Welcome to E-XFL.COM

Understanding Embedded - FPGAs (Field Programmable Gate Array)

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

Applications of Embedded - FPGAs

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

Details	
Product Status	Obsolete
Number of LABs/CLBs	100
Number of Logic Elements/Cells	238
Total RAM Bits	3200
Number of I/O	61
Number of Gates	3000
Voltage - Supply	4.75V ~ 5.25V
Mounting Type	Surface Mount
Operating Temperature	0°C ~ 85°C (TJ)
Package / Case	84-LCC (J-Lead)
Supplier Device Package	84-PLCC (29.31x29.31)
Purchase URL	https://www.e-xfl.com/product-detail/xilinx/xc4003-6pc84c

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

Speed Is Enhanced Two Ways

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

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

Improved Architecture

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

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

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

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

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

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

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

Table 3. Density and Performance for Several Common Circuit Functions

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

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

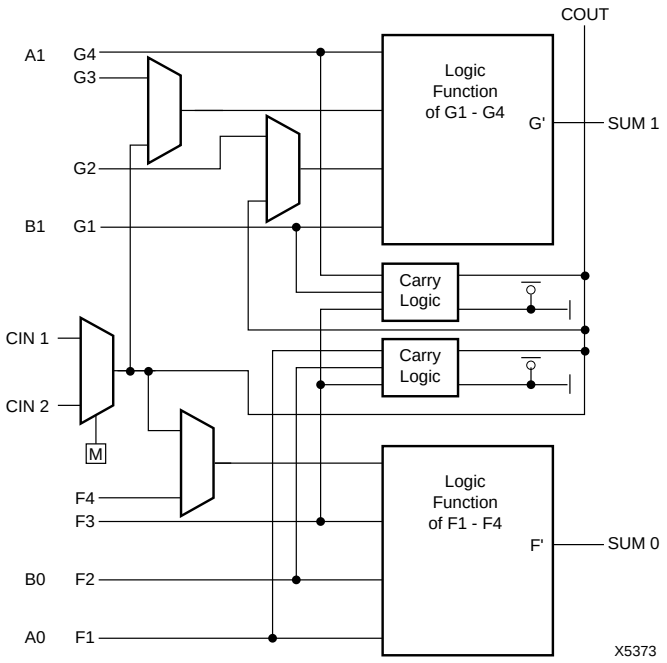


Figure 2. Fast Carry Logic in Each CLB

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.

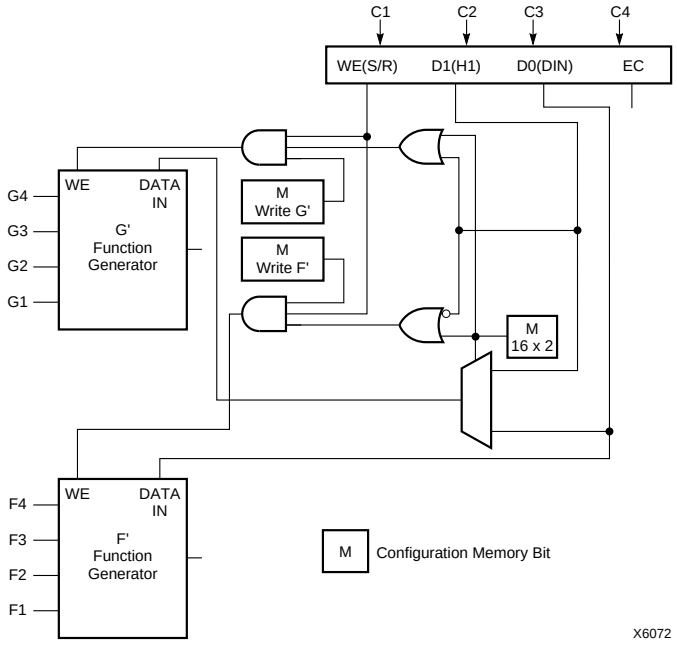


Figure 3. CLB Function Generators Can Be Used as Read/Write Memory Cells

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

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

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

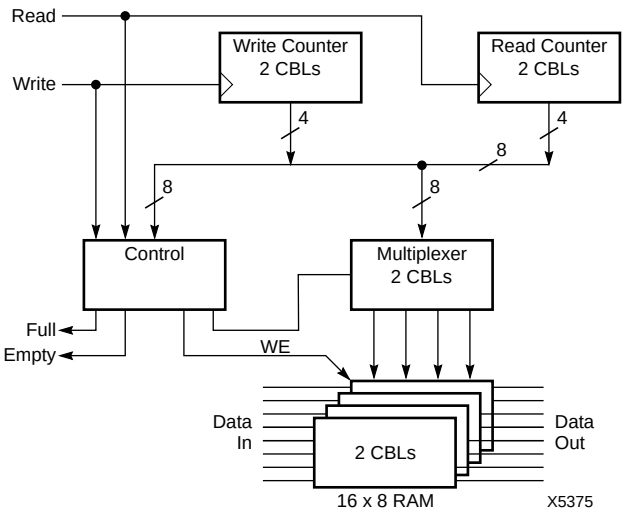


Figure 4. 16-byte FIFO

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 predetermined 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.

