Welcome to E-XFL.COM

Understanding Embedded - FPGAs (Field Programmable Gate Array)

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

Applications of Embedded - FPGAs

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

Details	
Product Status	Obsolete
Number of LABs/CLBs	100
Number of Logic Elements/Cells	238
Total RAM Bits	3200
Number of I/O	77
Number of Gates	3000
Voltage - Supply	4.75V ~ 5.25V
Mounting Type	Surface Mount
Operating Temperature	0°C ~ 85°C (TJ)
Package / Case	100-BQFP
Supplier Device Package	100-PQFP (20x14)
Purchase URL	https://www.e-xfl.com/product-detail/xilinx/xc4003-6pq100c

Speed Is Enhanced Two Ways

Delays in LCA-based designs are layout dependent. While this makes it hard to predict a worst-case guaranteed performance, there is a rule of thumb designers can consider — the system clock rate should not exceed one third to one half of the specified toggle rate. Critical portions of a design, shift registers and simple counters, can run faster — approximately two thirds of the specified toggle rate.

The XC4000 family can run at synchronous system clock rates of up to 60 MHz. This increase in performance over the previous families stems from two basic improvements: improved architecture and more abundant routing resources.

Improved Architecture

***More Inputs*:** The versatility of the CLB function generators improves system speed significantly. Table 3 shows how the XC4000 families implement many functions more efficiently and faster than is possible with XC3000 devices. A 9-bit parity checker, for example, can be implemented in one CLB with a propagation delay of 7 ns. Using a XC3000-family device, the same function requires two CLBs with a propagation delay of 2 x 5.5 ns = 11 ns. One XC4000 CLB can determine whether two 4-bit words are identical, again with a 7-ns propagation delay. The ninth input can be used for simple ripple expansion of this identity comparator (25.5 ns over 16 bits, 51.5 ns over 32 bits), or a 2-layer identity comparator can generate the result of a 32-bit comparison in 15 ns, at the cost of a single extra CLB. Simpler functions like multiplexers also benefit from the greater flexibility of the XC4000-families CLB. A 16-input multiplexer uses 5 CLBs and has a delay of only 13.5 ns.

More Outputs: The CLB can pass the combinatorial output(s) to the interconnect network, but can also store the combinatorial result(s) or other incoming data in one or two flip-flops, and connect their outputs to the interconnect network as well. With XC3000-families CLBs the designer has to make a choice, either output the combinatorial function or the stored value. In the XC4000 families, the flip flops can be used as registers or shift registers without blocking the function generators from performing a different, perhaps unrelated task. This increases the functional density of the devices.

When a function generator drives a flip-flop in a CLB, the combinatorial propagation delay *overlaps completely* with the set-up time of the flip-flop. The set-up time is specified between the function generator inputs and the clock input. This represents a performance advantage over competing technologies where combinatorial delays must be added to the flip-flop set-up time.

Fast Carry: As described earlier, each CLB includes high-speed carry logic that can be activated by configuration. The two 4-input function generators can be configured as a 2-bit adder with built-in hidden carry that can be expanded to any length. This dedicated carry circuitry is so fast and efficient that conventional speed-up methods like carry generate/propagate are meaningless even at the 16-bit level, and of marginal benefit at the 32-bit level.

A 16-bit adder requires nine CLBs and has a combinatorial carry delay of 20.5 ns. Compare that to the 30 CLBs and 50 ns, or 41 CLBs and 30 ns in the XC3000 family.

The fast-carry logic opens the door to many new applications involving arithmetic operation, where the previous generations of FPGAs were not fast and/or not efficient enough. High-speed address offset calculations in microprocessor or graphics systems, and high-speed addition in digital signal processing are two typical applications.

Faster and More Efficient Counters: The XC4000-families fast-carry logic puts two counter bits into each CLB and runs them at a clock rate of up to 42 MHz for 16 bits, whether the counters are loadable or not. For a 16-bit

Table 3. Density and Performance for Several Common Circuit Functions

		XC3000 (-125)		XC4000 (-5)	
16-bit Decoder From Input Pad		15 ns	4 CLBs	12 ns	0 CLBs
24-bit Accumulator		17 MHz	46 CLBs	32 MHz	13 CLBs
State Machine Benchmark*		18 MHz	34 CLBs	30 MHz	26 CLBs
16:1 Multiplexer		16 ns	8 CLBs	16 ns	5 CLBs
16-bit Unidirectional	Max Density	20 MHz	16 CLBs	40 MHz	8 CLBs
Loadable Counter	Max Speed	34 MHz	23 CLBs	42 MHz	9 CLBs
16-bit U/D Counter	Max Density	20 MHz	16 CLBs	40 MHz	8 CLBs
	Max Speed	30 MHz	27 CLBs	40 MHz	8 CLBs
16-bit Adder	Max Density	50 ns	30 CLBs	20.5 ns	9 CLBs
	Max Speed	30 ns	41 CLBs	20.5 ns	9 CLBs

* 16 states, 40 transitions, 10 inputs, 8 outputs

inputs could not be driven by all adjacent routing lines. In the XC4000 families, these constraints have been largely eliminated. This makes it easier for the software to complete the routing of complex interconnect patterns.

Chip architects and software designers worked closely together to achieve a solution that is not only inherently powerful, but also easy to utilize by the software-driven design tools for Partitioning, Placement and Routing. The goal was to provide automated push-button software tools that complete almost all designs, even large and dense ones, automatically, without operator assistance. But these tools will still give the designer the option to get involved in the partitioning, placement and, to a lesser extent, even the routing of critical parts of the design, if that is needed to optimize the performance.

On-Chip Memory

The XC4000, XC4000A and XC4000H family devices are the first programmable logic devices with RAM accessible to the user.

An optional mode for each CLB makes the memory look-up tables in the F' and G' function generators usable as either a 16 x 2 or 32 x 1 bit array of Read/Write memory cells (Figure 3). The F1-F4 and G1-G4 inputs to the function generators act as address lines, selecting a particular memory cell in each look-up table. The functionality of the CLB control signals change in this configuration; the H1, DIN, and S/R lines become the two data inputs and the Write Enable (WE) input for the 16 x 2 memory. When the 32 x 1 configuration is selected, D1 acts as the fifth address bit and D0 is the data input. The contents of the memory cell(s) being addressed are available at the F' and G' function-generator outputs, and can exit the CLB through its X and Y outputs, or can be pipelined using the CLB flip-flop(s).

Configuring the CLB function generators as Read/Write memory does not affect the functionality of the other portions of the CLB, with the exception of the redefinition of the control signals. 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.

The RAMs are very fast; read access is the same as logic delay, about 5.5 ns; write time is about 8 ns; both are several times faster than any off-chip solution. Such distributed RAM is a novel concept, creating new possibilities in system design: registered arrays of multiple accumulators, status registers, index registers, DMA counters, distributed shift registers, LIFO stacks, and FIFO buffers. The data path of a 16-byte FIFO uses four CLBs for storage, and six CLBs for address counting and multiplexing (Figure 4). With 32 storage locations per CLB, compared to two flip-flops per CLB, the cost of intelligent distributed memory has been reduced by a factor of 16.

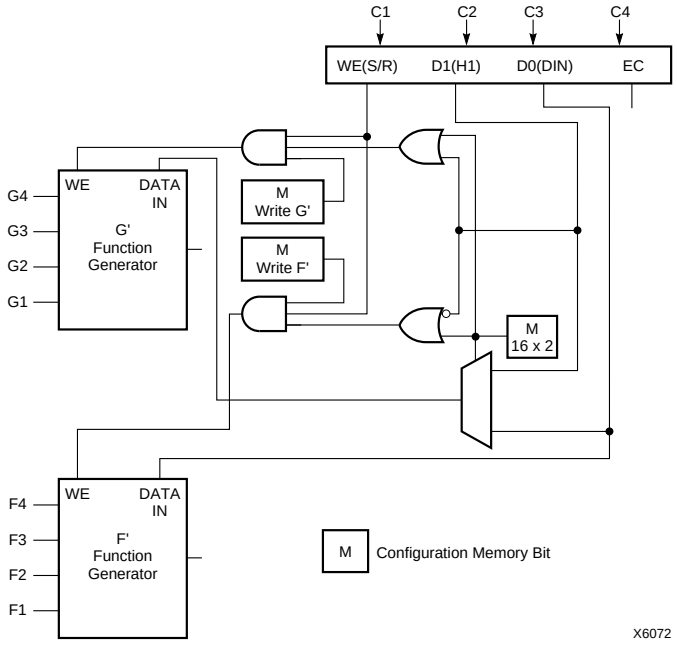


Figure 3. CLB Function Generators Can Be Used as Read/Write Memory Cells

Input/Output Blocks (IOBs), XC4000 and XC4000A Families (for XC4000H family, see page 2-82)

User-configurable IOBs provide the interface between external package pins and the internal logic (Figure 5). Each IOB controls one package pin and can be defined for input, output, or bidirectional signals.

Two paths, labeled I1 and I2, bring input signals into the array. Inputs are routed to an input register that can be programmed as either an edge-triggered flip-flop or a level-sensitive transparent latch. Optionally, the data input to the register can be delayed by several nanoseconds to compensate for the delay on the clock signal, that first must

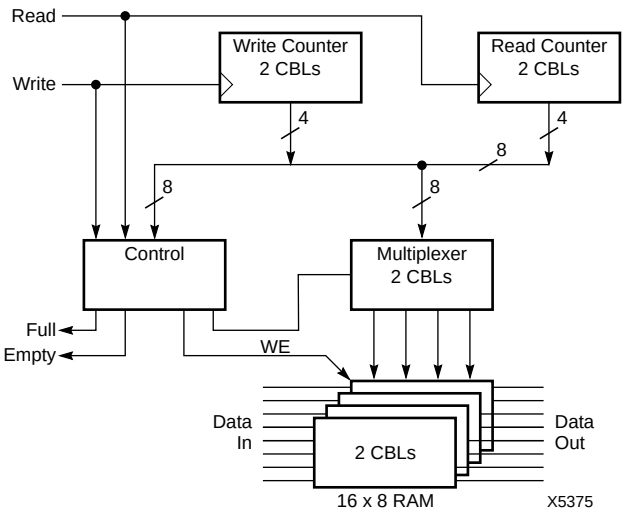


Figure 4. 16-byte FIFO

pass through a global buffer before arriving at the IOB. This eliminates the possibility of a data hold-time requirement at the external pin. The I1 and I2 signals that exit the block can each carry either the direct or registered input signal.

