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	576
Number of Logic Elements/Cells	1368
Total RAM Bits	18432
Number of I/O	192
Number of Gates	13000
Voltage - Supply	4.75V ~ 5.25V
Mounting Type	Surface Mount
Operating Temperature	0°C ~ 85°C (TJ)
Package / Case	240-BFQFP
Supplier Device Package	240-PQFP (32x32)
Purchase URL	https://www.e-xfl.com/product-detail/xilinx/xc4013-6pq240c

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

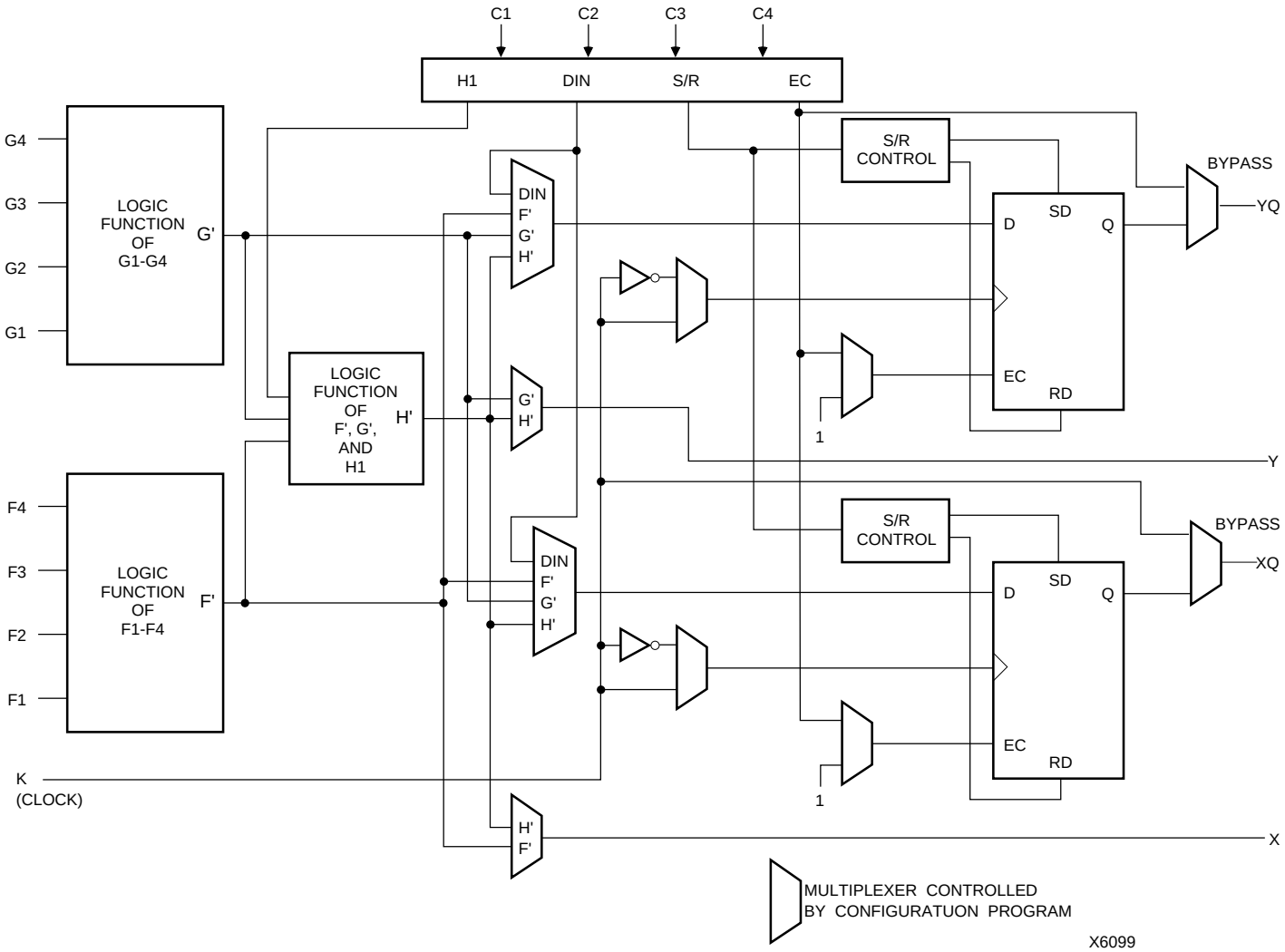


Figure 1. Simplified Block Diagram of XC4000-Families Configurable Logic Block

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, subtracters, 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.

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

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

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

The double-length lines (Figure 8) consist of a grid of metal segments twice as long as the single-length lines; i.e, a double-length line runs past two CLBs before entering a Switch Matrix. Double-length lines are grouped in pairs with the Switch Matrices staggered so that each line goes through a Switch Matrix at every other CLB location in that row or column. As with single-length lines, all the CLB inputs except K can be driven from any adjacent double-length line, and each CLB output can drive nearby double-length lines in both the vertical and horizontal planes. Double-length lines provide the most efficient implementation of intermediate length, point-to-point interconnections.

