



# XC4000, XC4000A, XC4000H Logic Cell Array Families

## Product Description

### Features

- Third Generation Field-Programmable Gate Arrays
  - Abundant flip-flops
  - Flexible function generators
  - On-chip ultra-fast RAM
  - Dedicated high-speed carry-propagation circuit
  - Wide edge decoders
  - Hierarchy of interconnect lines
  - Internal 3-state bus capability
  - Eight global low-skew clock or signal distribution network
- Flexible Array Architecture
  - Programmable logic blocks and I/O blocks
  - Programmable interconnects and wide decoders
- Sub-micron CMOS Process
  - High-speed logic and Interconnect
  - Low power consumption
- Systems-Oriented Features
  - IEEE 1149.1-compatible boundary-scan logic support
  - Programmable output slew rate
  - Programmable input pull-up or pull-down resistors
  - 12-mA sink current per output (XC4000 family)
  - 24-mA sink current per output (XC4000A and XC4000H families)
- Configured by Loading Binary File
  - Unlimited reprogrammability
  - Six programming modes
- XACT Development System runs on '386/'486-type PC, NEC PC, Apollo, Sun-4, and Hewlett-Packard 700 series
  - Interfaces to popular design environments like Viewlogic, Mentor Graphics and OrCAD
  - Fully automatic partitioning, placement and routing
  - Interactive design editor for design optimization
  - 288 macros, 34 hard macros, RAM/ROM compiler

### Description

The XC4000 families of Field-Programmable Gate Arrays (FPGAs) provide the benefits of custom CMOS VLSI, while avoiding the initial cost, time delay, and inherent risk of a conventional masked gate array.

The XC4000 families provide a regular, flexible, programmable architecture of Configurable Logic Blocks (CLBs), interconnected by a powerful hierarchy of versatile routing resources, and surrounded by a perimeter of programmable Input/Output Blocks (IOBs).

XC4000-family devices have generous routing resources to accommodate the most complex interconnect patterns. XC4000A devices have reduced sets of routing resources, sufficient for their smaller size. XC4000H high I/O devices maintain the same routing resources and CLB structure as the XC4000 family, while nearly doubling the available I/O.

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

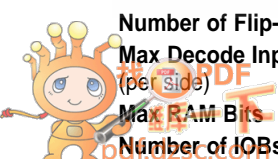
The XC4000 families are supported by powerful and sophisticated software, covering every aspect of design: from schematic entry, to simulation, to automatic block placement and routing of interconnects, and finally the creation of the configuration bit stream.

Since Xilinx FPGAs can be reprogrammed an unlimited number of times, they can be used in innovative designs where hardware is changed dynamically, or where hardware must be adapted to different user applications. FPGAs are ideal for shortening the design and development cycle, but they also offer a cost-effective solution for production rates well beyond 1000 systems per month.

**Table 1. The XC4000 Families of Field-Programmable Gate Arrays**

Device	XC4002A	4003/3A	4003H	4004A	4005/5A	4005H	4006	4008	4010/10D	4013/13D	4020	4025
Appr. Gate Count	2,000	3,000	3,000	4,000	5,000	5,000	6,000	8,000	10,000	13,000	20,000	25,000
CLB Matrix	8 x 8	10 x 10	10 x 10	12 x 12	14 x 14	14 x 14	16 x 16	18 x 18	20 x 20	24 x 24	28 x 28	32 x 32
Number of CLBs	64	100	100	144	196	196	256	324	400	576	784	1,024
Number of Flip-Flops	256	360	200	480	616	392	768	936	1,120	1,536	2,016	2,560
Max Decode Inputs (per Side)	24	30	30	36	42	42	48	54	60	72	84	96
Max RAM Bits	2,048	3,200	3,200	4,608	6,272	6,272	8,192	10,368	12,800*	18,432*	25,088	32,768
Number of IOBs	64	80	160	96	112	192	128	144	160	192	224	256

\*XC4010D and XC4013D have no RAM



## XC4000, XC4000A, XC4000H Logic Cell Array Families

### XC4000 Compared to XC3000A

For those readers already familiar with the XC3000A family of Xilinx Field Programmable Gate Arrays, here is a concise list of the major new features in the XC4000 family.

CLB has two **independent** 4-input function generators.

A **third** function generator combines the outputs of the two other function generators with a ninth input.

All function inputs are swappable, all have full access; none are mutually exclusive.

CLB has **very fast arithmetic carry** capability.

CLB function generator look-up table can also be used as high-speed **RAM**.

CLB flip-flops have asynchronous set **or** reset.

CLB has **four outputs**, two flip-flops, two combinatorial.

CLB connections symmetrically located on all **four** edges.

**IOB** has more versatile clocking polarity options.

**IOB** has programmable input set-up time:

**long** to avoid potential hold time problems,

**short** to improve performance.

**IOB** has Longline access through its own TBUF.

Outputs are **n-channel only**, lower  $V_{OH}$  increases speed.

XC4000 outputs can be paired to double sink current to **24 mA**. XC4000A and XC4000H outputs can each sink 24 mA, can be paired for **48 mA** sink current.

IEEE 1149.1- type **boundary scan** is supported in the I/O.

**Wide decoders** on all four edges of the LCA device.

Increased **number of interconnect resources**.

All CLB inputs and outputs have **access to most interconnect lines**.

**Switch Matrices** are simplified to increase speed.

**Eight global nets** can be used for clocking or distributing logic signals.

**TBUF** output configuration is more versatile and 3-state control less confined.

**Program** is single-function input pin, overrides everything.

**INIT pin** also acts as Configuration Error output.

**Peripheral Synchronous Mode** (8 bit) has been added.

**Peripheral Asynchronous Mode** has improved handshake.

**Start-up** can be **synchronized** to any user clock (this is a configuration option).

No Powerdown, but instead a **Global 3-state input** that does not reset any flip-flops.

No on-chip **crystal oscillator** amplifier.

Configuration Bit Stream includes **CRC error checking**.

**Configuration Clock** can be increased to **>8 MHz**.

Configuration Clock is **fully static**, no constraint on the maximum Low time.

**Readback** either ignores flip-flop content (avoids need for masking) or it takes a **snapshot** of all flip-flops at the start of Readback.

Readback has same **polarity** as Configuration and can be **aborted**.

Table 2. Three Generations of Xilinx Field-Programmable Gate Array Families

Parameter	XC4025	XC3195A	XC2018
Number of flip-flops	2,560	1,320	174
Max number of user I/O	256	176	74
Max number of RAM bits	32,768	0	0
Function generators per CLB	3	2	2
Number of logic inputs per CLB	9	5	4
Number of logic outputs per CLB	4	2	2
Number of low-skew global nets	8	2	2
Dedicated decoders	yes	no	no
Fast carry logic	yes	no	no
Internal 3-state drivers	yes	yes	no
Output slew-rate control	yes	yes	no
Power-down option	no	yes	yes
Crystal oscillator circuit	no	yes	yes

## Architectural Overview

The XC4000 families achieve high speed through advanced semiconductor technology and through improved architecture, and supports system clock rates of up to 50 MHz. Compared to older Xilinx FPGA families, the XC4000 families are more powerful, offering on-chip RAM and wide-input decoders. They are more versatile in their applications, and design cycles are faster due to a combination of increased routing resources and more sophisticated software. And last, but not least, they more than double the available complexity, up to the 20,000-gate level.

The XC4000 families have 16 members, ranging in complexity from 2,000 to 25,000 gates.

### Logic Cell Array Families

Xilinx high-density user-programmable gate arrays include three major configurable elements: configurable logic blocks (CLBs), input/output blocks (IOBs), and interconnections. The CLBs provide the functional elements for constructing the user's logic. The IOBs provide the interface between the package pins and internal signal lines. The programmable interconnect resources provide routing paths to connect the inputs and outputs of the CLBs and IOBs onto the appropriate networks. Customized configuration is established by programming internal static memory cells that determine the logic functions and interconnections implemented in the LCA device.

The first generation of LCA devices, the XC2000 family, was introduced in 1985. It featured logic blocks consisting of a combinatorial function generator capable of implementing 4-input Boolean functions and a single storage element. The XC2000 family has two members ranging in complexity from 800 to 1500 gates.

In the second-generation XC3000A LCA devices, introduced in 1987, the logic block was expanded to implement wider Boolean functions and to incorporate a second flip-flop in each logic block. Today, the XC3000 devices range in complexity from 1,300 to 10,000 usable gates. They have a maximum guaranteed toggle frequency ranging from 70 to 270 MHz, equivalent to maximum system clock frequencies of up to 80 MHz.

The third generation of LCA devices further extends this architecture with a yet more powerful and flexible logic block. I/O block functions and interconnection options have also been enhanced with each successive generation, further extending the range of applications that can be implemented with an LCA device.

This third-generation architecture forms the basis of the XC4000 families of devices that feature logic densities up to 25,000 usable gates and support system clock rates of

up to 50 MHz. The use of an advanced, sub-micron CMOS process technology as well as architectural improvements contribute to this increase in FPGA capabilities. However, achieving these high logic-density and performance levels also requires new and more powerful automated design tools. IC and software engineers collaborated during the definition of the third-generation LCA architecture to meet an important performance goal — an FPGA architecture and companion design tools for completely automatic placement and routing of 95% of all designs, plus a convenient way to complete the remaining few designs.

### Configurable Logic Blocks

A number of architectural improvements contribute to the increased logic density and performance levels of the XC4000 families. The most important one is a more powerful and flexible CLB surrounded by a versatile set of routing resources, resulting in more "effective gates per CLB." The principal CLB elements are shown in Figure 1. Each new CLB also packs a pair of flip-flops and two independent 4-input function generators. The two function generators offer designers plenty of flexibility because most combinatorial logic functions need less than four inputs. Consequently, the design-software tools can deal with each function generator independently, thus improving cell usage.

Thirteen CLB inputs and four CLB outputs provide access to the function generators and flip-flops. More than double the number available in the XC3000 families, these inputs and outputs connect to the programmable interconnect resources outside the block. Four independent inputs are provided to each of two function generators (F1 – F4 and G1 – G4). These function generators, whose outputs are labeled F' and G', are each capable of implementing any arbitrarily defined Boolean function of their four inputs. The function generators are implemented as memory look-up tables; therefore, the propagation delay is independent of the function being implemented. A third function generator, labeled H', can implement any Boolean function of its three inputs: F' and G' and a third input 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, and G' or H' can be connected to the Y output. Thus, a CLB can be used to implement any two independent functions of up-to-four variables, or any single function of five variables, or any function of four variables together with some functions of five variables, or it can implement even 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 density and speed.

The two storage elements in the CLB are edge-triggered D-type flip-flops with common clock (K) and clock enable (EC) inputs. A third common input (S/R) can be programmed as either an asynchronous set or reset signal

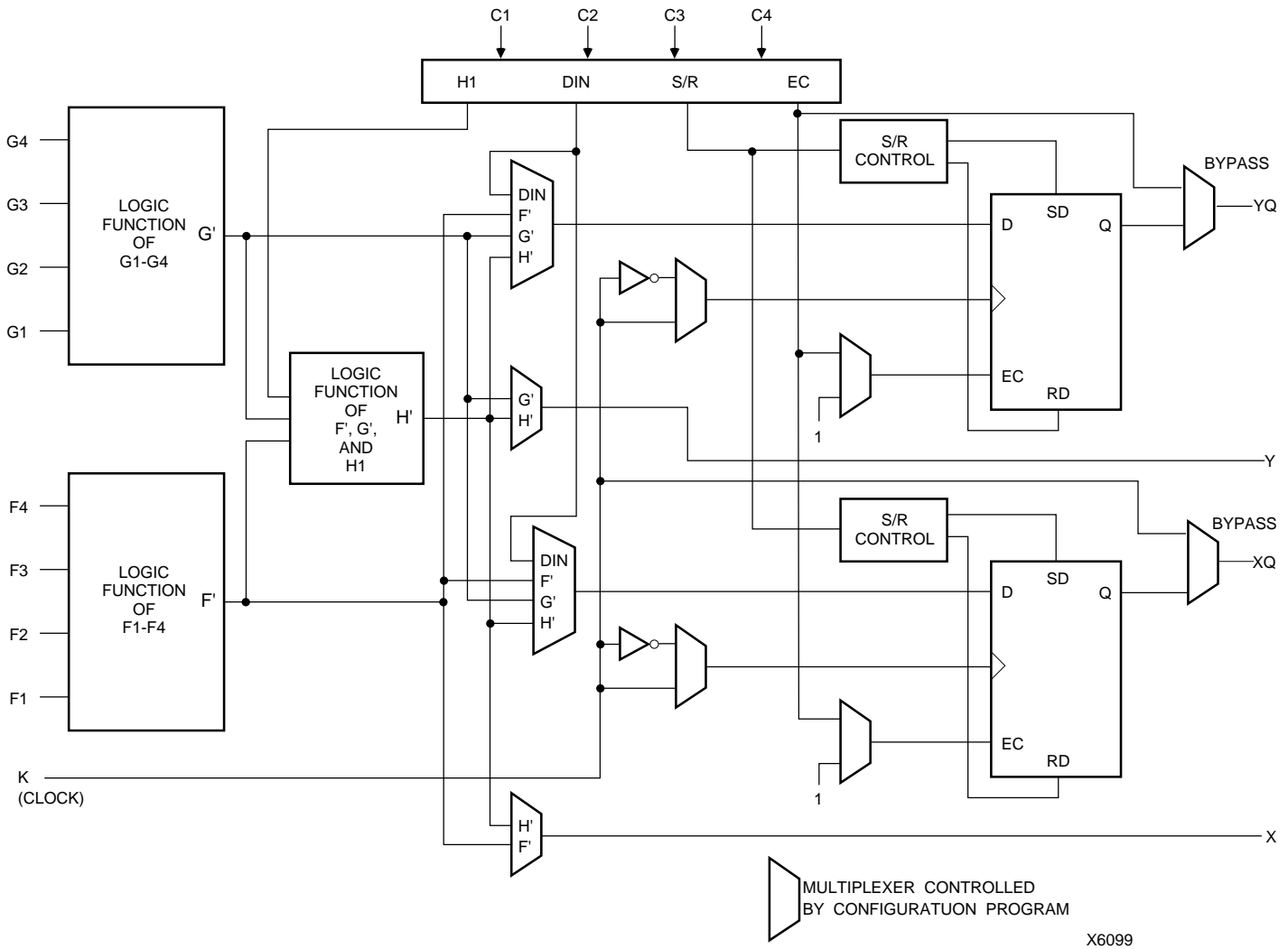


Figure 1. Simplified Block Diagram of XC4000-Families Configurable Logic Block

independently for each of the two registers; this input also can be disabled for either flip-flop. A separate global Set/Reset line (not shown in Figure 1) sets or clears each register during power-up, reconfiguration, or when a dedicated Reset net is driven active. This Reset net does not compete with other routing resources; it can be connected to any package pin as a global reset input.