Output signals can be inverted or not inverted, and can pass directly to the pad or be stored in an edge-triggered flip-flop. Optionally, an output enable signal can be used to place the output buffer in a high-impedance state, implementing 3-state outputs or bidirectional I/O. Under configuration control, the output (OUT) and output enable (OE) signals can be inverted, and the slew rate of the output buffer can be reduced to minimize power bus transients when switching non-critical signals. Each XC4000-families output buffer is capable of sinking 12 mA; two adjacent output buffers can be wire-ANDed externally to sink up to 24 mA. In the XC4000A and XC4000H families, each output buffer can sink 24 mA.

There are a number of other programmable options in the IOB. Programmable pull-up and pull-down resistors are useful for tying unused pins to V_{CC} or ground to minimize power consumption. Separate clock signals are provided for the input and output registers; these clocks can be inverted, generating either falling-edge or rising-edge triggered flip-flops. As is the case with the CLB registers, a global set/reset signal can be used to set or clear the input and output registers whenever the RESET net is active.

Embedded logic attached to the IOBs contains test structures compatible with IEEE Standard 1149.1 for boundary-scan testing, permitting easy chip and board-level testing.

Programmable Interconnect

All internal connections are composed of metal segments with programmable switching points to implement the desired routing. An abundance of different routing resources is provided to achieve efficient automated routing. The number of routing channels is scaled to the size of the array; i.e., it increases with array size.

In previous generations of LCAs, the logic-block inputs were located on the top, left, and bottom of the block; outputs exited the block on the right, favoring left-to-right data flow through the device. For the third-generation family, the CLB inputs and outputs are distributed on all four sides of the block, providing additional routing flexibility (Figure 6). In general, the entire architecture is more symmetrical and regular than that of earlier generations, and is more suited to well-established placement and routing algorithms developed for conventional mask- programmed gate-array design.

There are three main types of interconnect, distinguished by the relative length of their segments: single-length lines, double-length lines, and Longlines. Note: The number of routing channels shown in Figures 6 and 9 are for illustration purposes only; the actual number of routing channels varies with array size. The routing scheme was designed for minimum resistance and capacitance of the average routing path, resulting in significant performance improvements.

The single-length lines are a grid of horizontal and vertical lines that intersect at a Switch Matrix between each block. Figure 6 illustrates the single-length interconnect lines

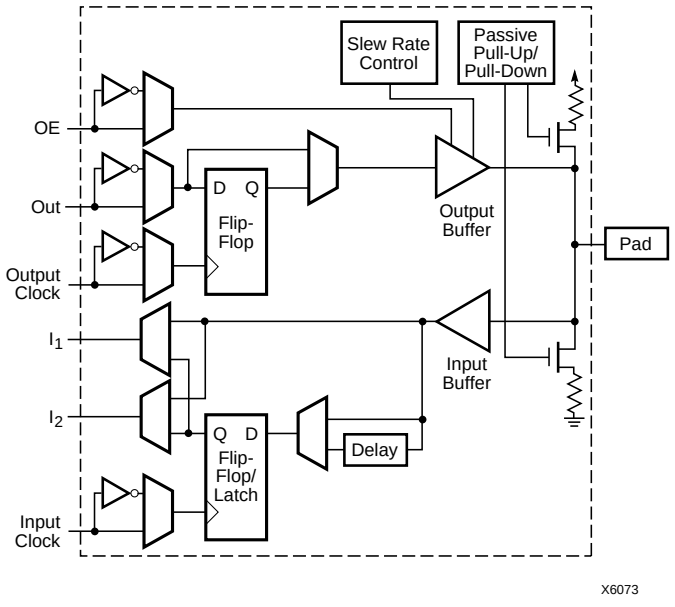


Figure 5. XC4000 and XC4000A Families Input/Output Block

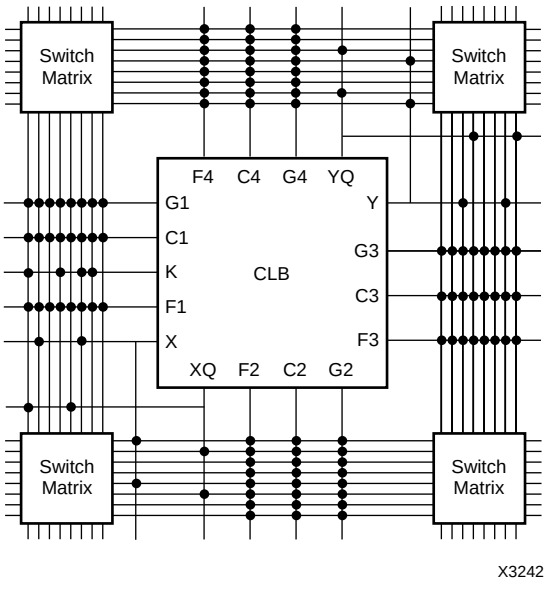


Figure 6. Typical CLB Connections to Adjacent Single-Length Lines

surrounding one CLB in the array. Each Switch Matrix consists of programmable n-channel pass transistors used to establish connections between the single-length lines (Figure 7). For example, a signal entering on the right side of the Switch Matrix can be routed to a single-length line on the top, left, or bottom sides, or any combination thereof, if multiple branches are required. Single-length lines are normally used to conduct signals within a localized area and to provide the branching for nets with fanout greater than one.

Compared to the previous generations of LCA architectures, the number of possible connections through the Switch Matrix has been reduced. This decreases capacitive loading and minimizes routing delays, thus increasing performance. However, a much more versatile set of connections between the single-length lines and the CLB inputs and outputs more than compensate for the reduction in Switch Matrix options, resulting in overall increased routability.

The function generator and control inputs to the CLB (F1-F4, G1-G4, and C1-C4) can be driven from any adjacent single-length line segment (Figure 6). The CLB clock (K) input can be driven from one-half of the adjacent single-length lines. Each CLB output can drive several of the single-length lines, with connections to both the horizontal and vertical Longlines.

The double-length lines (Figure 8) consist of a grid of metal segments twice as long as the single-length lines; i.e, a double-length line runs 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 CLB location in that row or column. As with single-length lines, all the CLB inputs except K can be driven from any adjacent double-length line, and each CLB output can drive nearby double-length lines in both the vertical and horizontal planes. Double-length lines provide the most efficient implementation of intermediate length, point-to-point interconnections.

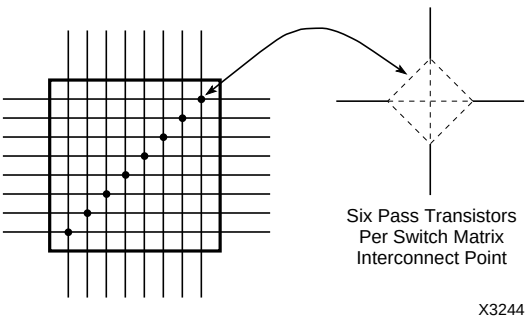


Figure 7. Switch Matrix

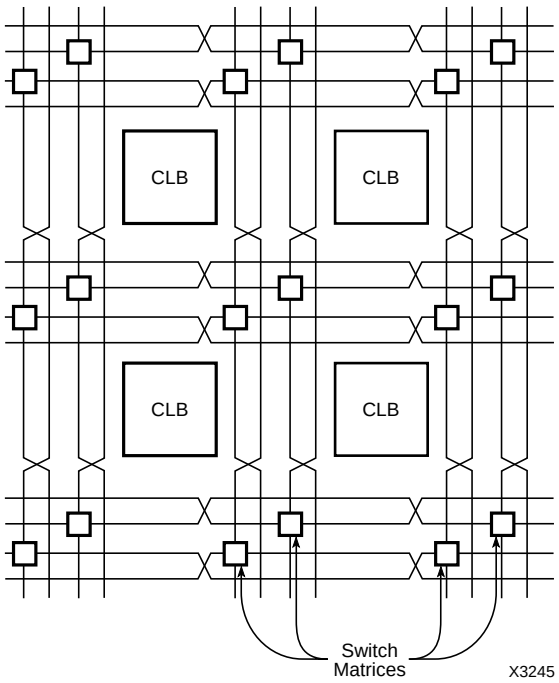


Figure 8. Double-Length Lines

Longlines form a grid of metal interconnect segments that run the entire length or width of the array (Figure 9). Additional vertical longlines can be driven by special global buffers, designed to distribute clocks and other high fanout control signals throughout the array with minimal skew. Longlines are intended for high fan-out, time-critical signal nets. Each Longline has a programmable splitter switch at its center, that can separate the line into two independent routing channels, each running half the width or height of the array. CLB inputs can be driven from a subset of the adjacent Longlines; CLB outputs are routed to the Longlines via 3-state buffers or the single-length interconnected lines.

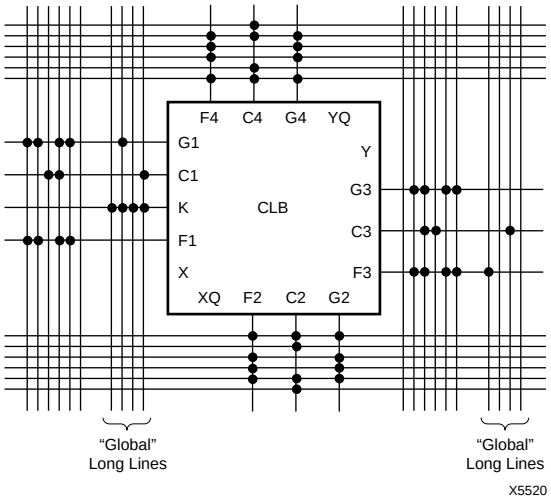


Figure 9. Longline Routing Resources with Typical CLB Connections

Communication between Longlines and single-length lines is controlled by programmable interconnect points at the line intersections. Double-length lines do not connect to other lines.

