Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of "Quality Parts,Customers Priority,Honest Operation,and Considerate Service",our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

XILINX ®

XC3000 Series
Field Programmable Gate Arrays
(XC3000A/L, XC3100A/L)

November 9, 1998 (Version 3.1)   Product Description

Features

- Complete line of four related Field Programmable Gate Array product families
  - XC3000A, XC3000L, XC3100A, XC3100L
- Ideal for a wide range of custom VLSI design tasks
  - Replaces TTL, MSI, and other PLD logic
  - Integrates complete sub-systems into a single package
  - Avoids the NRE, time delay, and risk of conventional masked gate arrays
- High-performance CMOS static memory technology
  - Guaranteed toggle rates of 70 to 370 MHz, logic delays from 7 to 1.5 ns
  - System clock speeds over 85 MHz
  - Low quiescent and active power consumption
- Flexible FPGA architecture
  - Compatible arrays ranging from 1,000 to 7,500 gate complexity
  - Extensive register, combinatorial, and I/O capabilities
  - High fan-out signal distribution, low-skew clock nets
  - Internal 3-state bus capabilities
  - TTL or CMOS input thresholds
  - On-chip crystal oscillator amplifier
- Unlimited reprogrammability
  - Easy design iteration
  - In-system logic changes
- Extensive packaging options
  - Over 20 different packages
  - Plastic and ceramic surface-mount and pin-grid-array packages
  - Thin and Very Thin Quad Flat Pack (TQFP and VQFP) options
- Ready for volume production
  - Standard, off-the-shelf product availability
  - 100% factory pre-tested devices
  - Excellent reliability record

- Complete Development System
  - Schematic capture, automatic place and route
  - Logic and timing simulation
  - Interactive design editor for design optimization
  - Timing calculator
  - Interfaces to popular design environments like Viewlogic, Cadence, Mentor Graphics, and others

Additional XC3100A Features

- Ultra-high-speed FPGA family with six members
  - 50-85 MHz system clock rates
  - 190 to 370 MHz guaranteed flip-flop toggle rates
  - 1.55 to 4.1 ns logic delays
- High-end additional family member in the 22 X 22 CLB array-size XC3195A device
- 8 mA output sink current and 8 mA source current
- Maximum power-down and quiescent current is 5 mA
- 100% architecture and pin-out compatible with other XC3000 families
- Software and bitstream compatible with the XC3000, XC3000A, and XC3000L families

XC3100A combines the features of the XC3000A and XC3100 families:

- Additional interconnect resources for TBUFs and CE inputs
- Error checking of the configuration bitstream
- Soft startup holds all outputs slew-rate limited during initial power-up
- More advanced CMOS process

Low-Voltage Versions Available

- Low-voltage devices function at 3.0 - 3.6 V
- XC3000L - Low-voltage versions of XC3000A devices
- XC3100L - Low-voltage versions of XC3100A devices

7

Device	Max Logic Gates	Typical Gate Range	CLBs	Array	User I/Os Max	Flip-Flops	Horizontal Longlines	Configuration Data Bits
XC3020A, 3020L, 3120A	1,500	1,000 - 1,500	64	8 x 8	64	256	16	14,779
XC3030A, 3030L, 3130A	2,000	1,500 - 2,000	100	10 x 10	80	360	20	22,176
XC3042A, 3042L, 3142A, 3142L	3,000	2,000 - 3,000	144	12 x 12	96	480	24	30,784
XC3064A, 3064L, 3164A	4,500	3,500 - 4,500	224	16 x 14	120	688	32	46,064
XC3090A, 3090L, 3190A, 3190L	6,000	5,000 - 6,000	320	16 x 20	144	928	40	64,160
XC3195A	7,500	6,500 - 7,500	484	22 x 22	176	1,320	44	94,984

Introduction

XC3000-Series Field Programmable Gate Arrays (FPGAs) provide a group of high-performance, high-density, digital integrated circuits. Their regular, extendable, flexible, user-programmable array architecture is composed of a configuration program store plus three types of configurable elements: a perimeter of I/O Blocks (IOBs), a core array of Configurable Logic Bocks (CLBs) and resources for interconnection. The general structure of an FPGA is shown in Figure 2. The development system provides schematic capture and auto place-and-route for design entry. Logic and timing simulation, and in-circuit emulation are available as design verification alternatives. The design editor is used for interactive design optimization, and to compile the data pattern that represents the configuration program.

The FPGA user logic functions and interconnections are determined by the configuration program data stored in internal static memory cells. The program can be loaded in any of several modes to accommodate various system requirements. The program data resides externally in an EEPROM, EPROM or ROM on the application circuit board, or on a floppy disk or hard disk. On-chip initialization logic provides for optional automatic loading of program data at power-up. The companion XC17XX Serial Configuration PROMs provide a very simple serial configuration program storage in a one-time programmable package.

The XC3000 Field Programmable Gate Array families provide a variety of logic capacities, package styles, temperature ranges and speed grades.

XC3000 Series Overview

There are now four distinct family groupings within the XC3000 Series of FPGA devices:

• XC3000A Family
• XC3000L Family
• XC3100A Family
• XC3100L Family

All four families share a common architecture, development software, design and programming methodology, and also common package pin-outs. An extensive Product Description covers these common aspects.

Detailed parametric information for the XC3000A, XC3000L, XC3100A, and XC3100L product families is then provided. (The XC3000 and XC3100 families are not recommended for new designs.)

Here is a simple overview of those XC3000 products currently emphasized:

• **XC3000A Family** — The XC3000A is an enhanced version of the basic XC3000 family, featuring additional interconnect resources and other user-friendly enhancements.
• **XC3000L Family** — The XC3000L is identical in architecture and features to the XC3000A family, but operates at a nominal supply voltage of 3.3 V. The XC3000L is the right solution for battery-operated and low-power applications.
• **XC3100A Family** — The XC3100A is a performance-optimized relative of the XC3000A family. While both families are bitstream and footprint compatible, the XC3100A family extends toggle rates to 370 MHz and in-system performance to over 80 MHz. The XC3100A family also offers one additional array size, the XC3195A.
• **XC3100L Family** — The XC3100L is identical in architectures and features to the XC3100A family, but operates at a nominal supply voltage of 3.3V.

Figure 1 illustrates the relationships between the families. Compared to the original XC3000 family, XC3000A offers additional functionality and increased speed. The XC3000L family offers the same additional functionality, but reduced speed due to its lower supply voltage of 3.3 V. The XC3100A family offers substantially higher speed and higher density with the XC3195A.

New XC3000 Series Compared to Original XC3000 Family

For readers already familiar with the original XC3000 family of FPGAs, the major new features in the XC3000A, XC3000L, XC3100A, and XC3100L families are listed in this section.

All of these new families are upward-compatible extensions of the original XC3000 FPGA architecture. Any bitstream used to configure an XC3000 device will configure the corresponding XC3000A, XC3000L, XC3100A, or XC3100L device exactly the same way.

The XC3100A and XC3100L FPGA architectures are upward-compatible extensions of the XC3000A and XC3000L architectures. Any bitstream used to configure an XC3000A or XC3000L device will configure the corresponding XC3100A or XC3100L device exactly the same way.

Improvements in the XC3000A and XC3000L Families

The XC3000A and XC3000L families offer the following enhancements over the popular XC3000 family:

The XC3000A and XC3000L families have additional interconnect resources to drive the I-inputs of TBUFs driving horizontal Longlines. The CLB Clock Enable input can be driven from a second vertical Longline. These two additions result in more efficient and faster designs when horizontal Longlines are used for data bussing.

During configuration, the XC3000A and XC3000L devices check the bit-stream format for stop bits in the appropriate positions. Any error terminates the configuration and pulls INIT Low.

When the configuration process is finished and the device starts up in user mode, the first activation of the outputs is automatically slew-rate limited. This feature, called Soft Startup, avoids the potential ground bounce when all out-puts are turned on simultaneously. After start-up, the slew rate of the individual outputs is, as in the XC3000 family, determined by the individual configuration option.

Improvements in the XC3100A and XC3100L Families

Based on a more advanced CMOS process, the XC3100A and XC3100L families are architecturally-identical, performance-optimized relatives of the XC3000A and XC3000L families. While all families are footprint compatible, the XC3100A family extends achievable system performance beyond 85 MHz.
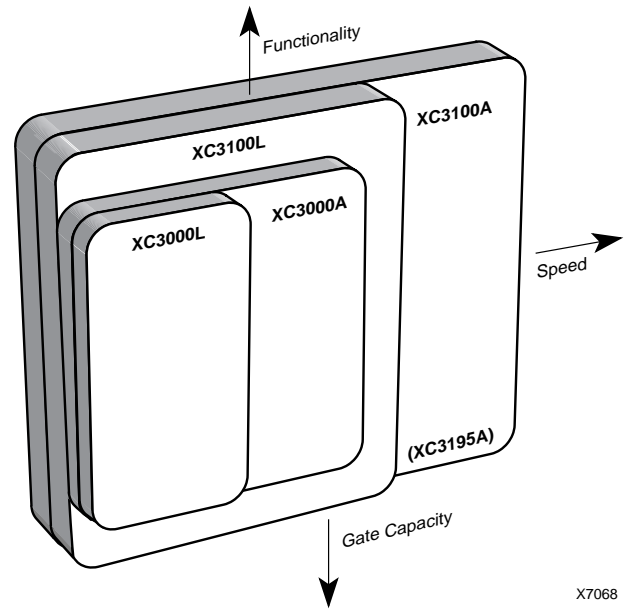
Figure 1: XC3000 FPGA Families

Detailed Functional Description

The perimeter of configurable Input/Output Blocks (IOBs) provides a programmable interface between the internal logic array and the device package pins. The array of Configurable Logic Blocks (CLBs) performs user-specified logic functions. The interconnect resources are programmed to form networks, carrying logic signals among blocks, analogous to printed circuit board traces connecting MSI/SSI packages.

The block logic functions are implemented by programmed look-up tables. Functional options are implemented by program-controlled multiplexers. Interconnecting networks between blocks are implemented with metal segments joined by program-controlled pass transistors.

These FPGA functions are established by a configuration program which is loaded into an internal, distributed array of configuration memory cells. The configuration program is loaded into the device at power-up and may be reloaded on command. The FPGA includes logic and control signals to implement automatic or passive configuration. Program data may be either bit serial or byte parallel. The development system generates the configuration program bitstream used to configure the device. The memory loading process is independent of the user logic functions.
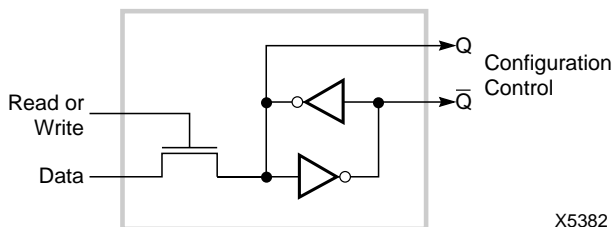
Configuration Memory