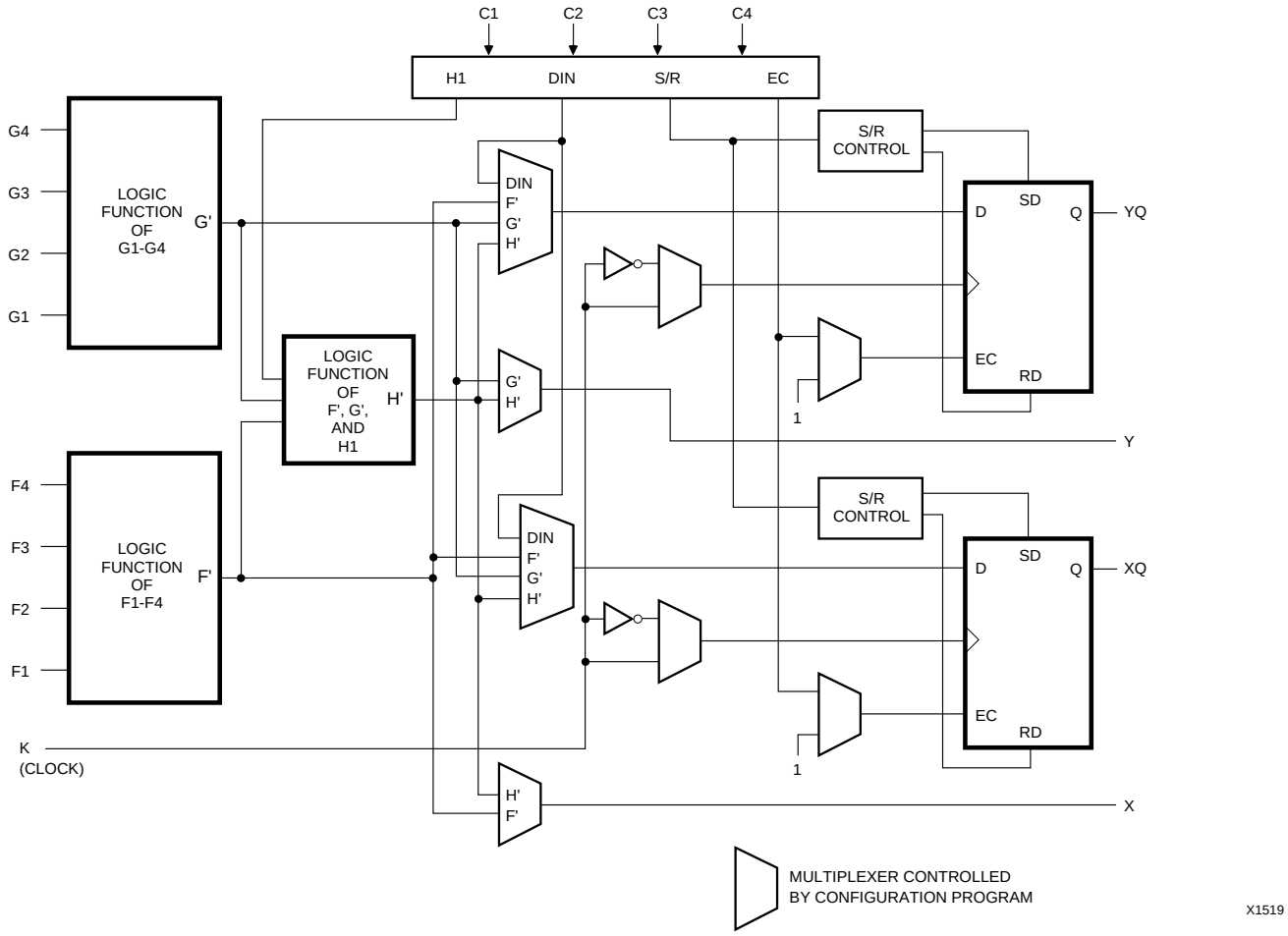


Figure 13. Simplified Block Diagram of XC4000 Configurable Logic Block

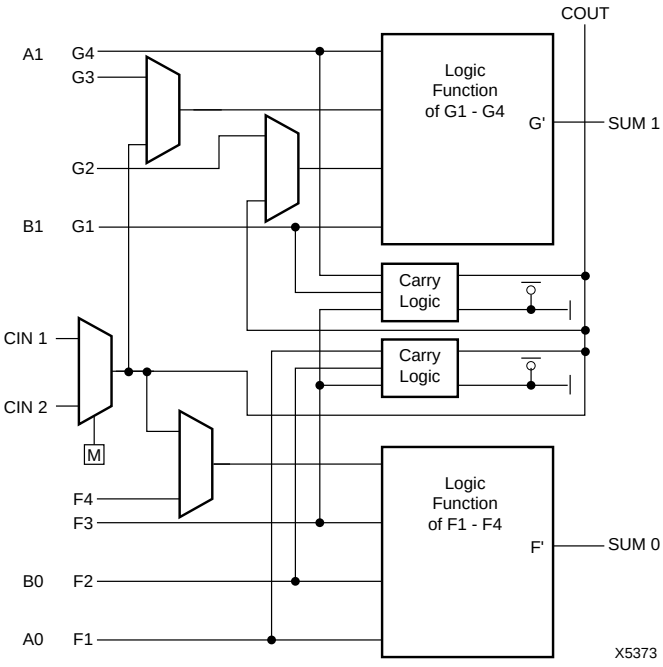


Figure 14. Fast Carry Logic in Each CLB

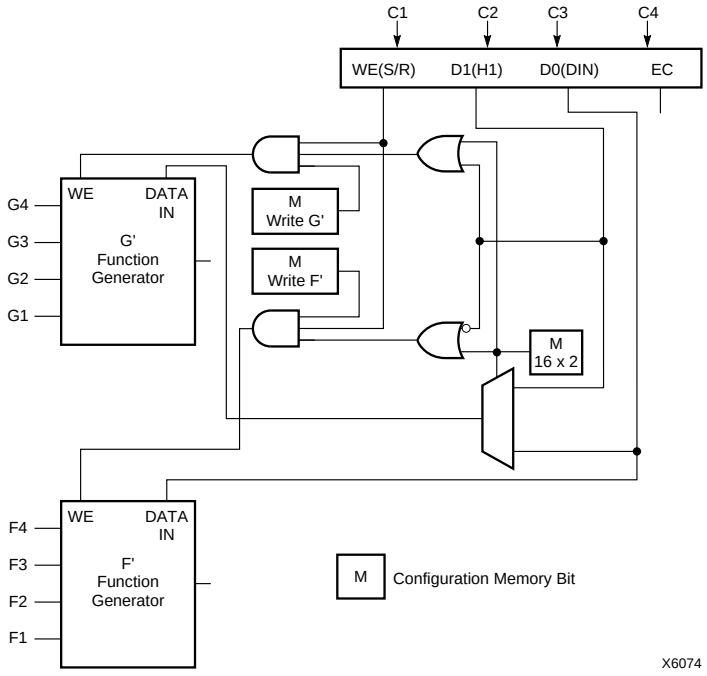


Figure 15. CLB Function Generators Can Be Used as Read/Write Memory Cells

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 overdrive 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