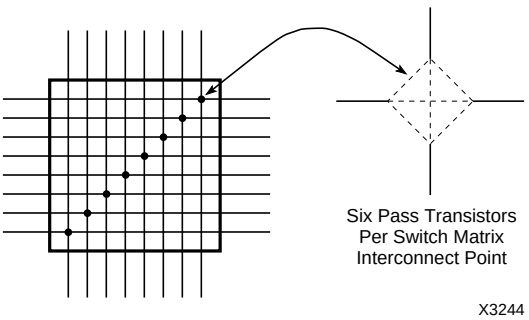


Figure 7. Switch Matrix

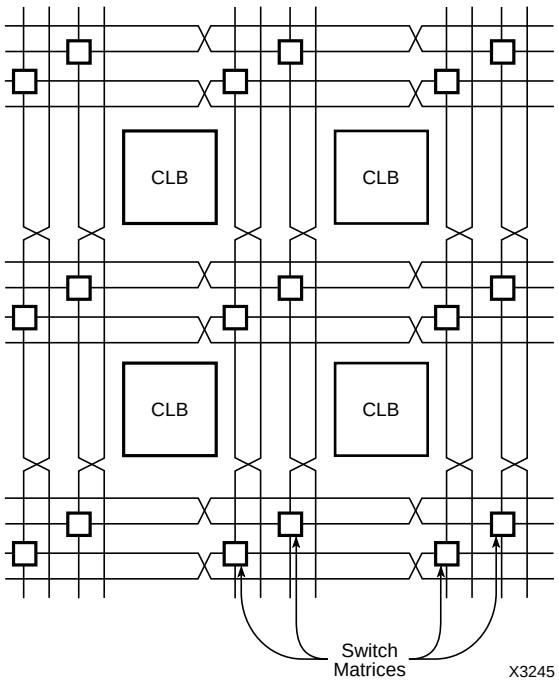


Figure 8. Double-Length Lines

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

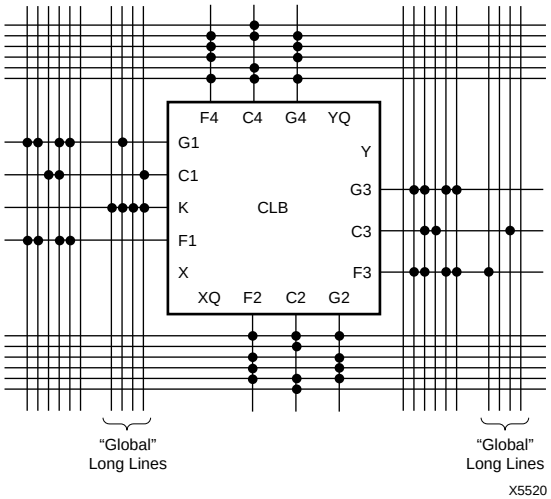


Figure 9. Longline Routing Resources with Typical CLB Connections

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

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

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

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

Development System

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

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

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

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

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

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

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

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

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

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

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.

