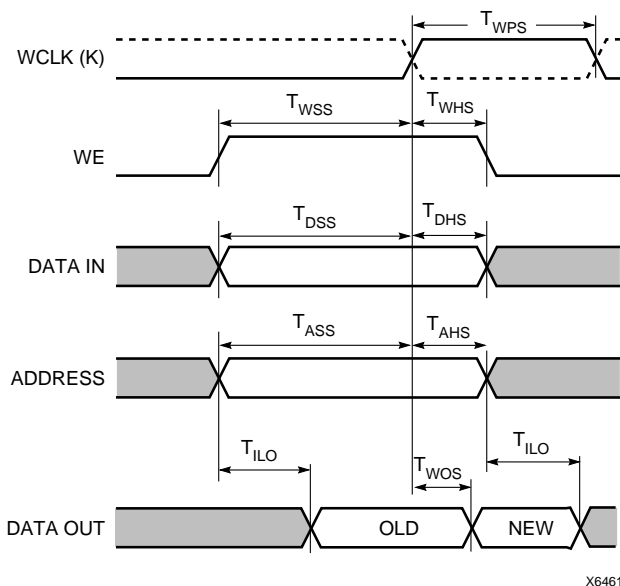


Figure 3: Edge-Triggered RAM Write Timing

Complex timing relationships between address, data, and write enable signals are not required, and the external write enable pulse becomes a simple clock enable. The active edge of WCLK latches the address, input data, and WE sig-

nals. An internal write pulse is generated that performs the write. See Figure 4 and Figure 5 for block diagrams of a CLB configured as 16x2 and 32x1 edge-triggered, single-port RAM.

The relationships between CLB pins and RAM inputs and outputs for single-port, edge-triggered mode are shown in Table 5.

The Write Clock input (WCLK) can be configured as active on either the rising edge (default) or the falling edge. It uses the same CLB pin (K) used to clock the CLB flip-flops, but it can be independently inverted. Consequently, the RAM output can optionally be registered within the same CLB either by the same clock edge as the RAM, or by the opposite edge of this clock. The sense of WCLK applies to both function generators in the CLB when both are configured as RAM.

The WE pin is active-High and is not invertible within the CLB.

Note: The pulse following the active edge of WCLK (T_{WPS} in Figure 3) must be less than one millisecond wide. For most applications, this requirement is not overly restrictive; however, it must not be forgotten. Stopping WCLK at this point in the write cycle could result in excessive current and even damage to the larger devices if many CLBs are configured as edge-triggered RAM.

Table 5: Single-Port Edge-Triggered RAM Signals

| RAM Signal | CLB Pin | Function |
|----------------|----------------------------------|----------------------------|
| D | D0 or D1 (16x2, 16x1), D0 (32x1) | Data In |
| A[3:0] | F1-F4 or G1-G4 | Address |
| A[4] | D1 (32x1) | Address |
| WE | WE | Write Enable |
| WCLK | K | Clock |
| SPO (Data Out) | F' or G' | Single Port Out (Data Out) |