The static memory cell used for the configuration memory in the Field Programmable Gate Array has been designed specifically for high reliability and noise immunity. Integrity of the device configuration memory based on this design is assured even under adverse conditions. As shown in Figure 3, the basic memory cell consists of two CMOS inverters plus a pass transistor used for writing and reading cell data. The cell is only written during configuration and only read during readback. During normal operation, the cell provides continuous control and the pass transistor is off and does not affect cell stability. This is quite different from the operation of conventional memory devices, in which the cells are frequently read and rewritten.
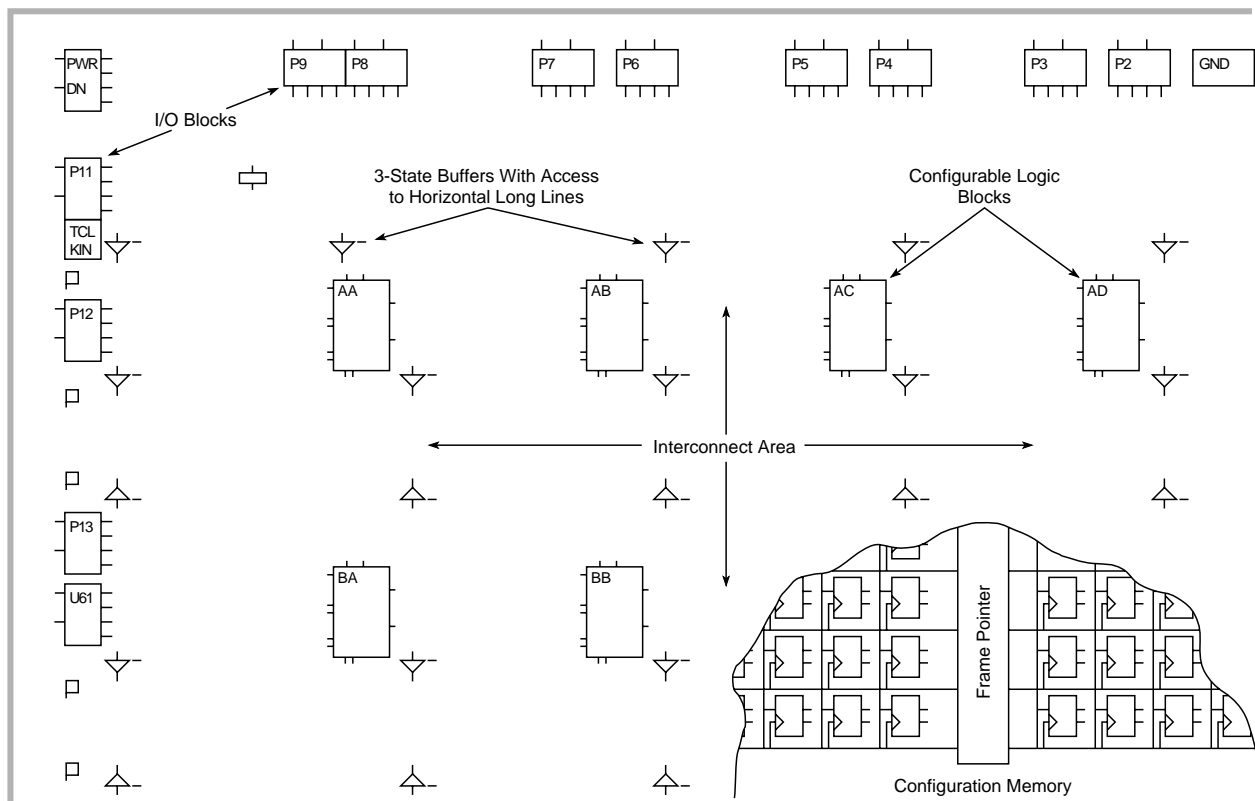
Figure 2: Field Programmable Gate Array Structure.
It consists of a perimeter of programmable I/O blocks, a core of configurable logic blocks and their interconnect resources. These are all controlled by the distributed array of configuration program memory cells.

Figure 3: Static Configuration Memory Cell.
It is loaded with one bit of configuration program and controls one program selection in the Field Programmable Gate Array.

The memory cell outputs Q and $\overline{Q}$ use ground and V_{CC} levels and provide continuous, direct control. The additional capacitive load together with the absence of address decoding and sense amplifiers provide high stability to the cell. Due to the structure of the configuration memory cells, they are not affected by extreme power-supply excursions or very high levels of alpha particle radiation. In reliability testing, no soft errors have been observed even in the presence of very high doses of alpha radiation.

The method of loading the configuration data is selectable. Two methods use serial data, while three use byte-wide data. The internal configuration logic utilizes framing information, embedded in the program data by the development system, to direct memory-cell loading. The serial-data framing and length-count preamble provide programming compatibility for mixes of various FPGA device devices in a synchronous, serial, daisy-chain fashion.

I/O Block

Each user-configurable IOB shown in Figure 4, provides an interface between the external package pin of the device and the internal user logic. Each IOB includes both registered and direct input paths. Each IOB provides a programmable 3-state output buffer, which may be driven by a registered or direct output signal. Configuration options allow each IOB an inversion, a controlled slew rate and a high impedance pull-up. Each input circuit also provides input clamping diodes to provide electrostatic protection, and circuits to inhibit latch-up produced by input currents.
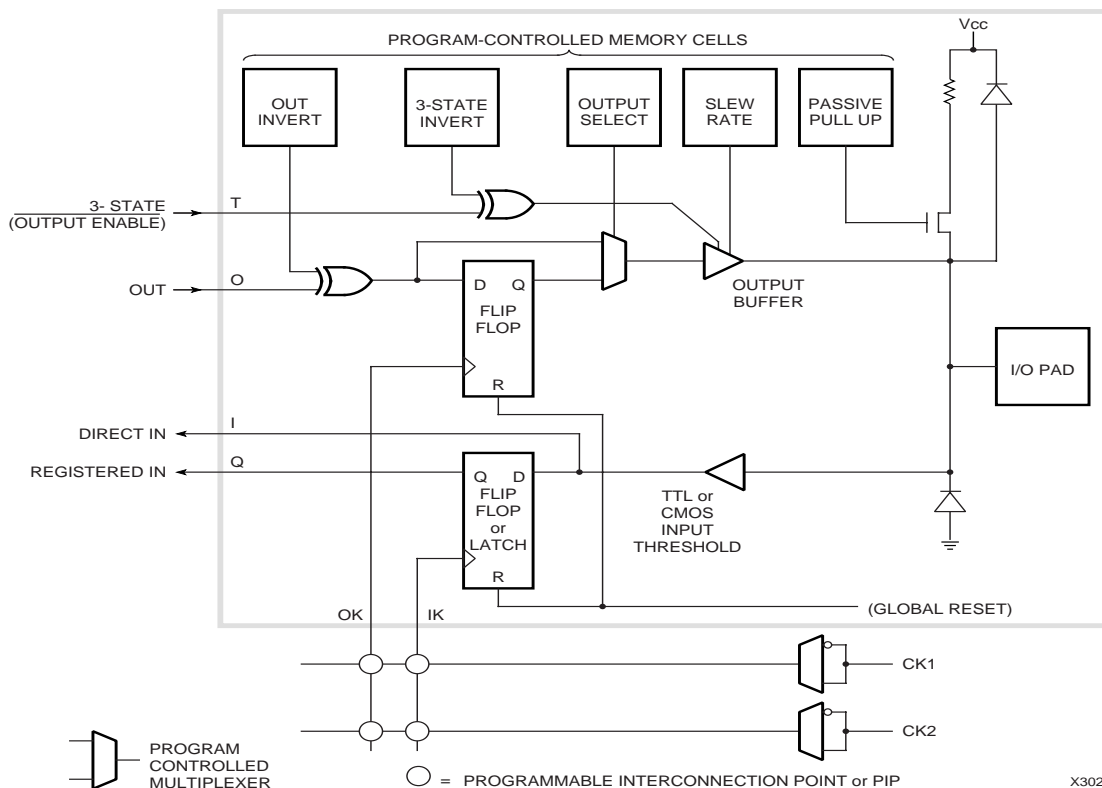
Figure 4: Input/Output Block.
Each IOB includes input and output storage elements and I/O options selected by configuration memory cells. A choice of two clocks is available on each die edge. The polarity of each clock line (not each flip-flop or latch) is programmable. A clock line that triggers the flip-flop on the rising edge is an active Low Latch Enable (Latch transparent) signal and vice versa. Passive pull-up can only be enabled on inputs, not on outputs. All user inputs are programmed for TTL or CMOS thresholds.

The input-buffer portion of each IOB provides threshold detection to translate external signals applied to the package pin to internal logic levels. The global input-buffer threshold of the IOBs can be programmed to be compatible with either TTL or CMOS levels. The buffered input signal drives the data input of a storage element, which may be configured as either a flip-flop or a latch. The clocking polarity (rising/falling edge-triggered flip-flop, High/Low transparent latch) is programmable for each of the two clock lines on each of the four die edges. Note that a clock line driving a *rising* edge-triggered flip-flop makes any latch driven by the same line on the same edge Low-level transparent and vice versa (*falling* edge, *High* transparent). All Xilinx primitives in the supported schematic-entry packages, however, are positive edge-triggered flip-flops or High transparent latches. When one clock line must drive flip-flops as well as latches, it is necessary to compensate for the difference in clocking polarities with an additional inverter either in the flip-flop clock input or the latch-enable input. I/O storage elements are reset during configuration or by the active-Low chip RESET input. Both direct input (from IOB pin I) and registered input (from IOB pin Q) signals are available for interconnect.

For reliable operation, inputs should have transition times of less than 100 ns and should not be left floating. Floating CMOS input-pin circuits might be at threshold and produce oscillations. This can produce additional power dissipation and system noise. A typical hysteresis of about 300 mV reduces sensitivity to input noise. Each user IOB includes a programmable high-impedance pull-up resistor, which may be selected by the program to provide a constant High for otherwise undriven package pins. Although the Field Programmable Gate Array provides circuitry to provide input protection for electrostatic discharge, normal CMOS handling precautions should be observed.

Flip-flop loop delays for the IOB and logic-block flip-flops are short, providing good performance under asynchronous clock and data conditions. Short loop delays minimize the probability of a metastable condition that can result from assertion of the clock during data transitions. Because of the short-loop-delay characteristic in the Field Programmable Gate Array, the IOB flip-flops can be used to synchronize external signals applied to the device. Once synchronized in the IOB, the signals can be used internally without further consideration of their clock relative timing, except as it applies to the internal logic and routing-path delays.

IOB output buffers provide CMOS-compatible 4-mA source-or-sink drive for high fan-out CMOS or TTL- compatible signal levels (8 mA in the XC3100A family). The network driving IOB pin O becomes the registered or direct data source for the output buffer. The 3-state control signal (IOB) pin T can control output activity. An open-drain output may be obtained by using the same signal for driving the output and 3-state signal nets so that the buffer output is enabled only for a Low.

Configuration program bits for each IOB control features such as optional output register, logic signal inversion, and 3-state and slew-rate control of the output.

The program-controlled memory cells of Figure 4 control the following options.

- Logic inversion of the output is controlled by one configuration program bit per IOB.
- Logic 3-state control of each IOB output buffer is determined by the states of configuration program bits that turn the buffer on, or off, or select the output buffer 3-state control interconnection (IOB pin T). When this IOB output control signal is High, a logic one, the buffer is disabled and the package pin is high impedance. When this IOB output control signal is Low, a logic zero, the buffer is enabled and the package pin is active. Inversion of the buffer 3-state control-logic sense (output enable) is controlled by an additional configuration program bit.
- Direct or registered output is selectable for each IOB. The register uses a positive-edge, clocked flip-flop. The clock source may be supplied (IOB pin OK) by either of two metal lines available along each die edge. Each of these lines is driven by an invertible buffer.
- Increased output transition speed can be selected to improve critical timing. Slower transitions reduce capacitive-load peak currents of non-critical outputs and minimize system noise.
- An internal high-impedance pull-up resistor (active by default) prevents unconnected inputs from floating.

Unlike the original XC3000 series, the XC3000A, XC3000L, XC3100A, and XC3100L families include the Soft Startup feature. When the configuration process is finished and the device starts up in user mode, the first activation of the outputs is automatically slew-rate limited. This feature avoids potential ground bounce when all outputs are turned on simultaneously. After start-up, the slew rate of the individual outputs is determined by the individual configuration option.

Summary of I/O Options
- Inputs
  - Direct
  - Flip-flop/latch
  - CMOS/TTL threshold (chip inputs)
  - Pull-up resistor/open circuit
- Outputs
  - Direct/registered
  - Inverted/not
  - 3-state/on/off
  - Full speed/slew limited
 - 3-state/output enable (inverse)

Configurable Logic Block

The array of CLBs provides the functional elements from which the user's logic is constructed. The logic blocks are arranged in a matrix within the perimeter of IOBs. For example, the XC3020A has 64 such blocks arranged in 8 rows and 8 columns. The development system is used to compile the configuration data which is to be loaded into the internal configuration memory to define the operation and interconnection of each block. User definition of CLBs and their interconnecting networks may be done by automatic translation from a schematic-capture logic diagram or optionally by installing library or user macros.

Each CLB has a combinatorial logic section, two flip-flops, and an internal control section. See Figure 5. There are: five logic inputs (A, B, C, D and E); a common clock input (K); an asynchronous direct RESET input (RD); and an enable clock (EC). All may be driven from the interconnect resources adjacent to the blocks. Each CLB also has two outputs (X and Y) which may drive interconnect networks.