Each flip-flop can be triggered on either the rising or falling clock edge. The source of a flip-flop data input is programmable: it is driven either by the functions F', G', and H', or the Direct In (DIN) block input. The flip-flops drive the XQ and YQ CLB outputs.

In addition, each CLB F' and G' function generator contains dedicated arithmetic logic for the fast generation of carry and borrow signals, greatly increasing the efficiency

and performance of adders, subtractors, accumulators, comparators and even counters.

Multiplexers in the CLB map the four control inputs, labeled C1 through C4 in Figure 1, into the four internal control signals (H1, DIN, S/R, and EC) in any arbitrary manner.

The flexibility and symmetry of the CLB architecture facilitates the placement and routing of a given application. Since the function generators and flip-flops have independent inputs and outputs, each can be treated as a separate entity during placement to achieve high packing density. Inputs, outputs, and the functions themselves can freely swap positions within a CLB to avoid routing congestion during the placement and routing operation.

### 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  $2 \times 5.5 \text{ ns} = 11 \text{ ns}$ . 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 micro-processor 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)	
<b>16-bit Decoder From Input Pad</b>		15 ns	4 CLBs	12 ns	0 CLBs
<b>24-bit Accumulator</b>		17 MHz	46 CLBs	32 MHz	13 CLBs
<b>State Machine Benchmark*</b>		18 MHz	34 CLBs	30 MHz	26 CLBs
<b>16:1 Multiplexer</b>		16 ns	8 CLBs	16 ns	5 CLBs
<b>16-bit Unidirectional</b>	Max Density	20 MHz	16 CLBs	40 MHz	8 CLBs
<b>Loadable Counter</b>	Max Speed	34 MHz	23 CLBs	42 MHz	9 CLBs
<b>16-bit U/D Counter</b>	Max Density	20 MHz	16 CLBs	40 MHz	8 CLBs
	Max Speed	30 MHz	27 CLBs	40 MHz	8 CLBs
<b>16-bit Adder</b>	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

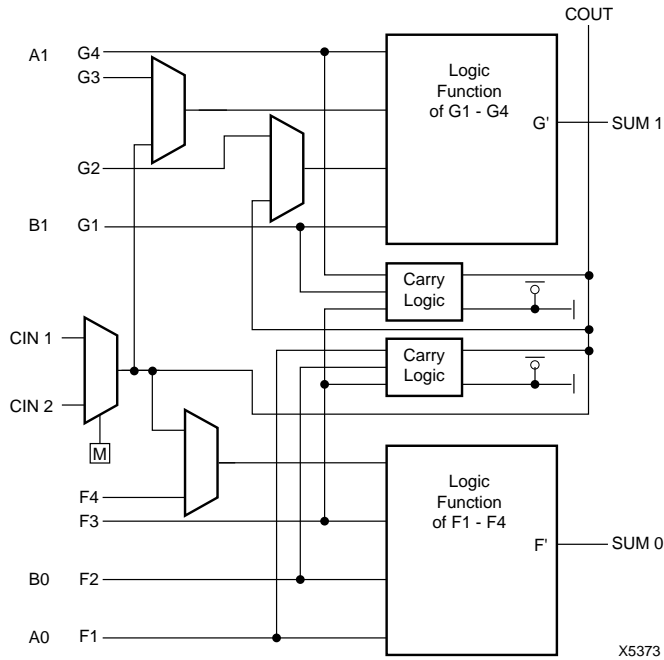


Figure 2. Fast Carry Logic in Each CLB

up/down counter, this means twice the speed in half the number of CLBs, compared with the XC3000 families.

**Pipelining Speeds Up The System:** The abundance of flip-flops in the CLBs invites pipelined designs. This is a powerful way of increasing performance by breaking the function into smaller subfunctions and executing them in parallel, passing on the results through pipeline flip-flops. This method should be seriously considered wherever total performance is more important than simple through-delay.

**Wide Edge Decoding:** For years, FPGAs have suffered from the lack of wide decoding circuitry. When the address or data field is wider than the function generator inputs (five bits in the XC3000 families), FPGAs need multi-level decoding and are thus slower than PALs. The XC4000-family CLBs have nine inputs; any decoder of up to nine inputs is, therefore, compact and fast. But, there is also a need for much wider decoders, especially for address decoding in large microprocessor systems. The XC4000 family has four programmable decoders located on each edge of each device. Each of these wired-AND gates is capable of accepting up to 42 inputs on the XC4005 and 72 on the XC4013. These decoders may also be split in two when a large number of narrower decoders are required for a maximum of 32 per device. These dedicated decoders accept I/O signals and internal signals as inputs and generate a decoded internal signal in 18 ns, pin-to-pin. The XC4000A family has only two decoder AND gates per edge which, when split provide a maximum of 16 per device. 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 FPGAs. Users often resorted to external PALs for simple but fast decoding functions. Now, the dedicated decoders in the XC4000 can implement these functions efficiently and fast.

**Higher Output Current:** The 4-mA maximum output current specification of today's FPGAs often forces the user to add external buffers, cumbersome especially on bidirectional I/O lines. The XC4000 families solve many of these problems by increasing the maximum output sink current to 12 mA. Two adjacent outputs may be interconnected to increase the output sink current to 24 mA. The FPGA can thus drive short buses on a pc board. The XC4000A and XC4000H outputs can sink 24 mA per output and can double up for 48 mA.

While the XC2000 and XC3000 families used complementary output transistors, the XC4000 outputs are n-channel for both pull-down and pull-up, somewhat analogous to the classical totem pole used in TTL. The reduced output High level (VOH) makes circuit delays more symmetrical for TTL-threshold systems. The XC4000H outputs have an optional p-channel output transistor.

### Abundant Routing Resources

Connections between blocks are made by metal lines with programmable switching points and switching matrices. Compared to the previous LCA families, these routing resources have been increased dramatically. The number of globally distributed signals has been increased from two to eight, and these lines have access to any clock or logic input. The designer of synchronous systems can now distribute not only several clocks, but also control signals, all over the chip, without having to worry about any skew.

There are more than twice as many horizontal and vertical Longlines that can carry signals across the length or width of the chip with minimal delay and negligible skew. The horizontal Longlines can be driven by 3-state buffers, and can thus be used as unidirectional or bidirectional data buses; or they can implement wide multiplexers or wired-AND functions.

Single-length lines connect the switching matrices that are located at every intersection of a row and a column of CLBs. These lines provide the greatest interconnect flexibility, but cause a delay whenever they go through a switching matrix. Double-length lines bypass every other matrix, and provide faster signal routing over intermediate distances.

Compared to the XC3000 family, the XC4000 families have more than double the routing resources, and they are arranged in a far more regular fashion. In older devices,

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.

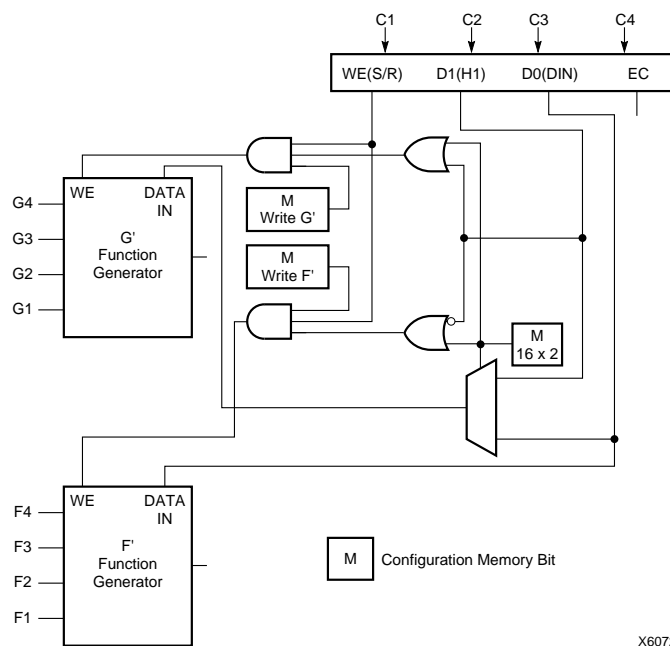
### On-Chip Memory

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.



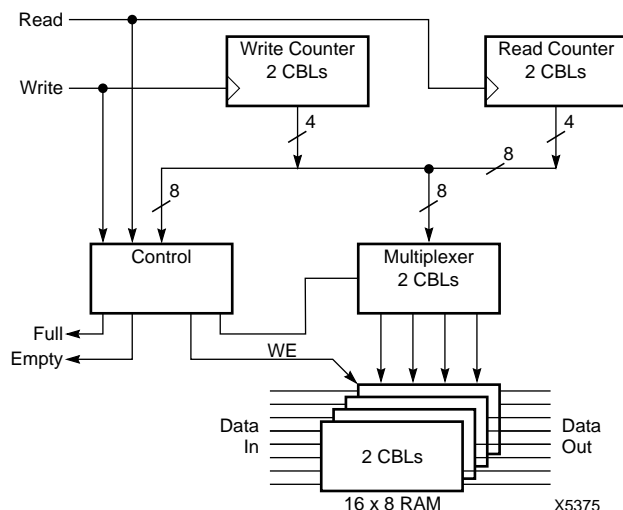
X6072

**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



X5375

**Figure 4. 16-byte FIFO**

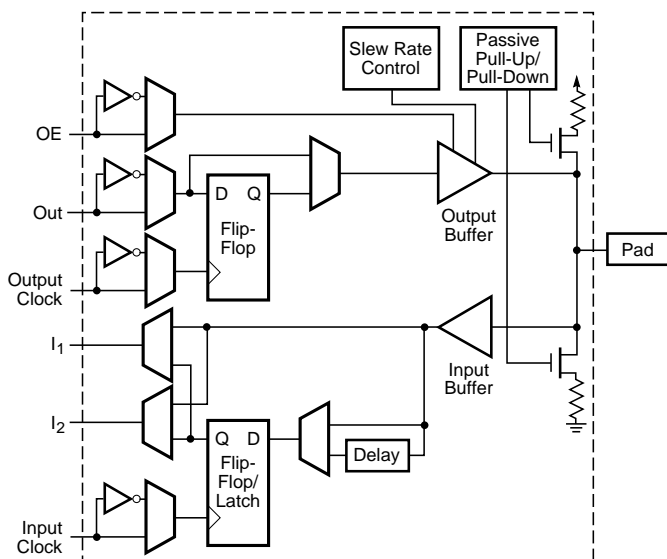
## XC4000, XC4000A, XC4000H Logic Cell Array Families

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.



X6073

**Figure 5. XC4000 and XC4000A Families Input/Output Block**

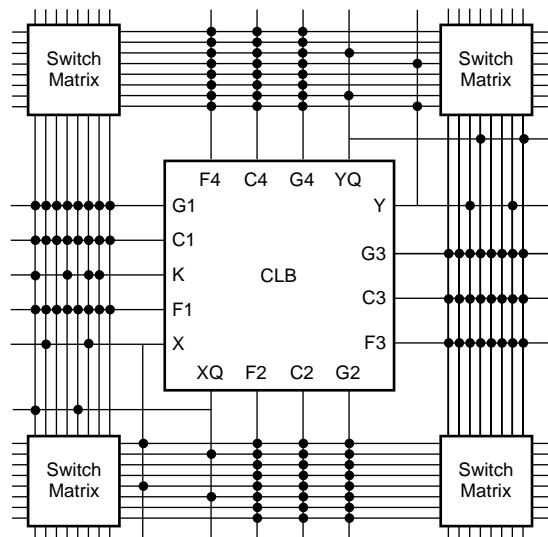
## Programmable Interconnect

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



X3242

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

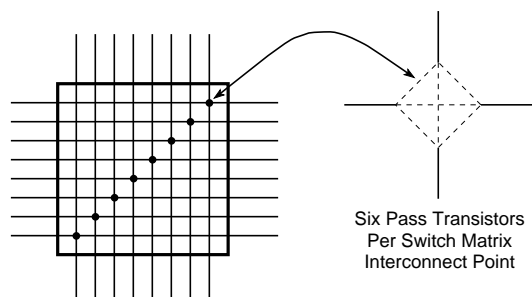


surrounding one CLB in the array. Each Switch Matrix consists of programmable n-channel pass transistors used to establish connections between the single-length lines (Figure 7). For example, a signal entering on the right side of the Switch Matrix can be routed to a single-length line on the top, left, 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.

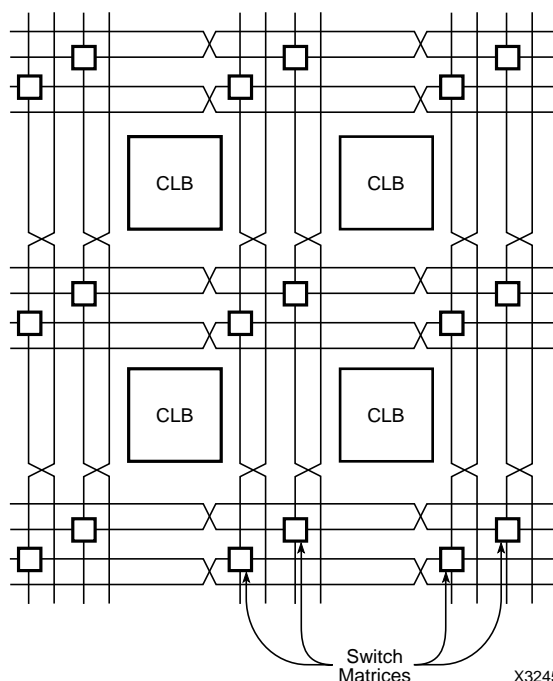
The function generator and control inputs to the CLB (F1-F4, G1-G4, and C1-C4) can be driven from any adjacent single-length line segment (Figure 6). The CLB clock (K) input can be driven from one-half of the adjacent single-length lines. Each CLB output can drive several of the single-length lines,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



X3244

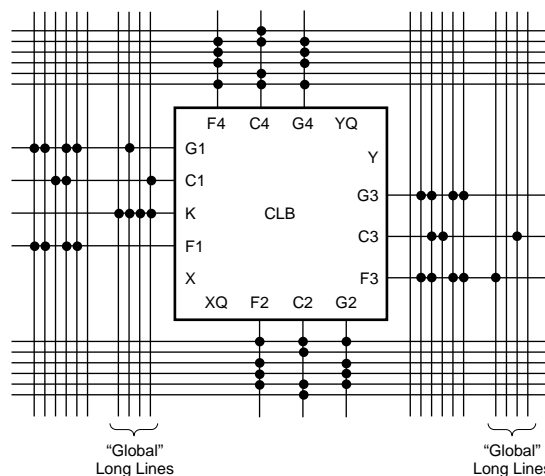
Figure 7. Switch Matrix



X3245

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 inter-connected lines.



X5520

Figure 9. Longline Routing Resources with Typical CLB Connections

## XC4000, XC4000A, XC4000H Logic Cell Array Families

---

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,

comparators, counters, data registers, decoders, encoders, I/O functions, latches, Boolean functions, RAM and ROM memory blocks, multiplexers, shift registers, and barrel shifters.