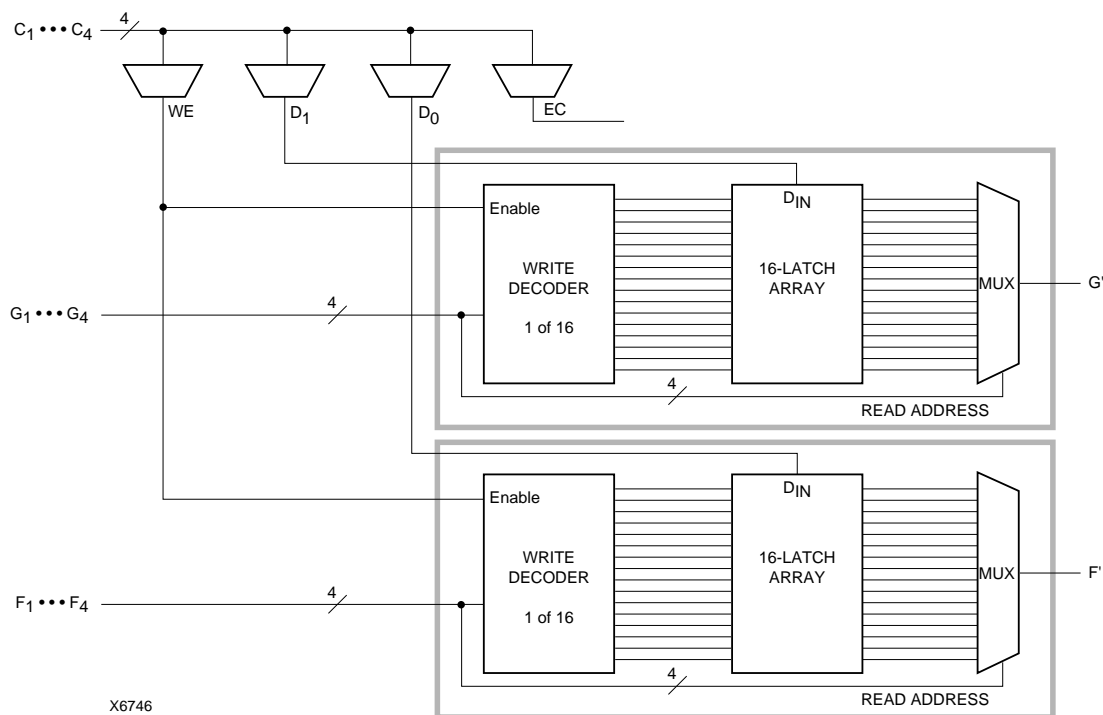


Figure 9: 16x2 (or 16x1) Level-Sensitive Single-Port RAM

6

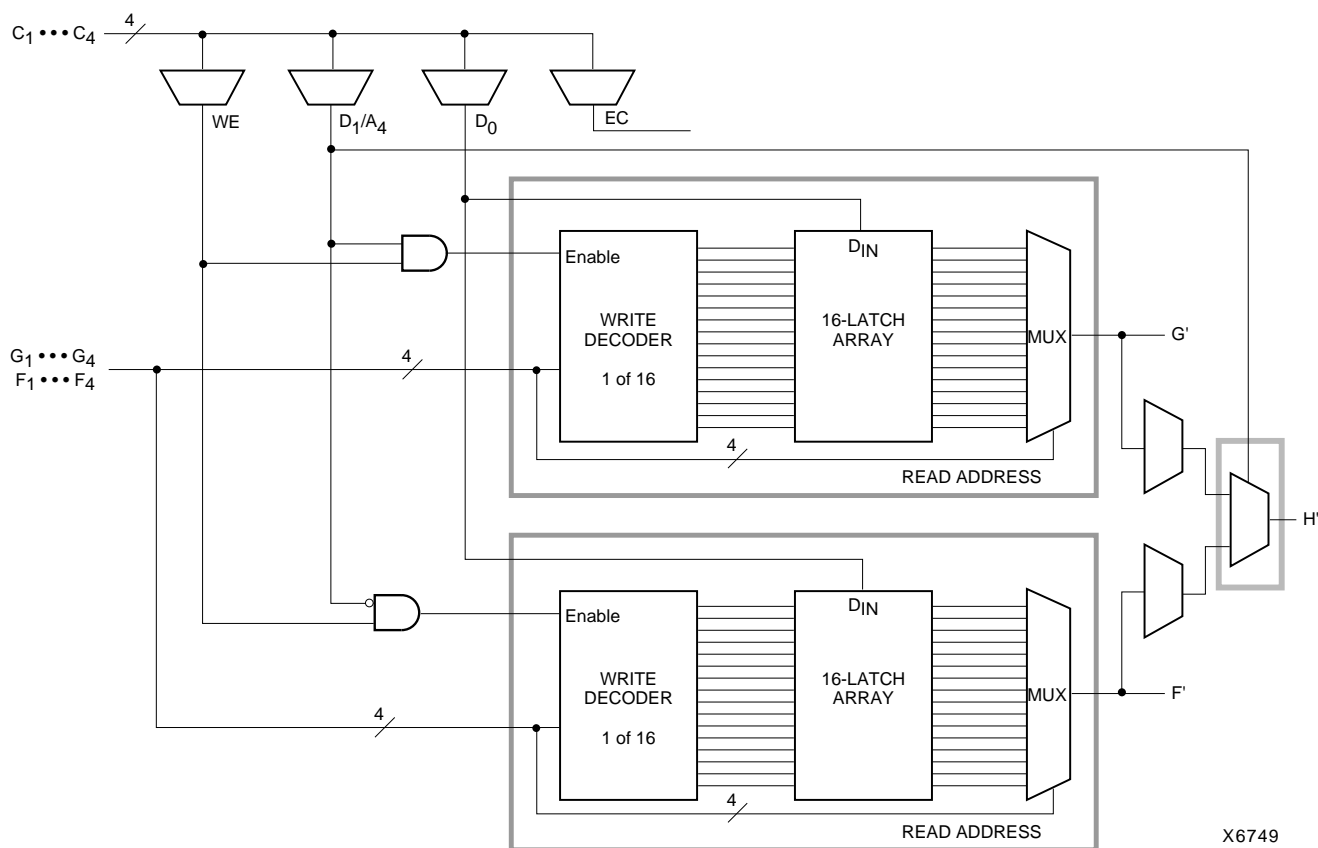


Figure 10: 32x1 Level-Sensitive Single-Port RAM (F and G addresses are identical)



X6699

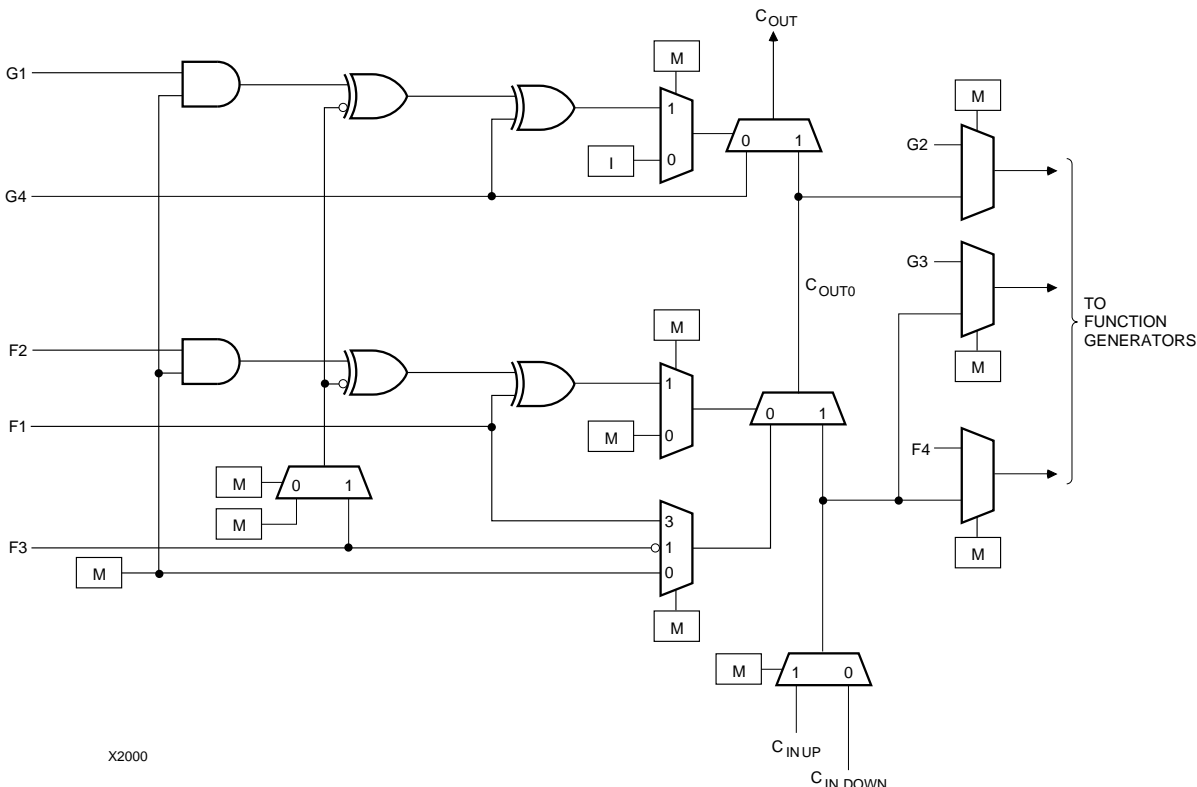


Figure 14: Detail of XC4000E Dedicated Carry Logic

Input/Output Blocks (IOBs)

User-configurable input/output blocks (IOBs) provide the interface between external package pins and the internal logic. Each IOB controls one package pin and can be configured for input, output, or bidirectional signals.

Figure 15 shows a simplified block diagram of the XC4000E IOB. A more complete diagram which includes the boundary scan logic of the XC4000E IOB can be found in Figure 40 on page 43, in the “Boundary Scan” section.