Data input for either flip-flop within a CLB is supplied from the function F or G outputs of the combinatorial logic, or the block input, DI. Both flip-flops in each CLB share the asynchronous RD which, when enabled and High, is dominant over clocked inputs. All flip-flops are reset by the active-Low chip input, RESET, or during the configuration process. The flip-flops share the enable clock (EC) which, when Low, recirculates the flip-flops' present states and inhibits response to the data-in or combinatorial function inputs on a CLB. The user may enable these control inputs and select their sources. The user may also select the clock net input (K), as well as its active sense within each CLB. This programmable inversion eliminates the need to route both phases of a clock signal throughout the device.
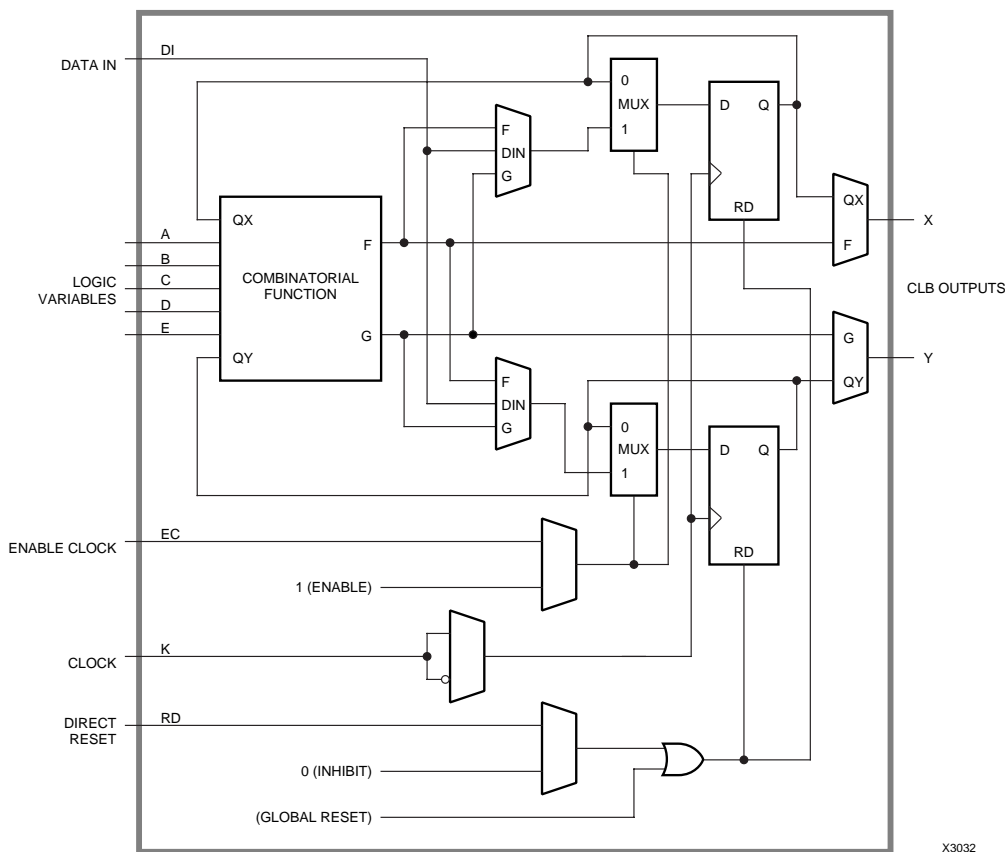
Figure 5: Configurable Logic Block.
Each CLB includes a combinatorial logic section, two flip-flops and a program memory controlled multiplexer selection of function. It has the following:
- five logic variable inputs A, B, C, D, and E
- a direct data in DI
- an enable clock EC
- a clock (invertible) K
- an asynchronous direct RESET RD
- two outputs X and Y

Flexible routing allows use of common or individual CLB clocking.

The combinatorial-logic portion of the CLB uses a 32 by 1 look-up table to implement Boolean functions. Variables selected from the five logic inputs and two internal block flip-flops are used as table address inputs. The combinatorial propagation delay through the network is independent of the logic function generated and is spike free for single input variable changes. This technique can generate two independent logic functions of up to four variables each as shown in Figure 6a, or a single function of five variables as shown in Figure 6b, or some functions of seven variables as shown in Figure 6c. Figure 7 shows a modulo-8 binary counter with parallel enable. It uses one CLB of each type. The partial functions of six or seven variables are implemented using the input variable (E) to dynamically select between two functions of four different variables. For the two functions of four variables each, the independent results (F and G) may be used as data inputs to either flip-flop or either logic block output. For the single function of five variables and merged functions of six or seven variables, the F and G outputs are identical. Symmetry of the F and G functions and the flip-flops allows the interchange of CLB outputs to optimize routing efficiencies of the networks interconnecting the CLBs and IOBs.

Programmable Interconnect

Programmable-interconnection resources in the Field Programmable Gate Array provide routing paths to connect inputs and outputs of the IOBs and CLBs into logic networks. Interconnections between blocks are composed of a two-layer grid of metal segments. Specially designed pass transistors, each controlled by a configuration bit, form programmable interconnect points (PIPs) and switching matrices used to implement the necessary connections between selected metal segments and block pins. Figure 8 is an example of a routed net. The development system provides automatic routing of these interconnections. Interactive routing is also available for design optimization. The inputs of the CLBs or IOBs are multiplexers which can be programmed to select an input network from the adjacent interconnect segments. *Since the switch connections to block inputs are unidirectional, as are block outputs, they are usable only for block input connection and not for routing.* Figure 9 illustrates routing access to logic block input variables, control inputs and block outputs. Three types of metal resources are provided to accommodate various network interconnect requirements.

- General Purpose Interconnect
- Direct Connection
- Longlines (multiplexed busses and wide AND gates)
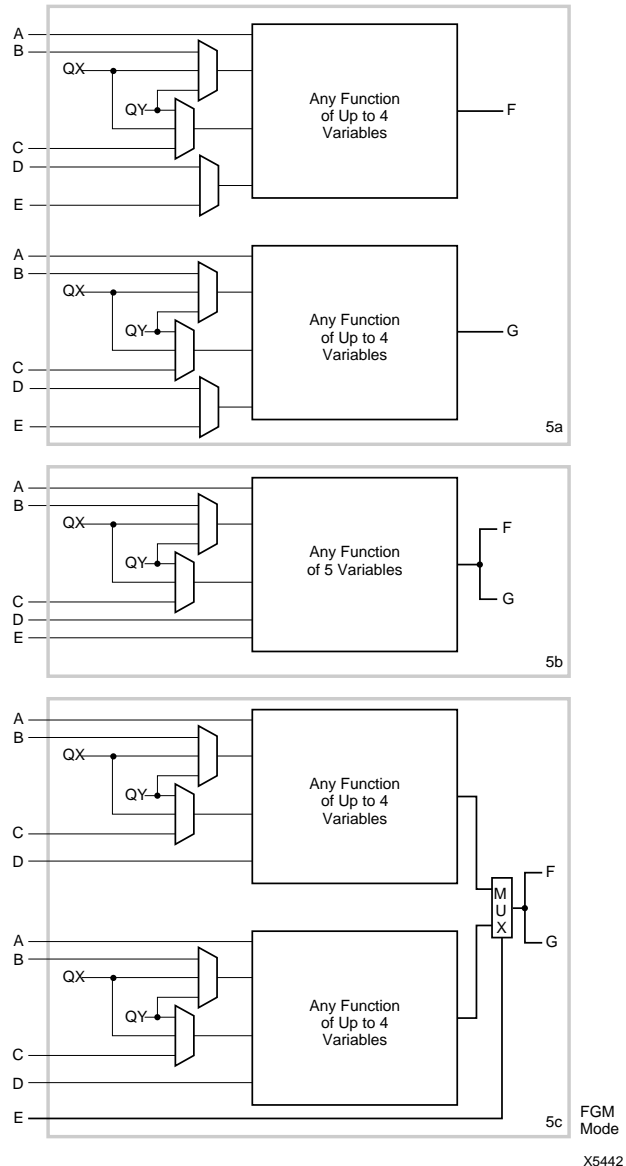
Figure 6: Combinational Logic Options

6a. Combinatorial Logic Option FG generates two functions of four variables each. One variable, A, must be common to both functions. The second and third variable can be any choice of B, C, QX and QY. The fourth variable can be any choice of D or E.

6b. Combinatorial Logic Option F generates any function of five variables: A, D, E and two choices out of B, C, QX, QY.

6c. Combinatorial Logic Option FGM allows variable E to select between two functions of four variables: Both have common inputs A and D and any choice out of B, C, QX and QY for the remaining two variables. Option 3 can then implement some functions of six or seven variables.
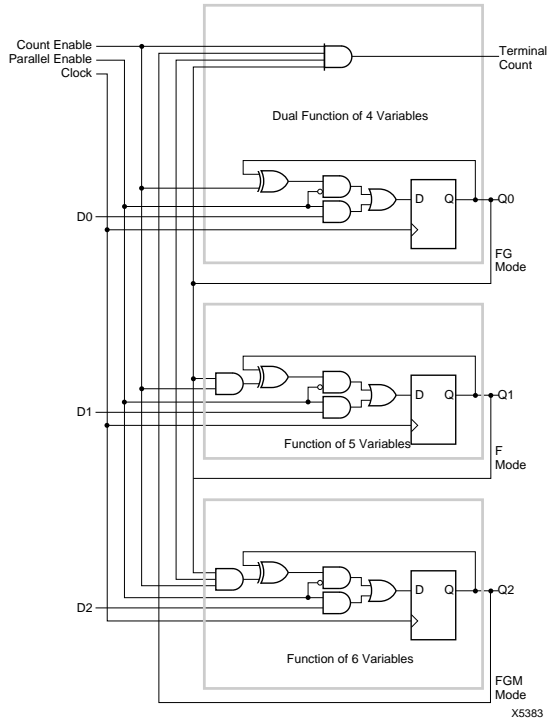
Figure 7: Counter.

The modulo-8 binary counter with parallel enable and clock enable uses one combinatorial logic block of each option.

General Purpose Interconnect

General purpose interconnect, as shown in Figure 10, consists of a grid of five horizontal and five vertical metal segments located between the rows and columns of logic and IOBs. Each segment is the height or width of a logic block. Switching matrices join the ends of these segments and allow programmed interconnections between the metal grid segments of adjoining rows and columns. The switches of an unprogrammed device are all non-conducting. The connections through the switch matrix may be established by the automatic routing or by selecting the desired pairs of matrix pins to be connected or disconnected. The legitimate switching matrix combinations for each pin are indicated in Figure 11.

Special buffers within the general interconnect areas provide periodic signal isolation and restoration for improved performance of lengthy nets. The interconnect buffers are available to propagate signals in either direction on a given general interconnect segment. These bidirectional (bidi) buffers are found adjacent to the switching matrices, above
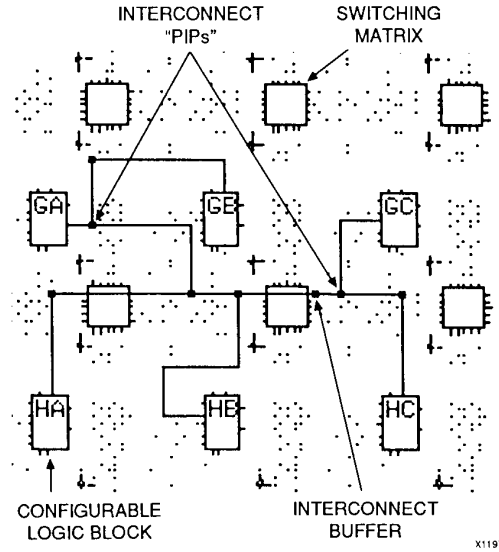

Figure 8: A Design Editor view of routing resources used to form a typical interconnection network from CLB GA.

and to the right. The other PIPs adjacent to the matrices are accessed to or from Longlines. The development system automatically defines the buffer direction based on the location of the interconnection network source. The delay calculator of the development system automatically calculates and displays the block, interconnect and buffer delays for any paths selected. Generation of the simulation netlist with a worst-case delay model is provided.

Direct Interconnect

Direct interconnect, shown in Figure 12, provides the most efficient implementation of networks between adjacent CLBs or I/O Blocks. Signals routed from block to block using the direct interconnect exhibit minimum interconnect propagation and use no general interconnect resources. For each CLB, the X output may be connected directly to the B input of the CLB immediately to its right and to the C input of the CLB to its left. The Y output can use direct interconnect to drive the D input of the block immediately above and the A input of the block below. Direct interconnect should be used to maximize the speed of high-performance portions of logic. Where logic blocks are adjacent to IOBs, direct connect is provided alternately to the IOB inputs (I) and outputs (O) on all four edges of the die. The right edge provides additional direct connects from CLB outputs to adjacent IOBs. Direct interconnections of IOBs with CLBs are shown in Figure 13.