Three-State Buffers
A pair of 3-state buffers, associated with each CLB in the array, can be used to drive signals onto the nearest horizontal Longlines above and below the block. This feature is also available in the XC3000 generation of LCA devices. The 3-state buffer input can be driven from any X, Y, XQ, or YQ output of the neighboring CLB, or from nearby single-length lines; the buffer enable can come from nearby vertical single-length or Longlines. Another 3-state buffer with similar access is located near each I/O block along the right and left edges of the array. These buffers can be used to implement multiplexed or bidirectional buses on the horizontal Longlines. Programmable pull-up resistors attached to both ends of these Longlines help to implement a wide wired-AND function.

Special Longlines running along the perimeter of the array can be used to wire-AND signals coming from nearby IOBs or from internal Longlines.

Taking Advantage of Reconfiguration
LCA devices can be reconfigured to change logic function while resident in the system. This gives the system designer a new degree of freedom, not available with any other type of logic. Hardware can be changed as easily as software. Design updates or modifications are easy. An LCA device can even be reconfigured dynamically to perform different functions at different times. Reconfigurable logic can be used to implement system self diagnostics, create systems capable of being reconfigured for different environments or operations, or implement dual-purpose hardware for a given application. As an added benefit, use of reconfigurable LCA devices simplifies hardware design and debugging and shortens product time-to-market.

Development System

The powerful features of the XC4000 device families require an equally powerful, yet easy-to-use set of development tools. Xilinx provides an enhanced version of the Xilinx Automatic CAE Tools (XACT) optimized for the XC4000 families.

As with other logic technologies, the basic methodology for XC4000 FPGA design consists of three inter-related steps: entry, implementation, and verification. Popular 'generic' tools are used for entry and simulation (for example, Viewlogic System's ViewDraw schematic editor and ViewSim simulator), but architecture-specific tools are needed for implementation.

All Xilinx development system software is integrated under the Xilinx Design Manager (XDM), providing designers with a common user interface regardless of their choice of entry and verification tools. XDM simplifies the selection of command-line options with pull-down menus and on-line help text. Application programs ranging from schematic capture to Partitioning, Placement, and Routing (PPR) can be accessed from XDM, while the program-command sequence is generated and stored for documentation prior to execution. The XMAKE command, a design compilation utility, automates the entire implementation process, automatically retrieving the design's input files and performing all the steps needed to create configuration and report files.

Several advanced features of the XACT system facilitate XC4000 FPGA design. The MEMGEN utility, a memory compiler, implements on-chip RAM within an XC4000 FPGA. Relationally Placed Macros (RPMs) – schematic-based macros with relative locations constraints to guide their placement within the FPGA – help ensure an optimized implementation for common logic functions. XACT-Performance, a feature of the Partition, Place, and Route (PPR) implementation program, allows designers to enter their exact performance requirements during design entry, at the schematic level.

Design Entry
Designs can be entered graphically, using schematic-capture software, or in any of several text-based formats (such as Boolean equations, state-machine descriptions, and high-level design languages).

Xilinx and third-party CAE vendors have developed library and interface products compatible with a wide variety of design-entry and simulation environments. A standard interface-file specification, XNF (Xilinx Netlist File), is provided to simplify file transfers into and out of the XACT development system.

Xilinx offers XACT development system interfaces to the following design environments.

- Viewlogic Systems (ViewDraw, ViewSim)
- Mentor Graphics V7 and V8 (NETED, Quicksim, Design Architect, Quicksim II)
- OrCAD (SDT , VST)
- Synopsys (Design Compiler, FPGA Compiler)
- Xilinx-ABEL
- X-BLOX

Many other environments are supported by third-party vendors. Currently, more than 100 packages are supported.

The schematic library for the XC4000 FPGA reflects the wide variety of logic functions that can be implemented in these versatile devices. The library contains over 400 primitives and macros, ranging from 2-input AND gates to 16-bit accumulators, and including arithmetic functions,

The XACT system also includes XDelay, a static timing analyzer. XDelay examines a design's logic and timing to calculate the performance along signal paths, identify possible race conditions, and detect set-up and hold-time violations. Timing analyzers do not require that the user generate input stimulus patterns or test vectors.

Summary

The result of eight years of FPGA design experience and feedback from thousands of customers, the XC4000 families combine architectural versatility, on-chip RAM, increased speed and gate complexity with abundant routing resources and new, sophisticated software to achieve fully automated implementation of complex, high-performance designs.

7400 Equivalents

	# of CLBs
'138	5
'139	2
'147	5
'148	6
'150	5
'151	3
'152	3
'153	2
'154	16
'157	2
'158	2
'160	5
'161	6
'162	8
'163	8
'164	4
'165s	9
'166	5
'168	7
'174	3
'194	5
'195	3
'280	3
'283	8
'298	2
'352	2
'390	3
'518	3
'521	3

Barrel Shifters

brlshft4	4
brlshft8	13

4-Bit Counters

cd4ce	3
cd4cle	5
cd4rle	6
cb4ce	3
cb4cle	6
cb4re	5

8- and 16-Bit Counters

cb8ce	6
cb8re	10
cc16ce	10
cc16cle	11
cc16cled	21

Identity Comparators

comp4	1
comp8	2
comp16	5

Magnitude Comparators

compm4	4
compm8	9
compm16	20

Decoders

d2-4e	2
d3-8e	4
d4-16e	16

Multiplexers

m2-1e	1
m4-1e	1
m8-1e	3
m16-1e	5

Registers

rd4r	2
rd8r	4
rd16r	8

Shift Registers

sr8ce	4
sr16re	8

RAMs

ram 16x4	2

Explanation of counter nomenclature

cb = binary counter
cd = BCD counter
cc = cascadable binary counter
d = bidirectional
l = loadable
x = cascadable
e = clock enable
r = synchronous reset
c = asynchronous clear

Figure 10. CLB Count of Selected XC4000 Soft Macros

Detailed Functional Description

XC4000 and XC4000A Input/Output Blocks

(For XC4000H family, see page 2-82)

The IOB forms the interface between the internal logic and the I/O pads of the LCA device. Under configuration control, the output buffer receives either the logic signal (.out) routed from the internal logic to the IOB, or the complement of this signal, or this same data after it has been clocked into the output flip-flop.

As a configuration option, each flip-flop (CLB or IOB) is initialized as either set or reset, and is also forced into this programmable initialization state whenever the global Set/Reset net is activated after configuration has been completed. The clock polarity of each IOB flip-flop can be configured individually, as can the polarity of the 3-state control for the output buffer.

Each output buffer can be configured to be either fast or slew-rate limited, which reduces noise generation and ground bounce. Each I/O pin can be configured with either an internal pull-up or pull down resistor, or with no internal resistor. Independent of this choice, each IOB has a pull-up resistor during the configuration process.

The 3-state output driver uses a totem pole n-channel output structure. V_{OH} is one n-channel threshold lower than V_{CC}, which makes rise and fall delays more symmetrical.

Family	Per IOB Source	Per IOB Sink	Per IOB Pair Sink	# Slew Modes
XC4000	4	12	24	2
XC4000A	4	24	48	4
XC4000H	4	24*	48	2

*XC4000H devices can sink only 4 mA configured for SoftEdge mode

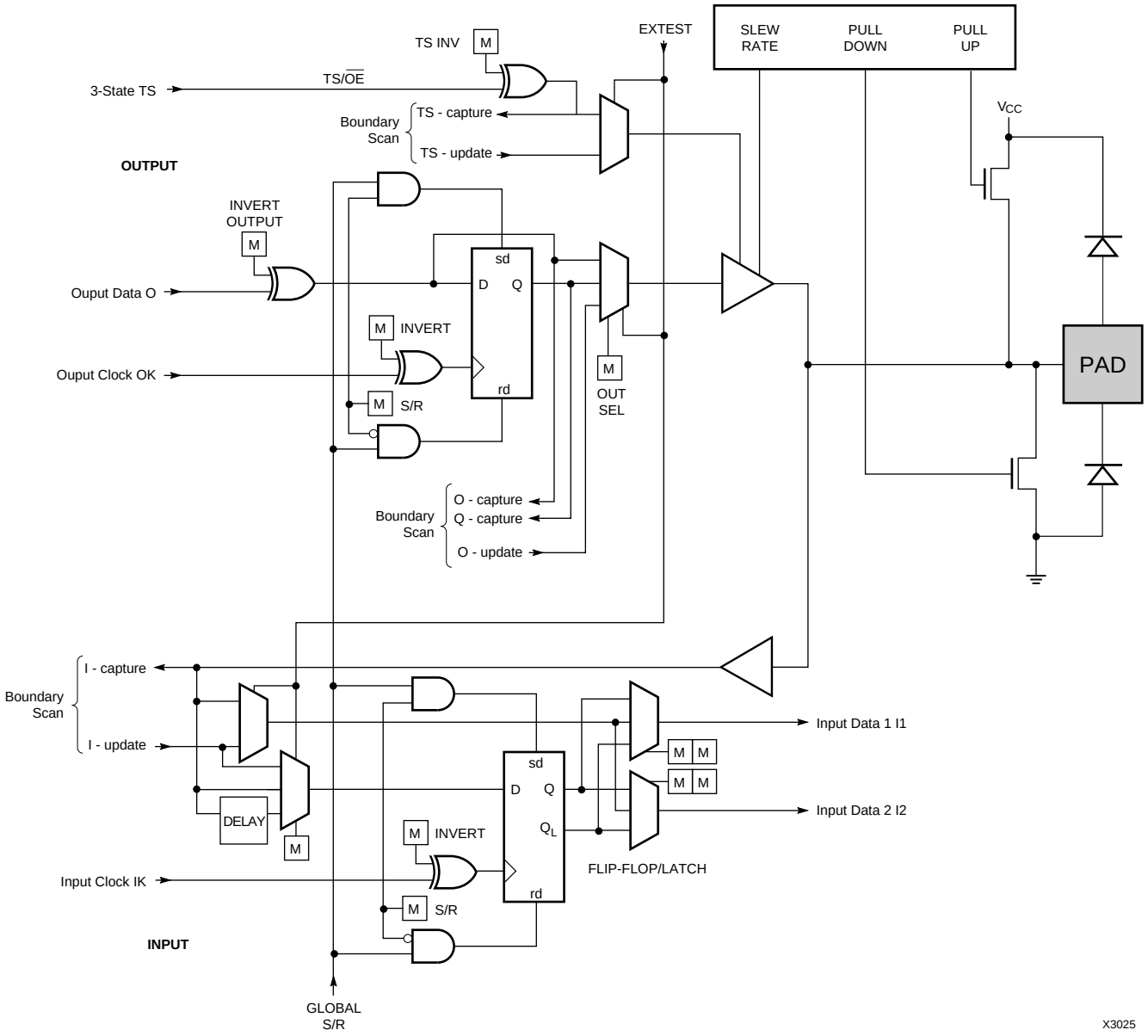


Figure 11. XC4000 and XC4000A I/O Block

The inputs drive TTL-compatible buffers with 1.2-V input threshold and a slight hysteresis of about 300 mV. These buffers drive the internal logic as well as the D-input of the input flip-flop.