The XC4000X IOB contains some special features not included in the XC4000E IOB. These features are highlighted in a simplified block diagram found in **Figure 16**, and discussed throughout this section. When XC4000X special features are discussed, they are clearly identified in the text. Any feature not so identified is present in both XC4000E and XC4000X devices.

I/O Input Signals

Two paths, labeled I1 and I2 in **Figure 15** and **Figure 16**, bring input signals into the array. Inputs also connect to an input register that can be programmed as either an edge-triggered flip-flop or a level-sensitive latch.

The choice is made by placing the appropriate library symbol. For example, IFD is the basic input flip-flop (rising edge triggered), and ILD is the basic input latch (transparent-High). Variations with inverted clocks are available, and some combinations of latches and flip-flops can be implemented in a single IOB, as described in the *XACT Libraries Guide*.

The XC4000E inputs can be globally configured for either TTL (1.2V) or 5.0 volt CMOS thresholds, using an option in the bitstream generation software. There is a slight input hysteresis of about 300mV. The XC4000E output levels are also configurable; the two global adjustments of input threshold and output level are independent.

Inputs on the XC4000XL are TTL compatible and 3.3V CMOS compatible. Outputs on the XC4000XL are pulled to the 3.3V positive supply.

The inputs of XC4000 Series 5-Volt devices can be driven by the outputs of any 3.3-Volt device, if the 5-Volt inputs are in TTL mode.

Supported sources for XC4000 Series device inputs are shown in [Table 8](#).

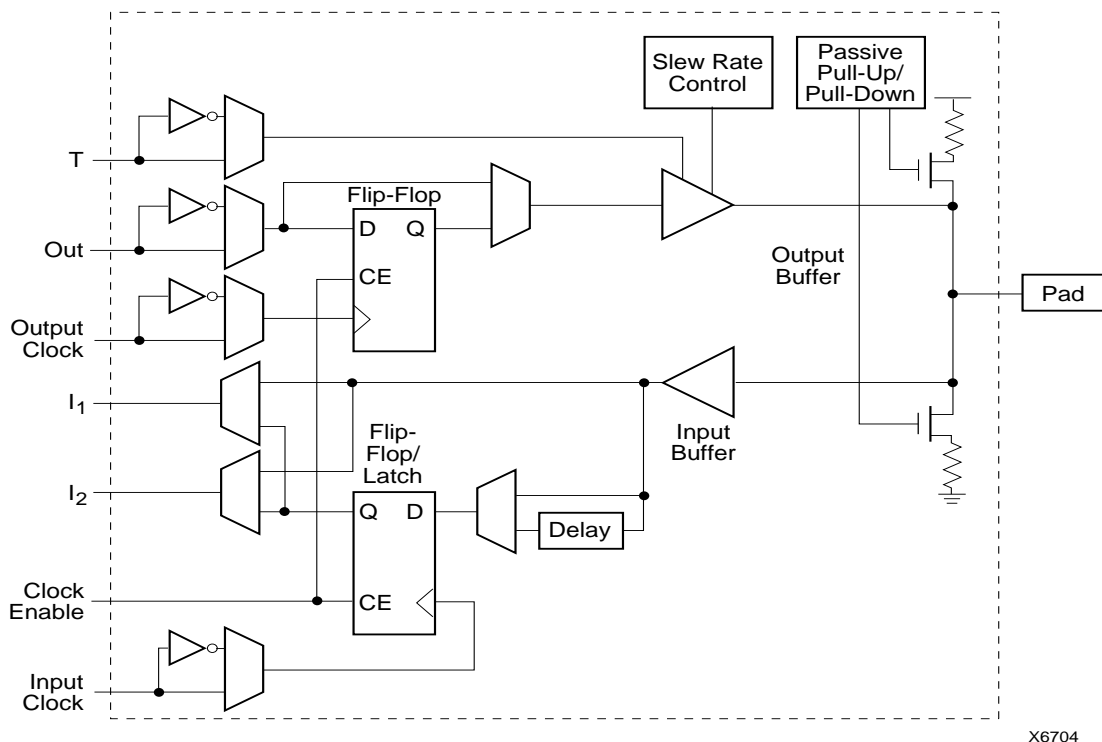


Figure 15: Simplified Block Diagram of XC4000E IOB

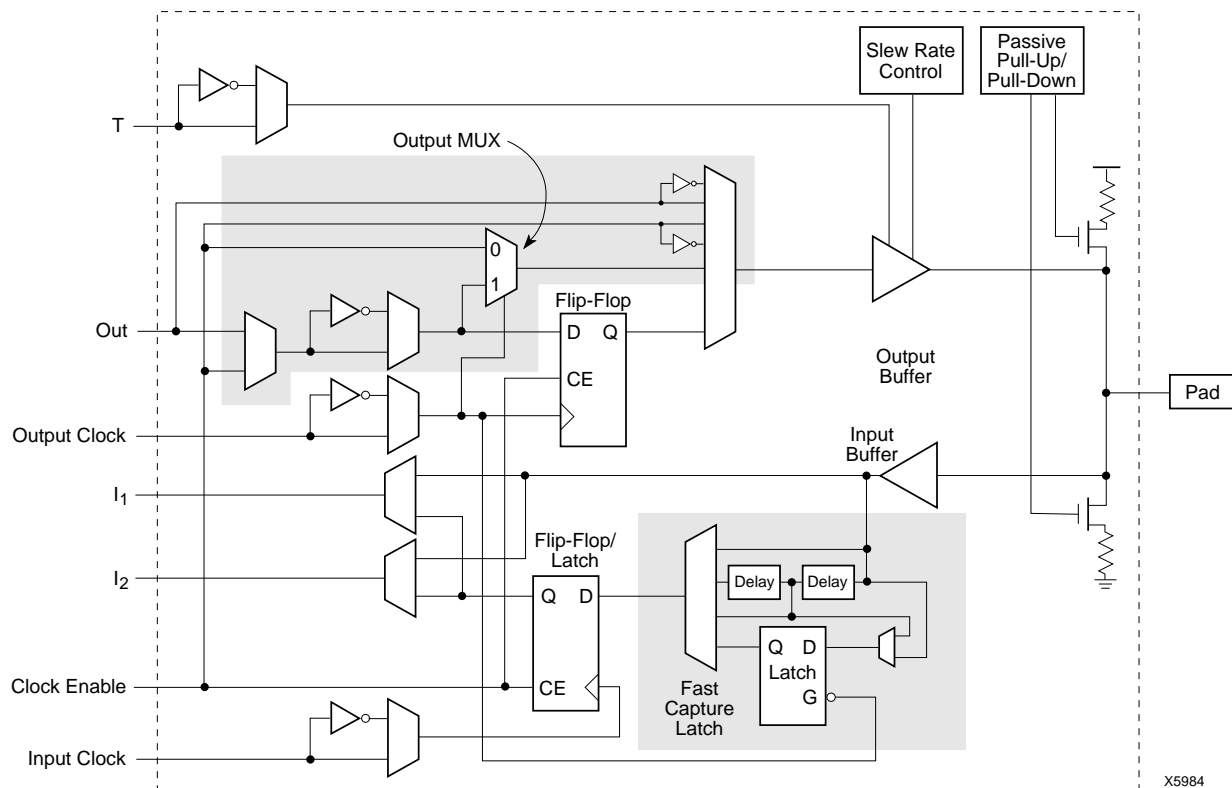


Figure 16: Simplified Block Diagram of XC4000X IOB (shaded areas indicate differences from XC4000E)