Designing with macros is as easy as designing with standard SSI/MSI functions. The 'soft macro' library contains detailed descriptions of common logic functions, but does not contain any partitioning or routing information. The performance of these macros depends, therefore, on how the PPR software processes the design. Relationally Placed Macros (RPMs), on the other hand, do contain pre-determined partitioning and relative placement information, resulting in an optimized implementation for these functions. Users can create their own library elements – either soft macros or RPMs – based on the macros and primitives of the standard library.

X-BLOX is a graphics-based high-level description language (HDL) that allows designers to use a schematic editor to enter designs as a set of generic modules. The X-BLOX compiler optimizes the modules for the target device architecture, automatically choosing the appropriate architectural resources for each function.

The XACT design environment supports hierarchical design entry, with top-level drawings defining the major functional blocks, and lower-level descriptions defining the logic in each block. The implementation tools automatically combine the hierarchical elements of a design. Different hierarchical elements can be specified with different design entry tools, allowing the use of the most convenient entry method for each portion of the design.

### **Design Implementation**

The design implementation tools satisfy the requirement for an automated design process. Logic partitioning, block placement and signal routing, encompassing the design implementation process, are performed by the Partition, Place, and Route program (PPR). The partitioner takes the logic from the entered design and maps the logic into the architectural resources of the FPGA (such as the logic blocks, I/O blocks, 3-state buffers, and edge decoders). The placer then determines the best locations for the blocks, depending on their connectivity and the required performance. The router finally connects the placed blocks together. The PPR algorithms result in the fully automatic implementation of most designs. However, for demanding applications, the user may exercise various degrees of control over the automated implementation process. Optionally, user-designated partitioning, placement, and routing information can be specified as part of the design entry process. The implementation of highly-structured designs can greatly benefit from the basic floorplanning techniques familiar to designers of large gate arrays.

The PPR program includes XACT-Performance, a feature that allows designers to specify the timing requirements

along entire paths during design entry. Timing path analysis routines in PPR then recognize and accommodate the user-specified requirements. Timing requirements can be entered on the schematic in a form directly relating to the system requirements (such as the targeted minimum clock frequency, or the maximum allowable delay on the data path between two registers). So, while the timing of each individual net is not predictable (nor does it need to be), the overall performance of the system along entire signal paths is automatically tailored to match user-generated specifications.

The automated implementation tools are complemented by the XACT Design Editor (XDE), an interactive graphics-based editor that displays a model of the actual logic and routing resources of the FPGA. XDE can be used to directly view the results achieved by the automated tools. Modifications can be made using XDE; XDE also performs checks for logic connectivity and possible design-rule violations.

### **Design Verification**

The high development cost associated with common mask-programmed gate arrays necessitates extensive simulation to verify a design. Due to the custom nature of masked gate arrays, mistakes or last-minute design changes cannot be tolerated. A gate-array designer must simulate and test all logic and timing using simulation software. Simulation describes what happens in a system under worst-case situations. However, simulation is tedious and slow, and simulation vectors must be generated. A few seconds of system time can take weeks to simulate.

Programmable-gate-array users, however, can use in-circuit debugging techniques in addition to simulation. Because Xilinx devices are reprogrammable, designs can be verified in the system in real time without the need for extensive simulation vectors.

The XACT development system supports both simulation and in-circuit debugging techniques. For simulation, the system extracts the post-layout timing information from the design database. This data can then be sent to the simulator to verify timing-critical portions of the design. Back-annotation – the process of mapping the timing information back into the signal names and symbols of the schematic – eases the debugging effort.

For in-circuit debugging, XACT includes a serial download and readback cable (XChecker) that connects the device in the system to the PC or workstation through an RS232 serial port. The engineer can download a design or a design revision into the system for testing. The designer can also single-step the logic, read the contents of the numerous flip-flops on the device and observe internal logic levels. Simple modifications can be downloaded into the system in a matter of minutes.

## XC4000, XC4000A, XC4000H Logic Cell Array Families

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		Barrel Shifters	Multiplexers
	# of CLBs		
'138	5	brlshft4	m2-1e
'139	2	brlshft8	m4-1e
'147	5		m8-1e
'148	6	<b>4-Bit Counters</b>	m16-1e
'150	5	cd4ce	
'151	3	cd4cle	<b>Registers</b>
'152	3	cd4rle	rd4r
'153	2	cb4ce	rd8r
'154	16	cb4cle	rd16r
'157	2	cb4re	
'158	2		<b>Shift Registers</b>
'160	5	<b>8- and 16-Bit Counters</b>	sr8ce
'161	6	cb8ce	sr16re
'162	8	cb8re	
'163	8	cc16ce	<b>RAMs</b>
'164	4	cc16cle	ram 16x4
'165s	9	cc16cled	
'166	5		<b>Explanation of counter nomenclature</b>
'168	7	<b>Identity Comparators</b>	cb = binary counter
'174	3	comp4	cd = BCD counter
'194	5	comp8	cc = cascadable binary counter
'195	3	comp16	d = bidirectional
'280	3		l = loadable
'283	8	<b>Magnitude Comparators</b>	x = cascadable
'298	2	compm4	e = clock enable
'352	2	compm8	r = synchronous reset
'390	3	compm16	c = asynchronous clear
'518	3		
'521	3	<b>Decoders</b>	
		d2-4e	
		d3-8e	
		d4-16e	

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

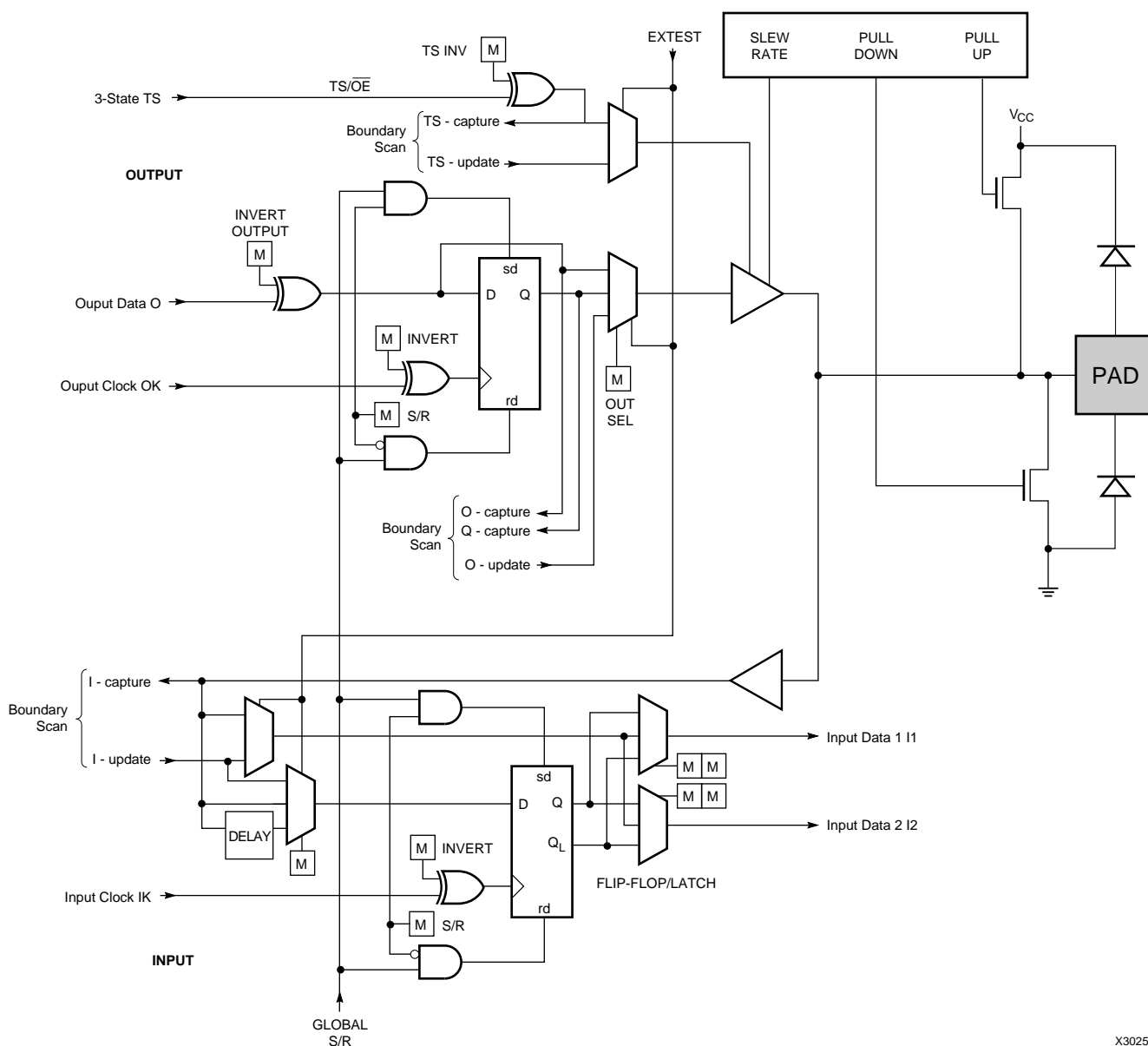


Figure 11. XC4000 and XC4000A I/O Block

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.

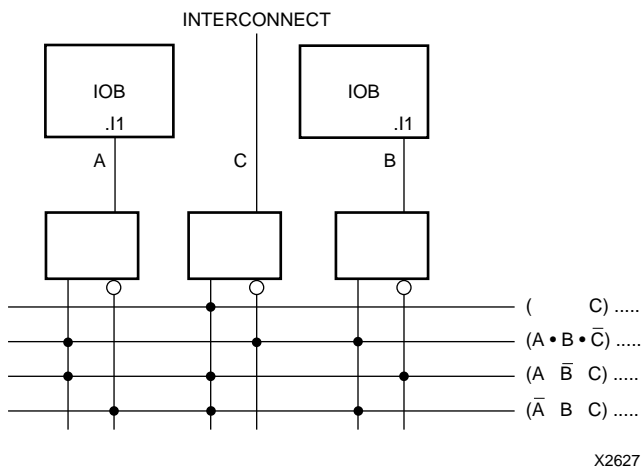
Under configuration control, the set-up time of this flip-flop can be increased so that normal clock routing does not result in a hold-time problem. Note that the input flip-flop set-up time is defined between the data measured at the device I/O pin and the clock input at the IOB. Any clock routing delay 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.

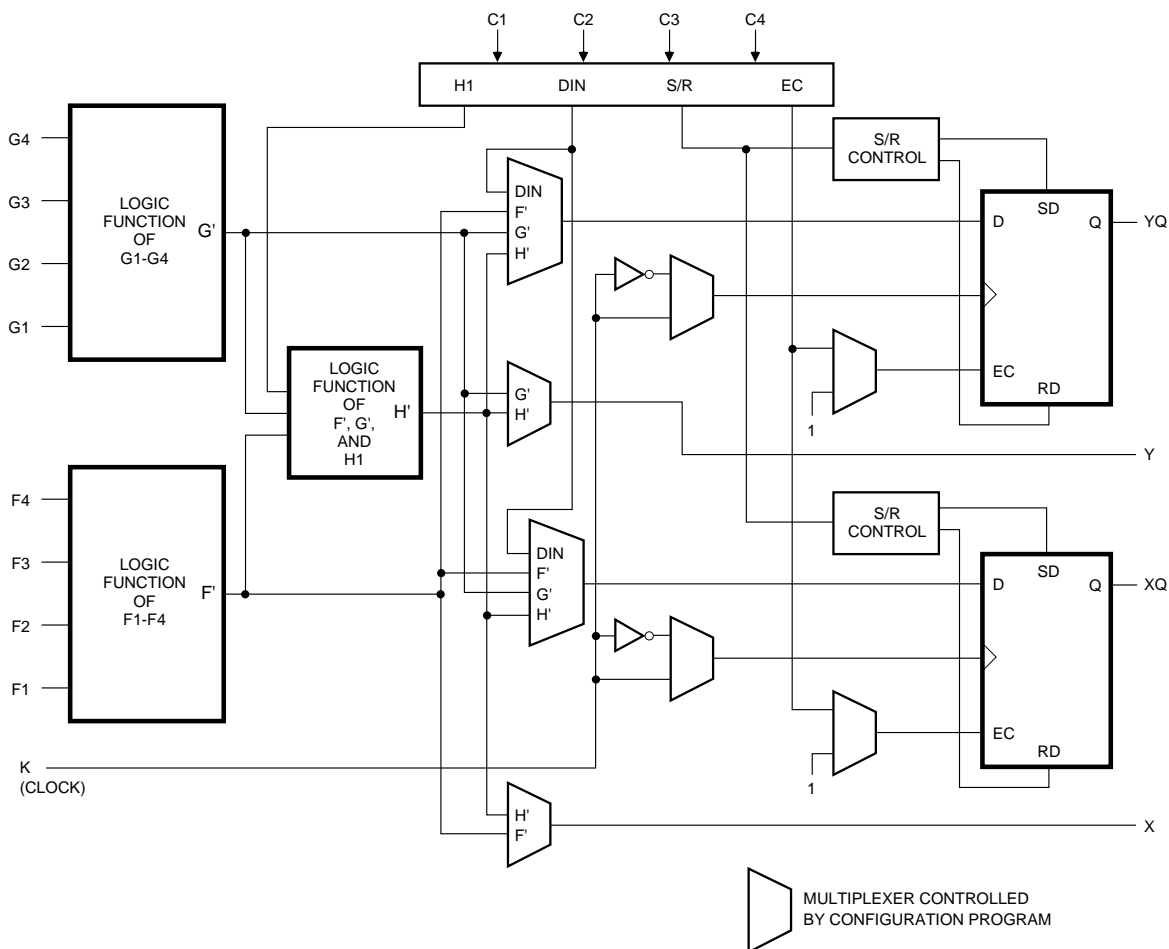
**Fast Carry Logic**

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



X1519

Figure 13. Simplified Block Diagram of XC4000 Configurable Logic Block

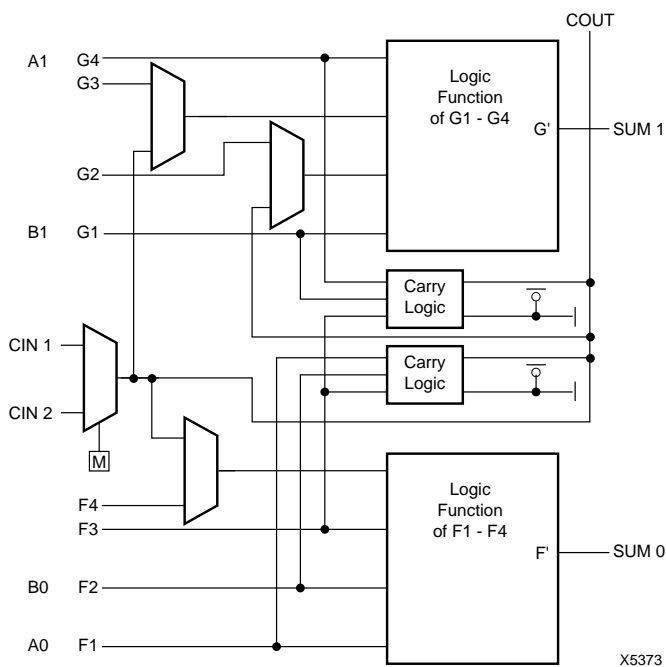
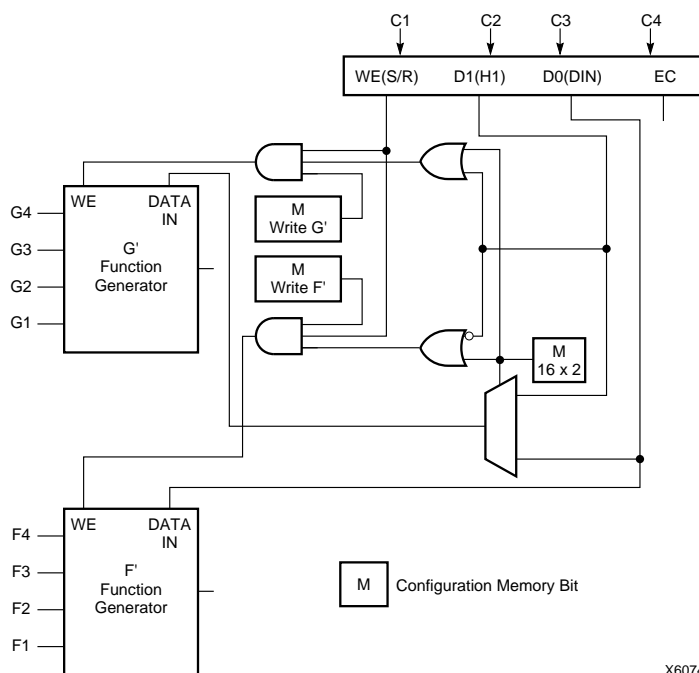


