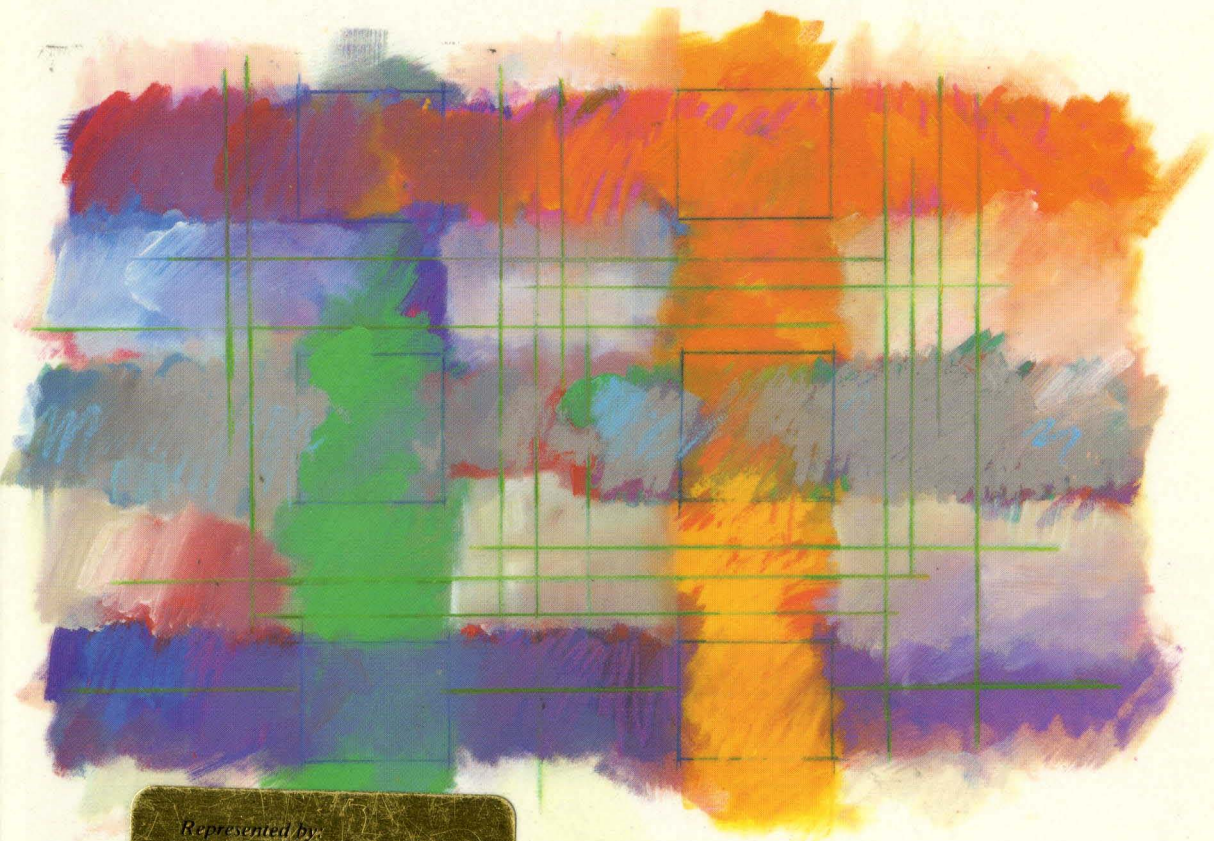
XILINX

The Programmable Gate Array Data Book



1991

The Programmable Gate Array Data Book

2100 Logic Drive
San Jose, California 95124
Telephone: (408) 559-7778

Contributors

Peter Alfke
Nick Camilleri
Jim Chumbley
Atsuko D'Amour
Rick Dudley
Abu Eghan
Chuck Erickson
Lee Farrell
Brad Fawcett
Chuck Fox
Dave Galli
Jim Hsieh
Clay Johnson
Steve Knapp
Dave Lautzenheiser
Mark Markham
Farid Mazouni
Bob McGrath
Bill O'Neill
George Nelson
Wes Patterson
Richard Ravel
Ed Resler
Randy Saldinger
Steve Schreifels
Robert Stransky
Mustafa Vezioroglu
Thomas Waugh
Perry Wu

XILINX

SECTION TITLES

XILINX

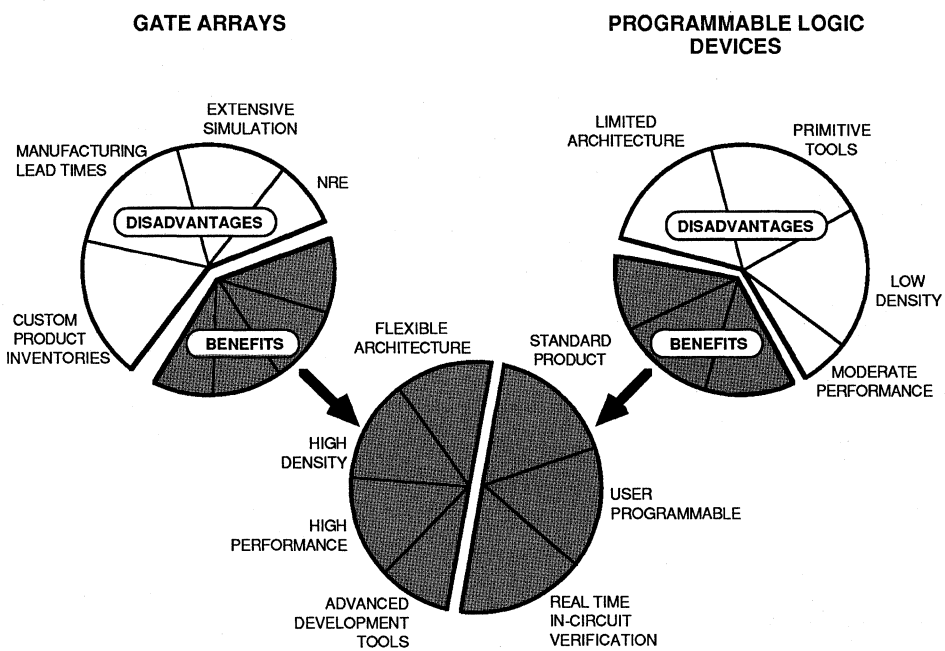
TABLE OF CONTENTS

6 Applications (Cont'd)

7 Article Reprints

8 Index

The Programmable Gate Array Company



1162 01

THE PROGRAMMABLE GATE ARRAY (LOGIC CELL™ ARRAY)

XILINX

Programmable Gate Arrays

About the Company...

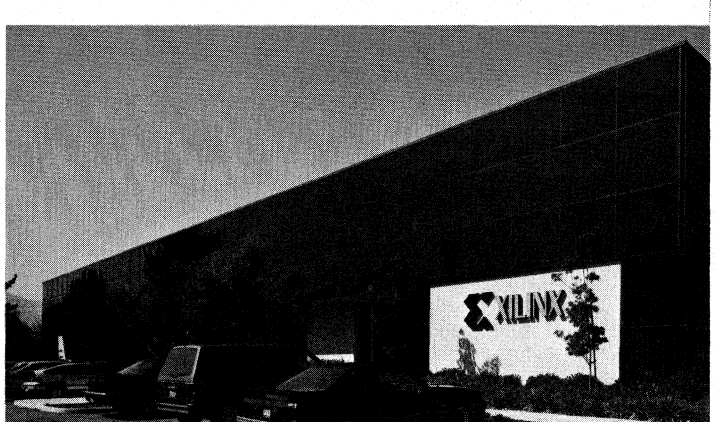
Xilinx was founded in 1984 based on the revolutionary idea, to combine the high logic density and versatility of gate arrays with the time-to-market advantages and off-the-shelf availability of user programmable standard parts. In 1985, Xilinx produced the world's first field-programmable gate array (FPGA). The company holds patents on FPGA architecture and technology, and today is the largest supplier of devices in this IC category, predicted to be the fastest growing segment of the semiconductor industry in the nineties. To date, the company has sold over 8500 development systems and five million FPGAs to more than 3500 system manufacturers worldwide.