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

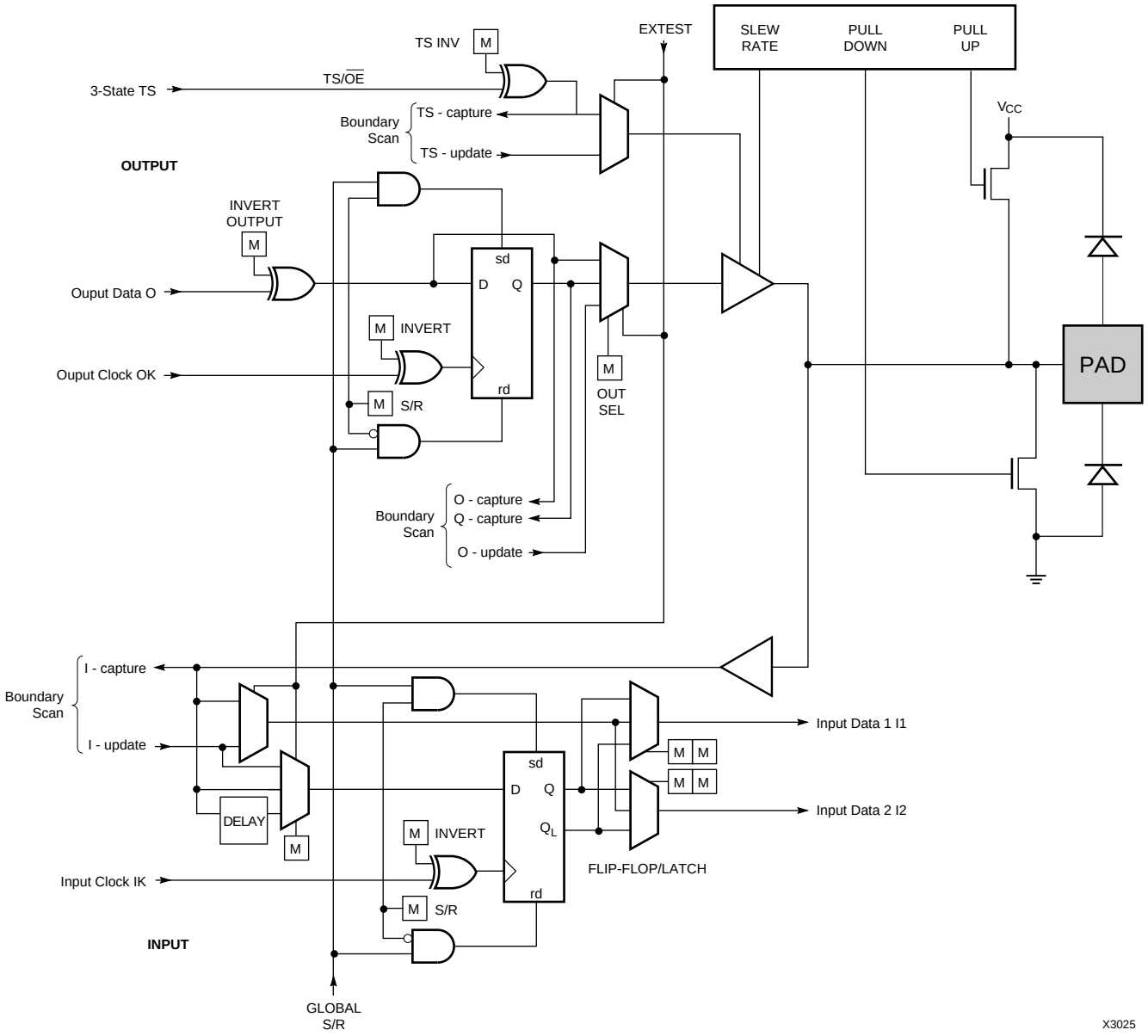


Figure 11. XC4000 and XC4000A I/O Block

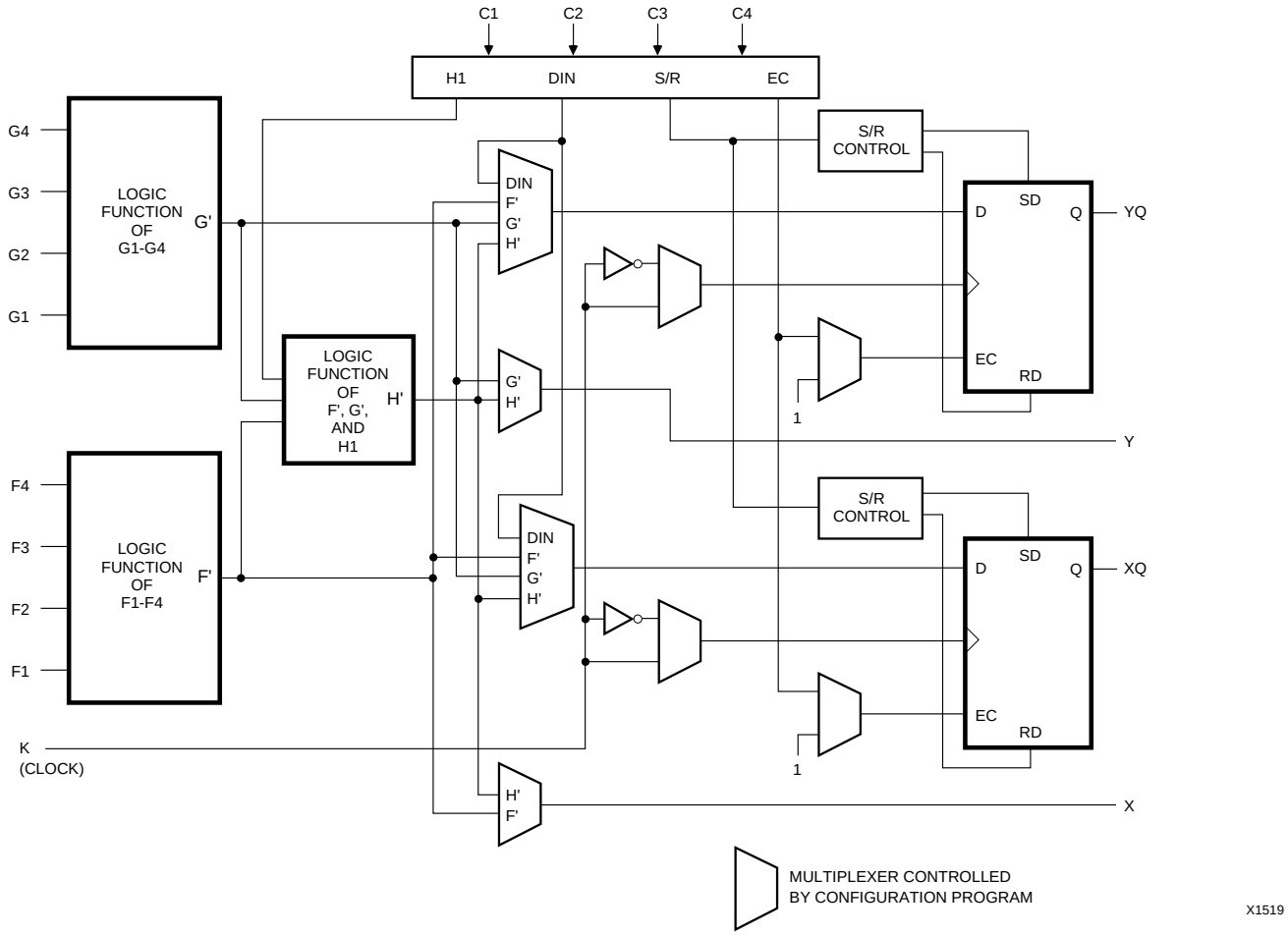


Figure 13. Simplified Block Diagram of XC4000 Configurable Logic Block

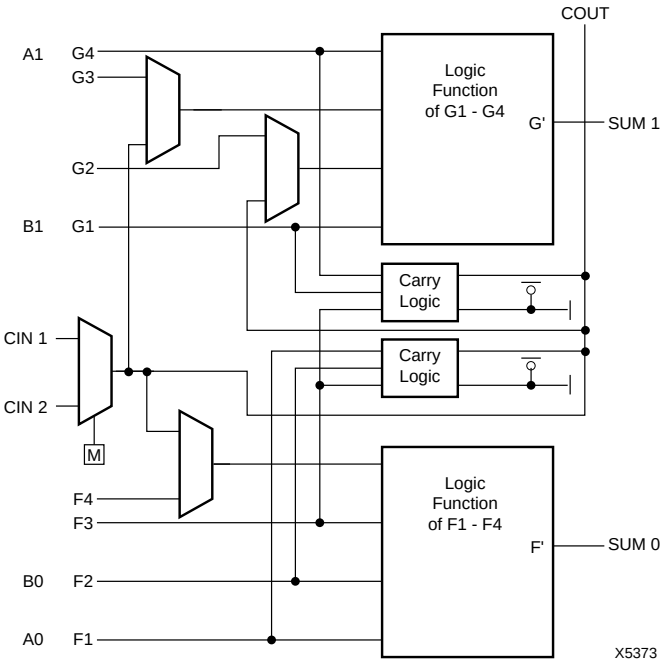


Figure 14. Fast Carry Logic in Each CLB

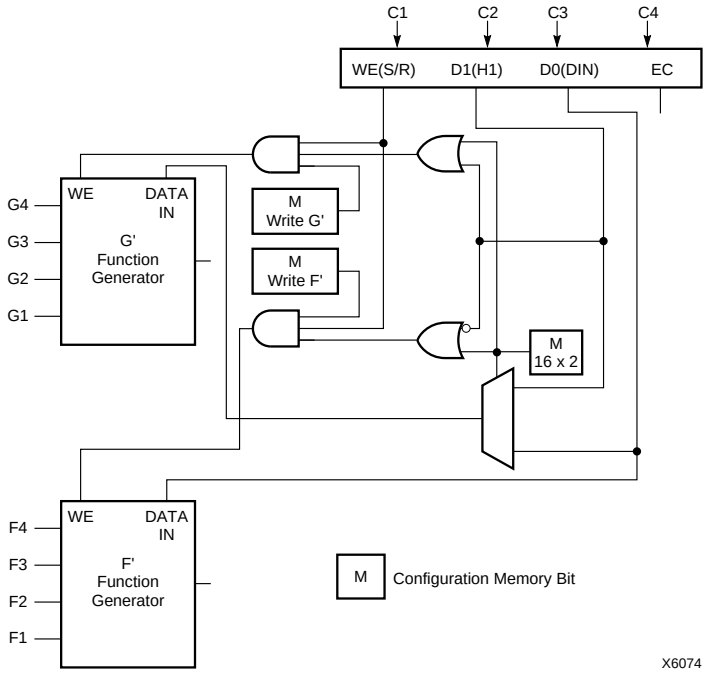


Figure 15. CLB Function Generators Can Be Used as Read/Write Memory Cells

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

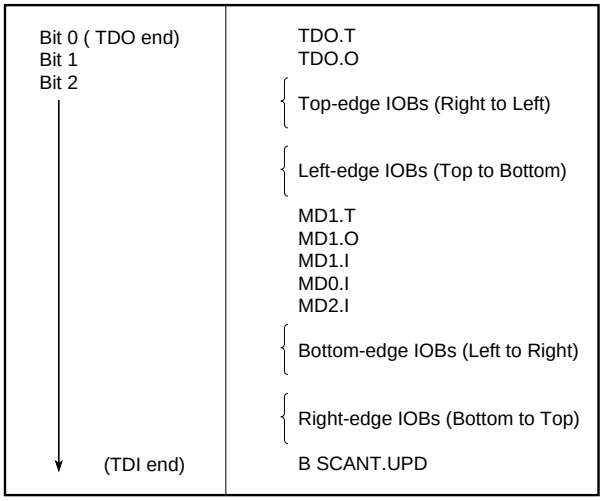
Instruction I_2	I_1	I_0	Test Selected	TDO Source	I/O Data Source
0	0	0	Extest	DR	DR
0	0	1	Sample/Preload	DR	Pin/Logic
0	1	0	User 1	TDO1	Pin/Logic
0	1	1	User 2	TDO2	Pin/Logic
1	0	0	Readback	Readback Data	Pin/Logic
1	0	1	Configure	DOUT	Disabled
1	1	0	Reserved	—	—
1	1	1	Bypass	Bypass Reg	Pin/Logic