Figure 14. Fast Carry Logic in Each CLB



X6074

Figure 15. CLB Function Generators Can Be Used as Read/Write Memory Cells

**Boundary Scan**

Boundary Scan is becoming an attractive feature that helps sophisticated systems manufacturers test their PC boards more safely and more efficiently. The XC4000 family 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.

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 electronic design. This structure is easily implemented with the serial and/or parallel connections of a four-pin interface on any Boundary-Scan-compatible IC. By exercising these 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 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 the Xilinx part.

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. A register operation begins with a *capture* where a set of data is parallel loaded into the designated register for shifting out. The next state is *shift*, where captured data are shifted out while the desired data are shifted in. A number of states are provided for Wait operations. The last state of a register sequence is the *update* where the shifted content of the register is loaded into the appropriate instruction- or data-holding register, either for instruction-register decode or for data-register pin control.

The primary data register is the Boundary-Scan register. For each IOB pin in the LCA device, it includes three bits of shift register and three *update* latches for: in, out and 3-state control. Non-IOB pins have appropriate partial bit population for in or out only. Each Extest Capture captures all available input pins.

The other standard data register is the single flip-flop *bypass* register. It resynchronizes data being passed through a device that need not be involved in the current scan operation. The LCA device provides two user nets (BSCAN.SEL1 and BSCAN.SEL2) which are the decodes of two user instructions. For these instructions, two corresponding nets (BSCAN.TDO1 and BSCAN.TDO2) allow

user scan data to be shifted out on TDO. The data register clock (BSCAN.DRCK) is available for control of test logic which the user may wish to implement with CLBs. The NAND of TCK and Run-test-idle is also provided (BSCAN.IDLE).

The XC4000 Boundary Scan instruction set also includes instructions to configure the device and read back the configuration data.

**Table 4. Boundary Scan Instruction**

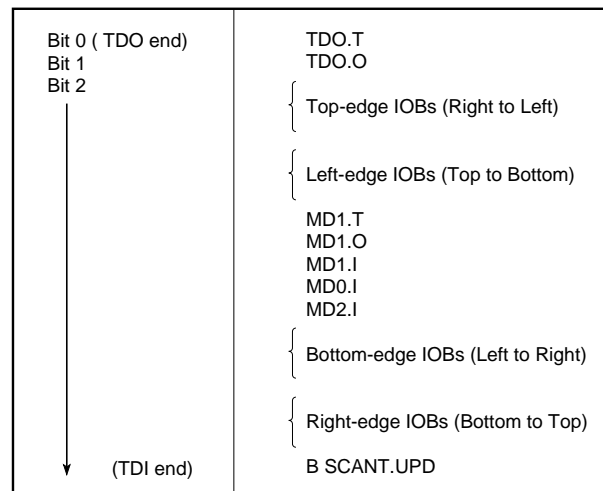
Instruction			Test Selected	TDO Source	I/O Data Source
I <sub>2</sub>	I <sub>1</sub>	I <sub>0</sub>			
0	0	0	Extest	DR	DR
0	0	1	Sample/Preload	DR	Pin/Logic
0	1	0	User 1	TDO1	Pin/Logic
0	1	1	User 2	TDO2	Pin/Logic
1	0	0	Readback	Readback Data	Pin/Logic
1	0	1	Configure	DOUT	Disabled
1	1	0	Reserved	—	—
1	1	1	Bypass	Bypass Reg	Pin/Logic

X2679

**Bit Sequence**

The bit sequence within each IOB is: in, out, 3-state. From a cavity-up (XDE) view of the chip, starting in the upper right chip corner, the Boundary-Scan data-register bits have the following order.

**Table 5. Boundary Scan Order**



X6075

The data register also includes the following non-pin bits: TDO.T, and TDO.I, which are always bits 0 and 1 of the data register, respectively, and BSCANT.UPD which is always the last bit of the data register. These three Boundary-Scan bits are special-purpose Xilinx test signals. PROGRAM, CCLK and DONE are not included in the Boundary-Scan register. For more information regarding Boundary Scan, refer to XAPP 017.001, *Boundary Scan in XC4000 Devices*.



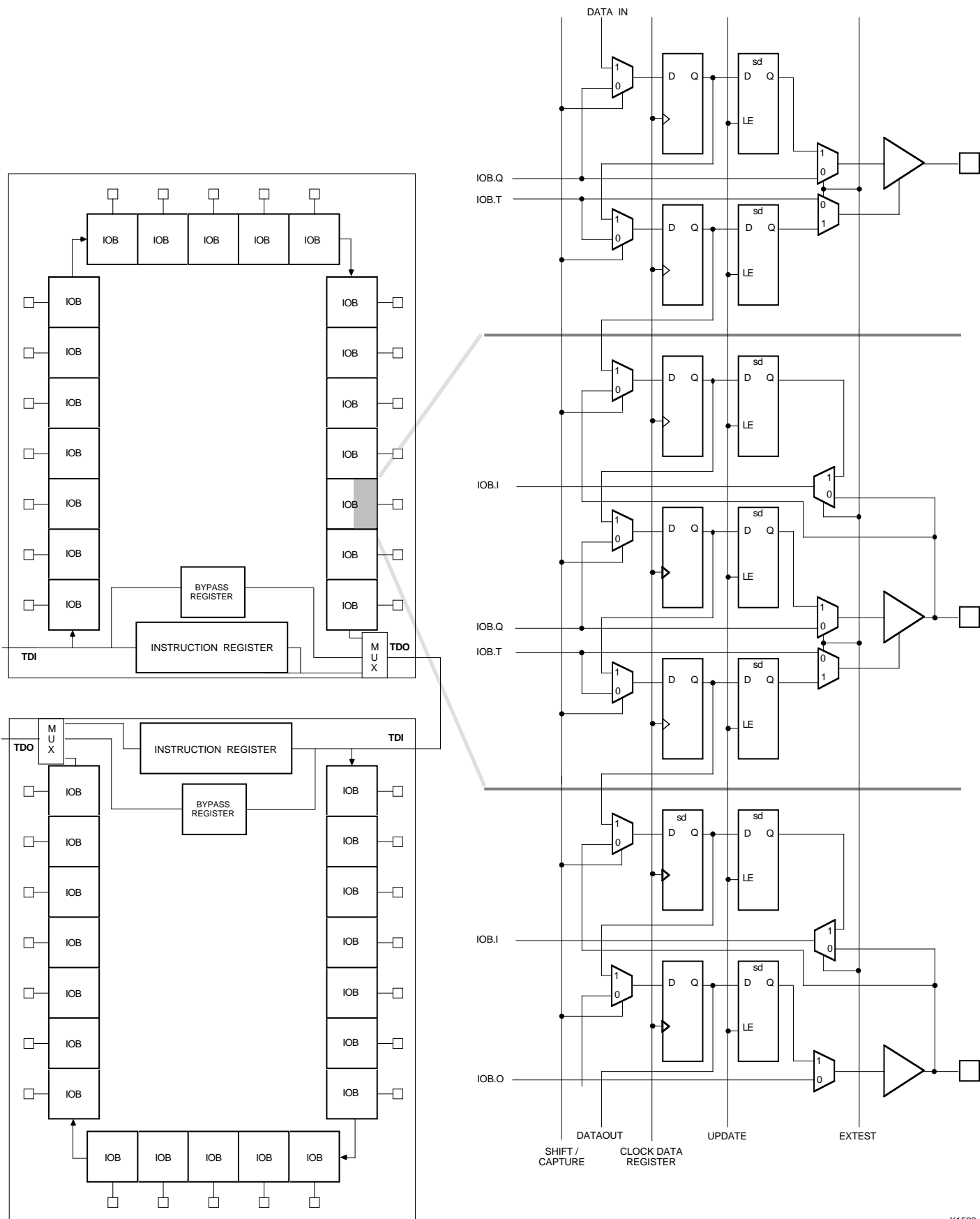


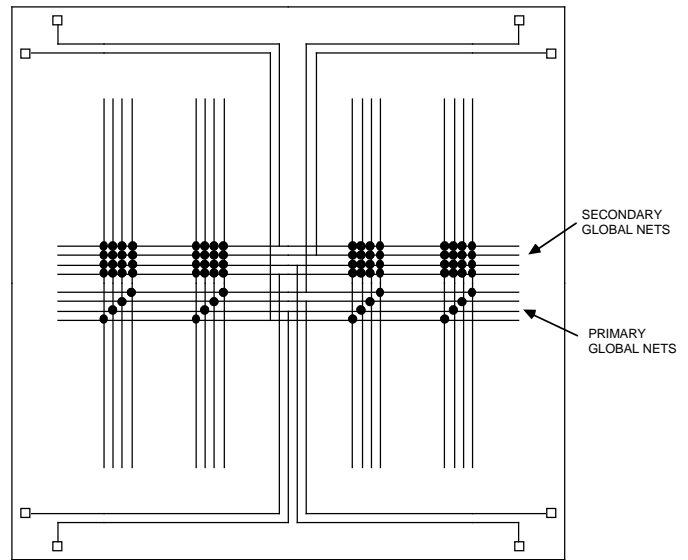
Figure 16. XC4000 Boundary Scan Logic. Includes three bits of Data Register per IOB, the IEEE 1149.1 Test Access Port controller, and the Instruction Register with decodes.

**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.

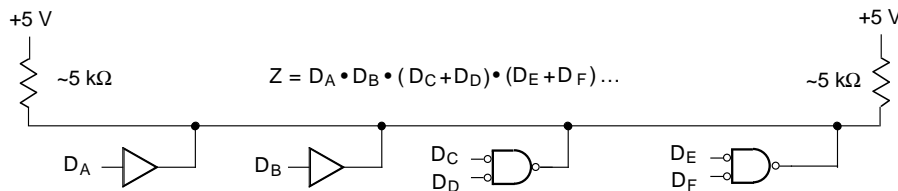
Each CLB column has four dedicated Vertical Longlines, each of these lines has access to a particular Primary Global Net, or to any one of the Secondary Global Nets. The Global Nets avoid clock skew and potential hold-time



X1027

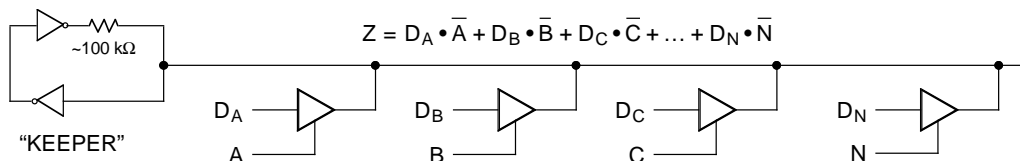
**Figure 17. XC4000 Global Net Distribution.** Four Lines per Column; Eight Inputs in the Four Chip Corners.

problems. The user must specify these Global Nets for all timing-sensitive global signal distribution.



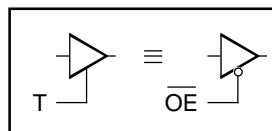
X1006

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



X1007

**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.**

## Oscillator

An internal oscillator is used for clocking of the power-on time-out, configuration memory clearing, and as the source of CCLK in Master modes. This oscillator signal runs at a nominal 8 MHz and varies with process,  $V_{CC}$  and temperature between 10 MHz max and 4 MHz min. This signal is available on an output control net (OSCO) in the upper right corner of the chip, if the oscillator-run control bit is enabled in the configuration memory. Two of four resynchronized taps of the power-on time-out divider are also available on OSC1 and OSC2. These taps are at the fourth, ninth, fourteenth and nineteenth bits of the ripple divider. This can provide output signals of approximately 500 kHz, 16 kHz, 490 Hz and 15 Hz.

## Special Purpose Pins

The mode pins are sampled prior to configuration to determine the configuration mode and timing options. After configuration, these pins can be used as auxiliary connections: Mode 0 (MD0.I) and Mode 2 (MD2.I) as inputs and Mode 1 (MD1.O and MD1.T) as an output. The XACT development system will not use these resources unless they are explicitly specified in the design entry. These dedicated nets are located in the lower left chip corner and are near the readback nets. This allows convenient routing if compatibility with the XC2000 and XC3000 family conventions of M0/RT, M1/RD is desired.

**Table 6. Configuration Modes**

Mode	M2	M1	M0	CCLK	Data
Master Serial	0	0	0	output	Bit-Serial
Slave Serial	1	1	1	input	Bit-Serial
Master Parallel up	1	0	0	output	Byte-Wide, 00000 ↑
Master Parallel down	1	1	0	output	Byte-Wide, 3FFFF ↓
Peripheral Synchr.	0	1	1	input	Byte-Wide
Peripheral Asynchr.	1	0	1	output	Byte-Wide
Reserved	0	1	0	—	—
Reserved	0	0	1	—	—

Peripheral Synchronous can be considered Slave Parallel

## Configuration

Configuration is the process of loading design-specific programming data into one or more LCA devices to define the functional operation of the internal blocks and their interconnections. This is somewhat like loading the command registers of a programmable peripheral chip. The XC4000 families use about 350 bits of configuration data per CLB and its associated interconnects. Each configuration bit defines the state of a static memory cell that controls either a function look-up table bit, a multiplexer input, or an interconnect pass transistor. The XACT development system translates the design into a netlist file. It automatically partitions, places and routes the logic and generates the configuration data in PROM format.