Xilinx has maintained market leadership with a succession of new products that have increased FPGA density sevenfold, improved FPGA speed fivefold, and reduced FPGA cost by a factor of four—all in less than three years.

Competitive pressures have forced manufacturers of electronic systems to bring increasingly complex products to market rapidly. Requirements for improved functionality, performance, reliability and lower cost are often addressed through the integration of ever larger numbers of transistors onto a single IC. In systems such as computers, telecommunications systems, medical diagnostic equipment and control systems, integration results in faster speed, smaller size, lower power consumption and lower costs. However, the length of time required to develop these more sophisticated systems is often incompatible with the stringent time-to-market requirements. With Xilinx FPGAs, design engineers can bring new products to market quickly without sacrificing the benefits of integration. Many systems can be manufactured with only three types of standard high-volume components—microprocessors, memories and FPGAs.

Xilinx strategy is to focus its resources on creating new ICs and development system software, on market development and creation of a diverse customer base across a broad range of geographic and market-application segments. The company avoids the large capital commitment and overhead burden associated with owning a wafer-fabrication facility by establishing a manufacturing alliance with Seiko Epson who has manufactured all of the company's FPGA production wafers to date. In 1989, Xilinx entered into an agreement with AT&T to provide additional production capacity. Each of these manufacturers uses the same proven CMOS processing used to manufacture high-speed static RAMs (SRAMs). Using a standard process is cost-effective and produces FPGAs with established reliability, and provides for early access to advances in CMOS technology.

The company markets its products in North America through a network of five direct-sales offices, 65 manufacturers' rep locations and six distributors. Outside North America, the company sells its products through direct-sales offices in England, Germany and Japan and through manufacturers' reps and distributors in 45 offices in 20 countries.



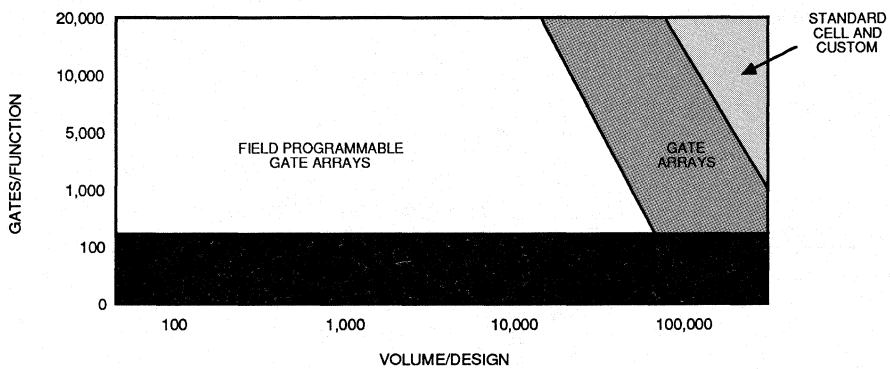
Introduction to Programmable Gate Arrays

Steady advances in the level of intergration in electronic circuits have improved many equipment features, reducing costs, power consumption, and system size, while increasing performance and reliability. Increasing levels of integration are most evident in microprocessor and memory ICs. With each process generation, the technology gap between these VLSI circuits and other standard logic ICs has widened. To achieve comparable densities for their proprietary logic functions, designers of digital equipment have been forced to consider factory-programmed custom and semicustom Application Specific Integrated Circuits (ASICs).

Field Programmable Gate Arrays (FPGAs), are high-density ASICs that can be configured by the user. They combine the logic integration benefits of custom VLSI with the design, production, and time-to-market advantages of standard products. Designers define the logic functions of the circuit and revise these functions as necessary. Thus FPGAs can be designed and verified in a few days, as opposed to several weeks for custom gate arrays; FPGA design changes can require as little as a few hours, compared to several weeks for a custom array. This results in significant cost savings by reducing the risks of design changes, rescheduling, and eliminating non-recurring engineering costs.

ASIC ALTERNATIVES

Application Specific ICs are the best solution for most logic functions.
The best ASIC solution depends on density requirements and production volumes.



1101 01

Field Programmable Gate Arrays

Unlike conventional gate arrays, FPGAs require no fixed costs, and no custom factory fabrication. Since each device is identical, manufacturing costs follow the same learning curve as other high-volume standard product ICs.

Programmable Logic Devices (PLDs)

PLDs are often used in place of five to ten SSI/MSI devices, and are the most efficient ASIC solution for densities up to a few hundred gates. Programmable Logic Devices (PLDs) include a number of competing alternatives, all based on variations of AND-OR plane architectures. The primary limitations of the PLD architecture are the number of flip-flops, the number of input/output signals, and the rigidity of the AND-OR plane logic and its interconnections. The use of one function often precludes the use of many other similar functions.

Standard Cell and Custom ICs

Standard cell and custom ICs require unique masks for all layers used in manufacturing. This imposes extra costs and delays for development, but results in the lowest production costs for high volume applications. Standard cell ICs offer the advantages of high level building blocks and analog functions.

Gate Arrays

Gate arrays implement user logic by interconnecting transistors or simple gates into more complex functions during the last stages of the manufacturing process. Gate arrays offer densities up to 100,000 gates or more, with utilization of 80-90% for smaller devices, and 40-60% for the largest.

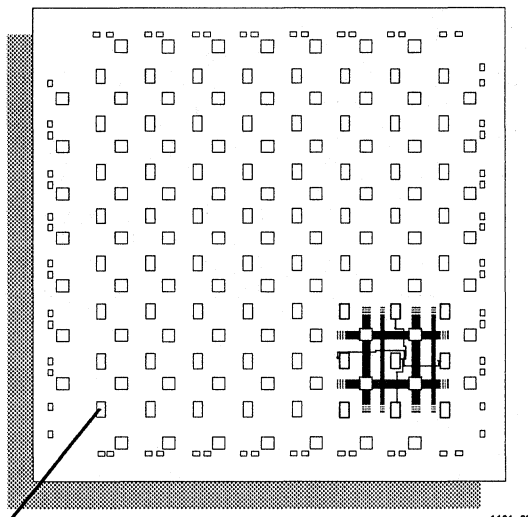
Unlike standard IC products, gate-array costs include fixed costs as well as production cost per unit. Gate arrays become cost effective when production volumes are high enough to provide a broad base to amortize fixed costs.