Under configuration control, the set-up time of this flip-flop can be increased so that normal clock routing does not result in a hold-time problem. Note that the input flip-flop set-up time is defined between the data measured at the device I/O pin and the clock input at the IOB. Any clock routing delay must, therefore, be subtracted from this set-up time to arrive at the real set-up time requirement on the device pins. A short specified set-up time might, therefore, result in a negative set-up time at the device pins, i.e. a hold-time requirement, which is usually undesirable. The default long set-up time can tolerate more clock delay without causing a hold-time requirement. For faster input register setup time, with non-zero hold, attach a "NODELAY" property to the flip-flop. The exact method to accomplish this depends on the design entry tool.

The input block has two connections to the internal logic, I1 and I2. Each of these is driven either by the incoming data, by the master or by the slave of the input flip-flop.

Wide Decoders

The periphery of the chip has four wide decoder circuits at each edge (two in the XC4000A). The inputs to each decoder are any of the I1 signals on that edge plus one local interconnect per CLB row or column. Each decoder generates High output (resistor pull-up) when the AND condition of the selected inputs, or their complements, is true. This is analogous to the AND term in typical PAL devices. Each decoder can be split at its center.

The decoder outputs can drive CLB inputs so they can be combined with other logic, or 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.

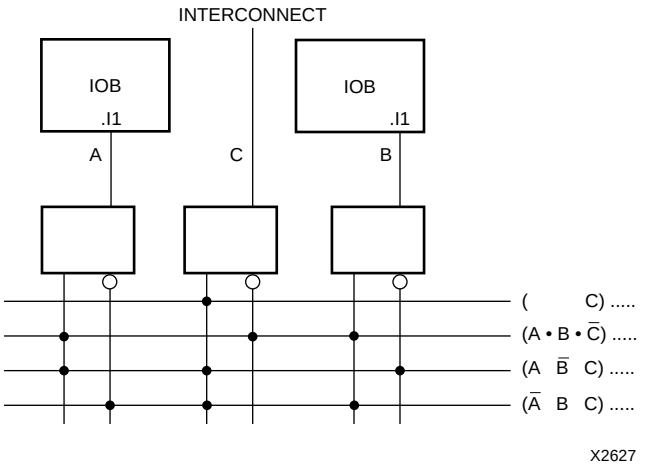


Figure 12. **Example of Edge Decoding.** Each row or column of CLBs provide up to three variables (or their complements)

Configurable Logic Blocks

Configurable Logic Blocks implement most of the logic in an LCA device. Two 4-input function generators (F and G) offer unrestricted versatility. A third function generator (H) can combine the outputs of F and G with a ninth input variable, thus implementing certain functions of up to nine variables, like parity check or expandable-identity comparison of two sets of four inputs.

The four control inputs C1 through C4 can each generate any one of four logic signals, used in the CLB.

- Enable Clock, Asynchronous Preset/Reset, DIN, and H1, when the memory function is disabled, or
- Enable Clock, Write Enable, D0, and D1, when the memory function is enabled.

Since the function-generator outputs are brought out independently of the flip-flop outputs, and DIN and H1 can be used as direct inputs to the two flip-flops, the two combinatorial and the two sequential functions in the CLB can be used independently. This versatility increases logic density and simplifies routing.

The asynchronous flip-flop input can be configured as either set or reset. This configuration option also determines the state in which the flip-flops become operational after configuration, as well as the effect of an externally or internally applied Set/Reset during normal operation.

Fast Carry Logic

The CLBs can generate the arithmetic-carry output for incoming operands, and can pass this extra output on to the next CLB function generator above or below. This connection is independent of normal routing resources and it is, presently, only supported by Hard Macros. A later software release will accommodate Soft Macros and will permit graphic editing of the fast logic circuitry. This fast carry logic is one of the most significant improvements in the XC4000 families, speeding up arithmetic and counting into the 60-MHz range.

Using Function Generators as RAMs

Using XC4000 devices, the designer can write into the latches that hold the configuration content of the function generators. Each function generator can thus be used as a small Read/Write memory, or RAM. The function generators in any CLB can be configured in three ways.

- Two 16 x 1 RAMs with two data inputs and two data outputs – identical or, if preferred, different addressing for each RAM
- One 32 x 1 RAM with one data input and one data output
- One 16 x 1 RAM plus one 5-input function generator

Interconnects

The XC4000 families use a hierarchy of interconnect resources.

- General purpose single-length and double-length lines offer fast routing between adjacent blocks, and highest flexibility for complex routes, but they incur a delay every time they pass through a switch matrix.
- Longlines run the width or height of the chip with negligible delay variations. They are used for signal distribution over long distances. Some Horizontal Longlines can be driven by 3-state or open-drain drivers, and can thus implement bidirectional buses or wired-AND decoding.
- Global Nets are optimized for the distribution of clock and time-critical or high-fan-out control signal. Four pad-driven Primary Global Nets offer shortest delay and negligible skew. Four pad-driven Secondary Global Nets have slightly longer delay and more skew due to heavier loading.

Each CLB column has four dedicated Vertical Longlines, each of these lines has access to a particular Primary Global Net, or to any one of the Secondary Global Nets. The Global Nets avoid clock skew and potential hold-time problems. The user must specify these Global Nets for all timing-sensitive global signal distribution.

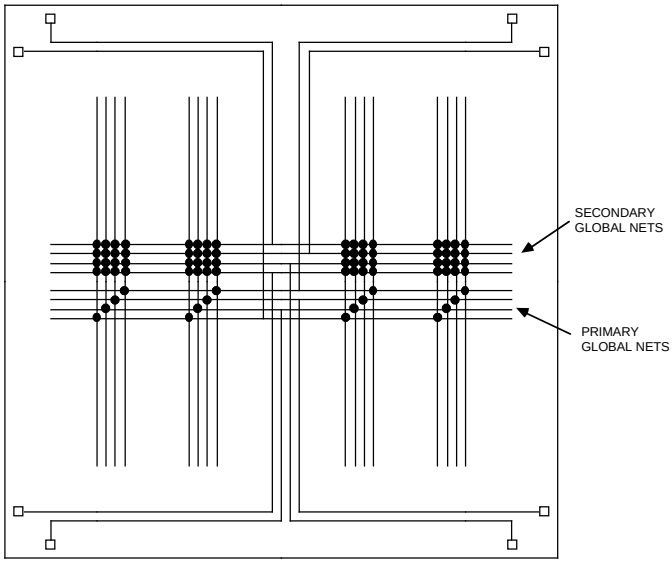
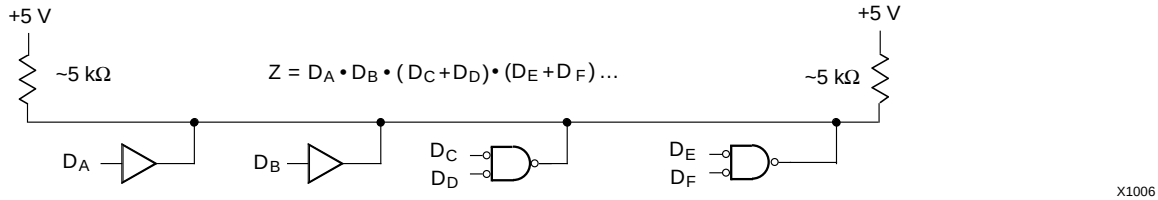
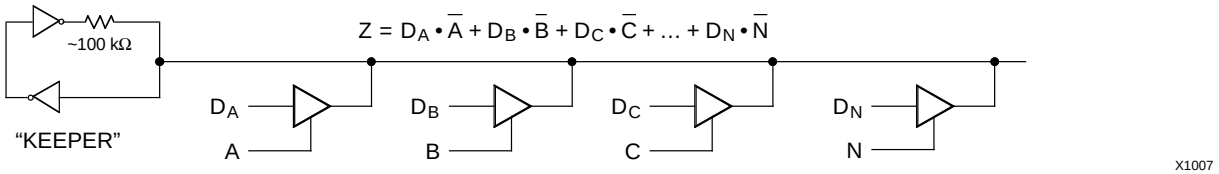


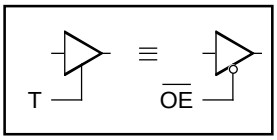
Figure 17. XC4000 Global Net Distribution. Four Lines per Column; Eight Inputs in the Four Chip Corners.



Open Drain Buffers Implement a Wired-AND Function. When all the buffer inputs are High the pull-up resistor(s) provide the High output.

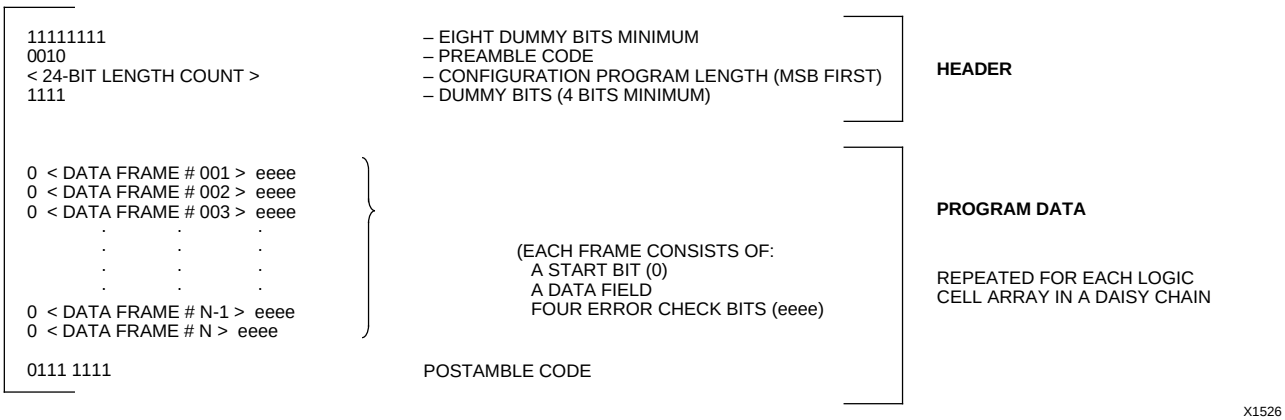


3-State Buffers Implement a Multiplexer. The selection is accomplished by the buffer 3-state signal.