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*.

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.

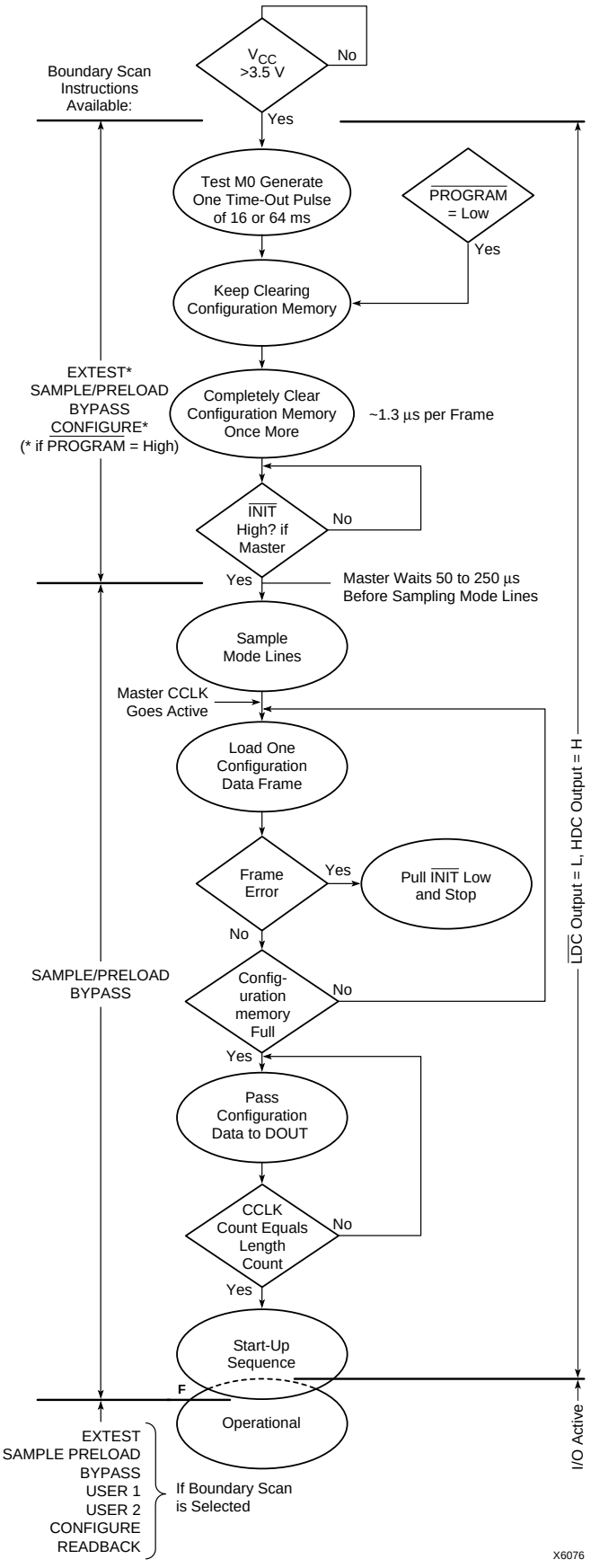


Figure 20. Start-up Sequence

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

After an LCA device has received its configuration data, it passes on any additional frame start bits and configuration data on DOUT. When the total number of configuration clocks applied after memory initialization equals the value of the 24-bit length count, the LCA device(s) begin the start-up sequence and become operational together.

Configuration Sequence

Configuration Memory Clear

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

Initialization

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

Configuration

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

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.