Programmable Gate Array Architecture

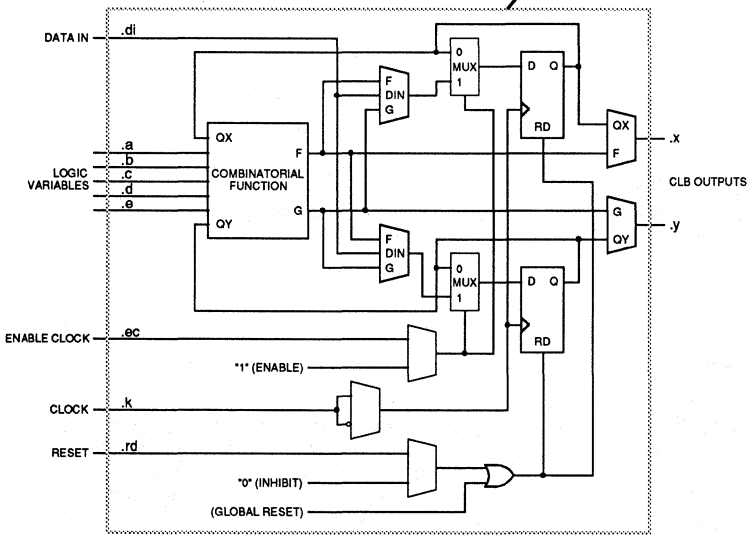
Xilinx's proprietary Logic Cell™ Array (LCA™) architecture is similar to that of other gate arrays, with an interior matrix of logic blocks and a surrounding ring of I/O interface blocks. Interconnect resources occupy the channels between the rows and columns of logic blocks, and between the logic blocks and the I/O blocks.

Like a microprocessor, the LCA device is a program-driven logic device. The functions of the LCA configurable logic blocks and I/O blocks, and their interconnection, are controlled by a configuration program stored in an on-chip memory. The configuration program is loaded automatically from an external memory on power-up or on command, or is programmed by a microprocessor as a part of system initialization.

LCA performance is determined by the logic speed, storage elements, and programmable interconnect. It is specified by the maximum toggle rate for a logic-block storage element configured as a toggle flip-flop. For typical applications, system clock rates are one-third to one-half the maximum flip-flop toggle rate.



1101 02



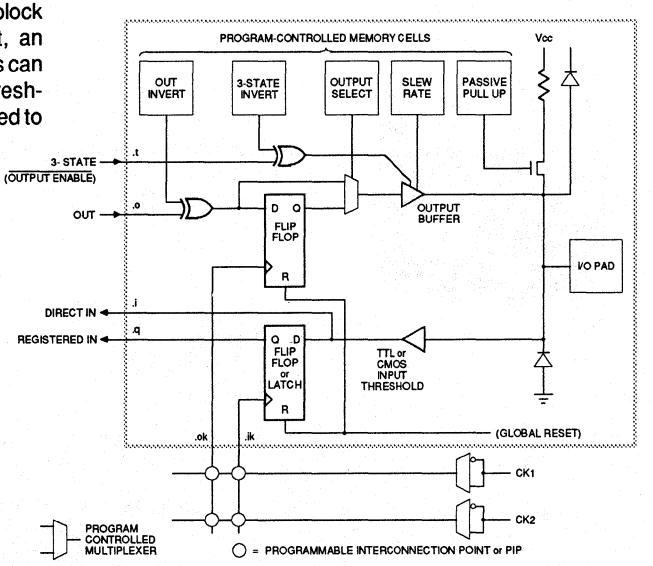
1101 03

Configurable Logic Block

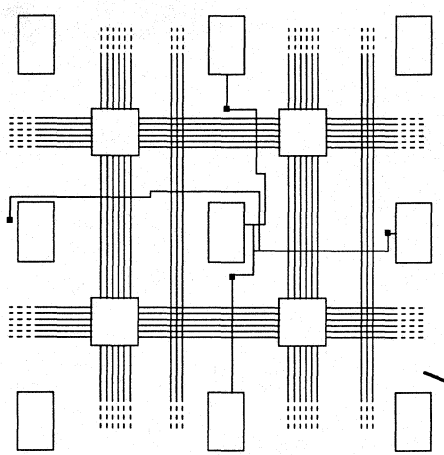
The core of the LCA device is a matrix of identical Configurable Logic Blocks (CLBs). Each CLB contains programmable combinatorial logic and storage registers. The combinatorial logic section of the block is capable of implementing any Boolean function of its input variables. The registers can be loaded from the combinatorial logic or directly from a CLB input. The register outputs can drive the combinatorial logic directly via an internal feedback path.

Input/Output Block

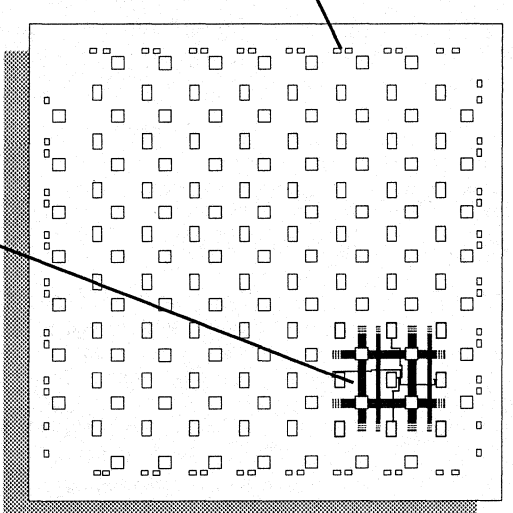
The periphery of the LCA device is made up of user programmable Input/Output Blocks (IOBs). Each block can be programmed independently to be an input, an output with 3-state control or a bidirectional pin. Inputs can be programmed to recognize either TTL or CMOS thresholds. Each IOB also includes flip-flops that can be used to buffer inputs and outputs.



1105 01A



1101 05



1101 02

Interconnect

The flexibility of the LCA device is due to the programmable resources that control the interconnection of any two points on the chip. Like other gate arrays, the LCA interconnection resources include a two-layer metal network of lines that run horizontally and vertically in the rows and columns between the CLBs. Programmable switches connect the inputs and outputs of IOBs and CLBs to nearby metal lines. Crosspoint switches and interchanges at the intersections of rows and columns can switch signals from one path to another. Long lines run the entire length or breadth of the chip, bypassing interchanges to provide distribution of critical signals with minimum delay or skew.

XC2000

Programmable Logic Cell Array Family

The XC2000 series of LCA devices was introduced in 1985. Price reductions since that time have reflected steadily increasing production volumes. The family includes two compatible arrays: the XC2064 with 1200 gates, and the XC2018 with 1800 gates.

Features

- Fully user-programmable:
  - I/O Functions
  - Logic and storage functions
  - Interconnections
- Three performance options: 50-, 70- and 100-MHz toggle rates
- Three package types: Dual in-line package
  Plastic leaded chip carrier
  Pin grid array
- TTL or CMOS input thresholds

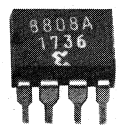
THE XC2000 Family Members

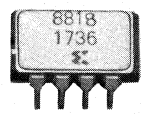
	XC2064	XC2018
Equivalent Gates	1200	1800
Configurable Logic Blocks	64	100
Combinatorial Logic Functions	128	200
Latches and Flip-Flops	122	174
Input/Outputs	58	74

XC1736A and XC1765

CMOS Serial Configuration PROM

The Serial Configuration PROMs are companion devices that provide permanent storage of LCA configuration programs. They can be used whenever a dedicated device is preferable to sharing of a larger EPROM, or to loading from a microprocessor.





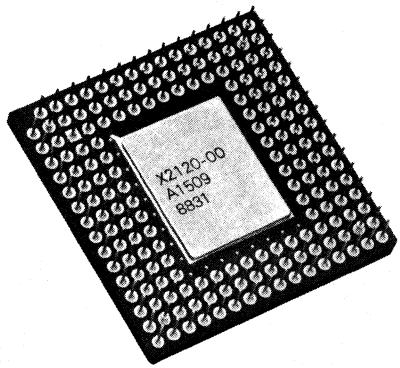
Features

- One-Time Programmable (OTP) 36,288- or 65,536-bit serial memory designed to store configuration programs for FPGAs
- Simple interface to a Logic Cell Array requires only two I/O pins
- Daisy-chain support for multiple devices
- Cascadable for large arrays or many LCA devices
- Storage of multiple configurations for a single LCA device
- Low-power CMOS EPROM process
- Space-efficient, low-cost 8-pin DIPs

XC3000

Logic Cell Array Family

The XC3000 series is a second generation family of CMOS Logic Cell Arrays that includes five compatible members with logic densities from 2000 to 9000 gates.