## Modes

The XC4000 families have six configuration modes selected by a 3-bit input code applied to the M0, M1, and M2 inputs. There are three self-loading Master modes, two Peripheral modes and the Serial Slave mode used primarily for daisy-chained devices. During configuration, some of the I/O pins are used temporarily for the configuration process. See Table 6.

For a detailed description of these configuration modes, see pages 2-32 through 2-41.

## Master

The Master modes use an internal oscillator to generate CCLK for driving potential slave devices, and to generate address and timing for external PROM(s) containing the configuration data. Master Parallel (up or down) modes generate the CCLK signal and PROM addresses and receive byte parallel data, which is internally serialized into the LCA data-frame format. The up and down selection generates starting addresses at either zero or 3FFFF, to be compatible with different microprocessor addressing conventions. The Master Serial mode generates CCLK and receives the configuration data in serial form from a Xilinx serial-configuration PROM.

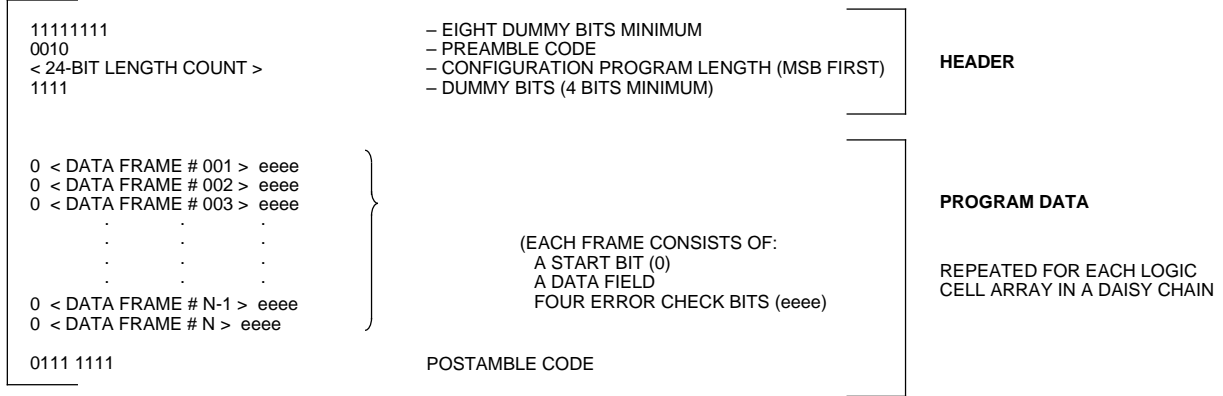
## Peripheral

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

## Serial Slave

In the Serial Slave mode, the LCA device receives serial-configuration data on the rising edge of CCLK and, after loading its configuration, passes additional data out, resynchronized on the next falling edge of CCLK. Multiple slave devices with identical configurations can be wired with parallel DIN inputs so that the devices can be configured simultaneously.

## XC4000, XC4000A, XC4000H Logic Cell Array Families



X1526

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

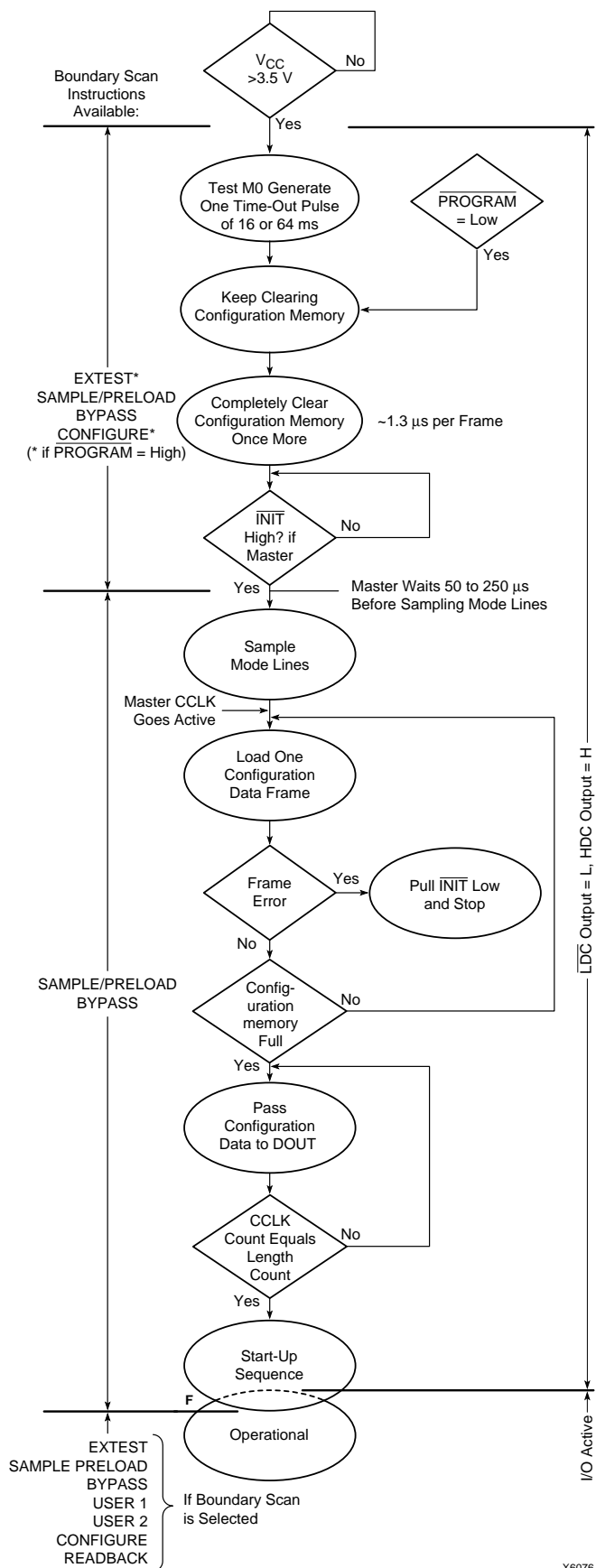
The user can add more "one" bits as leading dummy bits in the header, or, if CRC = off, as trailing dummy bits at the end of any frame, following the four error check bits, but the Length Count value **must** be adjusted for all such extra "one" bits, even for leading extra ones at the beginning of the 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



X6076

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  $\mu$ s 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 20. Start-up Sequence

tion data bits and a 4-bit frame error field. If a frame data error is detected, the LCA device halts loading, and signals the error by pulling the open-drain `INIT` pin Low.

After all configuration frames have been loaded into an LCA device, `DOUT` again follows the input data so that the remaining data is passed on to the next device.

### Start-Up

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

Figure 21 describes Start-up timing for the three Xilinx families in detail.

The **XC2000** family goes through a fixed sequence:

`DONE` goes High and the internal global Reset is de-activated one CCLK period after the I/O become active.

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

The **XC4000** family offers additional flexibility: The three events, `DONE` going High, the internal Reset/Set being de-activated, and the user I/O going active, can all occur in any arbitrary sequence, each of them one CCLK period before or after, or simultaneous with, any of the other.

The default option, and the most practical one, is for `DONE` to go High first, disconnecting the configuration data source and avoiding any contention when the I/Os become active one clock later. Reset/Set is then released another clock period later to make sure that user-operation starts from stable internal conditions. This is the most common sequence, shown with heavy lines in Figure 21, but the designer can modify it to meet particular requirements.

The XC4000 family offers another start-up clocking option: The three events described above don't have to be triggered by CCLK, they can, as a configuration option, be triggered by a user clock. This means that the device can wake up in synchronism with the user system.

The XC4000 family introduces an additional option: When this option is enabled, the user can externally hold the open-drain `DONE` output Low, and thus stall all further progress in the Start-up sequence, until `DONE` is released and has gone High. This option can be used to force synchronization of several LCA devices to a common user clock, or to guarantee that all devices are successfully configured before any I/Os go active.

### Start-up Sequence

The Start-up sequence begins when the configuration memory is full, and the total number of configuration clocks received since `INIT` went High equals the loaded value of the length count. The next rising clock edge sets a flip-flop `Q0` (see Figure 22), the leading bit of a 5-bit shift register.

The outputs of this register can be programmed to control three events.

- The release of the open-drain `DONE` output,
- The change of configuration-related pins to the user function, activating all IOBs.
- The termination of the global Set/Reset initialization of all CLB and IOB storage elements.

The `DONE` pin can also be wire-ANDed with `DONE` pins of other LCA devices or with other external signals, and can then be used as input to bit `Q3` of the start-up register. This is called “Start-up Timing Synchronous to Done In” and labeled: `CCLK_SYNC` or `UCLK_SYNC`. When `DONE` is not used as an input, the operation is called Start-up Timing Not Synchronous to `DONE` In, and is labeled `CCLK_NOSYNC` or `UCLK_NOSYNC`. These labels are not intuitively obvious.

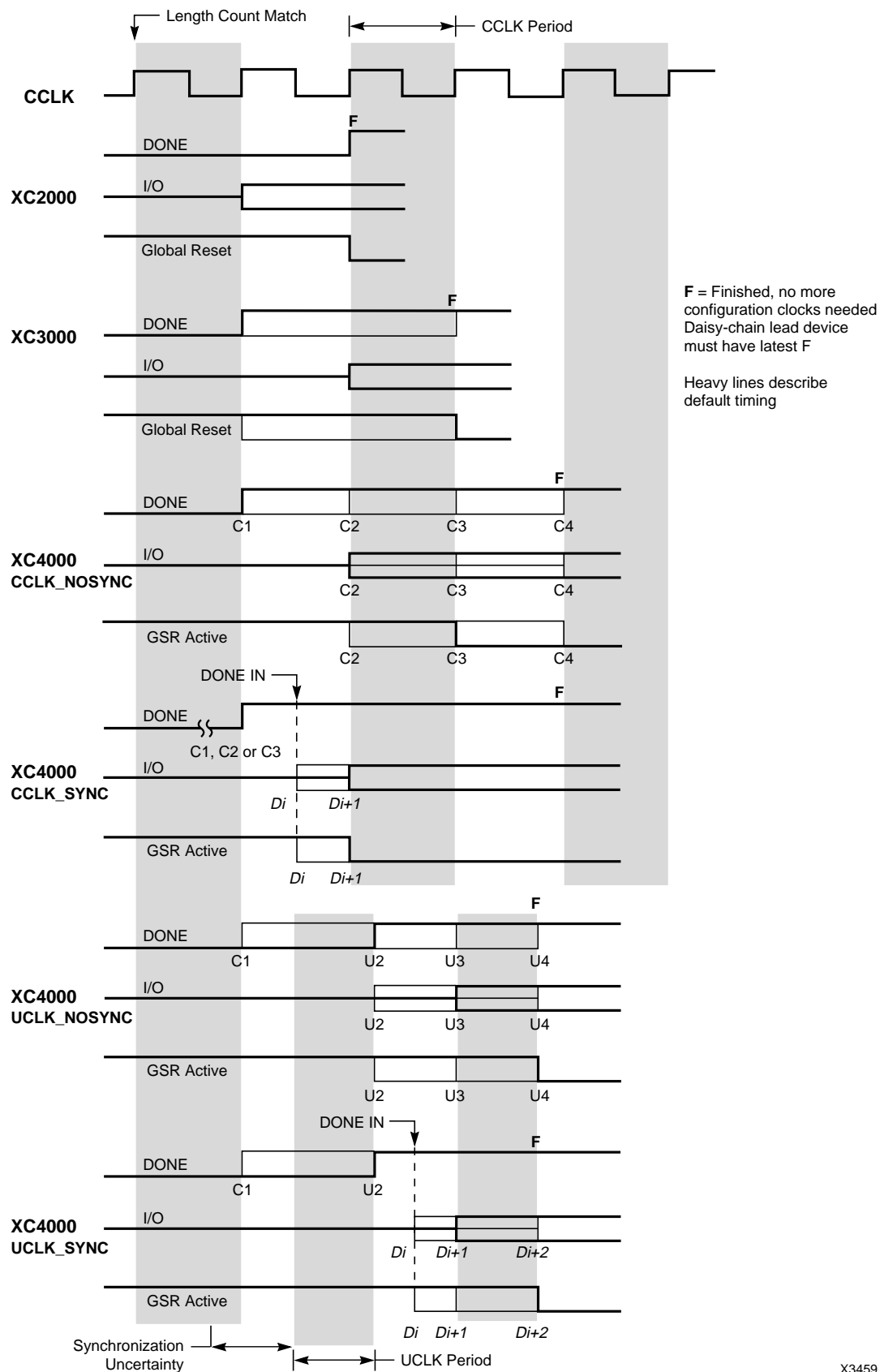
As a configuration option, the start-up control register beyond `Q0` can be clocked either by subsequent CCLK pulses or from an on-chip user net called `STARTUP.CLK`.

### Start-up from CCLK

If CCLK is used to drive the start-up, `Q0` through `Q3` provide the timing. Heavy lines in Figure 21 show the default timing which is compatible with XC2000 and XC3000 devices using early `DONE` and late Reset. The thin lines indicate all other possible timing options.

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

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



Note: Thick lines are default option.