Active High T is Identical to Active Low Output Enable.

Figure 18. TBUFs Driving Horizontal Longlines.



Device	XC4002A	XC4003A	XC4003/H	XC4004A	XC4005A	XC4005/H	XC4006	XC4008	XC4010/D	XC4013/D	XC4020	XC4025
Gates	2,000	3,000	3,000	4,000	5000	5,000	6,000	8,000	10,000	13,000	20,000	25,000
CLBs (Row x Col)	64 (8 x 8)	100 (10 x 10)	100 (10 x 10)	144 (12 x 12)	196 (14 x 14)	196 (14 x 14)	256 (16 x 16)	324 (18 x 18)	400 (20 x 20)	576 (24 x 24)	784 (28 x 28)	1,024 (32 x 32)
IOBs	64	80	80/.160	96	112	112 (192)	128	144	160	192	224	256
Flip-flops	256	360	360/300	480	616	616 (392)	768	936	1,120	1,536	2,016	2,560
Horizontal TBUF Longlines	16	20	20	24	28	28	32	36	40	48	56	64
TBUFs/Longline	10	12	12	14	16	16	18	20	22	26	30	34
Bits per Frame	102	122	126	142	162	166	186	206	226	266	306	346
Frames	310	374	428	438	502	572	644	716	788	932	1,076	1,220
Program Data	31,628	45,636	53,936	62,204	81,332	94,960	119,792	147,504	178,096	247,920	329,264	422,128
PROM size (bits)	31,668	45,676	53,976	62,244	81,372	95,000	119,832	147,544	178,136	247,960	329,304	422,168

XC4000, 4000H: 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
XC4000A: Bits per Frame = (10 x number of Rows) + 6 for the top + 10 for the bottom + 1 + 1 start bit + 4 error check bits
Number of Frames = (32 x number of Columns) + 21 for the left edge + 32 for the right edge + 1
Program Data = (Bits per Frame x Number of Frames) + 8 postamble bits
PROM Size = Program Data + 40
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, but the Length Count value **must** be adjusted for all such extra "one" bits, even for leading extra ones at the beginning of the header.

Figure 19. Internal Configuration Data Structure.

Format

The configuration-data stream begins with a string of ones, a 0010 preamble code, a 24-bit length count, and a four-bit separator field of ones. This is followed by the actual configuration data in frames, each starting with a zero bit and ending with a four-bit error check. For each XC4XXX device, the MakeBits software allows a selection of CRC or non-CRC error checking. The non-CRC error checking tests for a 0110 end of frame field for each frame of a selected LCA device. For CRC error checking, MakeBits software calculates a running CRC of inserts a unique four-bit partial check at the end of each frame. The 11-bit CRC check of the last frame of an LCA device includes the last seven data bits. Detection of an error results in suspension of data loading and the pulling down of the INIT pin. In master modes, CCLK and address signals continue to operate externally. The user must detect INIT and initialize a new configuration by pulsing the PROGRAM pin or cycling V_{CC}. The length and number of frames depend on the device type. Multiple LCA devices can be connected in a daisy chain by wiring their CCLK pins in parallel and connecting the DOUT of each to the DIN of the next. The lead-master LCA device and following slaves each passes resynchronized configuration data coming from a single source. The Header data, including the length count, is passed through and is captured by each LCA

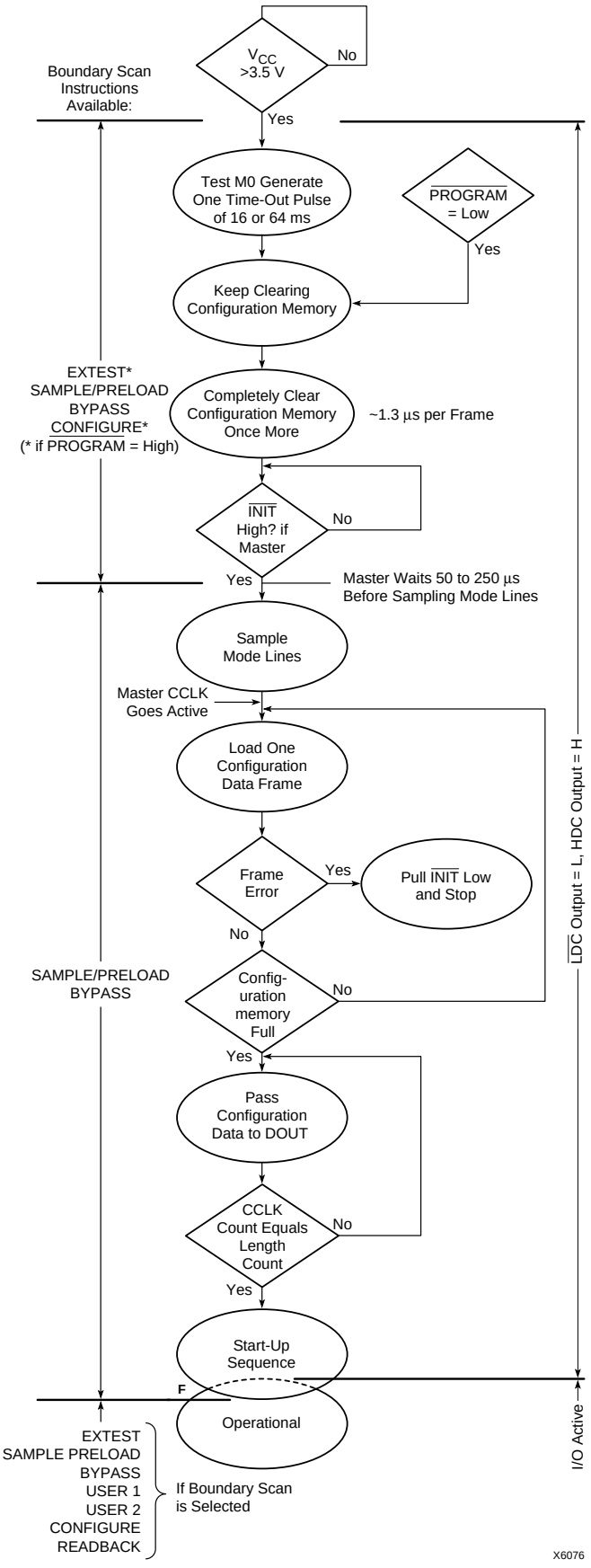


Figure 20. Start-up Sequence

device when it recognizes the 0010 preamble. Following the length-count data, any LCA device outputs a High on DOUT until it has received its required number of data frames.

After an LCA device 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 LCA device(s) begin the start-up sequence and become operational together.

Configuration Sequence

Configuration Memory Clear

When power is first applied or reapplied to an LCA device, an internal circuit forces initialization of the configuration logic. When V_{CC} reaches an operational level, and the circuit passes the write and read test of a sample pair of configuration bits, a nominal 16-ms time delay is started (four times longer when M0 is Low, i.e., in Master mode). During this time delay, or as long as the PROGRAM input is asserted, the configuration logic is held in a Configuration Memory Clear state. The configuration-memory frames are consecutively initialized, using the internal oscillator. At the end of each complete pass through the frame addressing, the power-on time-out delay circuitry and the level of the PROGRAM pin are tested. If neither is asserted, the logic initiates one additional clearing of the configuration frames and then tests the INIT input.

Initialization

During initialization and configuration, user pins HDC, LDC and INIT provide status outputs for system interface. The outputs, LDC, INIT and DONE are held Low and HDC is held High starting at the initial application of power. The open drain INIT pin is released after the final initialization pass through the frame addresses. There is a deliberate delay of 50 to 250 µs before a Master-mode device recognizes an inactive INIT. Two internal clocks after the INIT pin is recognized as High, the LCA device samples the three mode lines to determine the configuration mode. The appropriate interface lines become active and the configuration preamble and data can be loaded.

Configuration

The 0010 preamble code indicates that the following 24 bits represent the length count, i.e., the total number of configuration clocks needed to load the total configuration data. After the preamble and the length count have been passed through to all devices in the daisy chain, DOUT is held High to prevent frame start bits from reaching any daisy-chained devices. A specific configuration bit, early in the first frame of a master device, controls the configuration-clock rate and can increase it by a factor of eight. Each frame has a Low start bit followed by the frame-configura-