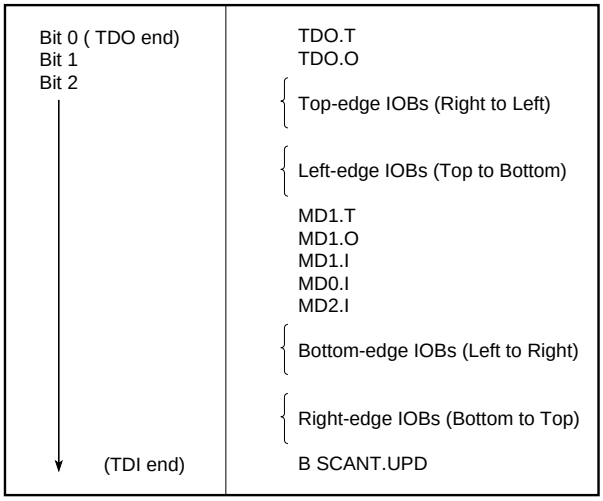
Instruction I_2	I_1	I_0	Test Selected	TDO Source	I/O Data Source
0	0	0	Extest	DR	DR
0	0	1	Sample/Preload	DR	Pin/Logic
0	1	0	User 1	TDO1	Pin/Logic
0	1	1	User 2	TDO2	Pin/Logic
1	0	0	Readback	Readback Data	Pin/Logic
1	0	1	Configure	DOUT	Disabled
1	1	0	Reserved	—	—
1	1	1	Bypass	Bypass Reg	Pin/Logic

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*.

Interconnects

The XC4000 families use a hierarchy of interconnect resources.

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

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

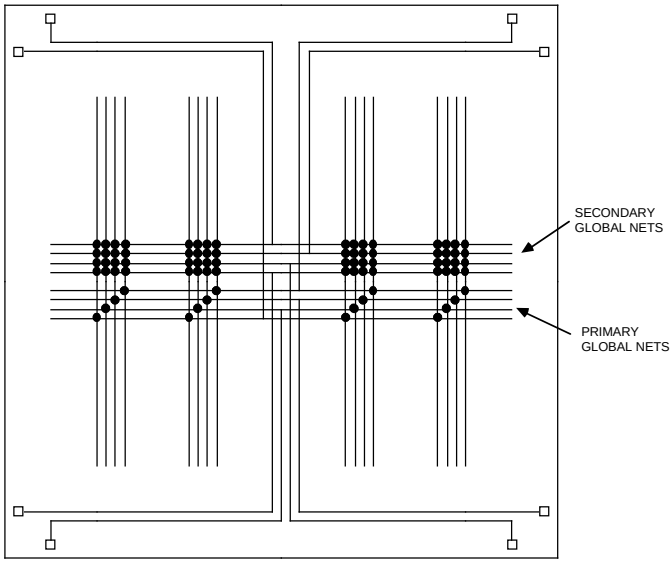
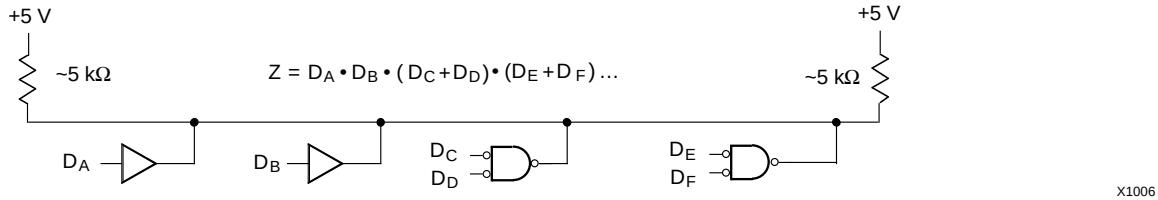
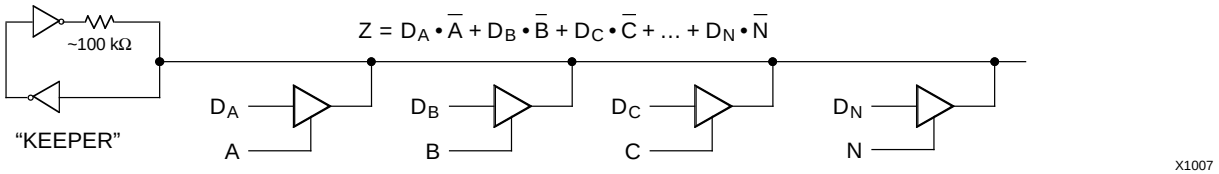


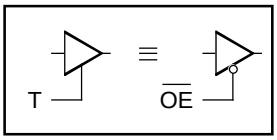
Figure 17. XC4000 Global Net Distribution. Four Lines per Column; Eight Inputs in the Four Chip Corners.



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



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



Active High T is Identical to Active Low Output Enable.

Figure 18. TBUFs Driving Horizontal Longlines.

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

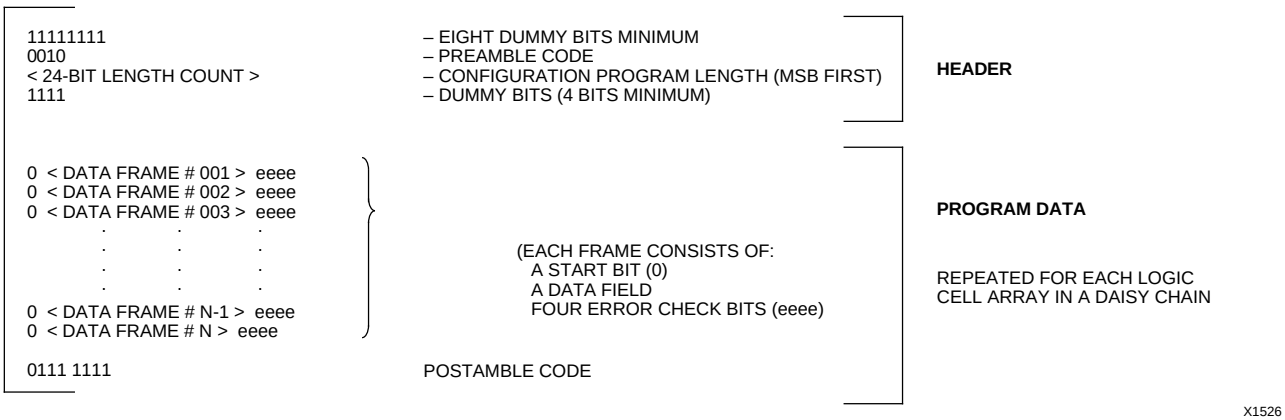
The Master modes use an internal oscillator to generate CCLK for driving potential slave devices, and to generate address and timing for external PROM(s) containing the configuration data. Master Parallel (up or down) modes generate the CCLK signal and PROM addresses and receive byte parallel data, which 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.