Figure 21. Start-up Timing

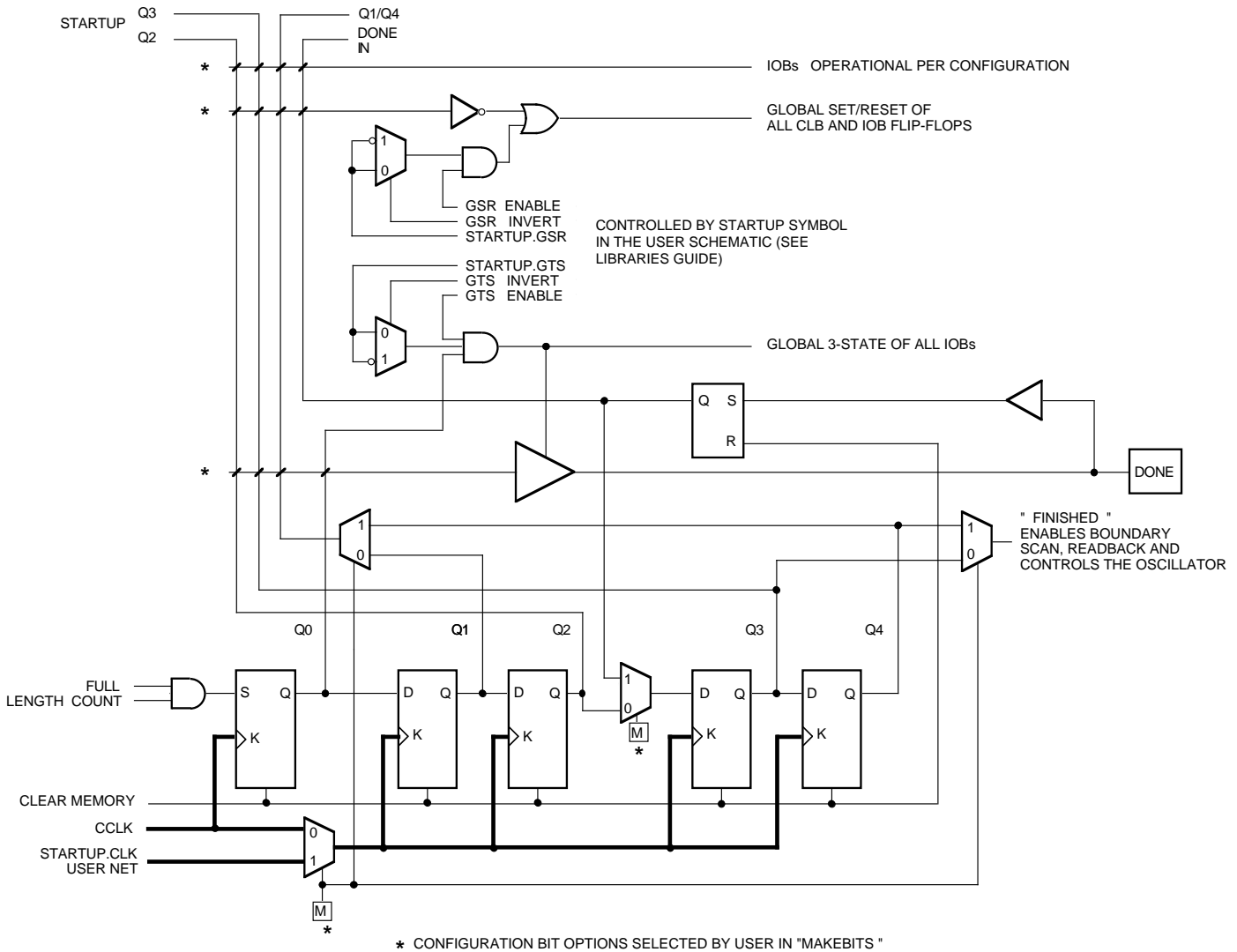


Figure 22. Start-up Logic

All Xilinx FPGAs of the XC2000, XC3000, XC4000 families 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.

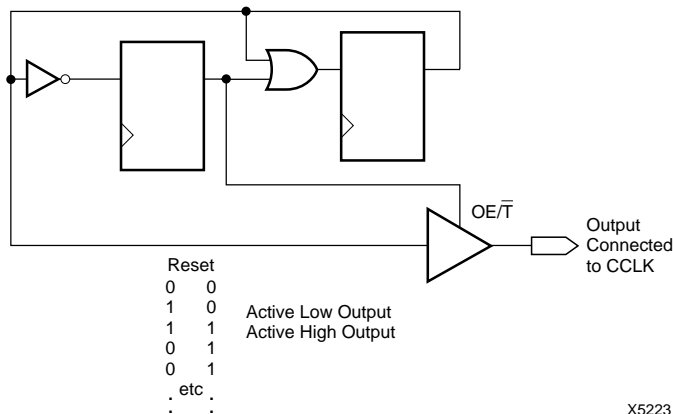
Not reaching F means that the device does not really finish its configuration, although DONE may have gone High, the

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 criticized 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).

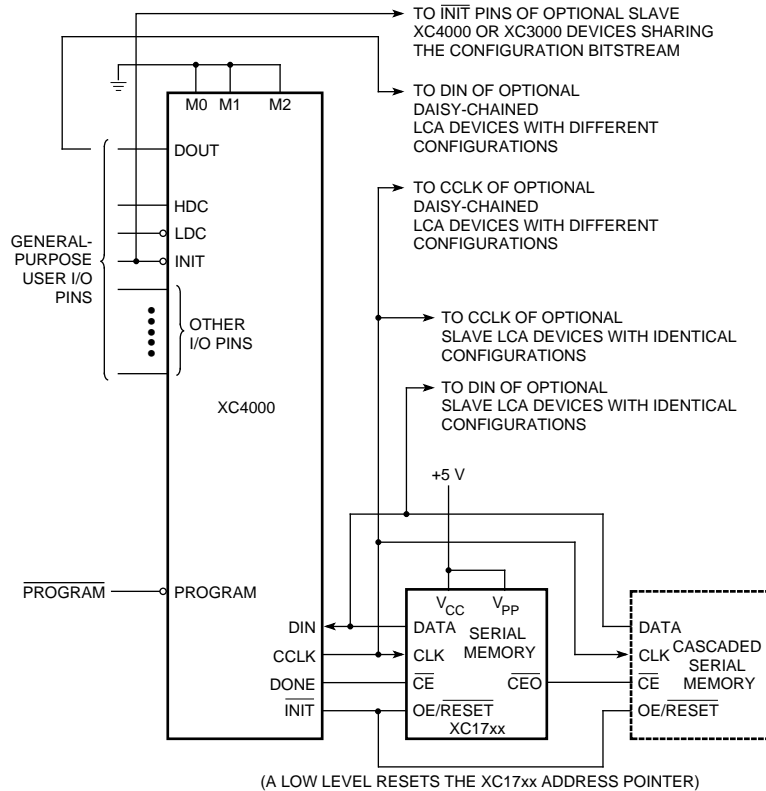
### Clock Select

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.

## Master Serial Mode



X6077

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

The lead LCA device 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 LCA device in the daisy-chain accepts data on the subsequent rising CCLK edge. The user can specify Fast ConfigRate, which starting somewhere in the first frame, increases the CCLK frequency eight times, from a value between 0.5 and 1.25 MHz, to a value between 4 and 10 MHz. Note that most Serial PROMs are not compatible with this high frequency.

The SPROM CE input can be driven from either LDC or DONE. Using LDC avoids potential contention on the DIN pin, if this pin is configured as user-I/O, but LDC is then

restricted to be a permanently High user output. Using DONE can also avoid contention on DIN, provided the early DONE option is invoked.

### 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.)

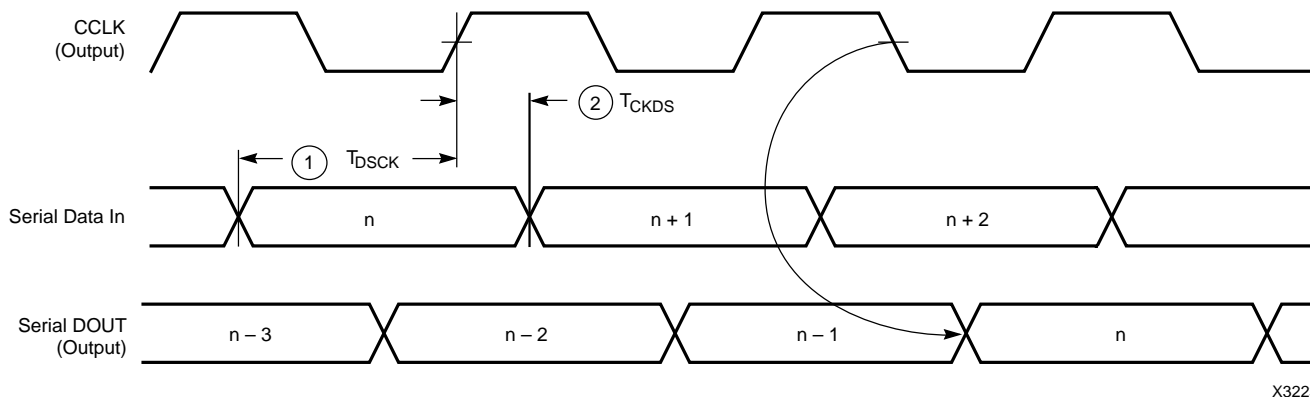
A Low on the PROGRAM input is the more radical approach, and is recommended when the power-supply rise time is excessive or poorly defined. As long as PROGRAM is Low, the XC4000 device keeps clearing its configuration memory. When PROGRAM goes High, the configuration memory is cleared one more time, 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

up to 250  $\mu$ s to make sure that all slaves in the potential daisy-chain have seen INIT being High.

### Master Serial Mode Programming Switching Characteristics

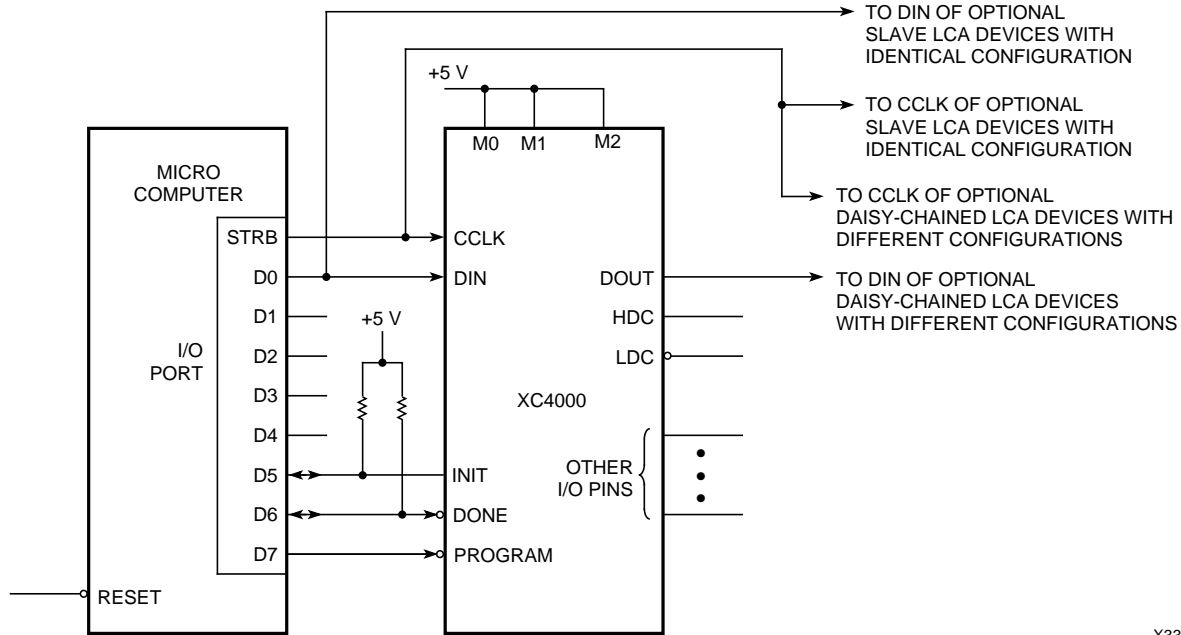


X3223

	Description	Symbol	Min	Max	Units
CCLK	Data In setup	1 $T_{DSCK}$	20		ns
	Data In hold	2 $T_{CKDS}$	0		ns

- Notes:
1. At power-up,  $V_{CC}$  must rise from 2.0 V to  $V_{CC}$  min in less than 25 ms, otherwise delay configuration by pulling PROGRAM Low until  $V_{CC}$  is valid.
  2. Configuration can be controlled by holding INIT Low with or until after the INIT of all daisy-chain slave mode devices is High.
  3. Master-serial-mode timing is based on testing in slave mode.

Slave Serial Mode



X3393

In Slave Serial mode, an external signal drives the CCLK input(s) of the LCA device(s). The serial configuration bitstream must be available at the DIN input of the lead LCA device a short set-up time before each rising CCLK edge. The lead LCA device 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 LCA device in the daisy-chain accepts data on the subsequent rising CCLK edge.

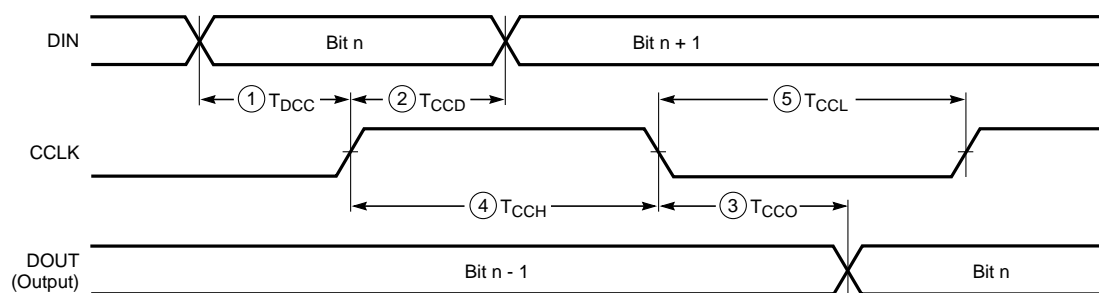
**How to Delay Configuration After Power-Up**

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.

## Slave Serial Mode Programming Switching Characteristics

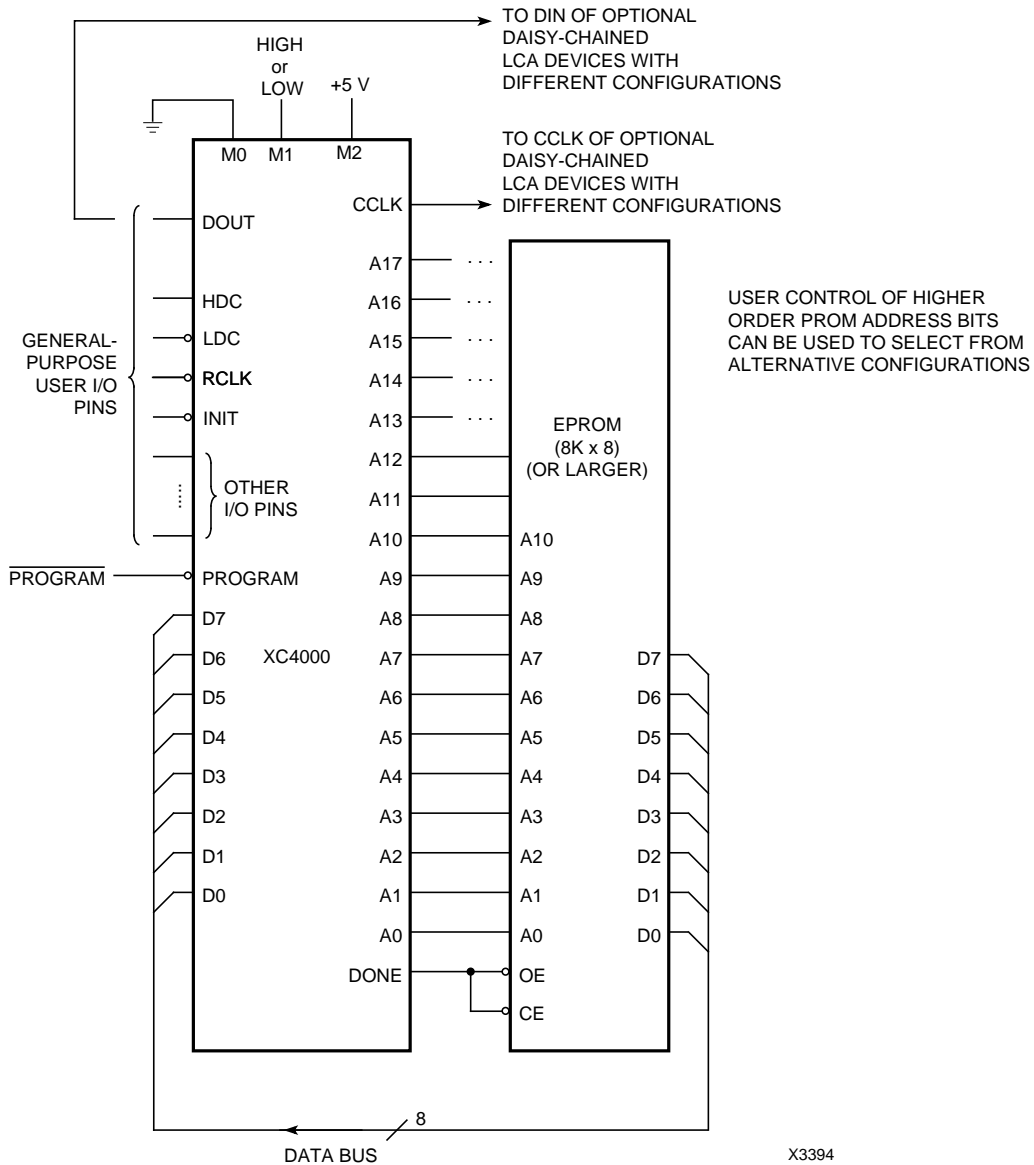


X5379

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

Note: Configuration must be delayed until the INIT of all daisy-chained LCA devices is High.