Features

- Fully user-programmable:
  - I/O Functions
  - Logic and storage functions
  - Interconnections
- Five member product family
  - 2000–9000 gates
  - Compatibility for ease of design migration
- Four performance options:
  - 50-, 70-, 100- and 125-MHz toggle rates
- Second generation architecture
  - 5-input logic functions
  - 2 flip-flops per CLB/IOB
  - Enhanced routing resources
  - 3-state drivers for wide ANDs
- Programmable voltage slew rates on outputs
- Three package types:
  - Plastic leaded chip carrier
  - Pin grid array
  - Quad flat package

The XC3000 Family Members

	XC3020	XC3030	XC3042	XC3064	XC3090
Equivalent Gates	2000	3000	4200	6400	9000
Configurable Logic Blocks	64	100	144	224	320
Combinatorial Logic Functions	128	200	288	448	640
Latches and Flip-Flops	256	360	480	688	928
Input/Outputs	64	80	96	120	144

XC4000

Logic Cell Array Family

The XC4000 series, the third-generation family of CMOS LCA devices, combines architectural versatility, on-chip RAM, increased speed and gate complexity with abundant routing resources and new, sophisticated software, to achieve fully automated implementation of complex, high-performance designs. It is the first FPGA family to break the 20,000-gate barrier; the first member of the XC4000 family will be sampled in late 1990.

Features

- Third generation user-programmable gate array
  - Abundant Flip-Flops
  - Flexible function generators
  - On-chip fast RAM
  - Dedicated high-speed Carry Propagation circuit
  - Fast, wide decoders
  - Unlimited number of logic levels
  - Hierarchy of interconnect lines
  - Internal 3-state bus capability
- Flexible array architecture
  - Programmable I/O blocks
  - Programmable logic blocks
  - Programmable interconnects
  - Programmable wide decoders
- Sub-micron CMOS process
  - High speed (toggle/shift rate >100 MHz, counters >50 MHz)
  - Low power consumption
- Systems-oriented features
  - Slew-rate limited outputs
  - Programmable input pull-up or pull-down resistors
- Configured by loading binary file
  - Unlimited reprogrammability
  - Six programming modes
- Development system runs on '386-based PC and on many popular workstations
  - Fully automatic placement and routing plus optional interactive enhancements

XC4000 Family Members

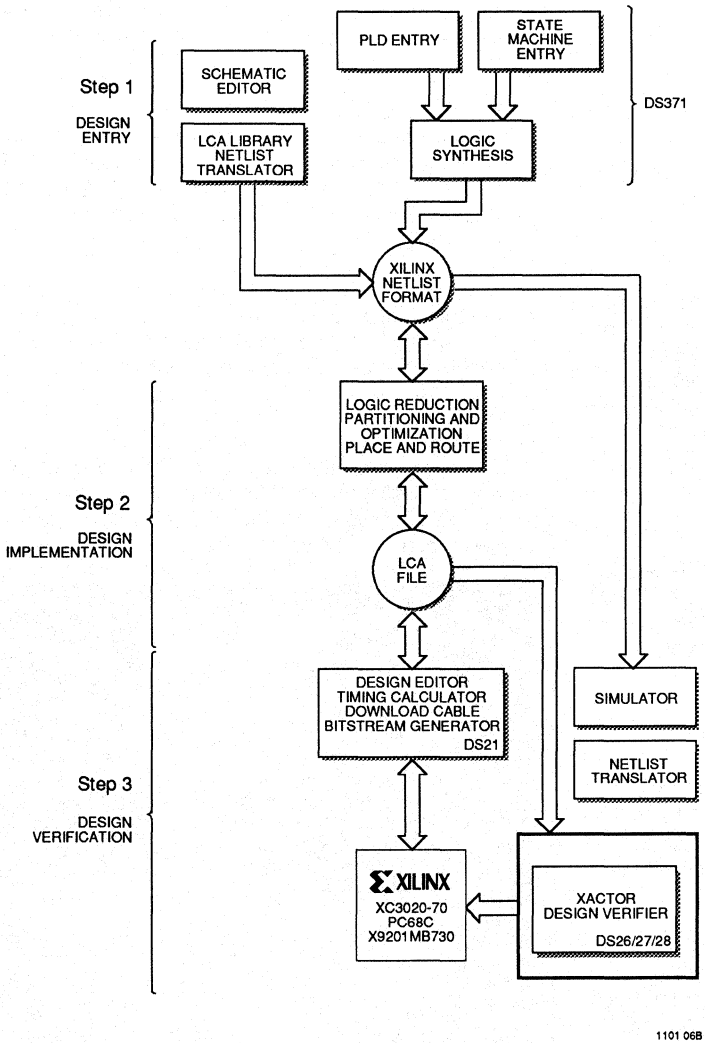
	XC4002	4003	4004	4005	4006	4008	4010	4013	4016	4020
Appr. Gate Count	2,000	3,000	4,000	5,000	6,000	8,000	10,000	13,000	16,000	20,000
CLB Matrix	8 x 8	10 x 10	12 x 12	14 x 14	16 x 16	18 x 18	20 x 20	24 x 24	26 x 26	30 x 30
Configurable Logic Blocks	64	100	144	196	256	324	400	576	676	900
Max RAM Bits	2,048	3,200	4,608	6,272	8,192	10,368	12,800	18,432	21,632	28,800
Input/Outpts	64	80	96	112	128	144	160	192	208	240

The XC4000 family of Logic Cell Arrays is not covered in this Data Book.

Ask for the separate XC4000 Product Description.

Development Systems

Designing with Xilinx FPGAs is similar to designing with other gate arrays. Designers can use familiar CAE tools for design entry and simulation. The open Xilinx development system includes a standard netlist format, the Xilinx Netlist File (XNF), that provides a bridge between schematic editors or simulators, and the XACT software for design implementation and real time design verification. The Xilinx software is supported on the PC/AT and compatibles as well as on popular engineering workstations.



Design Entry Software

consists of libraries and netlist interfaces for standard CAE software such as FutureNet, Schema, OrCAD, VIEWlogic, Mentor, Valid, CASE, and PALASM. Programmable gate array libraries permit design entry with standard TTL functions, with Boolean equations, and with user-defined macros.

Simulation Software

includes models and netlist interfaces to standard simulator software, such as SILOS and CADAT, that is used for logic and timing simulations.

Design Implementation Software

is used to convert schematic netlists and Boolean equations into efficient designs for programmable gate arrays. The software includes programs that perform partitioning, optimization, placement and routing, and interactive design editing.

In-circuit Design Verification Tools

permit real-time verification and debugging of a programmable gate array design as soon as it is placed and routed. Designers benefit from faster and more comprehensive design verification, and from reduced requirements to generate simulation vectors to exercise a design.

Technical Support

SOFTWARE UPDATES

Xilinx is continuing to improve the XACT development system software, and new versions are released two or three times per year. Updates are provided free of charge during the first year after purchase, provided the user returns the registration card. After the first year, users are encouraged to purchase a Software Maintenance Agreement to continue to receive software updates.