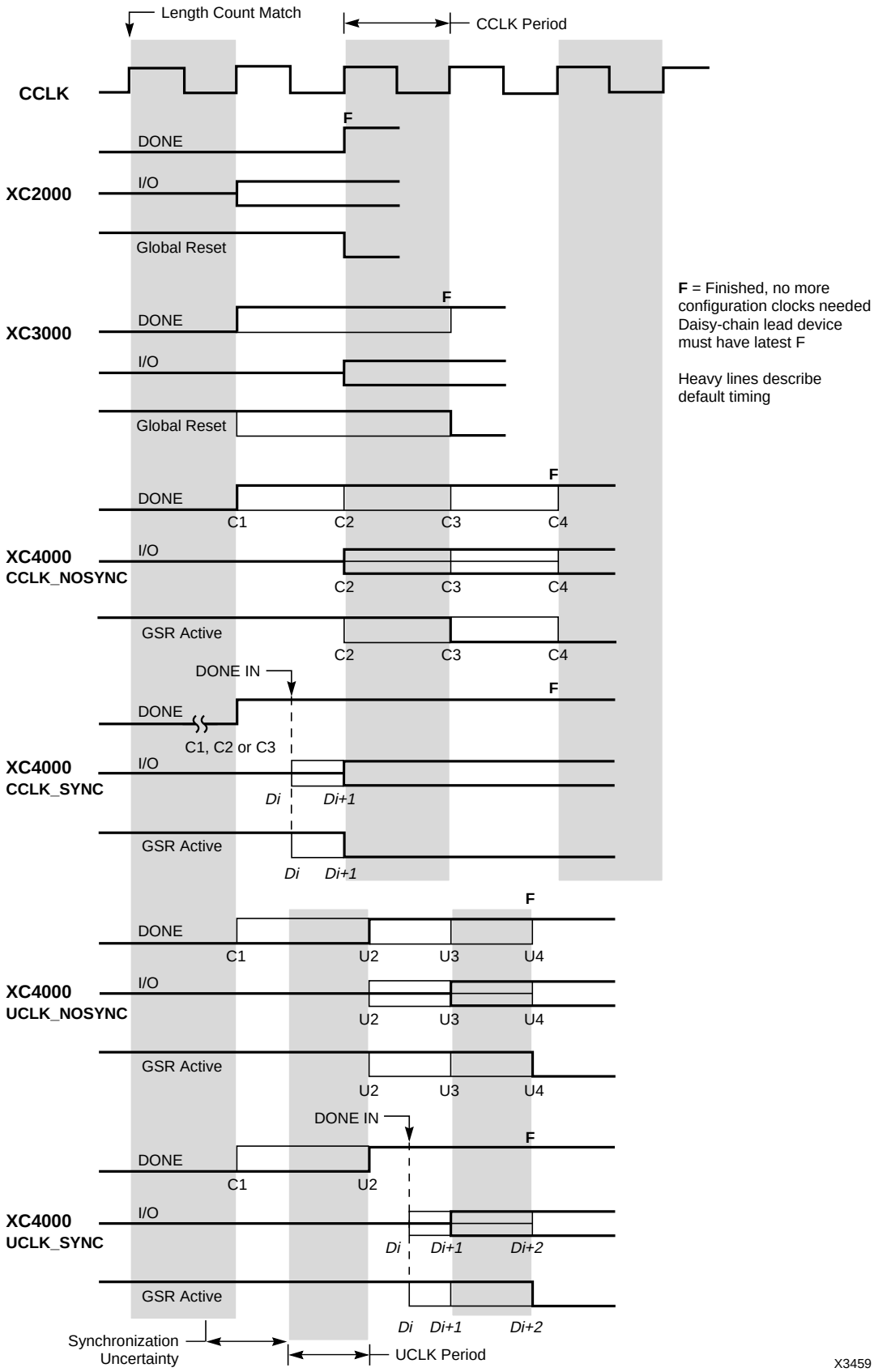
Device	XC4002A	XC4003A	XC4003/H	XC4004A	XC4005A	XC4005/H	XC4006	XC4008	XC4010/D	XC4013/D	XC4020	XC4025
Gates	2,000	3,000	3,000	4,000	5000	5,000	6,000	8,000	10,000	13,000	20,000	25,000
CLBs (Row x Col)	64 (8 x 8)	100 (10 x 10)	100 (10 x 10)	144 (12 x 12)	196 (14 x 14)	196 (14 x 14)	256 (16 x 16)	324 (18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

XC4000, 4000H: Bits per Frame = (10 x number of Rows) + 7 for the top + 13 for the bottom + 1 + 1 start bit + 4 error check bits
Number of Frames = (36 x number of Columns) + 26 for the left edge + 41 for the right edge + 1
XC4000A: Bits per Frame = (10 x number of Rows) + 6 for the top + 10 for the bottom + 1 + 1 start bit + 4 error check bits
Number of Frames = (32 x number of Columns) + 21 for the left edge + 32 for the right edge + 1
Program Data = (Bits per Frame x Number of Frames) + 8 postamble bits
PROM Size = Program Data + 40
The user can add more "one" bits as leading dummy bits in the header, or, if CRC = off, as trailing dummy bits at the end of any frame, following the four error check bits, but the Length Count value **must** be adjusted for all such extra "one" bits, even for leading extra ones at the beginning of the header.