**Master Parallel Mode**



X3394

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**

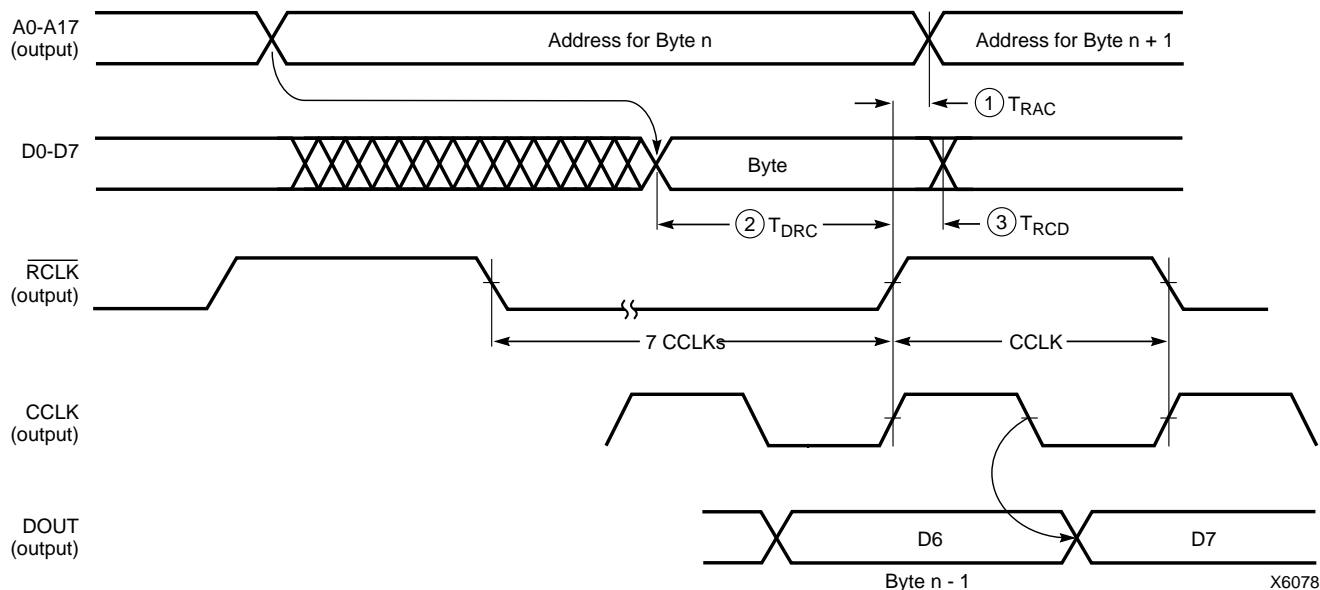
There are two methods to delay configuration after power-up: Put a logic Low on the PROGRAM input, or pull the bidirectional INIT pin Low, using an open-collector (open-drain) driver. (See also Figure 20 on page 2-27).

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  $\mu$ s to make sure that all slaves in the potential daisy-chain have seen **INIT** being High.

### Master Parallel Mode Programming Switching Characteristics

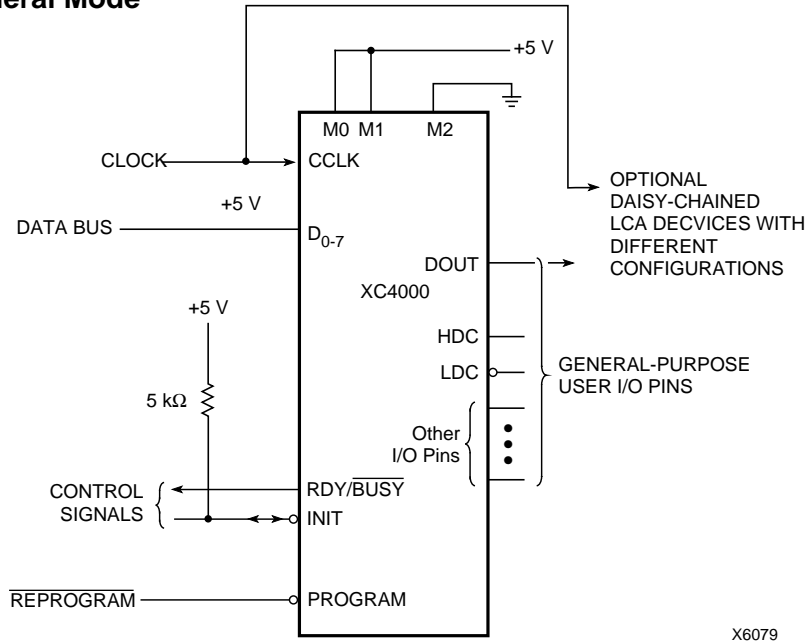


	Description	Symbol	Min	Max	Units
RCLK	Delay to Address valid	1   $T_{RAC}$	0	200	ns
	Data setup time	2   $T_{DRC}$	60		ns
	Data hold time	3   $T_{RCD}$	0		ns

- Notes:
1. At power-up,  $V_{CC}$  must rise from 2.0 V to  $V_{CC}$  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**

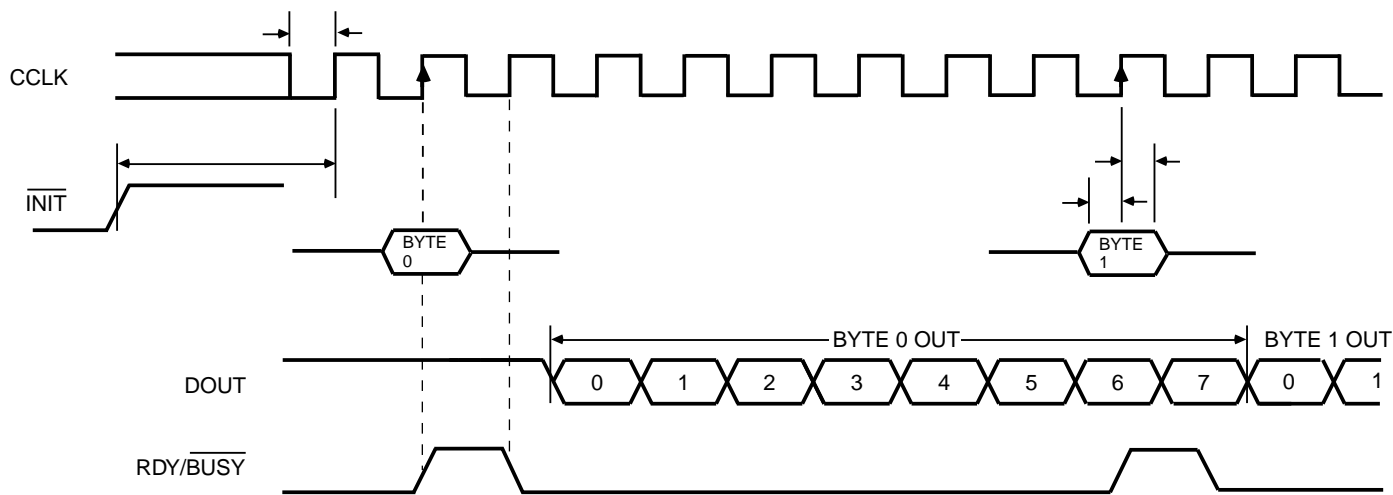
There are two methods to delay configuration after power-up: Put a logic Low on the PROGRAM input, or pull the bidirectional INIT pin Low, using an open-collector (open-drain) driver. (See also Figure 20 on page 2-27).

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.



## Synchronous Peripheral Mode Programming Switching Characteristics



X6096

	Description	Symbol	Min	Max	Units
CCLK	<u>INIT</u> (High) Setup time required	1 $T_{IC}$	5		$\mu$ s
	D0-D7 Setup time required	2 $T_{DC}$	60		ns
	D0-D7 Hold time required	3 $T_{CD}$	0		ns
	CCLK High time	$T_{CCH}$	50		ns
	CCLK Low time	$T_{CCL}$	60		ns
	CCLK Frequency	$F_{CC}$		8	MHz

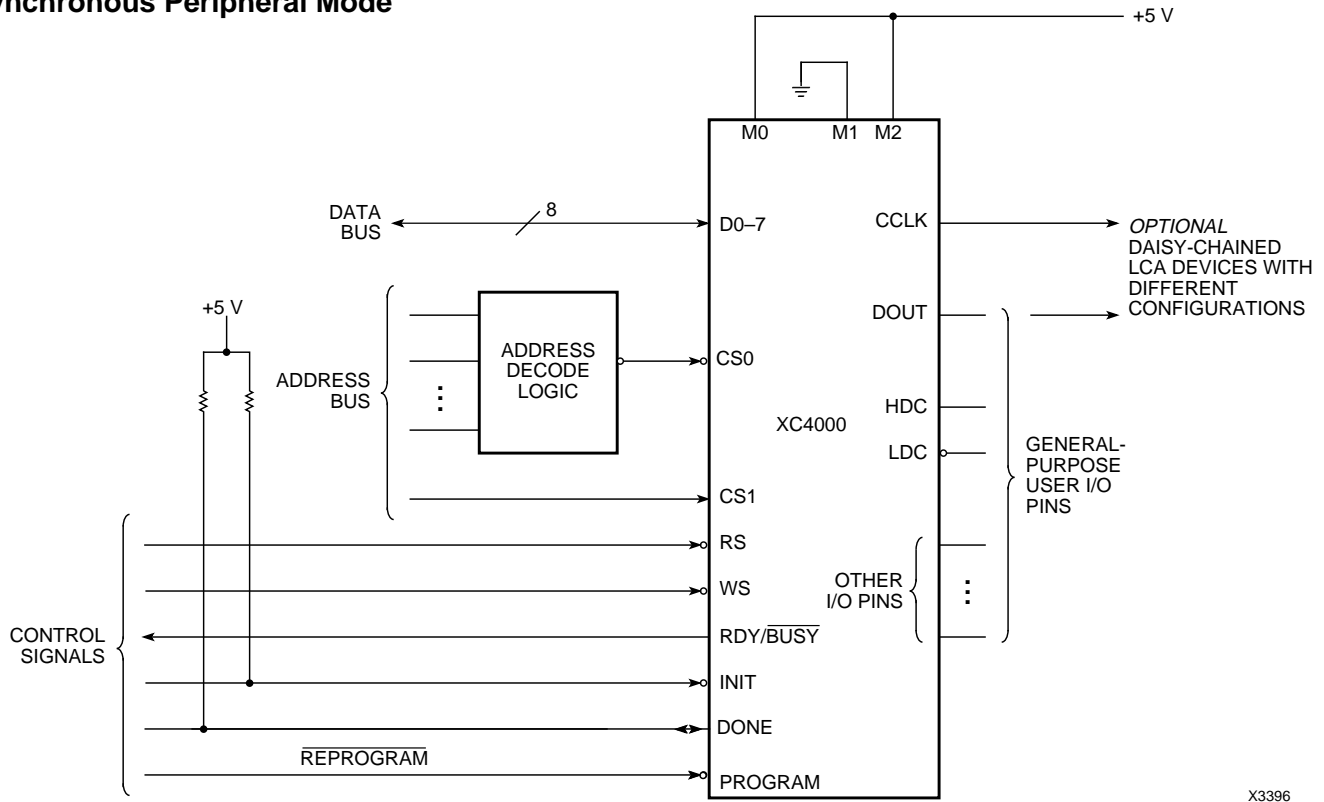
Notes: Peripheral Synchronous mode can be considered Slave Parallel mode. An external CCLK provides timing, clocking in the **first** data byte on the **second** rising edge of CCLK after INIT goes High. Subsequent data bytes are clocked in on every eighth consecutive rising edge of CCLK.

The RDY/BUSY line goes High for one CCLK period after data has been clocked in, although synchronous operation does not require such a response.

The pin name RDY/BUSY is a misnomer; in Synchronous Peripheral mode this is really an ACKNOWLEDGE signal.

Note that data starts to shift out serially on the DOUT pin 0.5 CLK periods after it was loaded in parallel. This obviously requires additional CCLK pulses after the last byte has been loaded.

## Asynchronous Peripheral Mode



X3396

### 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 RDY/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  $\mu$ s.

### Status Read

The logic AND condition of the CS0, CS1 and RS inputs puts the device status on the Data bus.

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

It is mandatory that the whole start-up sequence be started and completed by one byte-wide input. Otherwise, the pins used as Write Strobe or Chip Enable might become active outputs and interfere with the final byte transfer. If this transfer does not occur, the start-up sequence 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.