TRAINING COURSES

To get up-to-speed quickly, new Xilinx users are invited to attend comprehensive training classes. These classes are taught by factory experts and include the latest software and hardware advances.

XILINX USER GROUPS

Xilinx users are invited to attend training and information exchange sessions that are held two-to-three times per year in various locations worldwide. These User Group meetings are intended for experienced users of Xilinx Programmable Gate Arrays, and they emphasize the efficient use of the XACT development system.

FIELD APPLICATIONS ENGINEERS

Xilinx provides local technical support to customers through a network of Field Applications Engineers (FAEs). For the name and phone number of the nearest FAE, customers may call one of the Xilinx sales offices listed in the back of this book.

APPLICATIONS HOT LINE

Xilinx maintains an applications hot line to provide technical support to LCA users. This service is available from 7:30 am to 6:00 pm Pacific Time. Call (408) 879-5199 or (800) 255-7778 and ask for Applications Engineering.

BULLETIN BOARD

To provide customers with up-to-date information and an immediate response to questions, Xilinx provides 24-hour access to an electronic bulletin board. The Xilinx Technical bulletin board provides the following services to all registered XACT development-system customers.

- Read files from the bulletin board
- Check current software version numbers
- Download files
- Upload files
- Leave messages for other bulletin-board users.

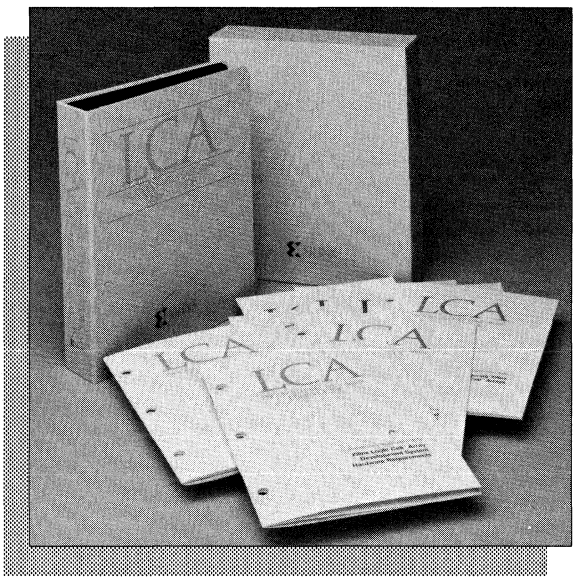
TECHNICAL LITERATURE

In addition to this databook, technical literature for the Xilinx programmable gate array includes four volumes that are delivered with every XACT development system.

- User's Guide
  The User's Guide is a collection of "how to" applications notes on such subjects as getting started with an LCA design, Boolean equation design entry, use of the simulator, placement and routing optimization, and LCA configuration.

- Reference Manuals (2 vols)
  The XACT Reference Manuals include a detailed description of each Xilinx software program.

- Macro library
  The Xilinx development system includes over 100 macros, including counters, registers, and multiplexers. The macro library manual includes schematics and documentation for each macro.



A Cost of Ownership Comparison

CONTENTS

EXECUTIVE SUMMARY

Introduction

Custom or mask-programmed gate arrays have many hidden costs beyond the obvious unit cost and NRE (non-recurring engineering) charges. Most of these additional costs are due to the fact that a gate array is a custom integrated circuit, one manufactured exclusively for a particular customer. Compared to a standard product, there are many hidden expenses, both during the design phase and after purchase, beyond the direct device cost.

Field-programmable gate arrays (FPGAs), on the other hand, are high-volume standard products—manufactured and fully tested devices that are used by all customers. There is no customization of the silicon.

Methodology

This analysis compares the total costs of custom gate arrays with those of field-programmable gate arrays. It looks at the various categories of costs, both fixed and variable, for devices from 2000 to 9000 gates, 80% of the gate-array market according to most studies.

Field Programmable Gate Arrays (FPGAs)	Gate Arrays
Standard product	Custom product
Off-the-Shelf delivery	Months to manufacture
Fast time to market	Manufacturing delays
Programmed by the user	Programmed in the factory
No NRE	NRE Costs
No inventory risk	Design specific
Fully factory tested	User develops test
Simulation useful	Simulation critical
In-circuit design verification	Not possible
Design changes anytime	NRE charge repeated
Second source exists	Additional cost and time

Because the gate array has fixed or up-front development costs (NRE, extra simulation time, generating test vectors, etc.) that the FPGA does not, its total cost of ownership is higher until a sufficient quantity is purchased. This analysis allows the user to calculate total cost of ownership at different quantities and derive break-even quantities — the volume below which it is more cost effective to use the FPGA (Break-even Analysis section). The overall objective is to determine the production volumes at which each product is most cost effective.

Conclusion

The choice between FPGA- and mask-programmed gate arrays must take into account more than the NRE and cents/gate unit cost. The use of a custom product entails many other costs and risks. Because of these fixed costs, it is less expensive at lower volumes to use a standard product: an FPGA. Since many of the hidden costs of using a custom gate array do not accrue to any one department, only the project manager can recognize the total cost.

Similar considerations have led to the widespread acceptance of EPROM memories as compared to ROMs, despite a higher EPROM cost per unit. The same factors can be applied in the choice of a gate array.

Figure 1 shows a representative break-even graph for a 2000-gate device using 1990 data. The vertical axis shows the total project cost—fixed costs plus unit costs multiplied by the number of units. At lower volumes, the custom gate array is more expensive because of fixed costs that are incurred even if no units are purchased. The FPGA project cost starts at zero, but rises faster because of a higher cost per-unit. In this case the break-even volume is between 10k and 20k units. The various components of this analysis are discussed in the following sections. Also, guidelines are given to help the user make a simple calculation for a specific solution.

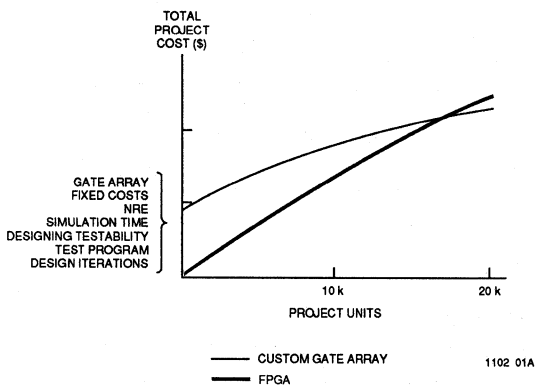


Figure 1. Typical Break-even Analysis 2000 Gates—1990

Several significant factors are omitted from Figure 1. First, the additional fixed costs (NRE, simulation) of bringing on a custom-gate-array second source are not included. Second, and much more important, the cost of the longer time to market when designing with the mask gate array is not included. This factor is reviewed in the Time to Market section. Both of these factors would raise the custom gate array curve and increase the break-even quantity. In other words, the FPGA would be more cost effective at an even higher production volume.

ROM VS EPROM ANALOGY

There is a relevant historical precedent for the use of a flexible standard product instead of a custom product with a lower direct cost per unit. While EPROMs have a cost per bit that is two to three times that of ROMs, they have consistently captured almost half the programmable memory market, measured in bits shipped. See Figure 2. Many of the reasons for the use of EPROMs are the same as those for the use of programmable gate arrays: faster time to market, lower inventory risks, easy design changes, faster delivery, and second sources. The higher price per bit is offset by the elimination of inventory and production risks.

Gate arrays have even more disadvantages versus programmable gate arrays than do ROMs versus EPROMs. The upfront design time, risk, and expense of ROMs is minimal, while that of gate arrays is substantial. ROM test tape generation is automatic, while that for gate arrays requires extensive engineering effort. Therefore, FPGAs may be even more widely used versus gate arrays than are EPROMs versus ROMs.