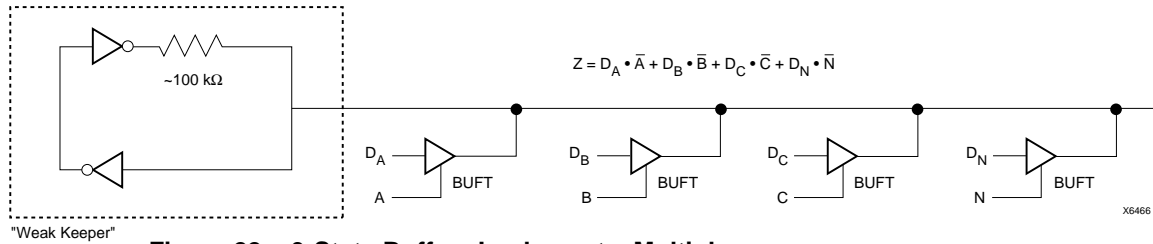


Figure 22: 3-State Buffers Implement a Multiplexer

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.

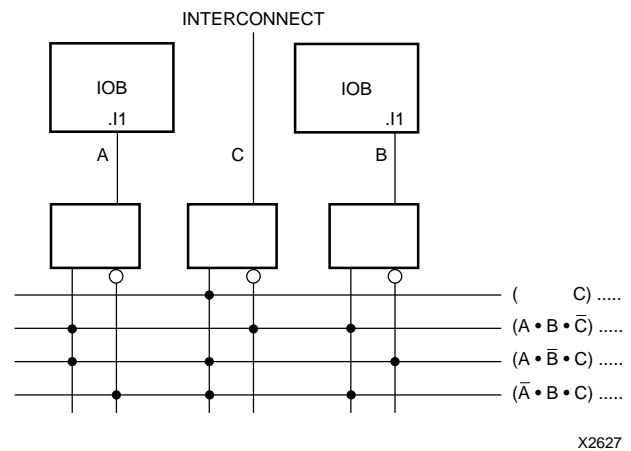


Figure 23: XC4000 Series Edge Decoding Example

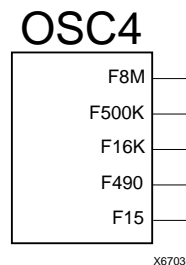


Figure 24: XC4000 Series Oscillator Symbol

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, V_{cc}, and temperature. The output frequency falls between 4 and 10 MHz.



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

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



Figure 29: Quad Lines (XC4000X only)