Figure 19. Internal Configuration Data Structure.

Format

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



Note: Thick lines are default option.

Figure 21. Start-up Timing

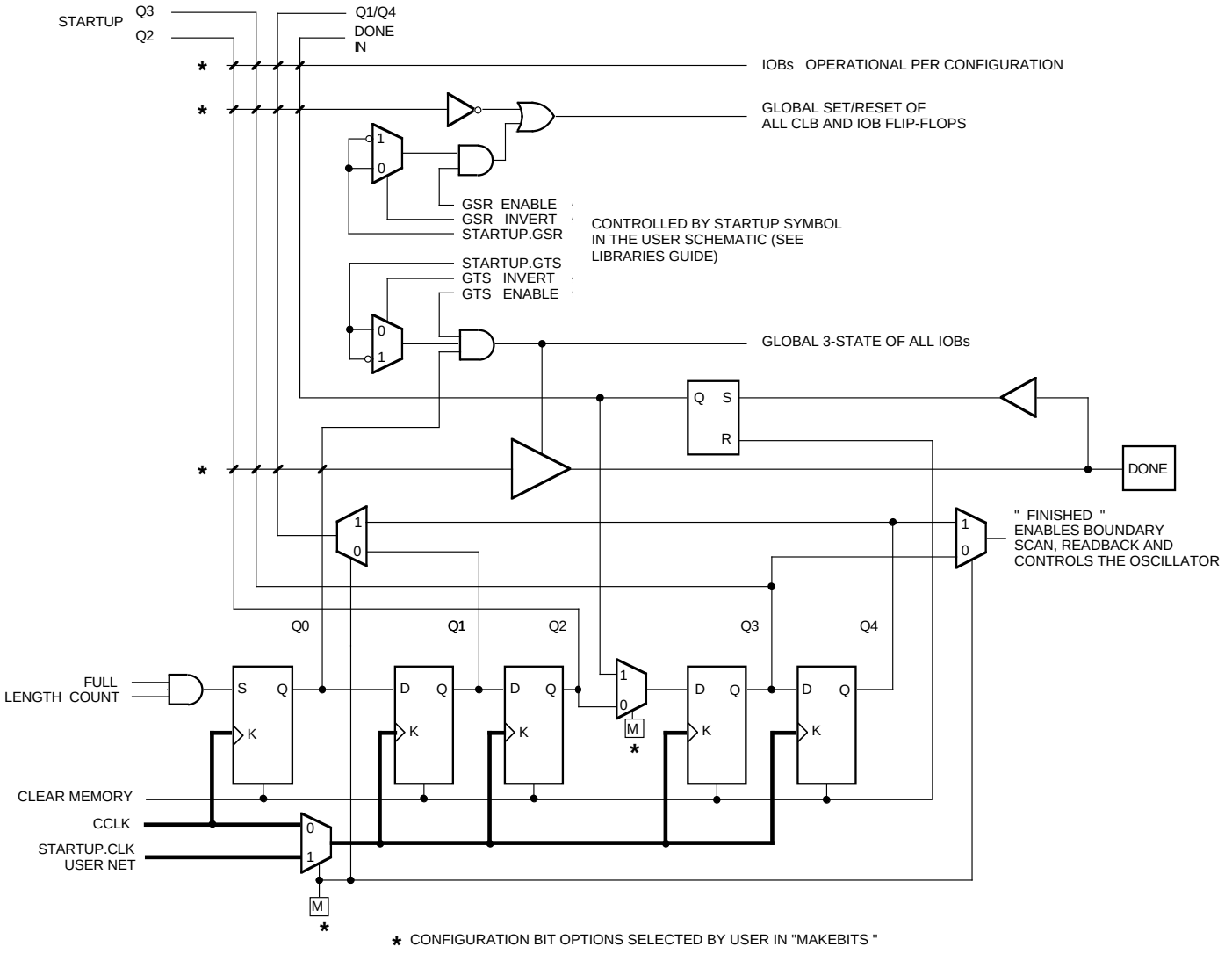


Figure 22. Start-up Logic

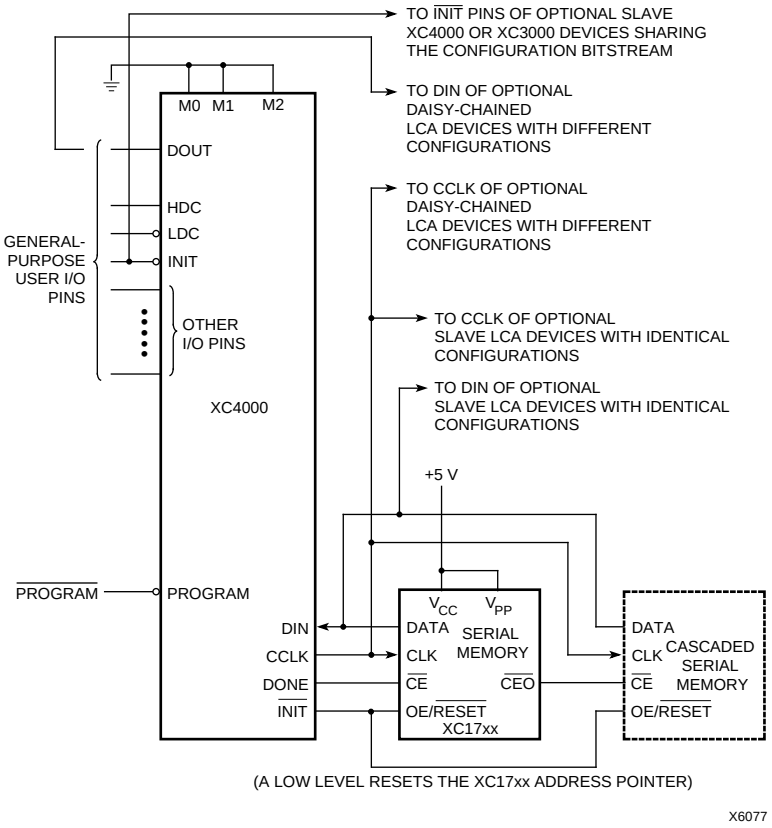
All Xilinx FPGAs of the XC2000, XC3000, XC4000 familiies use a compatible bitstream format and can, therefore, be connected in a daisy-chain in an arbitrary sequence. There is however one limitation. The lead device must belong to the highest family in the chain. If the chain contains XC4000 devices, the master cannot be an XC2000 or XC3000 device; if the daisy-chain contains XC3000 devices, the master cannot be an XC2000 device. The reason for this rule is shown in Figure 21 on the previous page. Since all devices in the chain store the same length count value and generate or receive one common sequence of CCLK pulses, they all recognize length-count match on the same CCLK edge, as indicated on the left edge of Figure 21. The master device will then drive additional CCLK pulses until it reaches its finish point F. The different families generate or require different numbers of additional CCLK pulses until they reach F.