7
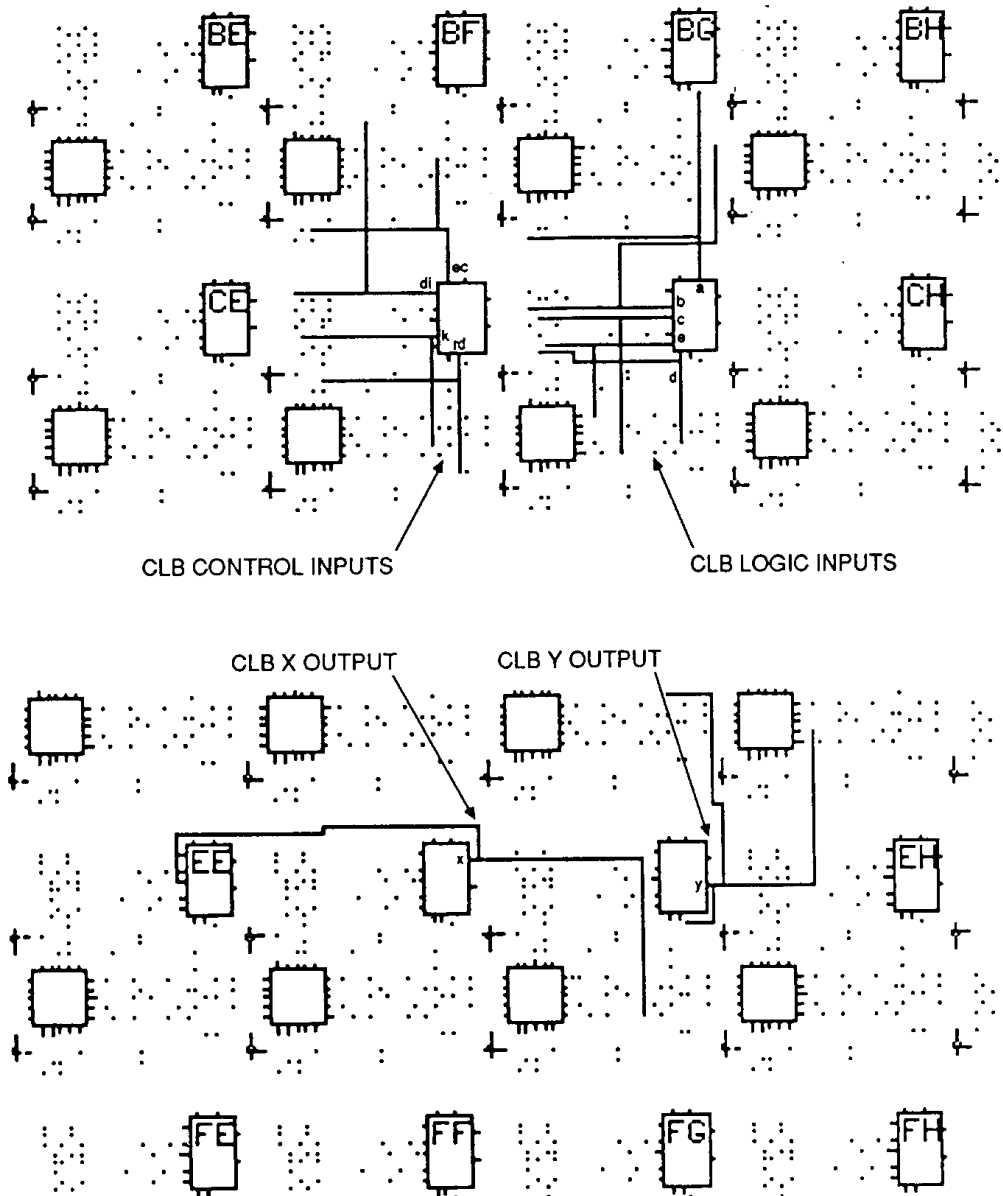
Figure 9:   Design Editor Locations of interconnect access, CLB control inputs, logic inputs and outputs. The dot pattern represents the available programmable interconnection points (PIPs).

Some of the interconnect PIPs are directional.
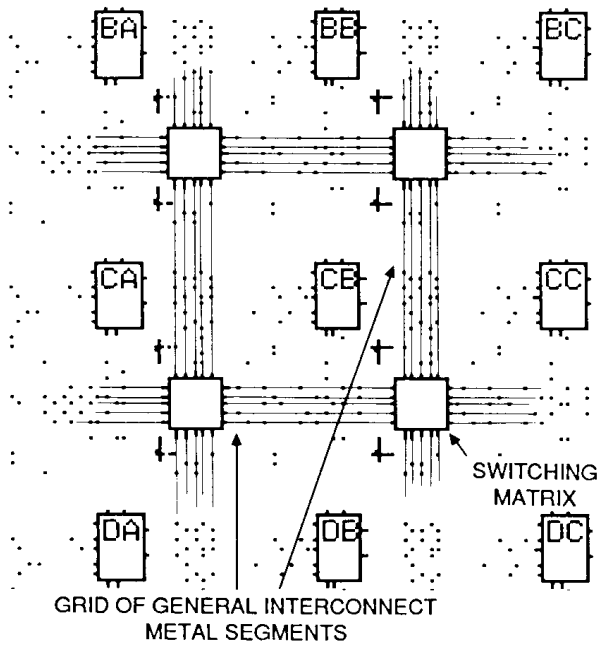
Figure 10: FPGA General-Purpose Interconnect.
Composed of a grid of metal segments that may be inter-connected through switch matrices to form networks for CLB and IOB inputs and outputs.
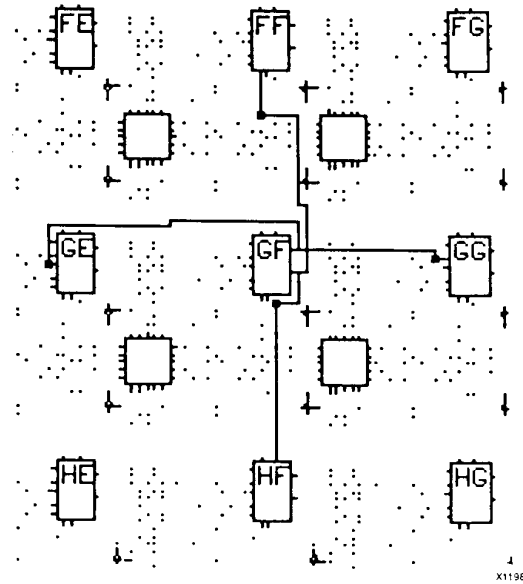
Figure 12: CLB X and Y Outputs.
The X and Y outputs of each CLB have single contact, direct access to inputs of adjacent CLBs

<div style="text-align:right">**7**</div>
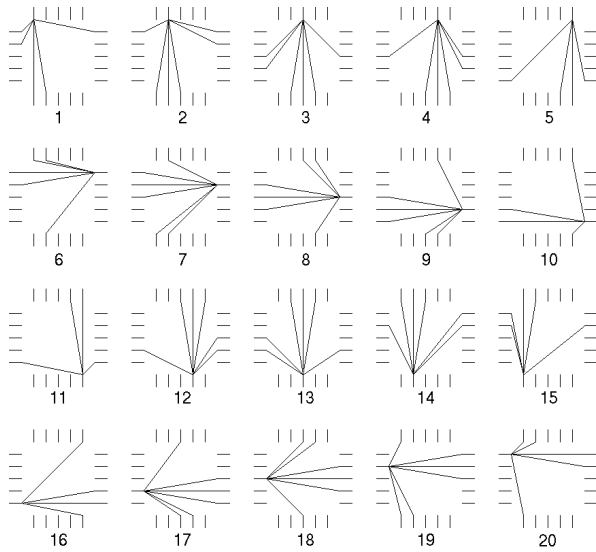
383 16

Figure 11: Switch Matrix Interconnection Options for Each Pin.
Switch matrices on the edges are different.

Global Buffer Direct Input

Global Buffer Inerconnect

* Unbonded IOBs (6 Places)

Alternate Buffer Direct Input
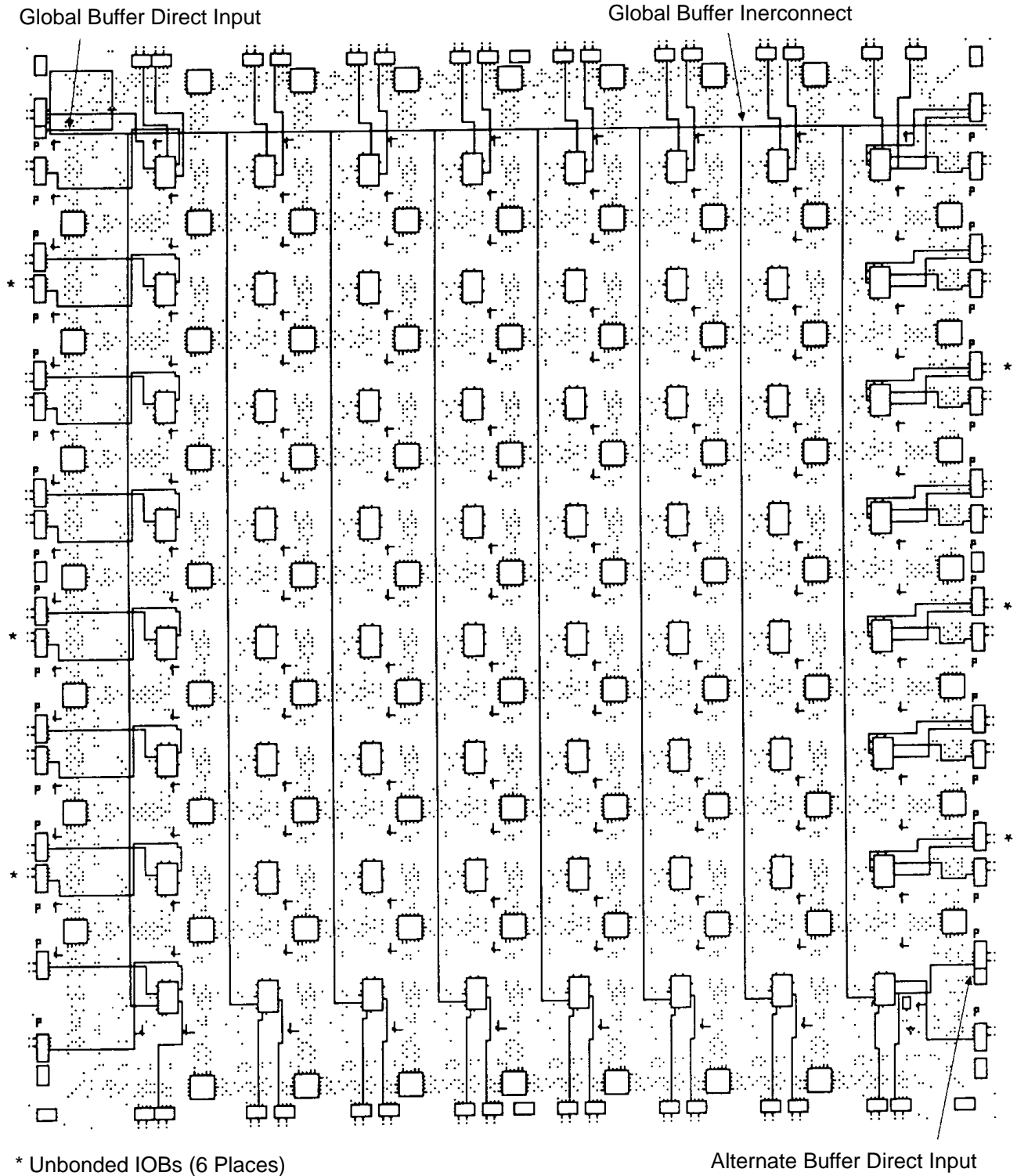
Figure 13:   XC3020A Die-Edge IOBs. The XC3020A die-edge IOBs are provided with direct access to adjacent CLBs.