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

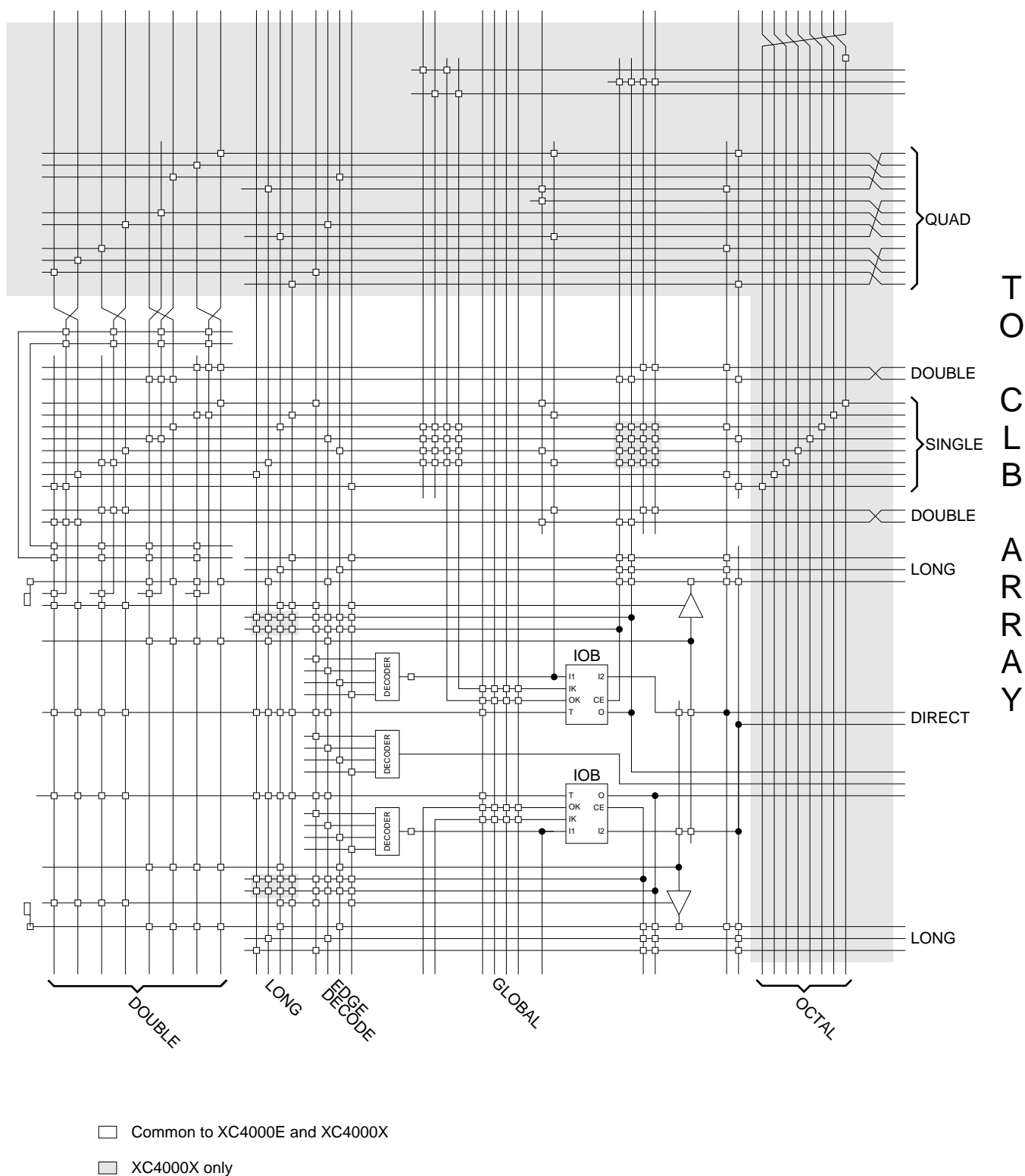
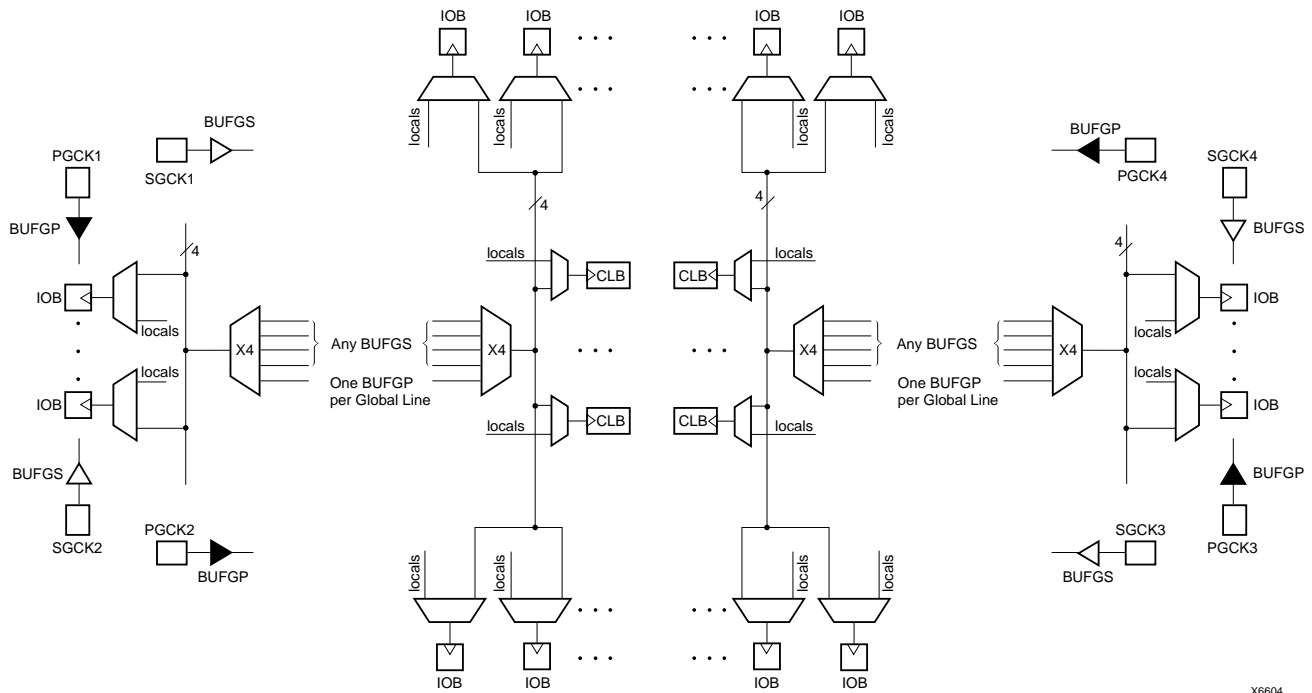
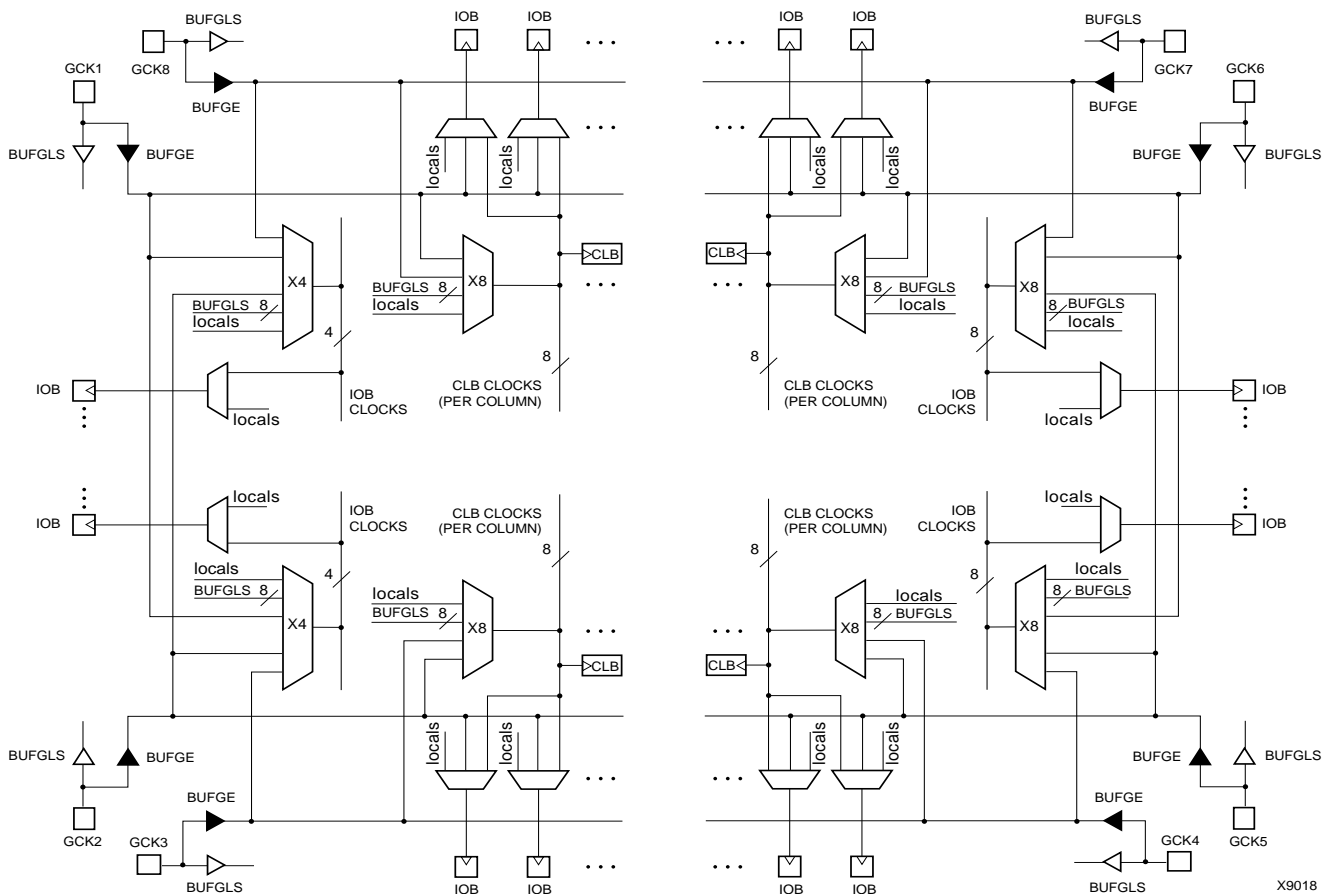


Figure 33: Detail of Programmable Interconnect Associated with XC4000 Series IOB (Left Edge)



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.