Not reaching F means that the device does not really finish its configuration, although DONE may have gone High, the outputs became active, and the internal RESET was released. The user has some control over the relative timing of these events and can, therefore, make sure that they occur early enough.

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

One CLB and one IOB in the lead XC3000 device are used to generate the additional CCLK pulse required by the XC4000 devices. When the lead device removes the internal RESET signal, the 2-bit shift register responds to its clock input and generates an active Low output signal for the duration of the subsequent clock period. An external connection between this output and CCLK thus creates

Master Serial Mode



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

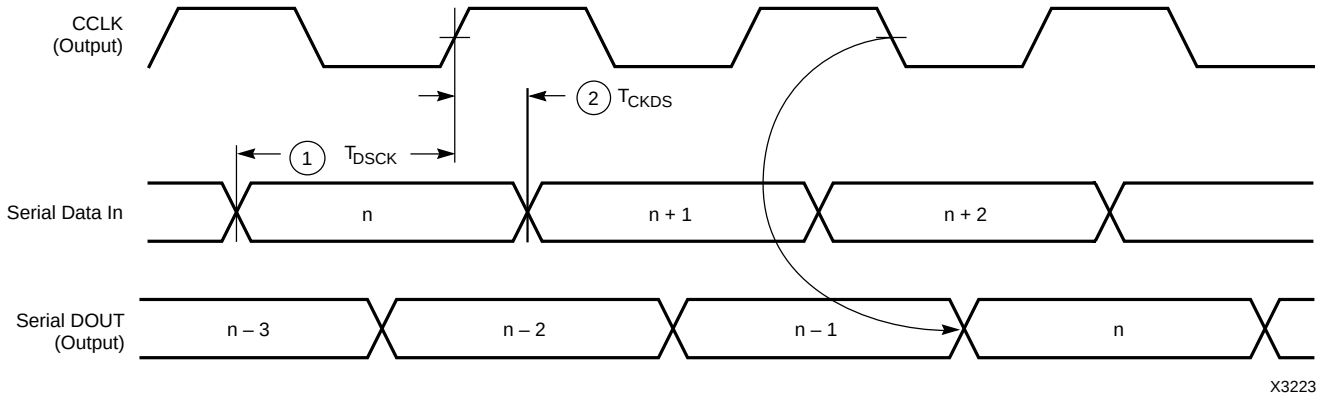
There are two methods to delay configuration after power-up: Put a logic Low on the PROGRAM input, or pull the bidirectional INIT pin Low, using an open-collector (open-drain) driver. (See also Figure 20 on page 2-27.)

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 µs to make sure that all slaves in the potential daisy-chain have seen INIT being High.

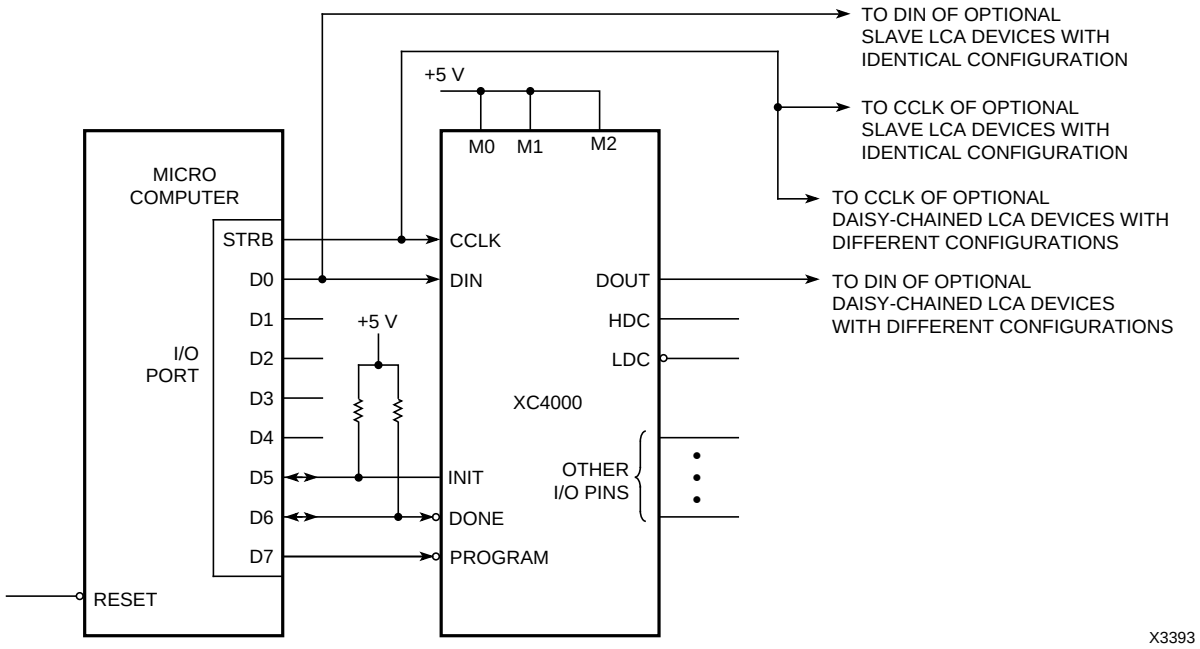
Master Serial Mode Programming Switching Characteristics