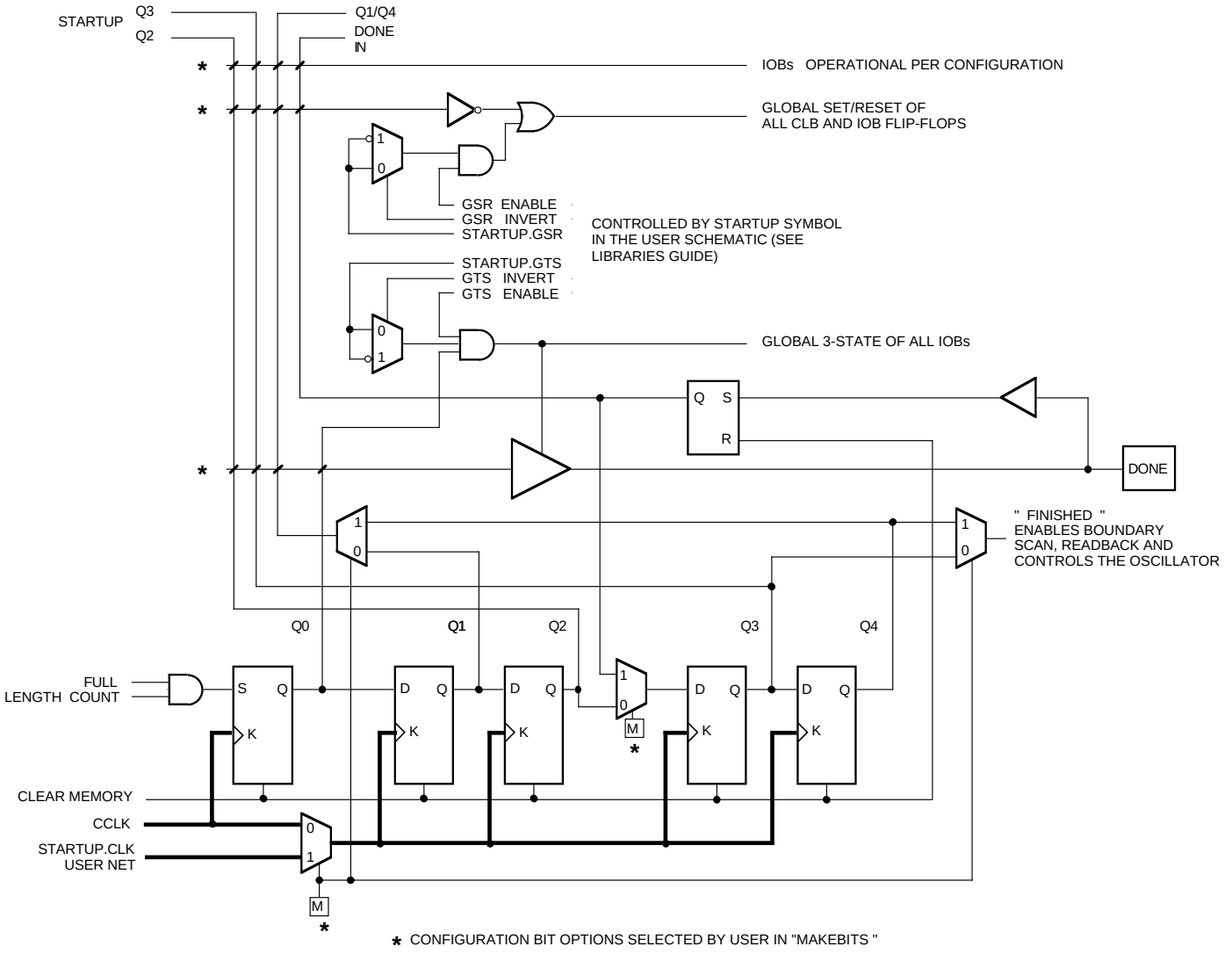


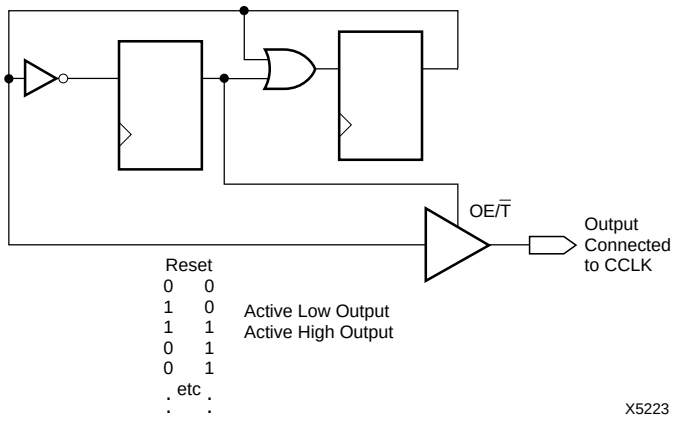
Figure 22. Start-up Logic

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

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

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

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



the extra CCLK pulse. This solution requires one CLB, one IOB and pin, and an internal oscillator with a frequency of up to 5 MHz as available clock source. Obviously, this XC3000 master device must be configured with late Internal Reset, which happens to be the default option.

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

Readback

The user can read back the content of configuration memory and the level of certain internal nodes without interfering with the normal operation of the device.

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

XC4000 Readback does not use any dedicated pins, but uses four internal nets (RDBK.TRIG, RDBK.DATA, RDBK.RIP and RDBK.CLK) that can be routed to any IOB.

After Readback has been initiated by a Low-to-High transition on RDBK.TRIG, the RDBK.RIP (Read In Progress) output goes High on the next rising edge of RDBK.CLK. Subsequent rising edges of this clock shift out Readback data on the RDBK.DATA net. Readback data does not include the preamble, but starts with five dummy bits (all High) followed by the Start bit (Low) of the first frame. The first two data bits of the first frame are always High.

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

Each frame ends with four error check bits. They are read back as High. The last seven bits of the last frame are also read back as High. An additional Start bit (Low) and an 11-bit Cyclic Redundancy Check (CRC) signature follow, before RIP returns Low.

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

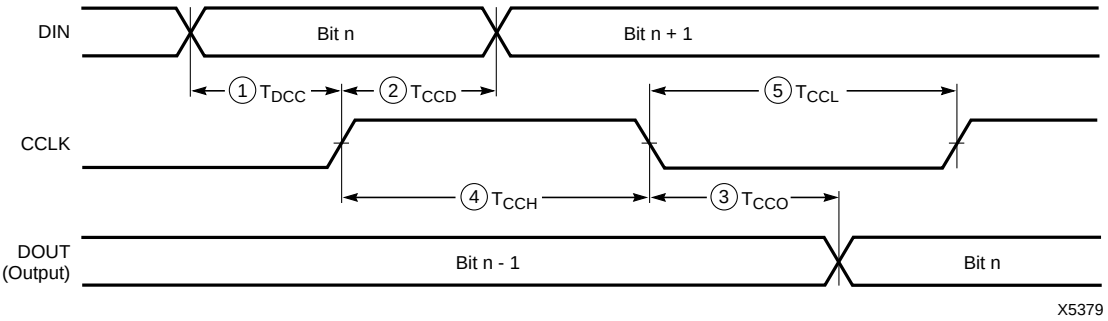
Read Capture
When the Readback Capture option is selected, the readback data stream includes sampled values of CLB and IOB signals imbedded in the data stream. The rising edge of RDBK.TRIG located in the lower-left chip corner, captures, in latches, the inverted values of the four CLB outputs and the IOB output flip-flops and the input signals I1, I2 . When the capture option is not selected, the values of the capture bits reflect the configuration data originally written to those memory locations. If the RAM capability of the CLBs is used, RAM data are available in readback, since they directly overwrite the F and G function-table configuration of the CLB.