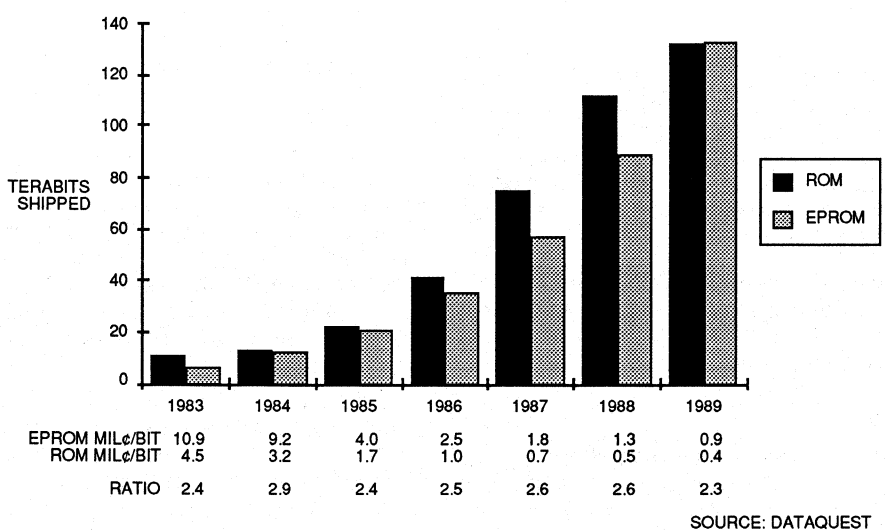


Figure 2. ROM/EPROM Analogy

WHO RECOGNIZES THE COSTS?

Many of the elements of the total cost of ownership for a gate array do not accrue to a single department, and often are not fully recognized. For example, the additional engineering time needed to design for testability may not be seen by purchasing. The inventory costs of a custom product may not be recognized by the design department. However, these are real costs, and they influence the profitability of the product and company. The person making the choice between custom gate arrays and FPGAs should consider the total costs of ownership for each alternative.

TOTAL COST = FIXED COST + (VARIABLE COST)(UNITS)

The total costs of using a product can be separated into two components. The first is the fixed costs: up-front development costs that are independent of volume. Some examples of these for gate arrays are the masking charge, simulation charge, and test program development. Due to amortization of these costs, the user's cost per unit can be very high until a sufficient volume of units is purchased. The second component of total cost is the variable cost, the incremental cost per unit. Besides the obvious unit cost, another element of variable cost is inventory cost.

This analysis will examine costs by these two categories. Fixed costs are summarized first, then variable costs. They are added to produce total cost.

FIXED DEVELOPMENT COSTS

Simulation

With a custom product, it is critical that the device work the first time. Otherwise, the user must pay to have the device prototyped a second time and will incur the manufacturing delay a second time. Custom gate arrays do not support a conventional, iterative, modular design process—the design is all-or-nothing. Simulation is a useful tool with FPGAs, but it is a critical one with gate arrays, and the designer can expect to spend more time simulating a custom gate array design. The programmable gate array designer can count on in-circuit verification and on-line changes if necessary.

Gate array simulation cost includes both computer time charges and the time of the engineer doing the simulation. While the gate-array vendor may or may not charge explicitly for computer time, an estimate would be $2,500 and 2.5 man weeks of simulation effort for a 2000-gate array, and $5,000 and seven man weeks for a 9000-gate array. This compares to 0.5 and 1 week for the FPGA, with no simulation charge.

Typically one fully burdened man week, including computer support, costs about $2000.

	2000 Gates	9000 Gates
Gate Array		
Simulation Charge	$2.5 k	$5 k
Man Weeks	2.5 MW	7 MW
FPGA		
Simulation Charge	None	None
Man Weeks	0.5 MW	1 MW

Time to Design for Testability

One key to getting a successful gate array the first time is to focus on testing issues. The user must guarantee that the device can be fully tested in a reasonable amount of time. Since the gate array vendor's only guarantee is that the device will pass the test program, the user must be certain that if the IC meets the user-generated test specifications, it will work in the circuit.

Spending extra time in the design phase provides insurance that the device can be tested. A Dataquest ASIC Market Report observes that "an engineer can sit down at a $20,000 CAE/CAD station and design a $1,000,000 test problem." Designing in testability may also be the only way to provide for testing of complex sequential circuitry, or elements like long counters. Therefore the gate array designer must spend additional time in the design phase. An estimate is one additional week for a 2000-gate array, and two additional weeks for a 9000-gate array.

The FPGA is a standard product with no incremental test costs. It is fully tested by Xilinx before shipment. No application-specific testing is needed.

Gate Array Incremental Cost

2000 Gates	9000 Gates
1 Man Week	2 Man Weeks

NRE Charges

NRE (Non-Recurring Engineering) charges cover the on-line vendor interface, design verification, mask charges, prototype samples and a nominal simulation (pre- and post-layout) time. The charges may vary with estimated production volumes. At volumes below 50,000 units, $10,000 to $20,000 is a competitive quote for lower density gate arrays. At the 9000-gate level, NRE charges may be in excess of $30,000.

There are no NRE charges for programmable gate arrays. The entire design process is done by the customer. FPGA software tools run on common workstations and personal computers, and are much less expensive than comparable tools for custom gate arrays.

Typical Gate Array NRE for 10,000 to 50,000 units

2000 Gates	9000 Gates
$10 k–$20 k	$20 k–$40 k

Design Iterations

The phrase "We need to add this feature" is all too common to the designer of electronic equipment. Designers often find themselves faced with the need to modify a design during prototyping or initial customer evaluation. Changes may be required to add features or reduce costs. As systems become more complex, "bugs" can be more prevalent.

Design iterations are almost never due to the failure of the gate-array vendor. Rather, there are risks associated with the choice of an inflexible technology in a very dynamic industry.

Industry data suggest that about half of all gate-array designs are modified before they are released to production. When a modification is required, NRE costs are incurred for the second pass. Since resimulation is likely to involve less effort than the initial simulation, 25% (50% probability times one half the effort) of the simulation cost is added.

Gate Array Incremental Cost

50% Probability	(original NRE time and cost + one half of original simulation time and cost)

Test Program Development

As noted in the Time to Design for Testability section, testability is critical to production success for gate arrays. Gate-array vendors rarely make production errors, but faulty devices may not be detected because the test vectors are not comprehensive.

The estimate for test-vector development is two weeks for a 2000-gate array, and four weeks for a 9000-gate array. Since the FPGA is a standard product, it is fully tested at the factory. No application-specific testing is needed.

A risk that the program manager should consider involves the level of experience or knowledge that the design team has with test development. If the first-pass design is unsuccessful, how much time and effort will be required to debug the problem? Both additional cost and time to market are at risk.

Gate Array Incremental Cost

2000 Gates	9000 Gates
2 Man Weeks	4 Man Weeks

Second Source

If a second source is required, the gate-array designer must identify a compatible vendor and resubmit the design. This involves another NRE charge and time for translating logic and resimulation. The model used here is the NRE charge plus one half the simulation cost.

Field-programmable gate arrays are standard products that already have a second source.

Gate Array Incremental Cost

	2000 Gates	9000 Gates
NRE	$10 k–$20 k	$20 k–$40 k
Simulation		
Charge	$1.25 k	$2.5 k
Man Weeks	1 MW	3 MW

Summary of Gate Array Fixed DevelopmentCosts

The summary in Table 1 shows typical fixed costs for both a 2000-gate and a 9000-gate array. Since assumptions may vary, a blank column is provided as a worksheet.