	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 Vcc 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



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

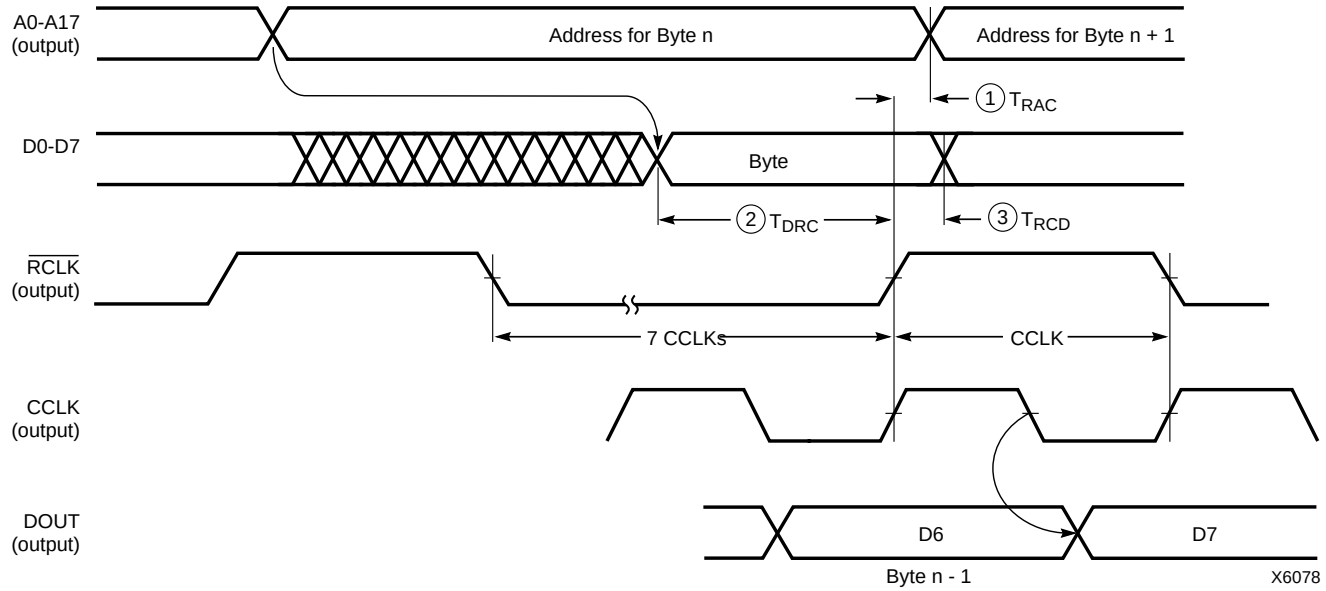
There are two methods to delay configuration after power-up: Put a logic Low on the PROGRAM input, or pull the bidirectional INIT pin Low, using an open-collector (open-drain) driver. (See also Figure 20 on page 2-27.)

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.

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

Master Parallel Mode Programming Switching Characteristics

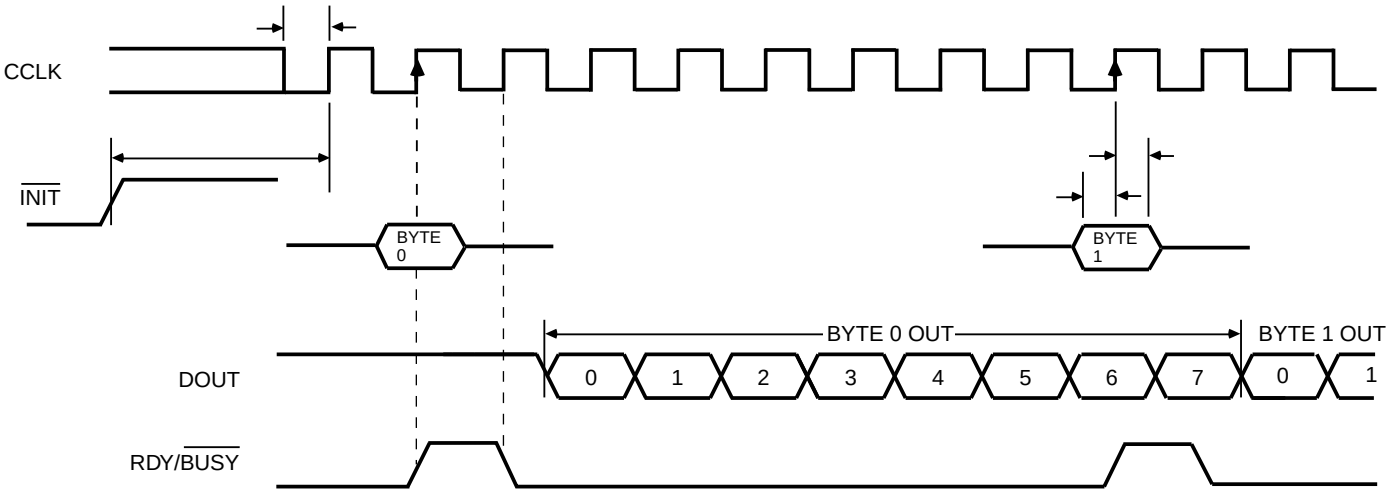


	Description	Symbol		Min	Max	Units
RCLK	Delay to Address valid	1	T_{RAC}	0	200	ns
	Data setup time	2	T_{DRC}	60		ns
	Data hold time	3	T_{RCD}	0		ns

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

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

Synchronous Peripheral Mode Programming Switching Characteristics



X6096

	Description	Symbol		Min	Max	Units
CCLK	INIT (High) Setup time required	1	T_{IC}	5		µs
	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