Read Abort
When the Readback Abort option is selected, a High-to-Low transition on RDBK.TRIG terminates the readback operation and prepares the logic to accept another trigger. After an aborted readback, additional clocks (up-to-one readback clock per configuration frame) may be required to re-initialize the control logic. The status of readback is indicated by the output control net (RDBK.RIP).

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

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

Slave Serial Mode Programming Switching Characteristics

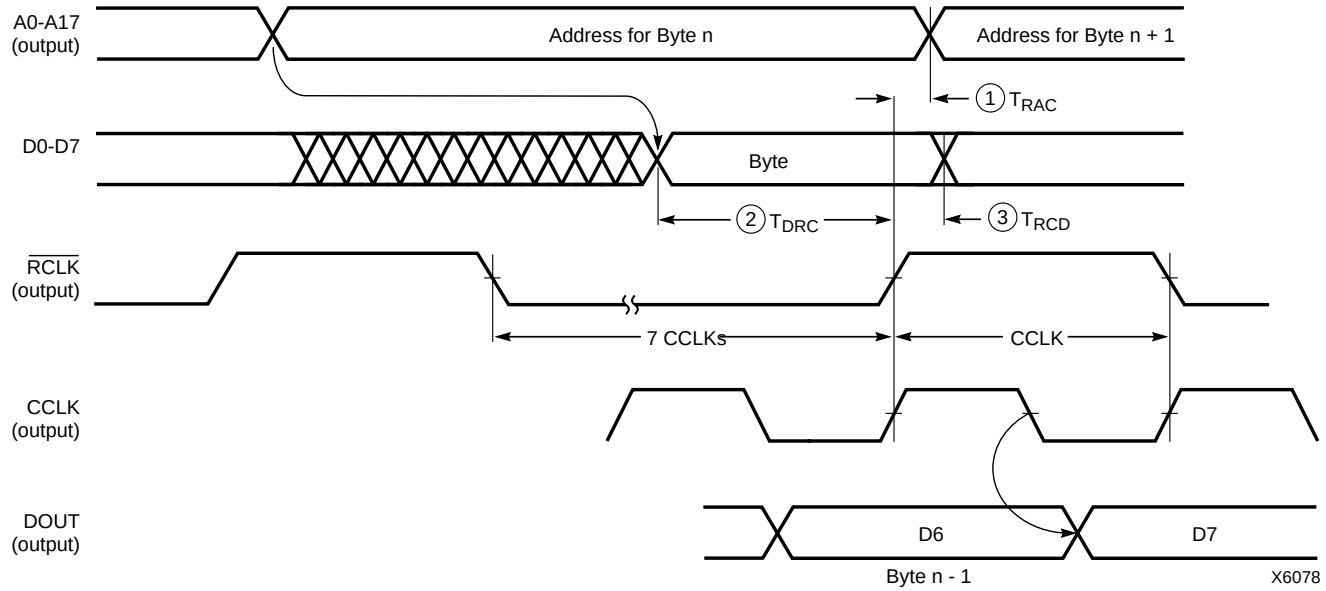


	Description	Symbol		Min	Max	Units
CCLK	DIN setup	1	T_{DCC}	20		ns
	DIN hold	2	T_{CCD}	0		ns
	to DOUT	3	T_{CCO}		30	ns
	High time	4	T_{CCH}	45		ns
	Low time	5	T_{CCL}	45		ns
	Frequency		F_{CC}		10	MHz

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

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

Master Parallel Mode Programming Switching Characteristics

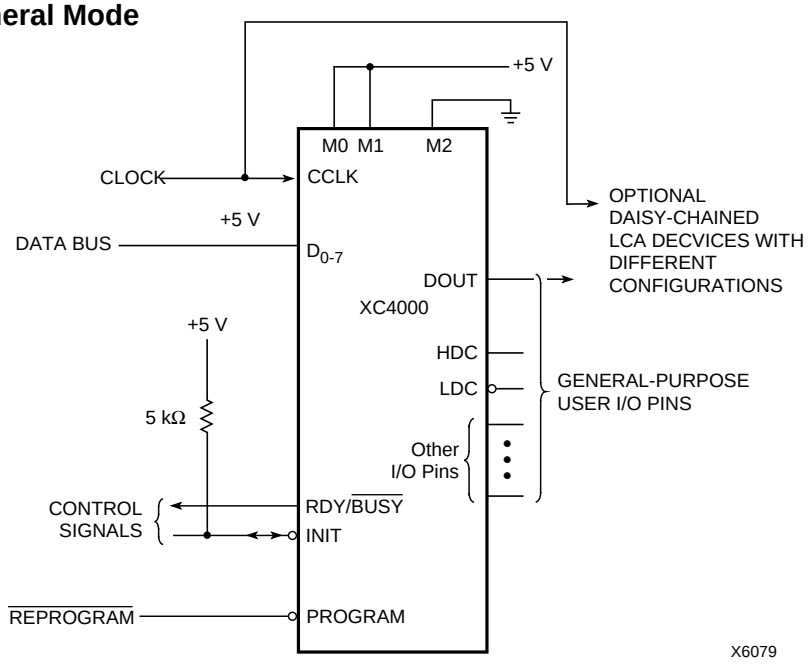


	Description	Symbol		Min	Max	Units
RCLK	Delay to Address valid	1	T_{RAC}	0	200	ns
	Data setup time	2	T_{DRC}	60		ns
	Data hold time	3	T_{RCD}	0		ns