Longlines

The Longlines bypass the switch matrices and are intended primarily for signals that must travel a long distance, or must have minimum skew among multiple destinations. Longlines, shown in Figure 14, run vertically and horizontally the height or width of the interconnect area. Each interconnection column has three vertical Longlines, and each interconnection row has two horizontal Longlines. Two additional Longlines are located adjacent to the outer sets of switching matrices. In devices larger than the XC3020A and XC3120A FPGAs, two vertical Longlines in each col-

umn are connectable half-length lines. On the XC3020A and XC3120A FPGAs, only the outer Longlines are connectable half-length lines.

Longlines can be driven by a logic block or IOB output on a column-by-column basis. This capability provides a common low skew control or clock line within each column of logic blocks. Interconnections of these Longlines are shown in Figure 15. Isolation buffers are provided at each input to a Longline and are enabled automatically by the development system when a connection is made.

7
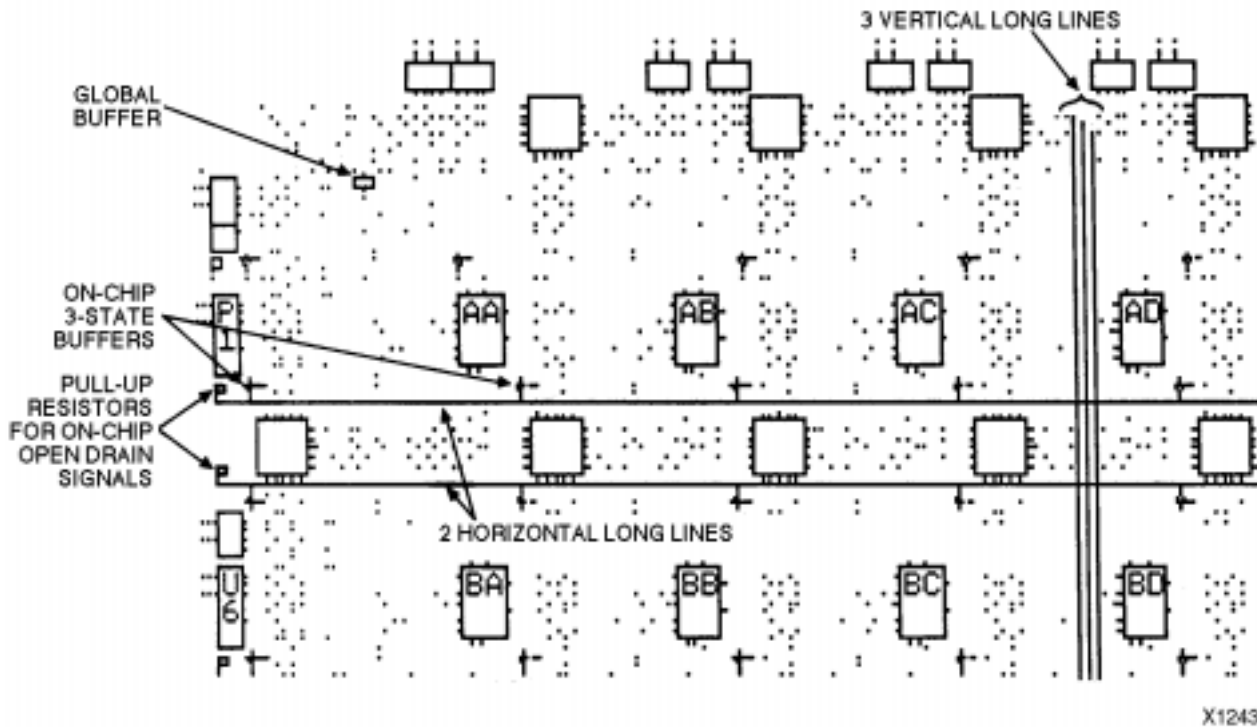
Figure 14:   Horizontal and Vertical Longlines. These Longlines provide high fan-out, low-skew signal distribution in each row and column. The global buffer in the upper left die corner drives a common line throughout the FPGA.

* FOUR OUTER LONG LINES ARE
CONNECTABLE HALF-LENGTH LINES

I/O BLOCK CLOCK NETS
(2 PER DIE EDGE)

HORIZONTAL
LONG LINES

3-STATE
BUFFERS

X1244 SCAN
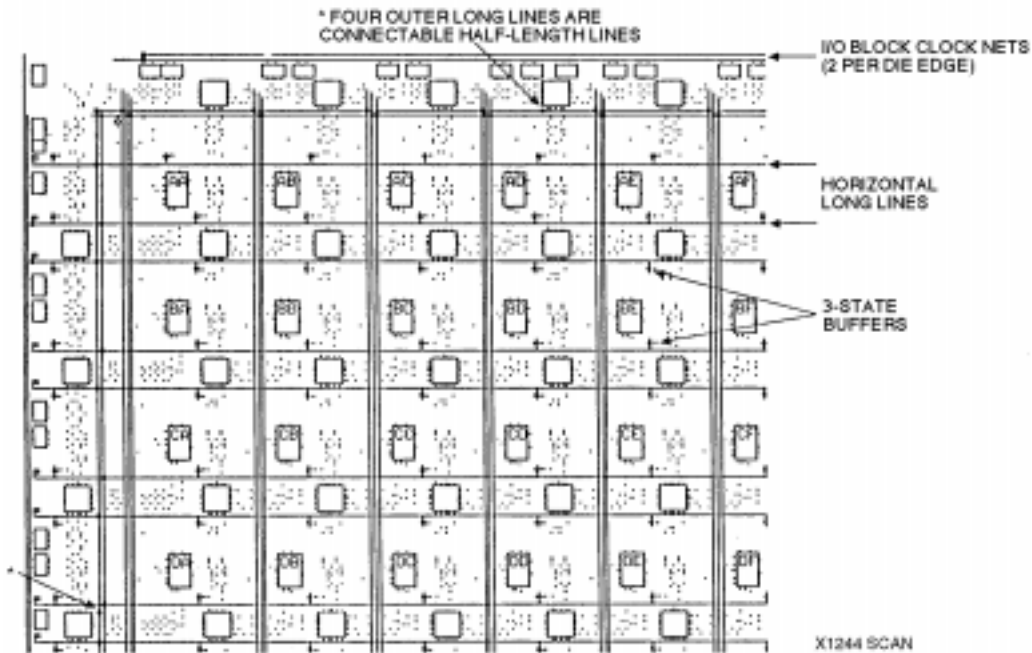
Figure 15: Programmable Interconnection of Longlines. This is provided at the edges of the routing area. Three-state buffers allow the use of horizontal Longlines to form on-chip wired AND and multiplexed buses. The left two non-clock vertical Longlines per column (except XC3020A) and the outer perimeter Longlines may be programmed as connectable half-length lines.

$$Z = D_A \bullet D_B \bullet D_C \bullet _{...} \bullet D_N$$

(LOW)

X3036
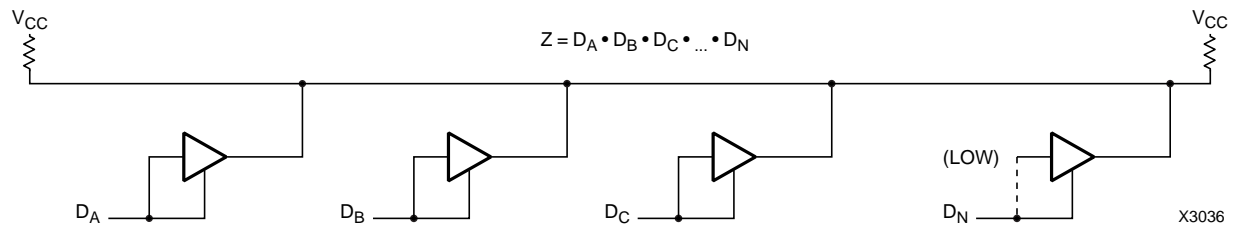
Figure 16: 3-State Buffers Implement a Wired-AND Function. When all the buffer 3-state lines are High, (high impedance), the pull-up resistor(s) provide the High output. The buffer inputs are driven by the control signals or a Low.

$$Z = D_A \bullet \overline{A} + D_B \bullet \overline{B} + D_C \bullet \overline{C} + ... + D_N \bullet \overline{N}$$

WEAK
KEEPER CIRCUIT

X1741A
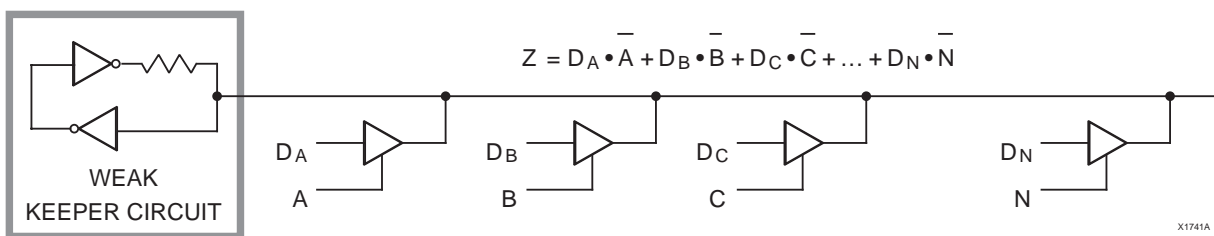
Figure 17: 3-State Buffers Implement a Multiplexer. The selection is accomplished by the buffer 3-state signal.

A buffer in the upper left corner of the FPGA chip drives a global net which is available to all K inputs of logic blocks. Using the global buffer for a clock signal provides a skew-free, high fan-out, synchronized clock for use at any or all of the IOBs and CLBs. Configuration bits for the K input to each logic block can select this global line or another routing resource as the clock source for its flip-flops. This net may also be programmed to drive the die edge clock lines for IOB use. An enhanced speed, CMOS threshold, direct access to this buffer is available at the second pad from the top of the left die edge.

A buffer in the lower right corner of the array drives a horizontal Longline that can drive programmed connections to a vertical Longline in each interconnection column. This alternate buffer also has low skew and high fan-out. The network formed by this alternate buffer's Longlines can be selected to drive the K inputs of the CLBs. CMOS threshold, high speed access to this buffer is available from the third pad from the bottom of the right die edge.