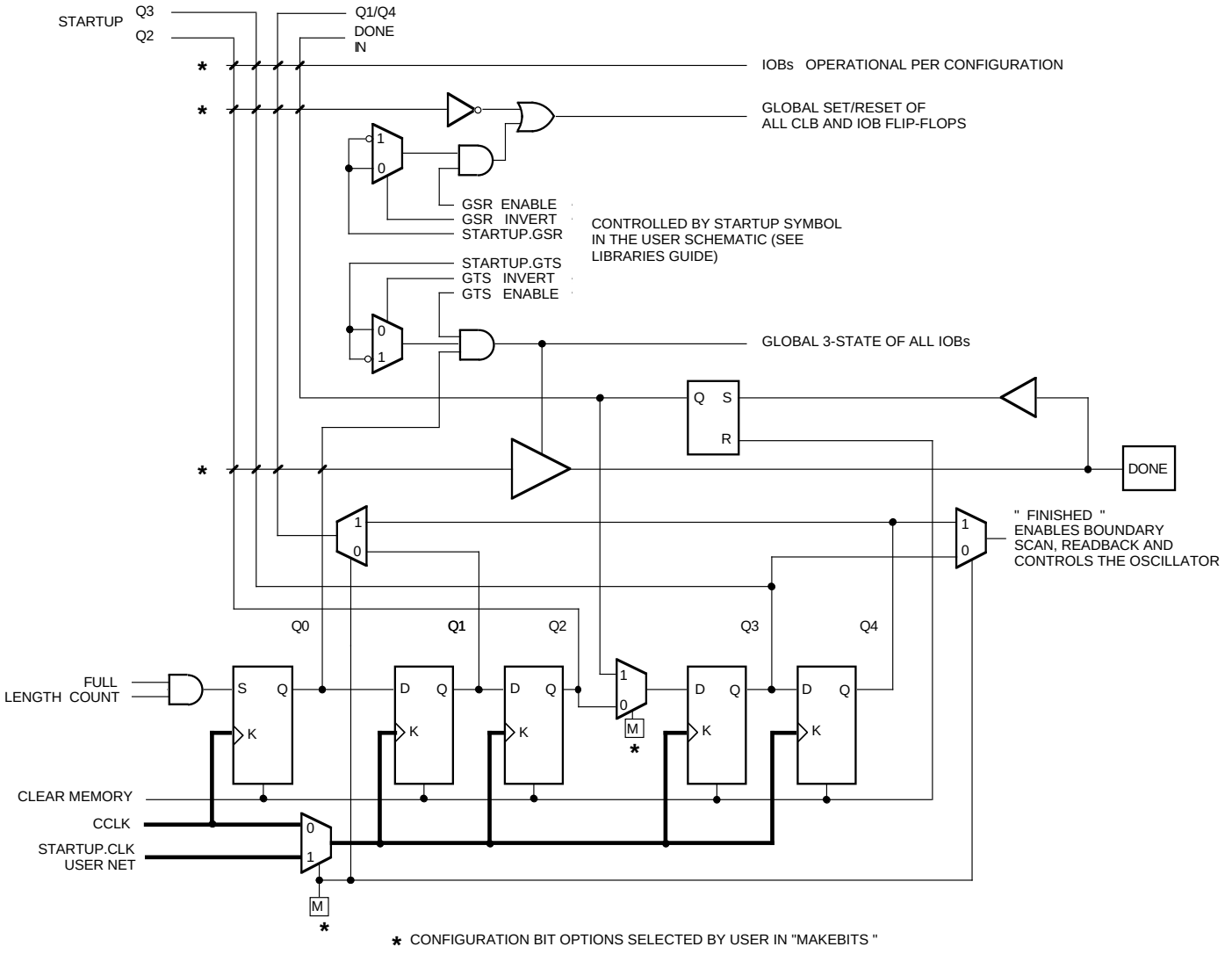


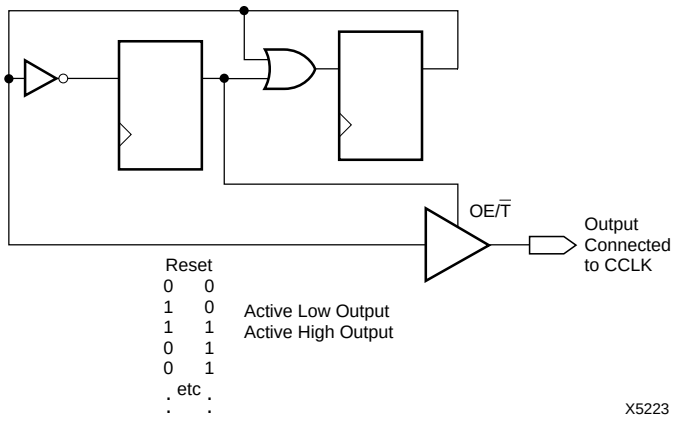
Figure 22. Start-up Logic

All Xilinx FPGAs of the XC2000, XC3000, XC4000 familiies 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 devices, the master cannot be an XC2000 or XC3000 device; if the daisy-chain contains XC3000 devices, the master cannot be an XC2000 device. The reason for this rule is shown in Figure 21 on the previous page. 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 21. The master device will then drive 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. The user has some control over the relative timing of these events and can, therefore, make sure that they occur early enough.

But, for XC4000, not reaching F means that READBACK cannot be initiated and most Boundary Scan instructions cannot be used.This limitation has been critized by designers who want to use an inexpensive lead device in peripheral mode and have the more precious I/O pins of the XC4000 devices all available for user I/O. Here is a solution for that case.

One CLB and one IOB in the lead XC3000 device are used to generate the additional CCLK pulse required by the XC4000 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. This solution requires one CLB, one IOB and pin, and an internal oscillator with a frequency of up to 5 MHz as available clock source. Obviously, this XC3000 master device must be configured with late Internal Reset, which happens to be the default option.

Using Global Set/Reset and Global 3-State Nets
The global Set/Reset (STARTUP.GSR) net can be driven by the user at any time to re-initialize all CLBs and IOBs to the same state they had at the end of configuration. For CLBs that is the same state as the one driven by the individually programmable asynchronous Set/Reset inputs. The global 3-state net (STARTUP.GTS), whenever activated after configuration is completed, forces all LCA outputs to the high-impedance state, unless Boundary Scan is enabled and is executing an EXTEST instruction.

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

XC4000 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.

After Readback has been initiated by a Low-to-High transition on RDBK.TRIG, 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.

Note that, in the XC4000 families, data is not inverted with respect to configuration the way it is in XC2000 and XC3000 families.

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 RIP returns Low.

Readback options are: Read Capture, Read Abort, and Clock Select.

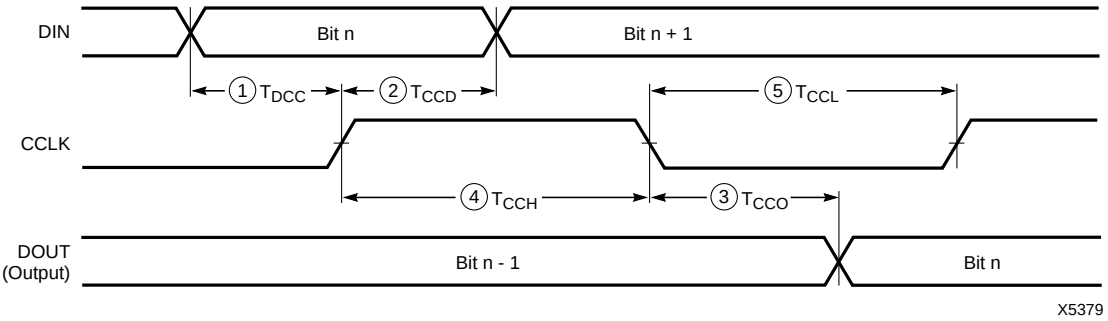
Read Capture
When the Readback Capture option is selected, the readback data stream includes sampled values of CLB and IOB signals imbedded in the data stream. The rising edge of RDBK.TRIG located in the lower-left chip corner, captures, in latches, the inverted values of the four CLB outputs and the IOB output flip-flops and the input signals I1, I2 . When the 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.

Read Abort
When the Readback 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).

Clock Select
Readback control and data are clocked on rising edges of RDBK.CLK located in the lower right chip corner. CCLK is an optional clock. If Readback must be inhibited for security reasons, the readback control nets are simply not connected.

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

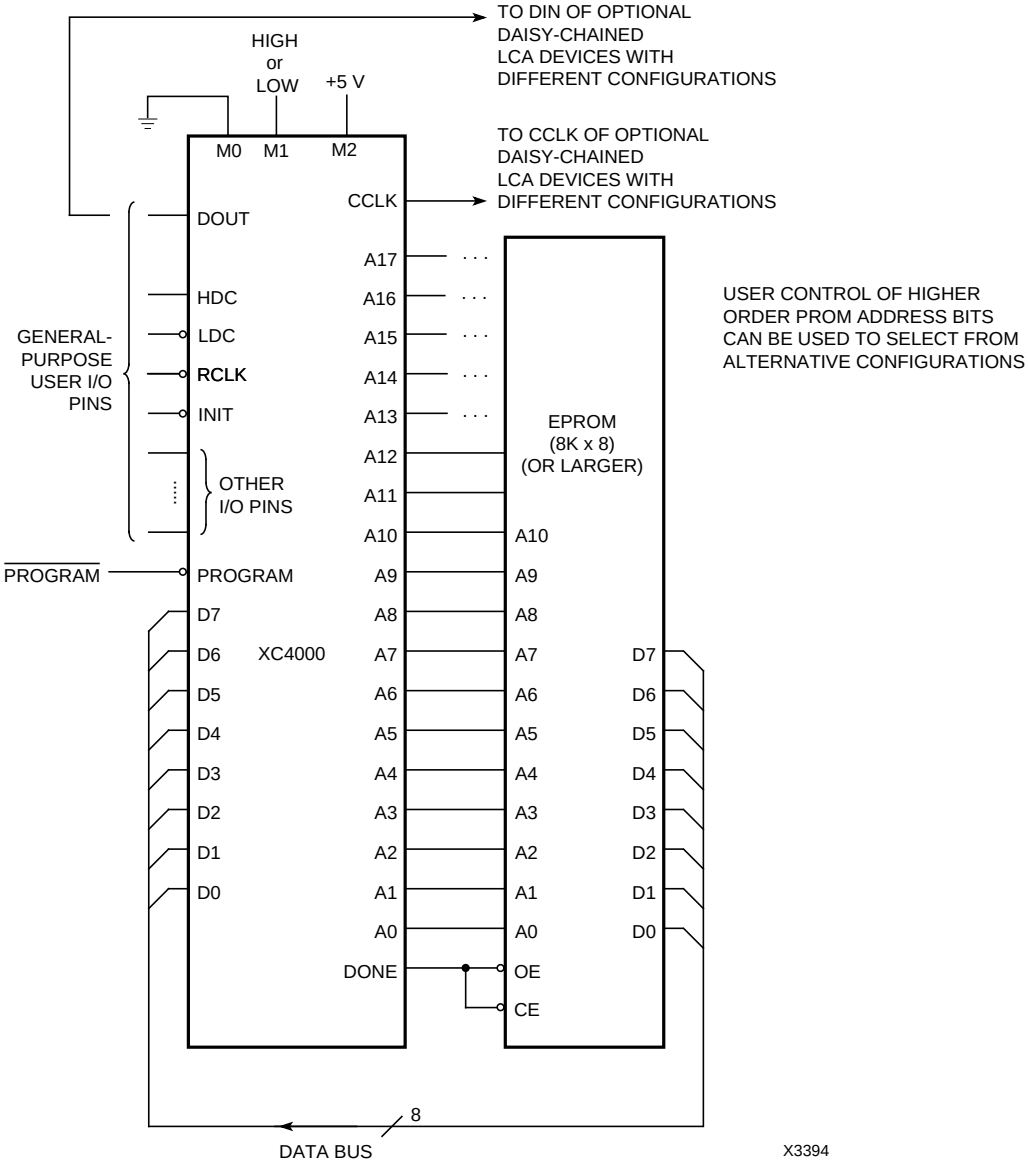
Slave Serial Mode Programming Switching Characteristics



	Description	Symbol		Min	Max	Units
CCLK	DIN setup	1	T_{DCC}	20		ns
	DIN hold	2	T_{CCD}	0		ns
	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 INIT of all daisy-chained LCA devices is High.