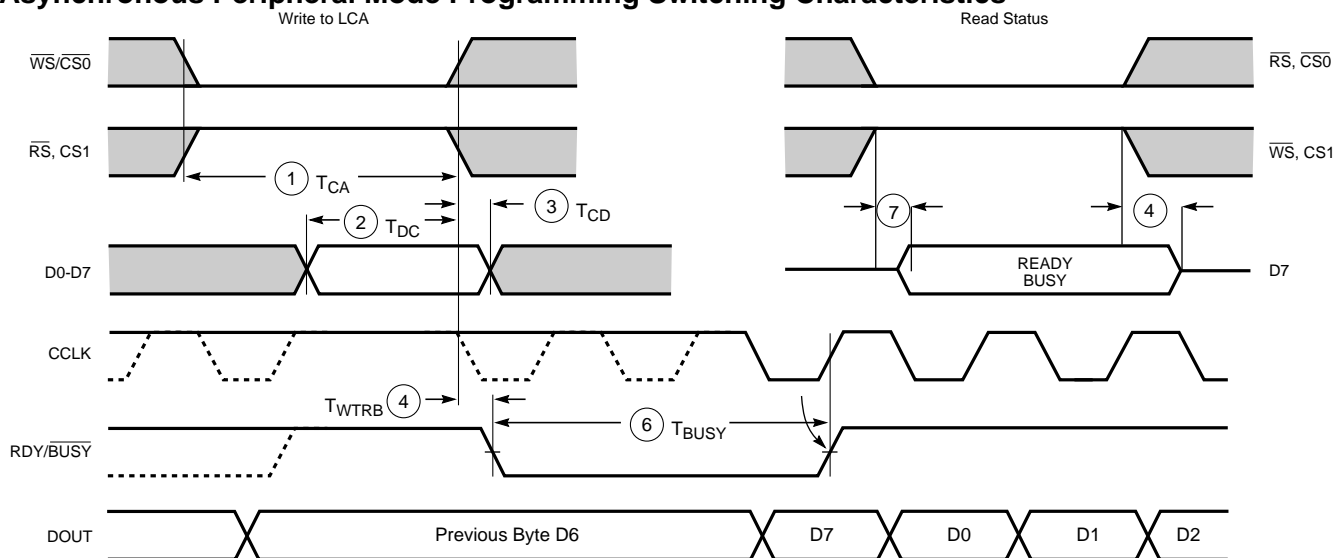
### How to Delay Configuration After Power-Up

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).

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  $\mu$ s to make sure that all slaves in the potential daisy-chain have seen INIT being High.

### Asynchronous Peripheral Mode Programming Switching Characteristics



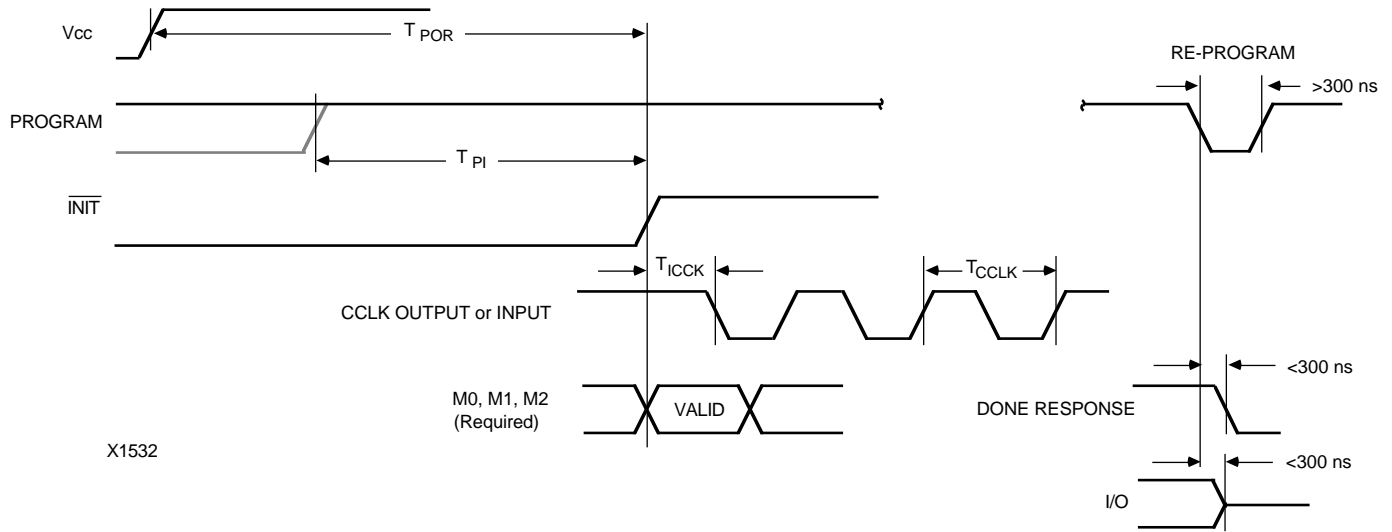
X6097

	Description	Symbol	Min	Max	Units
Write	Effective Write time required ( <u>CS0</u> , <u>WS</u> = Low, <u>RS</u> , <u>CS1</u> = High)	1   $T_{CA}$	100		ns
RDY	DIN Setup time required	2   $T_{DC}$	60		ns
	DIN Hold time required	3   $T_{CD}$	0		ns
	<u>RDY/BUSY</u> delay after end of Write or Read	4   $T_{WTRB}$		60	ns
	<u>RDY/BUSY</u> active after beginning of Read	7		60	ns
RDY	Earliest next <u>WS</u> after end of <u>BUSY</u>	5   $T_{RBWT}$	0		ns
	<u>BUSY</u> Low output (Note 4)	6   $T_{BUSY}$	2	9	CCLK Periods

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

***This timing diagram shows very relaxed requirements:  
Data need not be held beyond the rising edge of WS. BUSY will go active within 60 ns after the end of WS.  
WS may be asserted immediately after the end of BUSY.***

**General LCA Switching Characteristics**



X1532

**Master Modes**

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

**Slave and Peripheral Modes**

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

Note: At power-up, V<sub>CC</sub> must rise from 2.0 V to V<sub>CC</sub> min in less than 25 ms, otherwise delay configuration using PROGRAM until V<sub>CC</sub> 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
* INIT-ERROR	INIT-ERROR	INIT-ERROR	INIT-ERROR	INIT-ERROR	INIT-ERROR	I/O
						SGI-I/O
DONE	DONE	DONE	DONE	DONE	DONE	DONE
PROGRAM (I)	PROGRAM (I)	PROGRAM (I)	PROGRAM (I)	PROGRAM (I)	PROGRAM (I)	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
			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
			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
		RDY/BUSY	RDY/BUSY	RCLK	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)
			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.**

## Pin Descriptions

### Permanently Dedicated Pins

#### $V_{CC}$

Eight or more (depending on package type) connections to the nominal +5 V supply voltage. All must be connected.

---

#### GND

Eight or more (depending on package type) connections to ground. All must be connected.

---

#### CCLK

During configuration, Configuration Clock is an output of the LCA in Master modes or asynchronous Peripheral mode, but is an input to the LCA in Slave mode and Synchronous Peripheral mode.

After configuration, CCLK has a weak pull-up resistor and can be selected as Readback Clock.

---

#### DONE

This is a bidirectional signal, configurable with or without a pull-up resistor of 2 to 8 k $\Omega$ .

As an output, it indicates the completion of the configuration process. The configuration program determines the exact timing, the clock source for the Low-to-High transition, and enable of the pull-up resistor.

As an input, a Low level on DONE can be configured to delay the global logic initialization or the enabling of outputs

---

#### PROGRAM

This is an active Low input that forces the LCA to clear its configuration memory.

When PROGRAM goes High, the LCA finishes the current clear cycle and executes another complete clear cycle, before it goes into a WAIT state and releases INIT.

---

### User I/O Pins that can have Special Functions

#### RDY/BUSY

During peripheral modes, this pin indicates when it is appropriate to write another byte of data into the LCA device. The same status is also available on D7 in asynchronous peripheral mode, if a read operation is performed when the device is selected. After configuration, this is a user-programmable I/O pin.

---

#### RCLK

During Master Parallel configuration, each change on the A0-15 outputs is preceded by a rising edge on RCLK, a redundant output signal. After configuration, this is a user-programmable I/O pin.

---

#### M0, M1, M2

As Mode inputs, these pins are sampled before the start of configuration to determine the configuration mode to be used.

After configuration, M0 and M2 can be used as inputs, and M1 can be used as a 3-state output. These three pins have no associated input or output registers.

These pins can be user inputs or outputs only when called out by special schematic definitions.

---

#### TDO

If boundary scan is used, this is the Test Data Output.

If boundary scan is not used, this pin is a 3-state output without a register, after configuration is completed.

This pin can be user output only when called out by special schematic definitions.

---

#### TDI, TCK, TMS

If boundary scan is used, these pins are Test Data In, Test Clock, and Test Mode Select inputs respectively coming directly from the pads, bypassing the IOBs. These pins can also be used as inputs to the CLB logic after configuration is completed.

If the boundary scan option is not selected, all boundary scan functions are inhibited once configuration is completed, and these pins become user-programmable I/O.

### Note:

The XC4000 families have no Powerdown control input; use the global 3-state net instead.

The XC4000 families have no dedicated Reset input. Any user I/O can be configured to drive the global Set/Reset net.

---

### **HDC**

High During Configuration is driven High until configuration is completed. It is available as a control output indicating that configuration is not yet completed. After configuration, this is a user-programmable I/O pin.

---

### **LDC**

Low During Configuration is driven Low until configuration. It is available as a control output indicating that configuration is not yet completed. After configuration, this is a user-programmable I/O pin.

---

### **INIT**

Before and during configuration, this is a bidirectional signal. An external pull-up resistor is recommended.

As an active-Low open-drain output, INIT is held Low during the power stabilization and internal clearing of the configuration memory. As an active-Low input, it can be used to hold the LCA device in the internal WAIT state before the start of configuration. Master mode devices stay in a WAIT state an additional 30 to 300  $\mu$ s after INIT has gone High.

During configuration, a Low on this output indicates that a configuration data error has occurred. After configuration, this is a user-programmable I/O pin.

---

### **PGCK1 - PGCK4**

Four Primary Global Inputs each drive a dedicated internal global net with short delay and minimal skew. If not used for this purpose, any of these pins is a user-programmable I/O.

---

### **SGCK1 - SGCK4**

Four Secondary Global Inputs can each drive a dedicated internal global net, that alternatively can also be driven from internal logic. If not used for this purpose, any of these pins is a user-programmable I/O pin.

---

### **CS0, CS1, WS, RS**

These four inputs are used in Peripheral mode. The chip is selected when CS0 is Low and CS1 is High. While the chip is selected, a Low on Write Strobe (WS) loads the data present on the D0 - D7 inputs into the internal data buffer; a Low on Read Strobe (RS) changes D7 into a status output: High if Ready, Low if Busy, and D0...D6 are active Low. WS and 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**

During Master Parallel mode, these 18 output pins address the configuration EPROM. After configuration, these are user-programmable I/O pins.

---

### **D0 - D7**

During Master Parallel and Peripheral configuration modes, these eight input pins receive configuration data. After configuration, they are user-programmable I/O pins.

---

### **DIN**

During Slave Serial or Master Serial configuration modes, this is the serial configuration data input receiving data on the rising edge of CCLK.

During parallel configuration modes, this is the D0 input. After configuration, DIN is a user-programmable I/O pin.

---

### **DOUT**

During configuration in any mode, this is the serial configuration data output that can drive the DIN of daisy-chained slave LCA devices. DOUT data changes on the falling edge of CCLK, one-and-a-half CCLK periods after it was received at the DIN input. After configuration, DOUT is a user-programmable I/O pin.

---

### **Unrestricted User-Programmable I/O Pins**

#### **I/O**

A pin that 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 that defines the logic level as High.

## XC4000, XC4000A, XC4000H Logic Cell Array Families

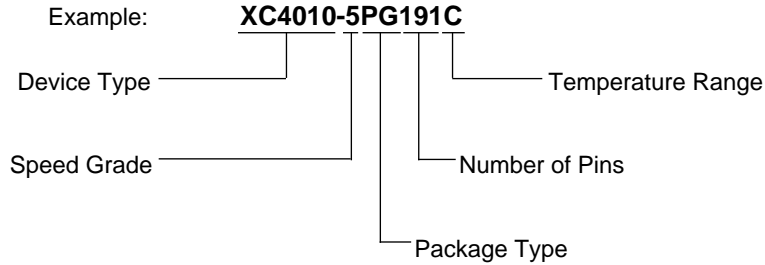
For a detailed description of the device architecture, see page 2-9 through 2-31.

For a detailed description of the configuration modes and their timing, see pages 2-32 through 2-55.

For detailed lists of package pinouts, see pages 2-57 through 2-67, 2-70, 2-81 through 2-85, and 2-100 through 2-101.

For package physical dimensions and thermal data, see Section 4.

### Ordering Information



### Component Availability

PINS	84		100		120	144	156	160	164	191	196	208		223	225	240		299	304
	PLAST. PLCC	PLAST. PQFP	PLAST. VQFP	TOP BRAZED CQFP	CERAM. PGA	PLAST. TQFP	CERAM. PGA	PLAST. PQFP	TOP BRAZED CQFP	CERAM. PGA	TOP BRAZED CQFP	PLAST. PQFP	METAL PQFP	CERAM. PGA	PLAST. BGA	PLAST. PQFP	METAL PQFP	METAL PQFP	HI QUAD
CODE	PC84	PQ100	VQ100	CB100	PG120	TQ144	PG156	PQ160	CB164	PG191	CB196	PQ208	MQ208	PG223	BG225	PQ240	MQ240	PG299	HQ304
XC4003	-6 -5 -4	C I C C	C I C C		C I C C														
XC4005	-10 -6 -5 -4	C I C I C	C I C I C				M B C I M B C	C I C C	M B M B			C I C I C							
XC4006	-6 -5 -4	C I C I C					C I C I C	C I C I C				C I C I C							
XC4008	-6 -5 -4	C I C I C					C I C I C			C I C I C		C I C I C	C I C I C						
XC4010	-10 -6 -5 -4	C I C I C					C I C I C			M B C I M B C	M B M B	C I C I C	C I C I C		C I C I C				
XC4010D	-6 -5 -4	C I C I					C I C I C					C I C I C			C I C I C				
XC4013	-6 -5 -4						C I C I C					C I C I C	C I C I C	C I (M B) C C	C I C C	C I C I C	C I C I C		
XC4013D	-6 -5 -4						C I C I C					C I C I C			C I C I C	C I C I C			
XC4020	-6 -5 -4											(C I) (C I) (C)		(C I) (C I) (C)		(C I) (C I) (C)		(C I) (C I) (C)	
XC4025	-6 -5 -4													C (C)		C I C I	C I C I	C I C I	C I C I
XC4002A	-6 -5 -4	C I C C	C I C C	C I C C	C I C C														
XC4003A	-10 -6 -5 -4	C I C C	C I C C	C I C C	M B M B C I M B														
XC4004A	-6 -5 -4	C I C			C I C	C I C		C I C											
XC4005A	-6 -5 -4	C I C I C				C I C I C	C I C I C	C I C I C				C I C I C							
XC4003H	-6 -5									C I C		C I C							
XC4005H	-6 -5													C I C		C I C	C I C		

C = Commercial = 0° to +85° C I = Industrial = -40° to +100° C M = Mil Temp = -55° to +125° C  
 B = MIL-STD-883C Class B Parentheses indicate future product plans