Internal Busses

A pair of 3-state buffers, located adjacent to each CLB, permits logic to drive the horizontal Longlines. Logic operation of the 3-state buffer controls allows them to implement wide multiplexing functions. Any 3-state buffer input can be selected as drive for the horizontal long-line bus by applying a Low logic level on its 3-state control line. See Figure 16. The user is required to avoid contention which can result from multiple drivers with opposing logic levels. Control of the 3-state input by the same signal that drives the buffer input, creates an open-drain wired-AND function. A logic High on both buffer inputs creates a high impedance, which represents no contention. A logic Low enables the buffer to drive the Longline Low. See Figure 17. Pull-up resistors are available at each end of the Longline to provide a High output when all connected buffers are non-conducting. This forms fast, wide gating functions. When data drives the inputs, and separate signals drive the 3-state control lines, these buffers form multiplexers (3-state busses). In this case, care must be used to prevent contention through multiple active buffers of conflicting levels on a common line. Each horizontal Longline is also driven by a weak keeper circuit that prevents undefined floating levels by maintaining the previous logic level when the line is not driven by an active buffer or a pull-up resistor. Figure 18 shows 3-state buffers, Longlines and pull-up resistors.
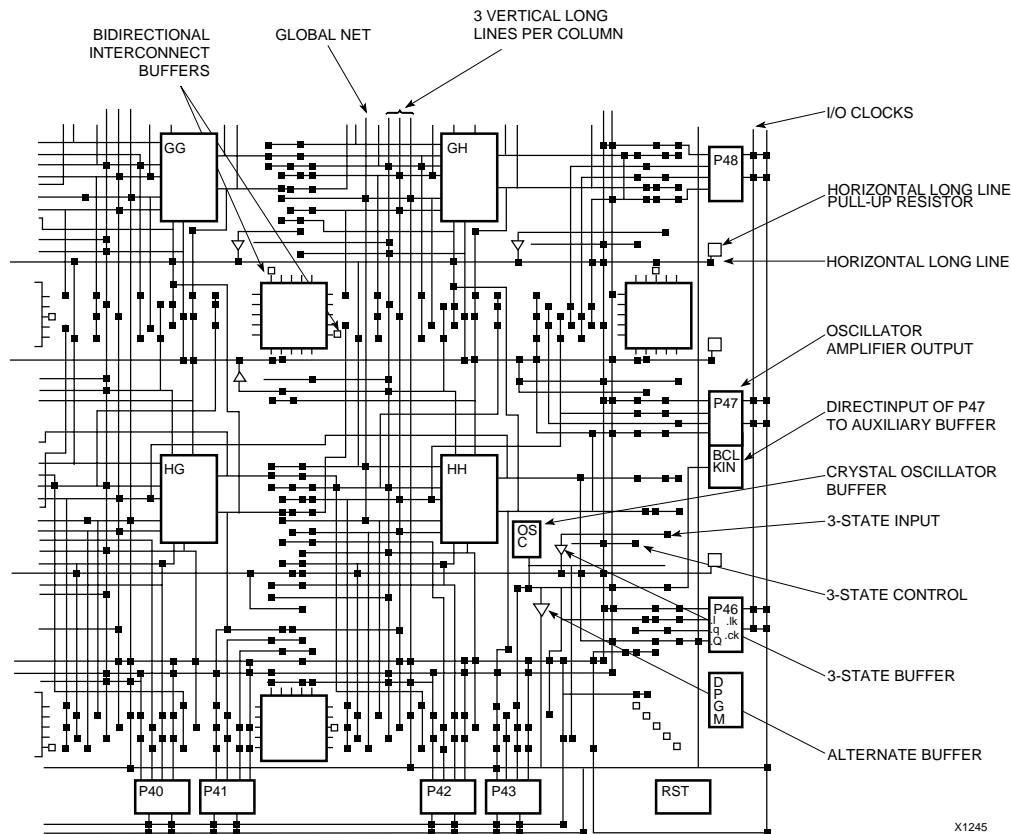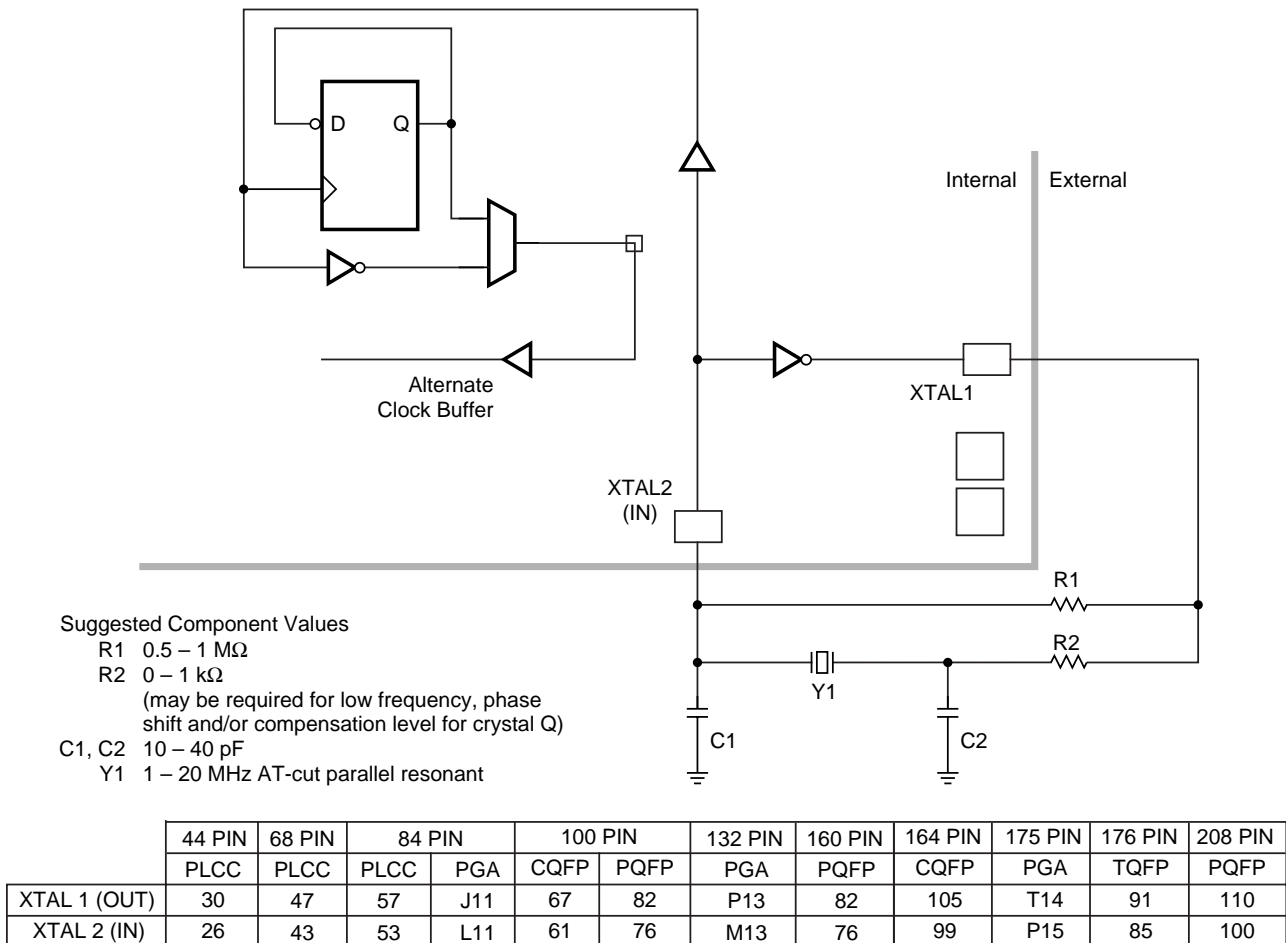

Figure 18: Design Editor.
An extra large view of possible interconnections in the lower right corner of the XC3020A.

Crystal Oscillator

Figure 18 also shows the location of an internal high speed inverting amplifier that may be used to implement an on-chip crystal oscillator. It is associated with the auxiliary buffer in the lower right corner of the die. When the oscillator is configured and connected as a signal source, two special user IOBs are also configured to connect the oscillator amplifier with external crystal oscillator components as shown in Figure 19. A divide by two option is available to assure symmetry. The oscillator circuit becomes active early in the configuration process to allow the oscillator to stabilize. Actual internal connection is delayed until completion of configuration. In Figure 19 the feedback resistor R1, between the output and input, biases the amplifier at threshold. The inversion of the amplifier, together with the R-C networks and an AT-cut series resonant crystal, produce the 360-degree phase shift of the Pierce oscillator. A

series resistor R2 may be included to add to the amplifier output impedance when needed for phase-shift control, crystal resistance matching, or to limit the amplifier input swing to control clipping at large amplitudes. Excess feedback voltage may be corrected by the ratio of C2/C1. The amplifier is designed to be used from 1 MHz to about one-half the specified CLB toggle frequency. Use at frequencies below 1 MHz may require individual characterization with respect to a series resistance. Crystal oscillators above 20 MHz generally require a crystal which operates in a third overtone mode, where the fundamental frequency must be suppressed by an inductor across C2, turning this parallel resonant circuit to double the fundamental crystal frequency, i.e., 2/3 of the desired third harmonic frequency network. When the oscillator inverter is not used, these IOBs and their package pins are available for general user I/O.

Suggested Component Values
R1   0.5 – 1 MΩ
R2   0 – 1 kΩ
(may be required for low frequency, phase shift and/or compensation level for crystal Q)
C1, C2   10 – 40 pF
Y1   1 – 20 MHz AT-cut parallel resonant

	44 PIN	68 PIN	84 PIN		100 PIN		132 PIN	160 PIN	164 PIN	175 PIN	176 PIN	208 PIN
	PLCC	PLCC	PLCC	PGA	CQFP	PQFP	PGA	PQFP	CQFP	PGA	TQFP	PQFP
XTAL 1 (OUT)	30	47	57	J11	67	82	P13	82	105	T14	91	110
XTAL 2 (IN)	26	43	53	L11	61	76	M13	76	99	P15	85	100

X7064

Figure 19:   Crystal Oscillator Inverter. When activated, and by selecting an output network for its buffer, the crystal oscillator inverter uses two unconfigured package pins and external components to implement an oscillator. An optional divide-by-two mode is available to assure symmetry.

ΣXILINX®

Configuration

Initialization Phase

An internal power-on-reset circuit is triggered when power is applied. When V_{CC} reaches the voltage at which portions of the FPGA device begin to operate (nominally 2.5 to 3 V), the programmable I/O output buffers are 3-stated and a high-impedance pull-up resistor is provided for the user I/O pins. A time-out delay is initiated to allow the power supply voltage to stabilize. During this time the power-down mode is inhibited. The Initialization state time-out (about 11 to 33 ms) is determined by a 14-bit counter driven by a self-generated internal timer. This nominal 1-MHz timer is subject to variations with process, temperature and power supply. As shown in Table 1, five configuration mode choices are available as determined by the input levels of three mode pins; M0, M1 and M2.

Table 1: Configuration Mode Choices

M0	M1	M2	CCLK	Mode	Data
0	0	0	output	Master	Bit Serial
0	0	1	output	Master	Byte Wide Addr. = 0000 up
0	1	0	—	reserved	—
0	1	1	output	Master	Byte Wide Addr. = FFFF down
1	0	0	—	reserved	—
1	0	1	output	Peripheral	Byte Wide
1	1	0	—	reserved	—
1	1	1	input	Slave	Bit Serial

In Master configuration modes, the device becomes the source of the Configuration Clock (CCLK). The beginning of configuration of devices using Peripheral or Slave modes must be delayed long enough for their initialization to be completed. An FPGA with mode lines selecting a Master configuration mode extends its initialization state using four times the delay (43 to 130 ms) to assure that all daisy-chained slave devices, which it may be driving, will be ready even if the master is very fast, and the slave(s) very slow. Figure 20 shows the state sequences. At the end of Initialization, the device enters the Clear state where it clears the configuration memory. The active Low, open-drain initialization signal INIT indicates when the Initialization and Clear states are complete. The FPGA tests for the absence of an external active Low RESET before it makes a final sample of the mode lines and enters the Configuration state. An external wired-AND of one or more INIT pins can be used to control configuration by the assertion of the active-Low RESET of a master mode device or to signal a processor that the FPGAs are not yet initialized.

If a configuration has begun, a re-assertion of RESET for a minimum of three internal timer cycles will be recognized and the FPGA will initiate an abort, returning to the Clear state to clear the partially loaded configuration memory words. The FPGA will then resample RESET and the mode lines before re-entering the Configuration state.