Master Parallel Mode



In Master Parallel mode, the lead LCA device directly addresses an industry-standard byte-wide EPROM, and accepts eight data bits right before incrementing (or decrementing) the address outputs.

The eight data bits are serialized in the lead LCA device, which then presents the preamble data (and all data that overflows the lead device) on the DOUT pin. There is an internal delay of 1.5 CCLK periods, after the rising CCLK edge that accepts a byte of data (and also changes the EPROM address) until the falling CCLK edge that makes the LSB (D0) of this byte appear at DOUT. This means that DOUT changes on the falling CCLK edge, and the next LCA device in the daisy-chain accepts data on the subsequent rising CCLK edge.

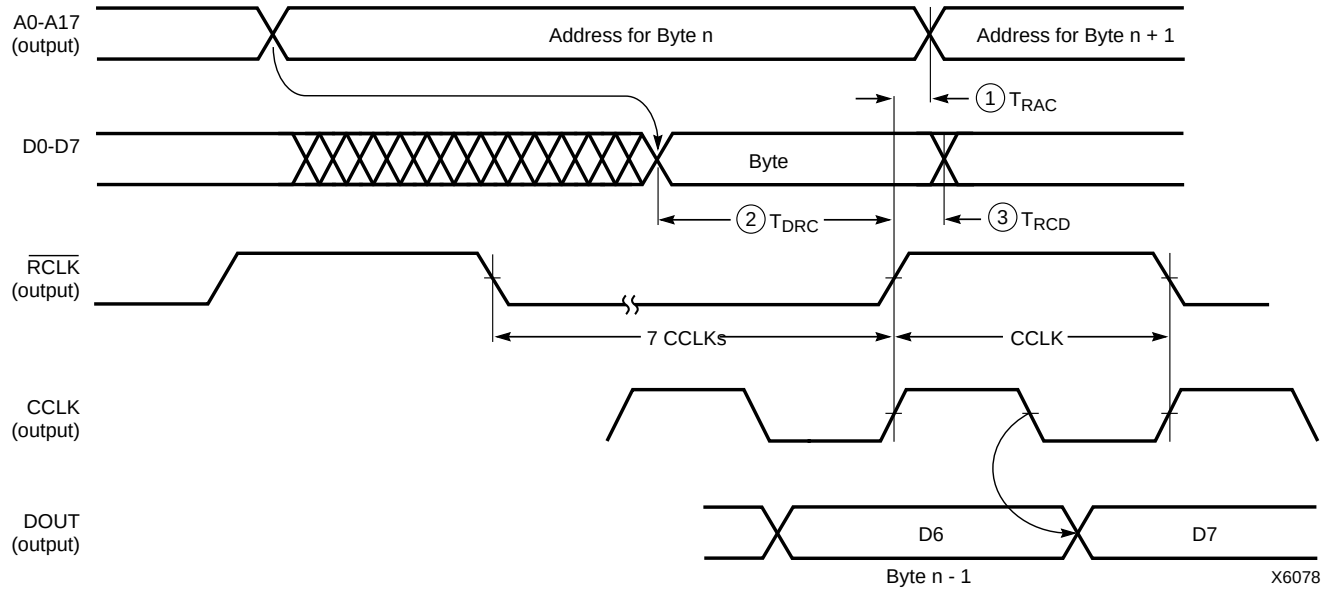
How to Delay Configuration After Power-Up

There are two methods to delay configuration after power-up: Put a logic Low on the PROGRAM input, or pull the bidirectional INIT pin Low, using an open-collector (open-drain) driver. (See also Figure 20 on page 2-27).

A Low on the PROGRAM input is the more radical approach, and is recommended when the power-supply rise time is excessive or poorly defined. As long as PROGRAM is Low, the XC4000 device keeps clearing its configuration memory. When PROGRAM goes High, the configuration memory is cleared one more time, followed by the beginning of configuration, provided the INIT input is not externally held Low. Note that a Low on the PROGRAM input automatically forces a Low on the INIT output.

Using an open-collector or open-drain driver to hold INIT Low before the beginning of configuration, causes the LCA device to wait after having completed the configuration memory clear operation. When INIT is no longer held Low externally, the device determines its configuration mode by capturing its status inputs, and is ready to start the configuration process. A master device waits an additional max 250 µs to make sure that all slaves in the potential daisy-chain have seen INIT being High.

Master Parallel Mode Programming Switching Characteristics

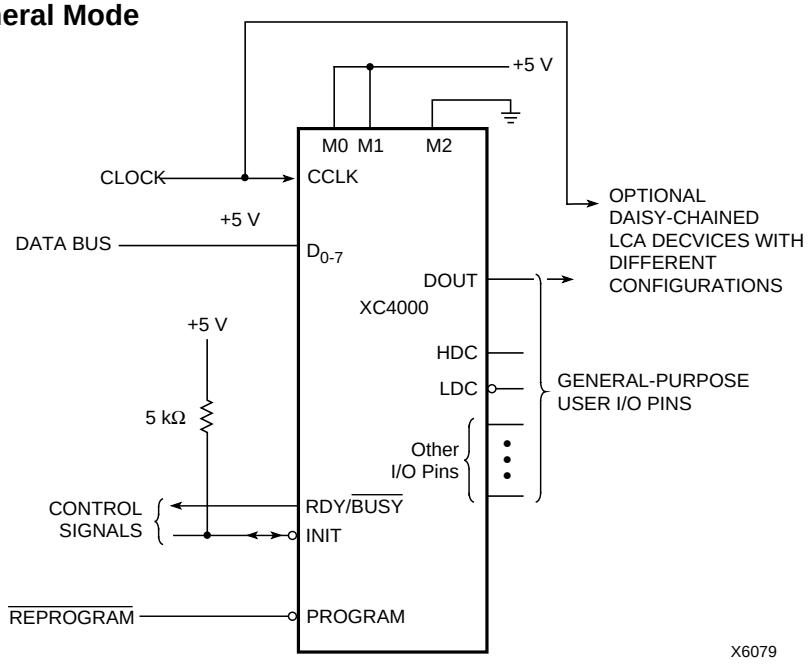


	Description	Symbol		Min	Max	Units
RCLK	Delay to Address valid	1	T_{RAC}	0	200	ns
	Data setup time	2	T_{DRC}	60		ns
	Data hold time	3	T_{RCD}	0		ns

Notes: 1. At power-up, V_{CC} must rise from 2.0 V to Vcc min in less than 25 ms, otherwise delay configuration using PROGRAM until V_{CC} is valid.
2. Configuration can be delayed by holding INIT Low with or until after the INIT of all daisy-chain slave mode devices is High.
3. The first Data byte is loaded and CCLK starts at the end of the first RCLK active cycle (rising edge).

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

Synchronous Peripheral Mode



Synchronous Peripheral mode can also be considered Slave Parallel mode. An external signal drives the CCLK input(s) of the LCA device(s). The first byte of parallel configuration data must be available at the D inputs of the lead LCA device a short set-up time before the rising CCLK edge. Subsequent data bytes are clocked in on every eighth consecutive rising CCLK edge. The same CCLK edge that accepts data, also causes the RDY/BUSY output to go High for one CCLK period. The pin name is a misnomer. In Synchronous Peripheral mode it is really an ACKNOWLEDGE signal. Synchronous operation does not require this response, but it is a meaningful signal for test purposes.

The lead LCA device serializes the data and presents the preamble data (and all data that overflows the lead device) on its DOUT pin. There is an internal delay of 1.5 CCLK periods, which means that DOUT changes on the falling CCLK edge, and the next LCA device in the daisy-chain accepts data on the subsequent rising CCLK edge. In order to complete the serial shift operation, 10 additional CCLK rising edges are required after the last data byte has been loaded, plus one more CCLK cycle for each daisy-chained device.

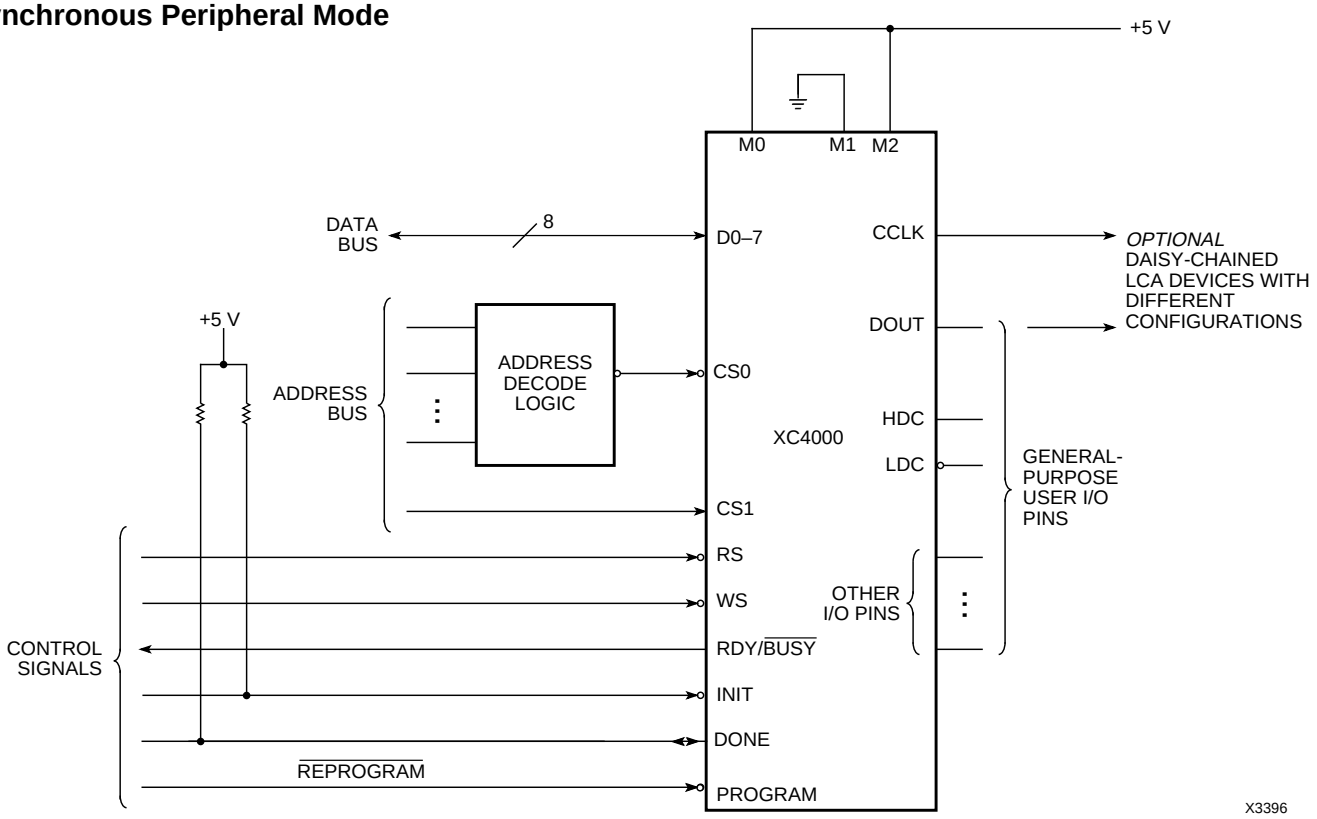
How to Delay Configuration After Power-Up