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 μ 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.

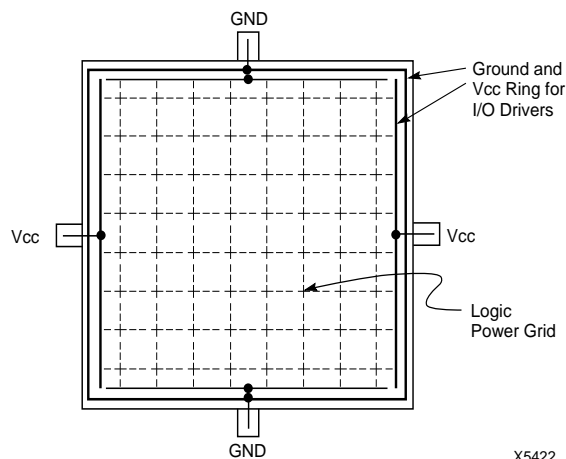


Figure 39: XC4000 Series Power Distribution

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 Ω - 100 k Ω pull-up resistor.

After configuration, if an IOB is unused it is configured as an input with a 50 k Ω - 100 k Ω 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. |
| Unrestricted User-Programmable I/O Pins | | | |
| 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 Ω - 100 k Ω) 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.



Figure 41: XC4000 Series Boundary Scan Logic

Instruction Set

The XC4000 Series boundary scan instruction set also includes instructions to configure the device and read back the configuration data. The instruction set is coded as shown in [Table 17](#).

Bit Sequence

The bit sequence within each IOB is: In, Out, 3-State. The input-only M0 and M2 mode pins contribute only the In bit to the boundary scan I/O data register, while the output-only M1 pin contributes all three bits.

The first two bits in the I/O data register are TDO.T and TDO.O, which can be used for the capture of internal signals. The final bit is BSCANT.UPD, which can be used to drive an internal net. These locations are primarily used by Xilinx for internal testing.

From a cavity-up view of the chip (as shown in XDE or Epic), starting in the upper right chip corner, the boundary scan data-register bits are ordered as shown in [Figure 42](#). The device-specific pinout tables for the XC4000 Series include the boundary scan locations for each IOB pin.

BSDL (Boundary Scan Description Language) files for XC4000 Series devices are available on the Xilinx FTP site.

Including Boundary Scan in a Schematic

If boundary scan is only to be used during configuration, no special schematic elements need be included in the schematic or HDL code. In this case, the special boundary scan pins TDI, TMS, TCK and TDO can be used for user functions after configuration.

To indicate that boundary scan remain enabled after configuration, place the BSCAN library symbol and connect the TDI, TMS, TCK and TDO pad symbols to the appropriate pins, as shown in [Figure 43](#).

Even if the boundary scan symbol is used in a schematic, the input pins TMS, TCK, and TDI can still be used as inputs to be routed to internal logic. Care must be taken not to force the chip into an undesired boundary scan state by inadvertently applying boundary scan input patterns to these pins. The simplest way to prevent this is to keep TMS High, and then apply whatever signal is desired to TDI and TCK.

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.