During configuration, the XC3000A, XC3000L, XC3100A, and XC3100L devices check the bit-stream format for stop bits in the appropriate positions. Any error terminates the configuration and pulls INIT Low.
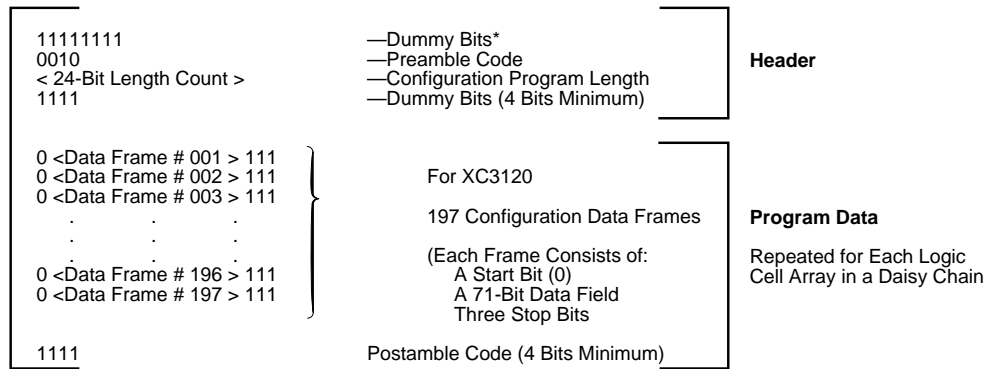
7
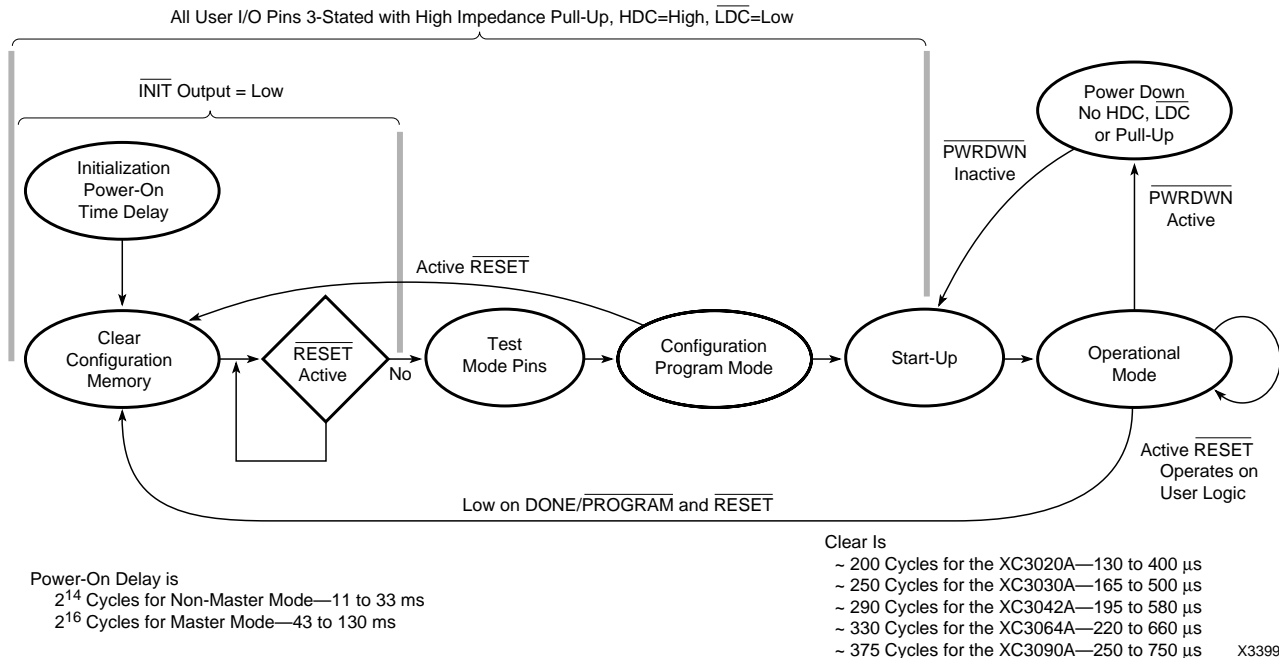
Figure 20: A State Diagram of the Configuration Process for Power-up and Reprogram.

A re-program is initiated.when a configured XC3000 series device senses a High-to-Low transition and subsequent >6 μs Low level on the DONE/PROG package pin, or, if this pin is externally held permanently Low, a High-to-Low transition and subsequent >6 μs Low time on the RESET package pin.

The device returns to the Clear state where the configuration memory is cleared and mode lines re-sampled, as for an aborted configuration. The complete configuration program is cleared and loaded during each configuration program cycle.

Length count control allows a system of multiple Field Programmable Gate Arrays, of assorted sizes, to begin operation in a synchronized fashion. The configuration program generated by the development system begins with a preamble of 111111110010 followed by a 24-bit length count representing the total number of configuration clocks needed to complete loading of the configuration program(s). The data framing is shown in Figure 21. All FPGAs connected in series read and shift preamble and length count in on positive and out on negative configuration clock edges. A device which has received the preamble and length count then presents a High Data Out until it has intercepted the appropriate number of data frames. When the configuration program memory of an FPGA is full and the length count does not yet compare, the device shifts any additional data through, as it did for preamble and length count. When the FPGA configuration memory is full and the length count compares, the device will execute

```
11111111          —Dummy Bits*
0010              —Preamble Code                              Header
< 24-Bit Length Count >  —Configuration Program Length
1111              —Dummy Bits (4 Bits Minimum)

0 <Data Frame # 001 > 111
0 <Data Frame # 002 > 111    For XC3120
0 <Data Frame # 003 > 111
      .    .    .             197 Configuration Data Frames    Program Data
      .    .    .
      .    .    .             (Each Frame Consists of:          Repeated for Each Logic
0 <Data Frame # 196 > 111        A Start Bit (0)               Cell Array in a Daisy Chain
0 <Data Frame # 197 > 111        A 71-Bit Data Field
                                 Three Stop Bits

1111                         Postamble Code (4 Bits Minimum)
```

*The LCA Device Require Four Dummy Bits Min; Software Generates Eight Dummy Bits

X5300_01

Device	XC3020A XC3020L XC3120A	XC3030A XC3030L XC3130A	XC3042A XC3042L XC3142A XC3142L	XC3064A XC3064L XC3164A	XC3090A XC3090L XC3190A XC3190L	XC3195A
Gates	1,000 to 1,500	1,500 to 2,000	2,000 to 3,000	3,500 to 4,500	5,000 to 6,000	6,500 to 7,500
CLBs	64	100	144	224	320	484
Row x Col	(8 x 8)	(10 x 10)	(12 x 12)	(16 x 14)	(20 x 16)	(22 x 22)
IOBs	64	80	96	120	144	176
Flip-flops	256	360	480	688	928	1,320
Horizontal Longlines	16	20	24	32	40	44
TBUFs/Horizontal LL	9	11	13	15	17	23
Bits per Frame (including1 start and 3 stop bits)	75	92	108	140	172	188
Frames	197	241	285	329	373	505
Program Data = Bits x Frames + 4 bits (excludes header)	14,779	22,176	30,784	46,064	64,160	94,944
PROM size (bits) = Program Data + 40-bit Header	14,819	22,216	30,824	46,104	64,200	94,984

Figure 21: Internal Configuration Data Structure for an FPGA. This shows the preamble, length count and data frames generated by the Development System.

The Length Count produced by the program = [(40-bit preamble + sum of program data + 1 per daisy chain device) rounded up to multiple of 8] – ($2 \leq K \leq 4$) where K is a function of DONE and RESET timing selected. An additional 8 is added if roundup increment is less than K. K additional clocks are needed to complete start-up after length count is reached.

a synchronous start-up sequence and become operational. See Figure 22. Two CCLK cycles after the completion of loading configuration data, the user I/O pins are enabled as configured. As selected, the internal user-logic RESET is released either one clock cycle before or after the I/O pins become active. A similar timing selection is programmable for the DONE/PROG output signal. DONE/PROG may also be programmed to be an open drain or include a pull-up resistor to accommodate wired ANDing. The High During Configuration (HDC) and Low During Configuration (LDC) are two user I/O pins which are driven active while an FPGA is in its Initialization, Clear or Configure states. They and DONE/PROG provide signals for control of external logic signals such as RESET, bus enable or PROM enable during configuration. For parallel Master configuration modes, these signals provide PROM enable control and allow the data pins to be shared with user logic signals.

User I/O inputs can be programmed to be either TTL or CMOS compatible thresholds. At power-up, all inputs have TTL thresholds and can change to CMOS thresholds at the completion of configuration if the user has selected CMOS thresholds. The threshold of PWRDWN and the direct clock inputs are fixed at a CMOS level.

If the crystal oscillator is used, it will begin operation before configuration is complete to allow time for stabilization before it is connected to the internal circuitry.

Configuration Data

Configuration data to define the function and interconnection within a Field Programmable Gate Array is loaded from an external storage at power-up and after a re-program signal. Several methods of automatic and controlled loading of the required data are available. Logic levels applied to mode selection pins at the start of configuration time determine the method to be used. See Table 1. The data may be either bit-serial or byte-parallel, depending on the configuration mode. The different FPGAs have different sizes and numbers of data frames. To maintain compatibility between various device types, the Xilinx product families use compatible configuration formats. For the XC3020A, configuration requires 14779 bits for each device, arranged in 197 data frames. An additional 40 bits are used in the header. See Figure 22. The specific data format for each device is produced by the development system and one or more of these files can then be combined and appended to a length count preamble and be transformed into a PROM format file by the development system. A compatibility exception precludes the use of an XC2000-series device as the master for XC3000-series devices if their DONE or RESET are programmed to occur after their outputs become active. The Tie Option defines output levels of unused blocks of a design and connects these to unused routing resources. This prevents indeterminate levels that might produce parasitic supply currents. If unused blocks are not sufficient to complete the tie, the user can indicate nets which must not

7
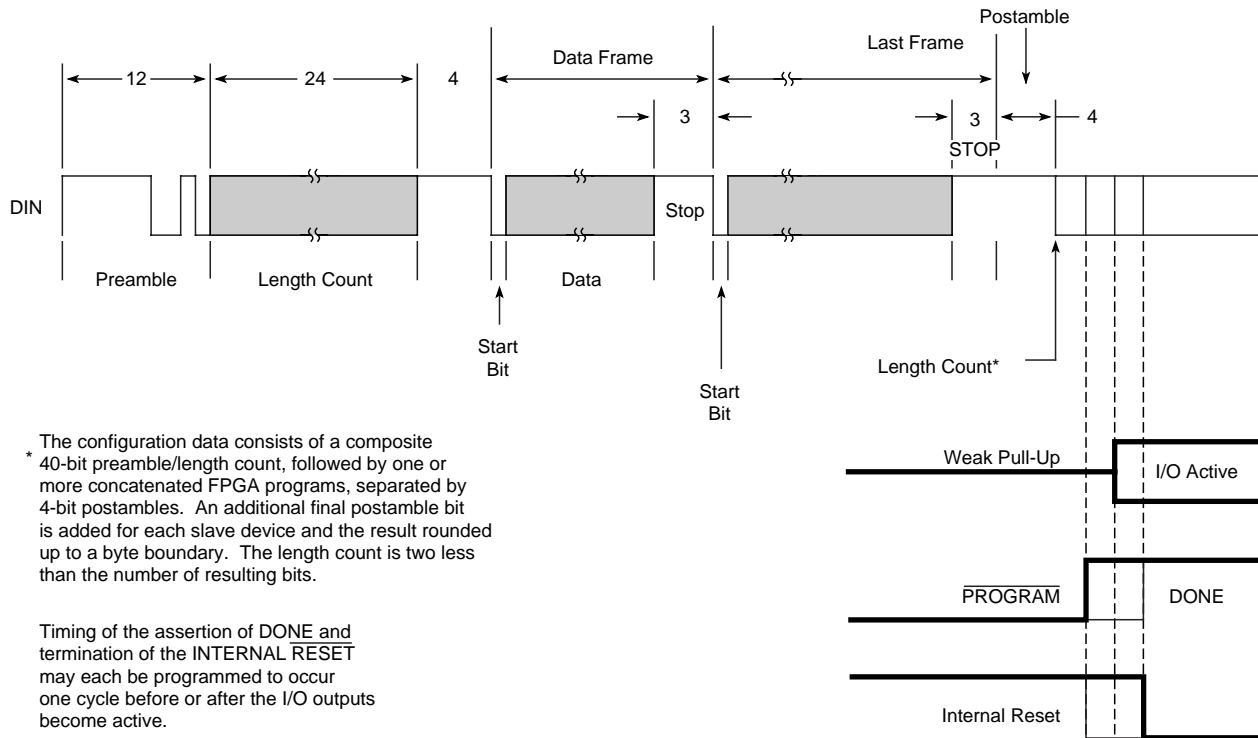
* The configuration data consists of a composite 40-bit preamble/length count, followed by one or more concatenated FPGA programs, separated by 4-bit postambles. An additional final postamble bit is added for each slave device and the result rounded up to a byte boundary. The length count is two less than the number of resulting bits.

Timing of the assertion of DONE and termination of the INTERNAL RESET may each be programmed to occur one cycle before or after the I/O outputs become active.

Heavy lines indicate the default condition

X5988

Figure 22: Configuration and Start-up of One or More FPGAs.

be used to drive the remaining unused routing, as that might affect timing of user nets. Tie can be omitted for quick breadboard iterations where a few additional milliamps of Icc are acceptable.

The configuration bitstream begins with eight High preamble bits, a 4-bit preamble code and a 24-bit length count. When configuration is initiated, a counter in the FPGA is set to zero and begins to count the total number of configuration clock cycles applied to the device. As each configuration data frame is supplied to the device, it is internally assembled into a data word, which is then loaded in parallel into one word of the internal configuration memory array. The configuration loading process is complete when the current length count equals the loaded length count and the required configuration program data frames have been written. Internal user flip-flops are held Reset during configuration.