There are two methods to delay configuration after power-up: Put a logic Low on the PROGRAM input, or pull the bidirectional INIT pin Low, using an open-collector (open-drain) driver. (See also Figure 20 on page 2-27).

A Low on the PROGRAM input is the more radical approach, and is recommended when the power-supply rise time is excessive or poorly defined. As long as PROGRAM is Low, the XC4000 device keeps clearing its configuration memory. When PROGRAM goes High, the configuration memory is cleared one more time, followed by the beginning of configuration, provided the INIT input is not externally held Low. Note that a Low on the PROGRAM input automatically forces a Low on the INIT output.

Using an open-collector or open-drain driver to hold INIT Low before the beginning of configuration, causes the LCA device to wait after having completed the configuration memory clear operation. When INIT is no longer held Low externally, the device determines its configuration mode by capturing its status inputs, and is ready to start the configuration process. A master device waits an additional max 250 µs to make sure that all slaves in the potential daisy-chain have seen INIT being High.

Asynchronous Peripheral Mode



Write to LCA

Asynchronous Peripheral mode uses the trailing edge of the logic AND condition of the CS0, CS1 and WS inputs to accept byte-wide data from a microprocessor bus. In the lead LCA device, this data is loaded into a double-buffered UART-like parallel-to-serial converter and is serially shifted into the internal logic. The lead LCA device presents the preamble data (and all data that overflows the lead device) on the DOUT pin.

The RDY/BUSY output from the lead LCA device acts as a handshake signal to the microprocessor. RDY/BUSY goes Low when a byte has been received, and goes High again when the byte-wide input buffer has transferred its information into the shift register, and the buffer is ready to receive new data. The length of the BUSY signal depends on the activity in the UART. If the shift register had been empty when the new byte was received, the BUSY signal lasts for only two CCLK periods. If the shift register was still full when the new byte was received, the BUSY signal can be as long as nine CCLK periods.

Note that after the last byte has been entered, only seven of its bits are shifted out. CCLK remains High with DOUT equal to bit 6 (the next-to-last bit) of the last byte entered. The READY/BUSY handshake can be ignored if the delay from any one Write to the end of the next Write is guaranteed to be longer than 10 CCLK periods,i.e. longer than 20 µs.

Status Read

The logic AND condition of the CS0, CS1and RS inputs puts the device status on the Data bus.

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

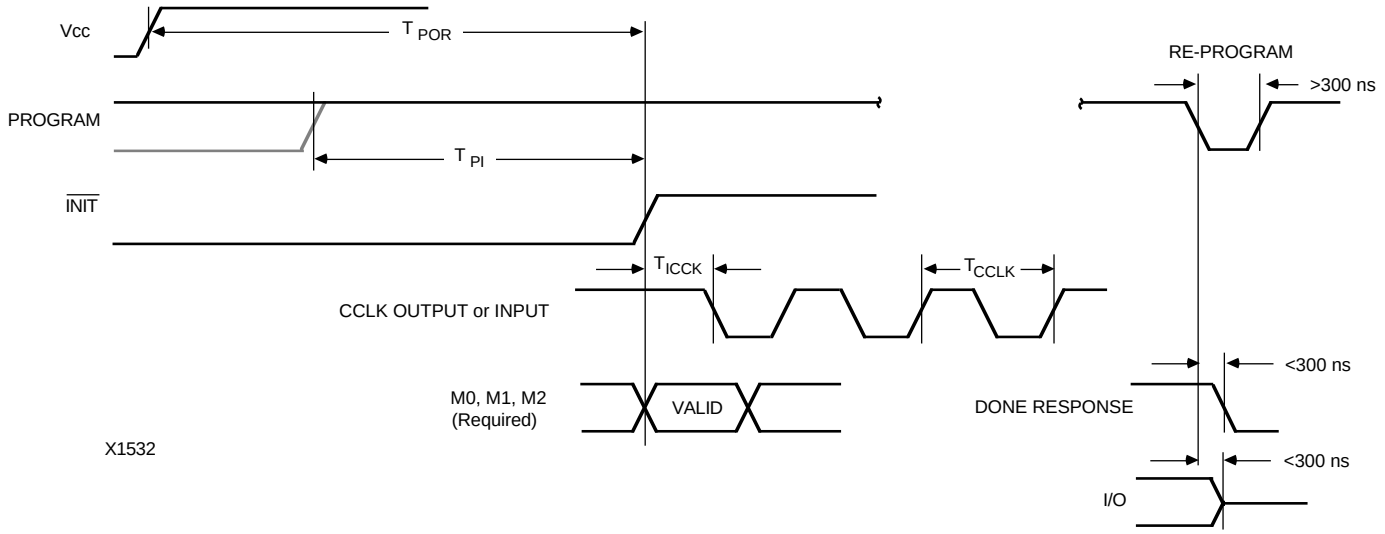
It is mandatory that the whole start-up sequence be started and completed by one byte-wide input. Otherwise, the pins used as Write Strobe or Chip Enable might become active outputs and inteffere with the final byte transfer. If this transfer does not occur, the start-up sequence will not be completed all the way to the finish (point F in Figure 21 on page 2-29). At worst, the internal reset will not be released; at best, Readback and Boundary Scan will be inhibited. The length-count value, as generated by MAKEPROM, is supposed to ensure that these problems never occur.

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

How to Delay Configuration After Power-Up

There are two methods to delay configuration after power-up: Put a logic Low on the PROGRAM input, or pull the bidirectional INIT pin Low, using an open-collector (open-drain) driver. (See also Figure 20 on page 2-27).

General LCA Switching Characteristics



Master Modes

	Symbol	Min	Max	Units
Power-On-Reset M0 = High M0 = Low	T_{POR} T_{POR}	10 40	40 130	ms ms
Program Latency	T_{PI}	30	200	µs per CLB column
CCLK (output) Delay period (slow) period (fast)	T_{ICCK} T_{CCLK} T_{CCLK}	40 640 100	250 2000 250	µs ns ns

Slave and Peripheral Modes

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

Note: At power-up, V_{CC} must rise from 2.0 V to V_{CC} min in less than 25 ms, otherwise delay configuration using PROGRAM until V_{CC} is valid.

Pin Functions During Configuration

CONFIGURATION MODE: <M2:M1:M0>						
SLAVE <1:1:1>	MASTER-SER <0:0:0>	SYN.PERIPH <0:1:1>	ASYN.PERIPH <1:0:1>	MASTER-HIGH <1:1:0>	MASTER-LOW <1:0:0>	USER OPERATION
				A16	A16	PGI-I/O
				A17	A17	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
						SGI-I/O
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)
M2 (HIGH) (I)	M2 (LOW) (I)	M2 (LOW) (I)	M2 (HIGH) (I)	M2 (HIGH) (I)	M2 (HIGH) (I)	(I)
						PGI-I/O
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
* $\overline{\text{INIT-ERROR}}$	$\overline{\text{INIT-ERROR}}$	$\overline{\text{INIT-ERROR}}$	$\overline{\text{INIT-ERROR}}$	$\overline{\text{INIT-ERROR}}$	$\overline{\text{INIT-ERROR}}$	I/O
						SGI-I/O
DONE	DONE	DONE	DONE	DONE	DONE	DONE
$\overline{\text{PROGRAM}}$ (I)	$\overline{\text{PROGRAM}}$ (I)	$\overline{\text{PROGRAM}}$ (I)	$\overline{\text{PROGRAM}}$ (I)	$\overline{\text{PROGRAM}}$ (I)	$\overline{\text{PROGRAM}}$ (I)	$\overline{\text{PROGRAM}}$
		DATA 7 (I)	DATA 7 (I)	DATA 7 (I)	DATA 7 (I)	I/O
						PGI-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
			$\overline{\text{CS0}}$ (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
			$\overline{\text{RS}}$ (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
		$\overline{\text{RDY/BUSY}}$	RDY/$\overline{\text{BUSY}}$	$\overline{\text{RCLK}}$	$\overline{\text{RCLK}}$	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	SGI-I/O
CCLK (I)	CCLK (O)	CCLK (I)	CCLK (O)	CCLK(O)	CCLK (O)	CCLK (I)
TDO	TDO	TDO	TDO	TDO	TDO	TDO-(O)
			$\overline{\text{WS}}$ (I)	A0	A0	I/O
				A1	A1	PGI-I/O
			CS1 (I)	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	SGI-I/O
						ALL OTHERS

X6081

Represents a 50 kΩ to 100 kΩ pull-up before and during configuration

* INIT is an open-drain output during configuration

(I) Represents an input

Before and during configuration, all outputs that are not used for the configuration process are 3-stated with a 50 kΩ to 100 kΩ pull-up resistor.