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

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

Synchronous Peripheral Mode



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

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

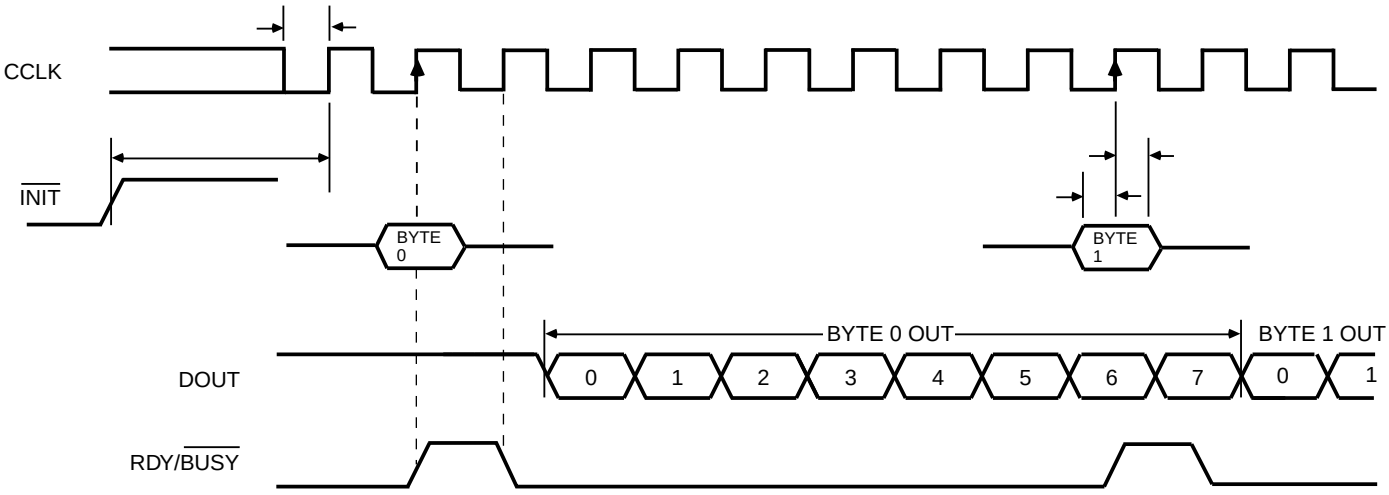
How to Delay Configuration After Power-Up

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

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

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

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.

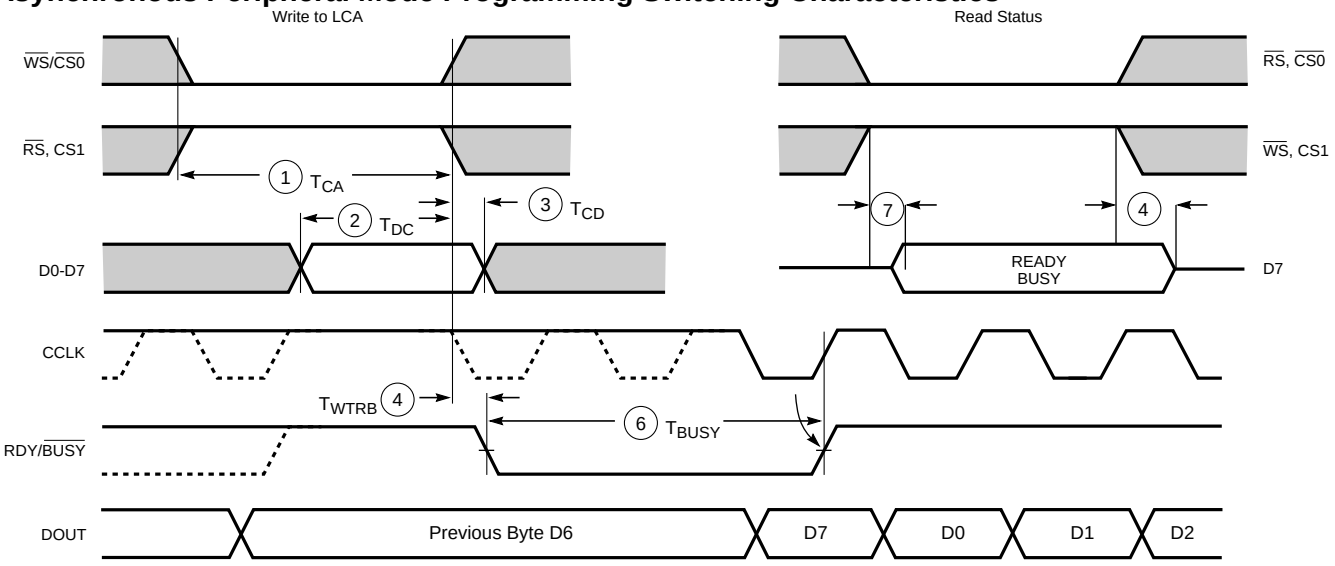
The pin name RDY/BUSY is a misnomer; in Synchronous Peripheral mode this is really an ACKNOWLEDGE signal.

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.

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.