Two user-programmable pins are defined in the unconfigured Field Programmable Gate Array. High During Configuration (HDC) and Low During Configuration (LDC) as well as DONE/PROG may be used as external control signals during configuration. In Master mode configurations it is convenient to use LDC as an active-Low EPROM Chip Enable. After the last configuration data bit is loaded and the length count compares, the user I/O pins become active. Options allow timing choices of one clock earlier or later for the timing of the end of the internal logic RESET and the assertion of the DONE signal. The open-drain DONE/PROG output can be AND-tied with multiple devices and used as an active-High READY, an active-Low PROM enable or a RESET to other portions of the system. The state diagram of Figure 20 illustrates the configuration process.

Configuration Modes

Master Mode

In Master mode, the FPGA automatically loads configuration data from an external memory device. There are three Master modes that use the internal timing source to supply the configuration clock (CCLK) to time the incoming data. Master Serial mode uses serial configuration data supplied to Data-in (DIN) from a synchronous serial source such as the Xilinx Serial Configuration PROM shown in Figure 23. Master Parallel Low and High modes automatically use parallel data supplied to the D0–D7 pins in response to the 16-bit address generated by the FPGA. Figure 25 shows an example of the parallel Master mode connections required. The HEX starting address is 0000 and increments for Master Low mode and it is FFFF and decrements for Master High mode. These two modes provide address compatibility with microprocessors which begin execution from opposite ends of memory.

Peripheral Mode

Peripheral mode provides a simplified interface through which the device may be loaded byte-wide, as a processor peripheral. Figure 27 shows the peripheral mode connections. Processor write cycles are decoded from the common assertion of the active low Write Strobe (WS), and two active low and one active high Chip Selects (CS0, CS1, CS2). The FPGA generates a configuration clock from the internal timing generator and serializes the parallel input data for internal framing or for succeeding slaves on Data Out (DOUT). A output High on READY/BUSY pin indicates the completion of loading for each byte when the input register is ready for a new byte. As with Master modes, Peripheral mode may also be used as a lead device for a daisy-chain of slave devices.

Slave Serial Mode

Slave Serial mode provides a simple interface for loading the Field Programmable Gate Array configuration as shown in Figure 29. Serial data is supplied in conjunction with a synchronizing input clock. Most Slave mode applications are in daisy-chain configurations in which the data input is driven from the previous FPGA's data out, while the clock is supplied by a lead device in Master or Peripheral mode. Data may also be supplied by a processor or other special circuits.

Daisy Chain

The development system is used to create a composite configuration for selected FPGAs including: a preamble, a length count for the total bitstream, multiple concatenated data programs and a postamble plus an additional fill bit per device in the serial chain. After loading and passing-on the preamble and length count to a possible daisy-chain, a lead device will load its configuration data frames while providing a High DOUT to possible down-stream devices as shown in Figure 25. Loading continues while the lead device has received its configuration program and the current length count has not reached the full value. The additional data is passed through the lead device and appears on the Data Out (DOUT) pin in serial form. The lead device also generates the Configuration Clock (CCLK) to synchronize the serial output data and data in of down-stream FPGAs. Data is read in on DIN of slave devices by the positive edge of CCLK and shifted out the DOUT on the negative edge of CCLK. A parallel Master mode device uses its internal timing generator to produce an internal CCLK of 8 times its EPROM address rate, while a Peripheral mode device produces a burst of 8 CCLKs for each chip select and write-strobe cycle. The internal timing generator continues to operate for general timing and synchronization of inputs in all modes.

Special Configuration Functions

The configuration data includes control over several special functions in addition to the normal user logic functions and interconnect.

- Input thresholds
- Readback disable
- DONE pull-up resistor
- DONE timing
- RESET timing
- Oscillator frequency divided by two

Each of these functions is controlled by configuration data bits which are selected as part of the normal development system bitstream generation process.

Input Thresholds

Prior to the completion of configuration all FPGA input thresholds are TTL compatible. Upon completion of configuration, the input thresholds become either TTL or CMOS compatible as programmed. The use of the TTL threshold option requires some additional supply current for threshold shifting. The exception is the threshold of the PWRDWN input and direct clocks which always have a CMOS input. Prior to the completion of configuration the user I/O pins each have a high impedance pull-up. The configuration program can be used to enable the IOB pull-up resistors in the Operational mode to act either as an input load or to avoid a floating input on an otherwise unused pin.

Readback

The contents of a Field Programmable Gate Array may be read back if it has been programmed with a bitstream in which the Readback option has been enabled. Readback may be used for verification of configuration and as a method of determining the state of internal logic nodes during debugging. There are three options in generating the configuration bitstream.

- "Never" inhibits the Readback capability.
- "One-time," inhibits Readback after one Readback has been executed to verify the configuration.
- "On-command" allows unrestricted use of Readback.

Readback is accomplished without the use of any of the user I/O pins; only M0, M1 and CCLK are used. The initiation of Readback is produced by a Low to High transition of the M0/RTRIG (Read Trigger) pin. The CCLK input must then be driven by external logic to read back the configuration data. The first three Low-to-High CCLK transitions clock out dummy data. The subsequent Low-to-High CCLK transitions shift the data frame information out on the M1/RDATA (Read Data) pin. Note that the logic polarity is always inverted, a zero in configuration becomes a one in Readback, and vice versa. Note also that each Readback frame has one Start bit (read back as a one) but, unlike in configuration, each Readback frame has only one Stop bit (read back as a zero). The third leading dummy bit mentioned above can be considered the Start bit of the first frame. All data frames must be read back to complete the process and return the Mode Select and CCLK pins to their normal functions.

Readback data includes the current state of each CLB flip-flop, each input flip-flop or latch, and each device pad. These data are imbedded into unused configuration bit positions during Readback. This state information is used by the development system In-Circuit Verifier to provide visibility into the internal operation of the logic while the system is operating. To readback a uniform time-sample of all storage elements, it may be necessary to inhibit the system clock.

Reprogram

To initiate a re-programming cycle, the dual-function pin DONE/$\overline{\text{PROG}}$ must be given a High-to-Low transition. To reduce sensitivity to noise, the input signal is filtered for two cycles of the FPGA internal timing generator. When reprogram begins, the user-programmable I/O output buffers are disabled and high-impedance pull-ups are provided for the package pins. The device returns to the Clear state and clears the configuration memory before it indicates 'initialized'. Since this Clear operation uses chip-individual internal timing, the master might complete the Clear operation and then start configuration before the slave has completed the Clear operation. To avoid this problem, the slave $\overline{\text{INIT}}$ pins must be AND-wired and used to force a $\overline{\text{RESET}}$ on the master (see Figure 25). Reprogram control is often implemented using an external open-collector driver which pulls DONE/$\overline{\text{PROG}}$ Low. Once a stable request is recognized, the DONE/$\overline{\text{PROG}}$ pin is held Low until the new configuration has been completed. Even if the re-program request is externally held Low beyond the configuration period, the FPGA will begin operation upon completion of configuration.

DONE Pull-up

DONE/$\overline{\text{PROG}}$ is an open-drain I/O pin that indicates the FPGA is in the operational state. An optional internal pull-up resistor can be enabled by the user of the development system. The DONE/$\overline{\text{PROG}}$ pins of multiple FPGAs in a daisy-chain may be connected together to indicate all are DONE or to direct them all to reprogram.

DONE Timing

The timing of the DONE status signal can be controlled by a selection to occur either a CCLK cycle before, or after, the outputs going active. See Figure 22. This facilitates control of external functions such as a PROM enable or holding a system in a wait state.

7

RESET Timing

As with DONE timing, the timing of the release of the internal reset can be controlled to occur either a CCLK cycle before, or after, the outputs going active. See Figure 22. This reset keeps all user programmable flip-flops and latches in a zero state during configuration.

Crystal Oscillator Division

A selection allows the user to incorporate a dedicated divide-by-two flip-flop between the crystal oscillator and the alternate clock line. This guarantees a symmetrical clock signal. Although the frequency stability of a crystal oscillator is very good, the symmetry of its waveform can be affected by bias or feedback drive.

Bitstream Error Checking

Bitstream error checking protects against erroneous configuration.

Each Xilinx FPGA bitstream consists of a 40-bit preamble, followed by a device-specific number of data frames. The number of bits per frame is also device-specific; however, each frame ends with three stop bits (111) followed by a start bit for the next frame (0).

All devices in all XC3000 families start reading in a new frame when they find the first 0 after the end of the previous frame. An original XC3000 device does not check for the correct stop bits, but XC3000A, XC3100A, XC3000L, and XC3100L devices check that the last three bits of any frame are actually 111.

Under normal circumstances, all these FPGAs behave the same way; however, if the bitstream is corrupted, an XC3000 device will always start a new frame as soon as it finds the first 0 after the end of the previous frame, even if the data is completely wrong or out-of-sync. Given sufficient zeros in the data stream, the device will also go Done, but with incorrect configuration and the possibility of internal contention.

An XC3000A/XC3100A/XC3000L/XC3100L device starts any new frame only if the three preceding bits are all ones. If this check fails, it pulls INIT Low and stops the internal configuration, although the Master CCLK keeps running. The user must then start a new configuration by applying a >6 µs Low level on RESET.

This simple check does not protect against random bit errors, but it offers almost 100 percent protection against erroneous configuration files, defective configuration data sources, synchronization errors between configuration source and FPGA, or PC-board level defects, such as broken lines or solder-bridges.

Reset Spike Protection

A separate modification slows down the RESET input before configuration by using a two-stage shift register driven from the internal clock. It tolerates submicrosecond High spikes on RESET before configuration. The XC3000 master can be connected like an XC4000 master, but with its RESET input used instead of INIT. (On XC3000, INIT is output only).

Soft Start-up

After configuration, the outputs of all FPGAs in a daisy-chain become active simultaneously, as a result of the same CCLK edge. In the original XC3000/3100 devices, each output becomes active in either fast or slew-rate limited mode, depending on the way it is configured. This can lead to large ground-bounce signals. In XC3000A, XC3000L, XC3100A, and XC3100L devices, all outputs become active first in slew-rate limited mode, reducing the ground bounce. After this soft start-up, each individual output slew rate is again controlled by the respective configuration bit.

Configuration Timing

This section describes the configuration modes in detail.

Master Serial Mode

In Master Serial mode, the CCLK output of the lead FPGA drives a Xilinx Serial PROM that feeds the 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 DIN pin. The lead FPGA accepts this data on the subsequent rising CCLK edge.

The lead FPGA then presents the preamble data (and all data that overflows the lead device) on its DOUT pin. There is an internal delay of 1.5 CCLK periods, which means that

DOUT changes on the falling CCLK edge, and the next device in the daisy-chain accepts data on the subsequent rising CCLK edge.

The SPROM CE input can be driven from either $\overline{LDC}$ or DONE. Using $\overline{LDC}$ avoids potential contention on the DIN pin, if this pin is configured as user-I/O, but $\overline{LDC}$ is then restricted to be a permanently High user output. Using DONE also avoids contention on DIN, provided the early DONE option is invoked.

Figure 23: Master Serial Mode Circuit Diagram

X3223

	Description		Symbol	Min	Max	Units
CCLK	Data In setup	1	T_{DSCK}	60		ns
	Data In hold	2	C_{KDS}	0		ns

Notes: 1. At power-up, V_{CC} must rise from 2.0 V to V_{CC} min in less than 25 ms. If this is not possible, configuration can be delayed by holding RESET Low until V_{CC} has reached 4.0 V (2.5 V for the XC3000L). A very long V_{CC} rise time of >100 ms, or a non-monotonically rising V_{CC} may require >6-μs High level on RESET, followed by a >6-μs Low level on $\overline{RESET}$ and D/$\overline{P}$ after VCC has reached 4.0 V (2.5 V for the XC3000L).
2. Configuration can be controlled by holding $\overline{RESET}$ Low with or until after the $\overline{INIT}$ of all daisy-chain slave-mode devices is High.
3. Master-serial-mode timing is based on slave-mode testing.

Figure 24: Master Serial Mode Programming Switching Characteristics

Master Parallel Mode

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

The eight data bits are serialized in the lead FPGA, which then presents the preamble data (and all data that overflows the lead device) on the DOUT pin. There is an inter-

nal delay of 1.5 CCLK periods, after the rising CCLK edge that accepts a byte of data, and also changes the EPROM address, until the falling CCLK edge that makes the LSB (D0) of this byte appear at DOUT. This means that DOUT changes on the falling CCLK edge, and the next device in the daisy chain accepts data on the subsequent rising CCLK edge.

Figure 25: Master Parallel Mode Circuit Diagram