	Typical 2000 Gates	Typical 9000 Gates	Customer Application
1. Simulation			
NRE	$2,500	$5,000	_____
Man Weeks	2 MW	7 MW	_____
2. Design for Testability	1 MW	2 MW	_____
3. NRE Charges	$10 k–$20 k	$20 k–$40 k	_____
4. Design Iterations @ 50% probability			
NRE	$8,125	$16,250	_____
Man Weeks	0.5 MW	1.5 MW	_____
5. Test Program Development	2 MW	4 MW	_____
6. Second Source (NRE + 50% SIM)			
NRE	$16,250	$32,500	_____
Man Weeks	1 MW	3 MW	_____
Total Without Second Source			
NRE	$25,625	$51,250	_____
Man Weeks	5.5 MW	14.5 MW	_____
Total With Second Source			_____
NRE	$41,875	$83,750	_____
Man Weeks	6.5 MW	17.5 MW	_____
			_____
Total Fixed Costs @ $2 k/MW			@ $__/MW
Without Second Source	$36,265	$80,250	_____
With Second Source	$54,875	$118,750	_____

Table 1. Typical Fixed Costs

VARIABLE COSTS

Production Unit Cost (Cents/Gate)

Gate-array prices are often quoted in terms of cents per gate. For 1.2 micron, 2000-gate arrays, at the volumes considered in this analysis (10,000 to 30,000 units), a figure of 0.15 – 0.20 cents/gate (without package) is typical. At similar volumes, the cost per gate (without package) for an FPGA is two to three times the cost of a custom gate array. For reasons explained below, this gap is expected to narrow over the next few years. All of the cents/gate numbers are for die only. Since CMOS gate arrays and FPGAs use the same packages, the package adders are equivalent.

An important consideration in calculating the total cost of ownership is the year during which most of the production volume will be purchased. Since FPGAs are newer products, their cost is declining at a steeper rate than gate arrays. They are in the introduction phase of the life cycle, while gate arrays are in a more mature phase of the cycle. Price comparisions should be based on projections over the production life of the product.

A standard product has more silicon content and less factory overhead than a custom product. Since all customers buy the same product, there is more of the semiconductor learning curve with cumulative volume. Given the profitability levels of array manufacturers, gate array prices may decline only slightly over time and could even rise.

1991 FPGA Unit Costs— Without Package

	2000 Gates	4000 Gates	9000 Gates
	20 k Qty	10 k Qty	10 k Qty
Programmable (Cents/Gate)	0.30–0.40	0.40–0.50	0.50–0.60

Process Technology

There are also technology reasons for the steeper decline in FPGA cost. Figure 3 shows that the processes used for logic ICs, including gate arrays, typically lag behind those used for memory ICs. Since the FPGA is a standard IC built on a memory process, it can take advantage of each new process to shrink the die and reduce costs.

With a conventional gate array, the process that is available at the time of design is usually used throughout the production lifetime of the product. Except for very high-volume applications, few gate arrays are retooled to take advantage of process advances. The time from design start to end of production lifetime is usually several years. Over this period, the FPGA will move to successively more advanced processes, resulting in steadily decreasing costs. By the end of the production lifetime, the FPGA will be several processes ahead and the cost difference will be reduced significantly.

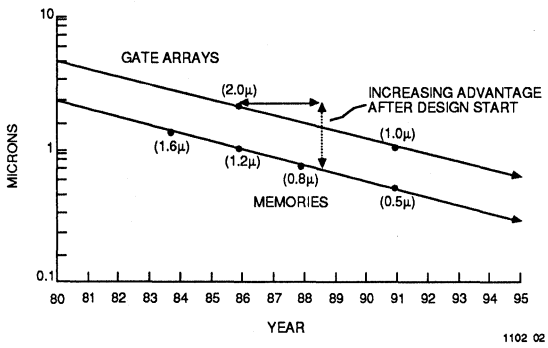


Figure 3. Process Evolution

Pad-Limited Die Sizes

As gate arrays and FPGAs grow in I/O pin count, a phenomenon known as "pad-limiting" is more likely to occur. The spacing between I/O pads is determined by mechanical limitations of the equipment used for lead bonding. In I/O-intensive applications the number of pads around the outer edge of the die determines the die size, instead of the number of gates. See Figure 4. In I/O-intensive applications, a "cost per I/O" may be a more useful measure than "cost per gate."

For a given I/O count, in the pad-limited case the FPGA and the gate array would be the same die size. As a result, the higher volume, standard product, FPGA could actually be less expensive on a per-unit basis than the custom-product gate array. There would be no break-even quantity — the FPGA would have a lower cost of ownership at all volumes.

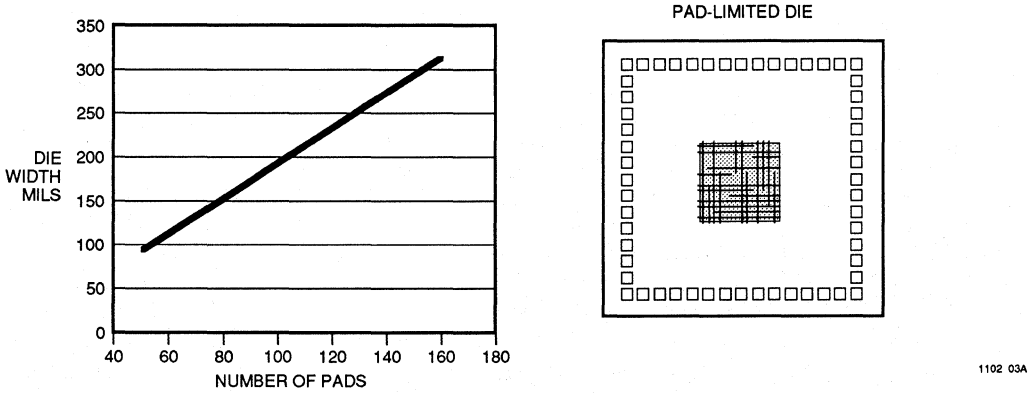


Figure 4. Minimum Die Size vs I/O

Effect of Die Cost on Total Cost

Figure 5 illustrates a third point about the capability of FPGAs to narrow the cost difference with custom gate arrays. The chart shows the contribution to total device cost of wafer, die, assembly and test. Wafer cost represents about 20% to 40% of the total device cost, and die cost about 30% to 50%. A 50% difference in die cost – between a gate array and a FPGA – shown in the chart translates to only a 20% difference (80 vs 100) in total cost by the time the device has been tested. This comparison is based on production of the FPGA in a more advanced process than the custom gate array, as discussed in the Unit Cost (Cents/Gate) section.

Inventory Reserves

Inventories include extra devices ordered and stocked to cover contingencies. For a custom product, this is the only way parts can be delivered in less than the normal production time (2–4 months). Contingencies are often thought of in terms of negative events like a defective lot or manufacturing shortfalls.

However, contingencies also include positive events like stocking for large, upside orders or where demand is difficult to estimate. This can be especially true during a product's introduction, when design changes and demand spikes occur simultaneously.

With a custom product it is also necessary to build inventory as the product nears the end of its life cycle. Demand is low and difficult to forecast, and it may not be possible to reorder a small quantity. Spares and replacements must be stocked. A JIT(just in time) inventory system is less practical.