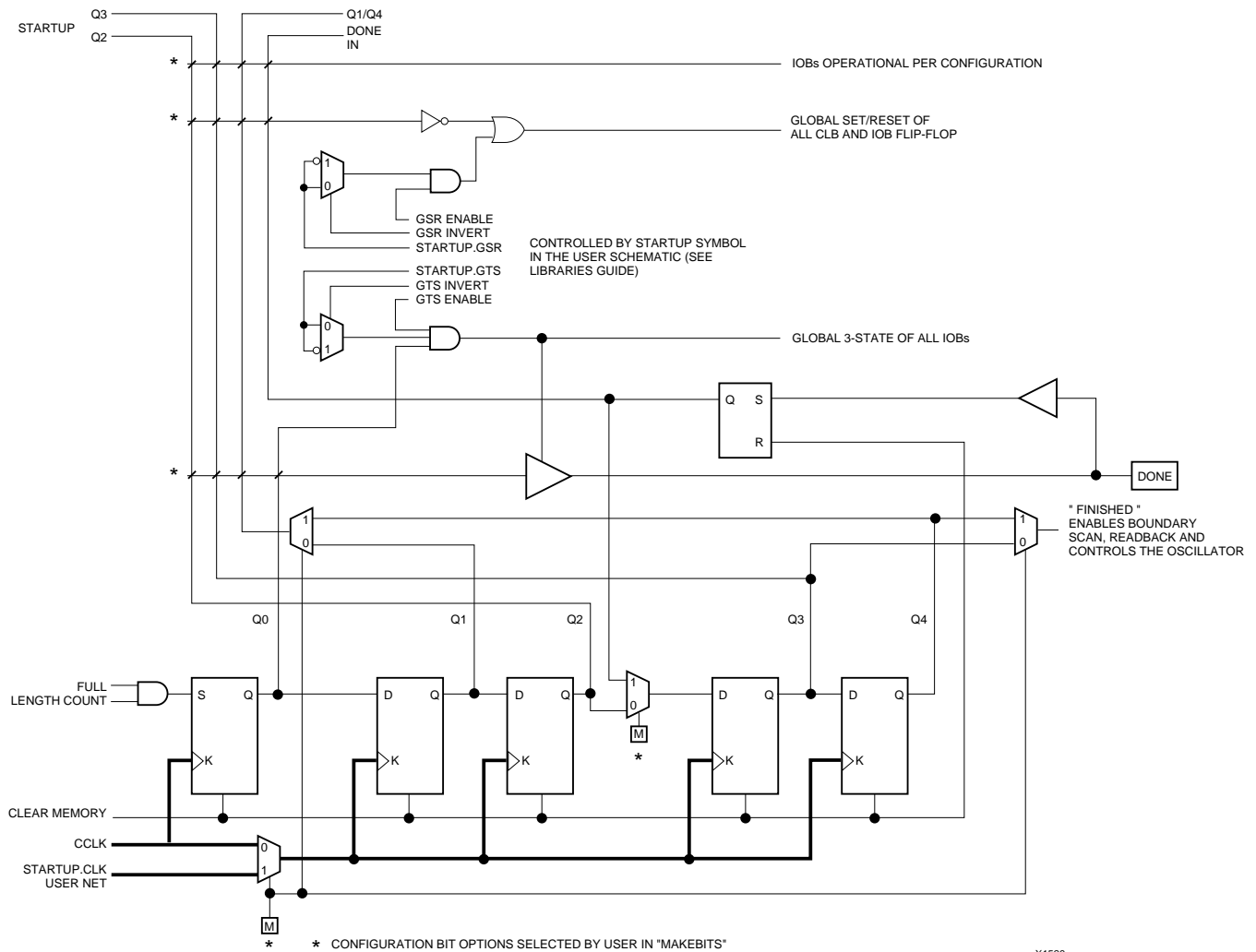


Figure 48: Start-up Logic

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 READ-

BACK 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.

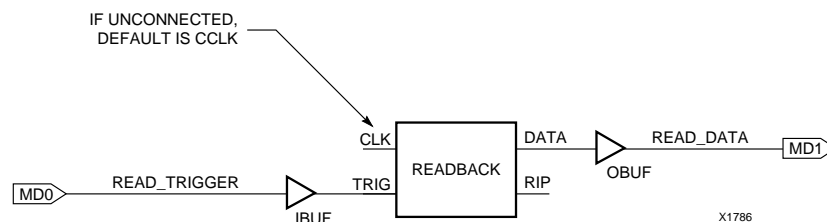


Figure 49: Readback Schematic Example

Readback Options

Readback options are: Read Capture, Read Abort, and Clock Select. They are set with the bitstream generation software.

Read Capture

When the Read Capture option is selected, the readback data stream includes sampled values of CLB and IOB signals. The rising edge of RDBK.TRIG latches the inverted values of the four CLB outputs, the IOB output flip-flops and the input signals I1 and I2. Note that while the bits describing configuration (interconnect, function generators, and RAM content) are *not* inverted, the CLB and IOB output signals *are* inverted.

When the Read Capture option is not selected, the values of the capture bits reflect the configuration data originally written to those memory locations.

If the RAM capability of the CLBs is used, RAM data are available in readback, since they directly overwrite the F and G function-table configuration of the CLB.

RDBK.TRIG is located in the lower-left corner of the device, as shown in [Figure 50](#).

Read Abort

When the Read Abort option is selected, a High-to-Low transition on RDBK.TRIG terminates the readback operation and prepares the logic to accept another trigger.

After an aborted readback, additional clocks (up to one readback clock per configuration frame) may be required to re-initialize the control logic. The status of readback is indicated by the output control net RDBK.RIP. RDBK.RIP is High whenever a readback is in progress.

Clock Select

CCLK is the default clock. However, the user can insert another clock on RDBK.CLK. Readback control and data are clocked on rising edges of RDBK.CLK. If readback must be inhibited for security reasons, the readback control nets are simply not connected.

RDBK.CLK is located in the lower right chip corner, as shown in [Figure 50](#).

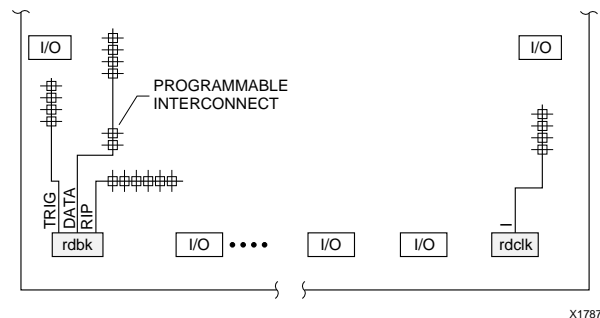


Figure 50: READBACK Symbol in Graphical Editor

Violating the Maximum High and Low Time Specification for the Readback Clock

The readback clock has a maximum High and Low time specification. In some cases, this specification cannot be met. For example, if a processor is controlling readback, an interrupt may force it to stop in the middle of a readback. This necessitates stopping the clock, and thus violating the specification.

The specification is mandatory only on clocking data at the end of a frame prior to the next start bit. The transfer mechanism will load the data to a shift register during the last six clock cycles of the frame, prior to the start bit of the following frame. This loading process is dynamic, and is the source of the maximum High and Low time requirements.

Therefore, the specification only applies to the six clock cycles prior to and including any start bit, including the clocks before the first start bit in the readback data stream. At other times, the frame data is already in the register and the register is not dynamic. Thus, it can be shifted out just like a regular shift register.

The user must precisely calculate the location of the readback data relative to the frame. The system must keep track of the position within a data frame, and disable interrupts before frame boundaries. Frame lengths and data formats are listed in [Table 19](#), [Table 20](#) and [Table 21](#).

Readback with the XChecker Cable

The XChecker Universal Download/Readback Cable and Logic Probe uses the readback feature for bitstream verification. It can also display selected internal signals on the PC or workstation screen, functioning as a low-cost in-circuit emulator.

Configuration Timing

The seven configuration modes are discussed in detail in this section. Timing specifications are included.

Slave Serial Mode

In Slave Serial mode, an external signal drives the CCLK input of the FPGA. The serial configuration bitstream must be available at the DIN input of the lead FPGA a short setup time before each 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 delay of 0.5 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.

Figure 51 shows a full master/slave system. An XC4000 Series device in Slave Serial mode should be connected as shown in the third device from the left.

Slave Serial mode is selected by a <111> on the mode pins (M2, M1, M0). Slave Serial is the default mode if the mode pins are left unconnected, as they have weak pull-up resistors during configuration.