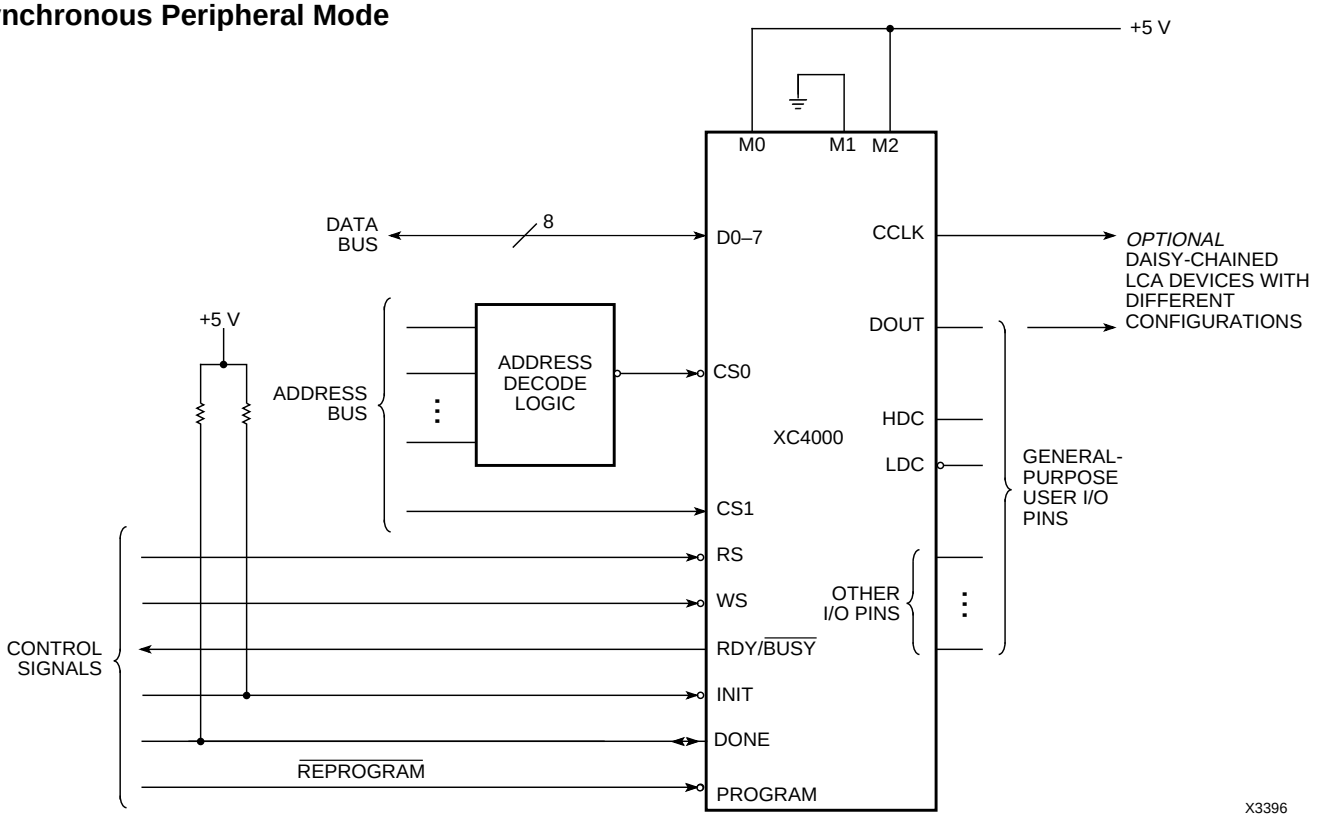
Notes: Peripheral Synchronous mode can be considered Slave Parallel mode. An external CCLK provides timing, clocking in the **first** data byte on the **second** rising edge of CCLK after INIT goes High. Subsequent data bytes are clocked in on every eighth consecutive rising edge of CCLK.

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



Write to LCA

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

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

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

Status Read

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

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

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

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

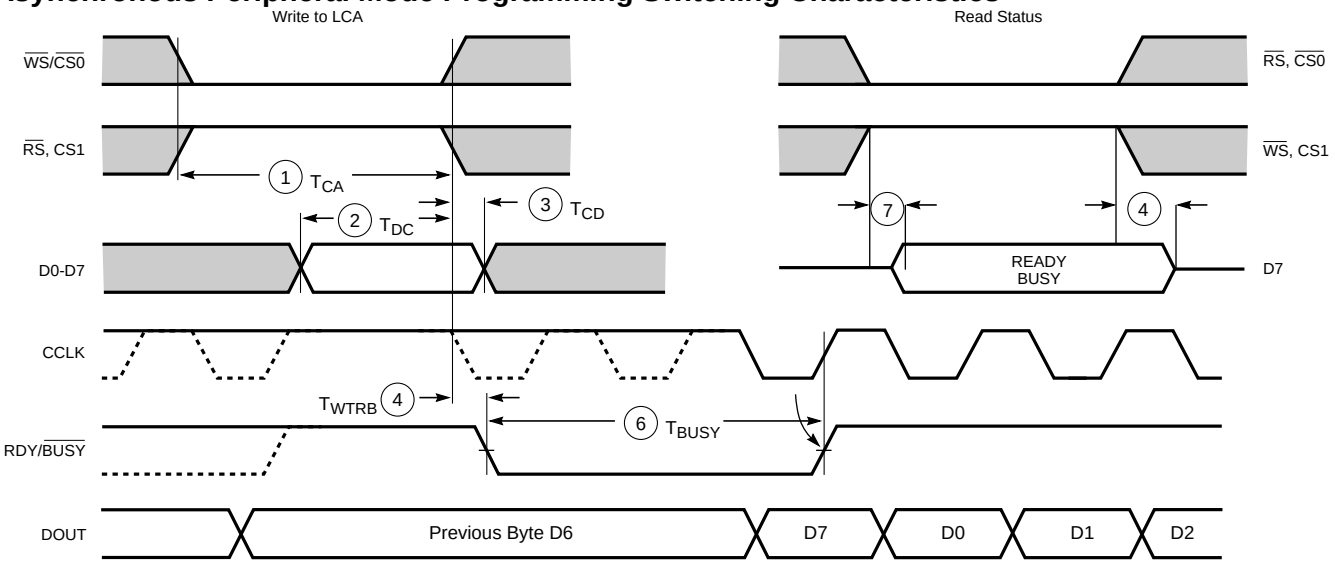
How to Delay Configuration After Power-Up

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

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

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

Asynchronous Peripheral Mode Programming Switching Characteristics



X6097

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