Pin Functions During Configuration

CONFIGURATION MODE: <M2:M1:M0>						
SLAVE <1:1:1>	MASTER-SER <0:0:0>	SYN.PERIPH <0:1:1>	ASYN.PERIPH <1:0:1>	MASTER-HIGH <1:1:0>	MASTER-LOW <1:0:0>	USER OPERATION
				A16	A16	PGI-I/O
				A17	A17	I/O
TDI	TDI	TDI	TDI	TDI	TDI	TDI-I/O
TCK	TCK	TCK	TCK	TCK	TCK	TCK-I/O
TMS	TMS	TMS	TMS	TMS	TMS	TMS-I/O
						SGI-I/O
M1 (HIGH) (I)	M1 (LOW) (I)	M1 (HIGH) (I)	M1 (LOW) (I)	M1 (HIGH) (I)	M1 (LOW) (I)	(O)
M0 (HIGH) (I)	M0 (LOW) (I)	M0 (HIGH) (I)	M0 (HIGH) (I)	M0 (LOW) (I)	M0 (LOW) (I)	(I)
M2 (HIGH) (I)	M2 (LOW) (I)	M2 (LOW) (I)	M2 (HIGH) (I)	M2 (HIGH) (I)	M2 (HIGH) (I)	(I)
						PGI-I/O
HDC (HIGH)	HDC (HIGH)	HDC (HIGH)	HDC (HIGH)	HDC (HIGH)	HDC (HIGH)	I/O
LDC (LOW)	LDC (LOW)	LDC (LOW)	LDC (LOW)	LDC (LOW)	LDC (LOW)	I/O
* $\overline{\text{INIT-ERROR}}$	$\overline{\text{INIT-ERROR}}$	$\overline{\text{INIT-ERROR}}$	$\overline{\text{INIT-ERROR}}$	$\overline{\text{INIT-ERROR}}$	$\overline{\text{INIT-ERROR}}$	I/O
						SGI-I/O
DONE	DONE	DONE	DONE	DONE	DONE	DONE
$\overline{\text{PROGRAM}}$ (I)	$\overline{\text{PROGRAM}}$ (I)	$\overline{\text{PROGRAM}}$ (I)	$\overline{\text{PROGRAM}}$ (I)	$\overline{\text{PROGRAM}}$ (I)	$\overline{\text{PROGRAM}}$ (I)	$\overline{\text{PROGRAM}}$
		DATA 7 (I)	DATA 7 (I)	DATA 7 (I)	DATA 7 (I)	I/O
						PGI-I/O
		DATA 6 (I)	DATA 6 (I)	DATA 6 (I)	DATA 6 (I)	I/O
		DATA 5 (I)	DATA 5 (I)	DATA 5 (I)	DATA 5 (I)	I/O
			$\overline{\text{CS0}}$ (I)			I/O
		DATA 4 (I)	DATA 4 (I)	DATA 4 (I)	DATA 4 (I)	I/O
		DATA 3 (I)	DATA 3 (I)	DATA 3 (I)	DATA 3 (I)	I/O
			$\overline{\text{RS}}$ (I)			I/O
		DATA 2 (I)	DATA 2 (I)	DATA 2 (I)	DATA 2 (I)	I/O
		DATA 1 (I)	DATA 1 (I)	DATA 1 (I)	DATA 1 (I)	I/O
		$\overline{\text{RDY/BUSY}}$	RDY/$\overline{\text{BUSY}}$	$\overline{\text{RCLK}}$	$\overline{\text{RCLK}}$	I/O
DIN (I)	DIN (I)	DATA 0 (I)	DATA 0 (I)	DATA 0 (I)	DATA 0 (I)	I/O
DOUT	DOUT	DOUT	DOUT	DOUT	DOUT	SGI-I/O
CCLK (I)	CCLK (O)	CCLK (I)	CCLK (O)	CCLK(O)	CCLK (O)	CCLK (I)
TDO	TDO	TDO	TDO	TDO	TDO	TDO-(O)
			$\overline{\text{WS}}$ (I)	A0	A0	I/O
				A1	A1	PGI-I/O
			CS1 (I)	A2	A2	I/O
				A3	A3	I/O
				A4	A4	I/O
				A5	A5	I/O
				A6	A6	I/O
				A7	A7	I/O
				A8	A8	I/O
				A9	A9	I/O
				A10	A10	I/O
				A11	A11	I/O
				A12	A12	I/O
				A13	A13	I/O
				A14	A14	I/O
				A15	A15	SGI-I/O
						ALL OTHERS

X6081

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

* INIT is an open-drain output during configuration

(I) Represents an input

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

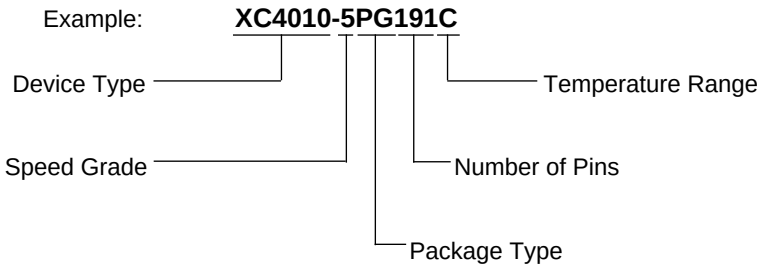
For a detailed description of the device architecture, see page 2-9 through 2-31.

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
TYPE		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	C I	C I			C I														
	-5	C	C			C														
	-4	C	C			C														
XC4005	-10							M B		M B										
	-6	C I	C I					C I M B	C I	M B			C I							
	-5	C I	C I					C I	C I				C I							
	-4	C	C					C	C				C							
XC4006	-6	C I						C I	C I				C I							
	-5	C I						C I	C I				C I							
	-4	C						C	C				C							
XC4008	-6	C I							C I		C I		C I	C I						
	-5	C I							C I		C I		C I	C I						
	-4	C							C		C		C	C						
XC4010	-10										M B	M B								
	-6	C I							C I		C I M B	M B	C I	C I		C I				
	-5	C I							C I		C I		C I	C I		C I				
	-4	C							C		C		C	C		C				
XC4010D	-6	C I							C I				C I			C I				
	-5	C I							C I				C I			C I				
	-4								C				C			C				
XC4013	-6								C I				C I	C I	CI (M B)	C I	CI	C I		
	-5								C I				C I	C I	C I	C I	C I	C I		
	-4								C				C	C	C	C	CI	C		
XC4013D	-6								C I				C I			C I	C I			
	-5								C I				C I			C I	C I			
	-4								C				C			C	C			
XC4020	-6												(C I)		(C I)		(C I)		(C I)	
	-5												(C I)		(C I)		(C I)		(C I)	
	-4												(C)		(C)		(C)		(C)	
XC4025	-6														C			C I	C I	C I
	-5														C			C I	C I	C I
	-4																			
XC4002A	-6	C I	C I	C I		C I														
	-5	C	C	C		C														
	-4																			
XC4003A	-10				M B	M B														
	-6	C I	C I	C I	M B	C I M B														
	-5	C	C	C		C														
	-4	C	C	C		C														
XC4004A	-6	C I				C I	C I		C I											
	-5	C				C	C		C											
	-4																			
XC4005A	-6	C I					C I	C I	C I				C I							
	-5	C I					C I	C I	C I				C I							
	-4	C					C	C	C				C							
XC4003H	-6										C I		C I							
	-5										C		C							
XC4005H	-6														C I		C I	C I		
	-5														C		C	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