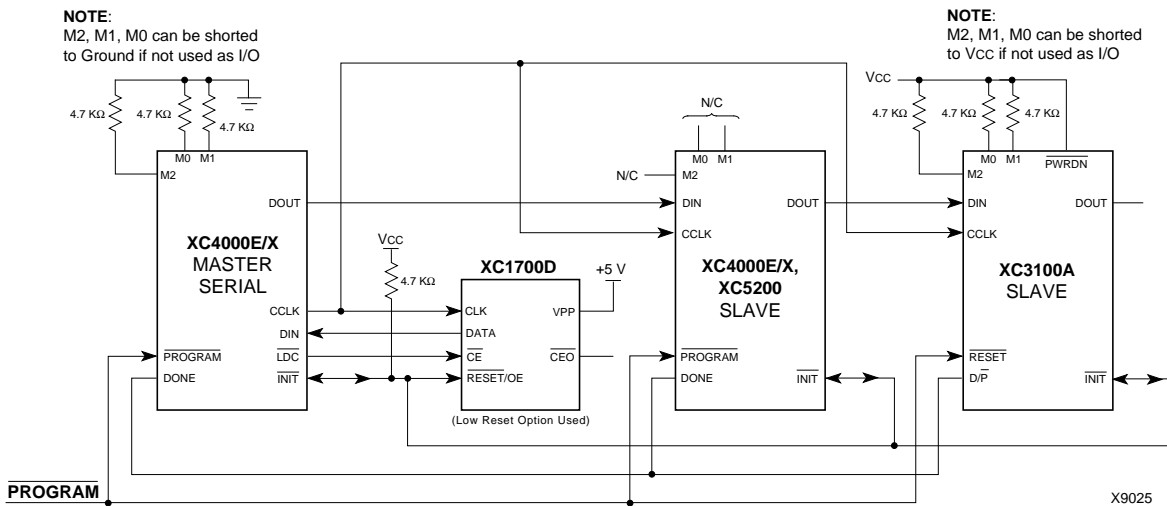
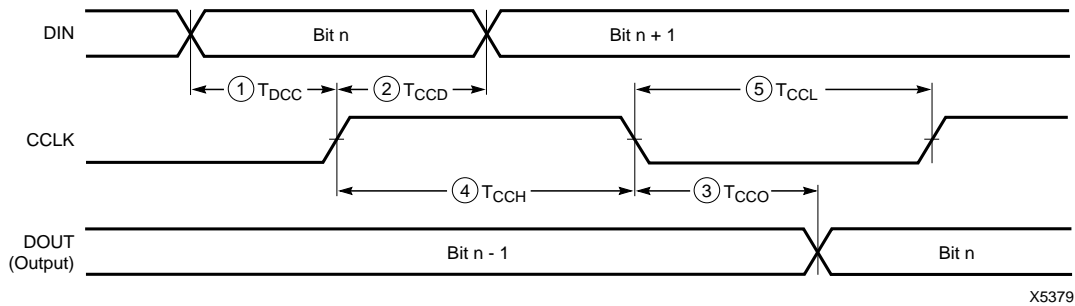


Figure 51: Master/Slave Serial Mode Circuit Diagram



| | Description | Symbol | | Min | Max | Units |
|------|-------------|--------|------------------|-----|-----|-------|
| CCLK | DIN setup | 1 | T _{DCC} | 20 | | ns |
| | DIN hold | 2 | T _{CCD} | 0 | | ns |
| | DIN to DOUT | 3 | T _{CCO} | | 30 | ns |
| | High time | 4 | T _{CCH} | 45 | | ns |
| | Low time | 5 | T _{CCL} | 45 | | ns |
| | Frequency | | F _{CC} | | 10 | MHz |

Note: Configuration must be delayed until the $\overline{\text{INIT}}$ pins of all daisy-chained FPGAs are High.

Figure 52: Slave Serial Mode Programming Switching Characteristics

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.

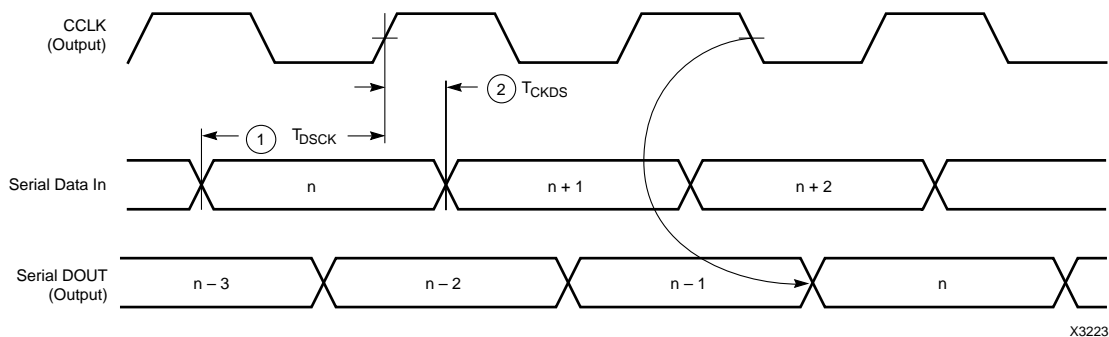
In the bitstream generation software, the user can specify Fast ConfigRate, which, starting several bits into the first frame, increases the CCLK frequency by a factor of eight.

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_{DSCK} | 20 | | ns |
| | DIN hold | 2 T_{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

XC4000 Series Electrical Characteristics and Device-Specific Pinout Table

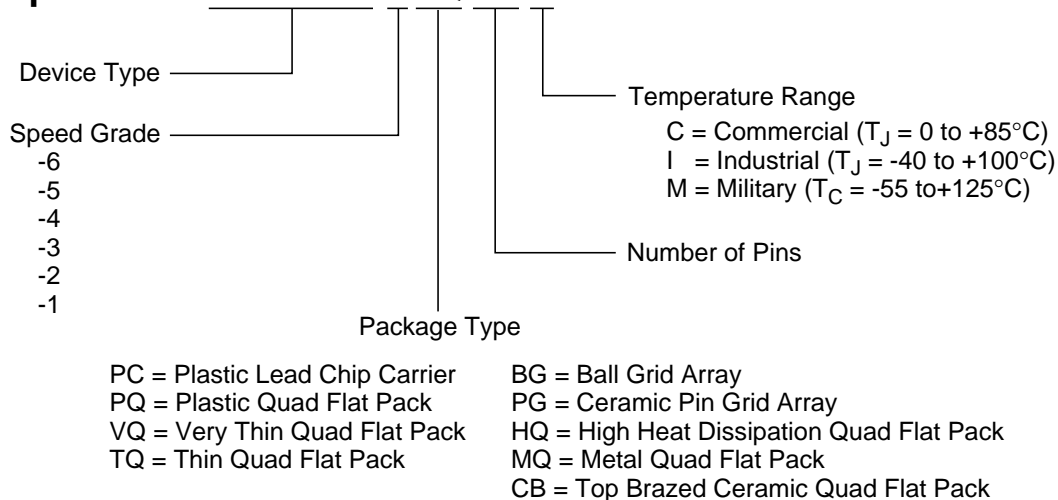
For the latest Electrical Characteristics and package/pinout information for each XC4000 Family, see the Xilinx web site at

http://www.xilinx.com/xlnx/xweb/xil_publications_index.jsp

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 |