Since minimum manufacturing quantities for semiconductors are determined by wafer lots, a custom product will have excess WIP (work in process) or finished goods inventory if the desired order quantity is less than the minimum economical wafer-lot quantity. Inventory is created and costs are incurred. Moreover, there is the problem of inventory ownership if the parts are never ordered by the customer.

Although the safety stock reserve is a function of the cost of the product itself, a figure of 10% is reasonable for gate arrays that have unit costs under $25.00. In comparison, since changes to FPGAs can be made in software in minutes, and since only one part type is widely stocked, the comparative safety stock reserve is 0%.

Gate Array Incremental Inventory Cost
10% Additional Unit Cost

YIELD TO PRODUCTION

Due to rapidly changing markets, many designs never go into production. Sometimes a company will develop competing projects, with only one moving to production. Many times the market will change, or competition will emerge, and projects will be cancelled or redirected. Of course each design team expects that its project will succeed, but in the aggregate this is not true. If a company chooses gate arrays as the primary logic technology, and starts many designs, this factor will occur.

According to Dataquest ASIC and Standard Logic Semiconductor Volume 1,only 50% of gate-array designs go into production. Therefore, the true cost of the gate array should recognize additional costs for simulation, designing for testability, and NRE. For 2000 gates, using the numbers in the Summary of Fixed Development Costs section, this would mean an additional ($2,500 + 3 MW + $15,000). For 9000 gates the number is ($5,000 + 8 MW + $30,000).

Gate Array Incremental Cost
Simulation Cost + Time to Design for Testability + NRE Cost

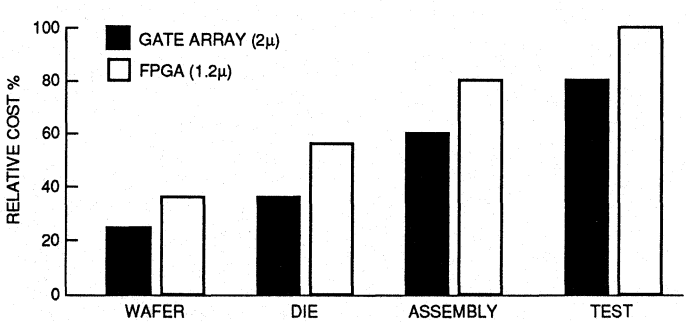


Figure 5. Relative Manufacturing Cost by Stage of Completion

COST OF OWNERSHIP ANALYSIS

While gate arrays have a lower unit cost, they have incremental fixed costs that must be incurred before the first unit is received. Example costs are shown in Table 1 (Summary of Fixed Development Costs). Therefore, at lower unit volumes the FPGA is less expensive, until the gate array can amortize the up-front fixed costs.

Table 2 is a form that can be used for calculating the total cost of ownership at various volumes. Table 2 points to the "break-even quantity"—the quantity where the unit cost of the two devices is the same—of the next section.

Project Quantity	**1,000**	**5,000**	**10,000**	**20,000**
Gate Array—No second source				
1. Fixed costs from Table 1				
2. Unit cost				
3. Inventory reserves: (Line 2)(1.1)				
4. Total variable cost = (Line 3)(Qty)				
5. Total cost = Line 1 + Line 4				
6. Unit cost = Line 5/Qty				
Gate Array—second source				
7. Fixed costs from Table 1				
8. Second source costs				
9. Total fixed costs				
10. Total variable cost—Line 4 above				
11. Total cost = Line 9 + Line 10				
12. Unit cost = Line 11/Qty				
Field Programmable Gate Array				
13. Unit cost				

Table 2. Total Cost vs Volume Purchased

BREAK EVEN ANALYSIS

Figure 6 is a graphic representation of the break-even calculation for the case of 2000 gates, 1990 pricing, and no second source. Up to the break-even unit volume, the programmable gate array solution has a lower total project cost.

Similar graphs can be built for different assumptions by filling in Table 1. For the gate array, the break-even graph is merely line 5 or line 11 plotted versus quantity. For the FPGA it is line 13 times the quantity plotted versus the quantity.

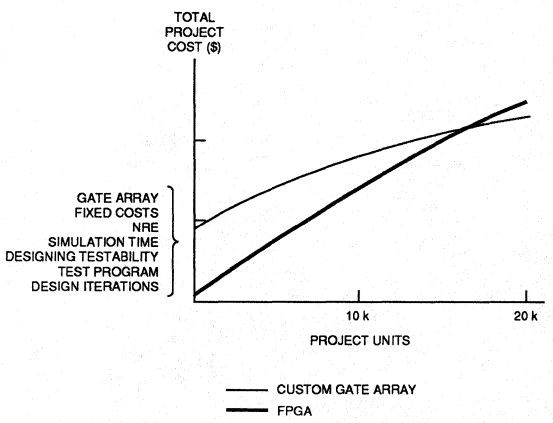


Figure 6. Typical Break-even Analysis 2000 Gates—1990

TIME TO MARKET

There are numerous examples of products that failed due to late market entry. A study by McKinsey & Co. stated that a product that is six months late to market will miss out on 1/3 of the potential profit over the product lifetime. If there is any problem in simulation, or any iteration of the gate array design, a gate array would easily add six months to a product schedule.

At the 2,000 gate level, assume the gate array is used in a $2,000 product that has 15% profit margins. For 10,000 units sold:

Lost Profit = $2,000 x 10,000 x 15% x 1/3 = $1.0 million or $100 per device

At the 9000 gate level, assume the gate array is used in a $10,000 product that has 20% profit margins. For 2000 units sold:

Lost Profit = $10,000 x 2,000 x 20% x 1/3 = $1.33 million or $667 per device

Note that these catastrophic costs are not included in any of the previous sections. They are a quantitative estimate of the risk of using a custom product.

PRODUCT LIFE CYCLES

Throughout the electronics industry, the product lifetimes are shrinking. In the personal computer industry, it is not uncommon to find product upgrades within 6 to12 months. This means that the volumes associated with any one gate- array design can be much smaller than anticipated, even if the end product still exists. It also means that it is critical to achieve a rapid design time.

Figure 6 shows that 2000-gate FPGAs are more economical at volumes up to 10,000 to 20,000 units. These volumes will represent an increasing number of products.

REFERENCES

1. *Technology Research Group Letter,* March 1986, page 7.
2. Dataquest Inc., *Gate Arrays—Product Analysis,* page 3, ASIC and Standard Logic Semiconductors, 1987.
3. Reinertsen, Donald G.,"Whodunit? The Search for the New-Product Killers," *Electronic Business,* July 1983, pages 62–66.
4. Integrated Circuit Engineering Corp., *ASIC Outlook,* 1987.


XILINX

XILINX

Product Specifications

XC3000
Logic Cell™ Array Family

Product Specification

FEATURES

- High Performance—70-, 100- and 125-MHz Toggle Rates
- Second Generation Field-Programmable Gate Array
  - I/O functions
  - Digital logic functions
  - Interconnections
- Flexible array architecture
  - Compatible arrays, 2000 to 9000 gate logic complexity
  - Extensive register and I/O capabilities
  - High fan-out signal distribution
  - Internal 3-state bus capabilities
  - TTL or CMOS input thresholds
  - On-chip oscillator amplifier
- Standard product availability
  - Low-power, CMOS, static-memory technology
  - Performance equivalent to TTL SSI/MSI
  - 100% factory pre-tested
  - Selectable configuration modes
- Complete XACT™ development system
  - Schematic Capture
  - Automatic Place/Route
  - Logic and Timing Simulation
  - Design Editor
  - Library and User Macros
  - Timing Calculator
  - XACTOR In-Circuit Verifier
  - Standard PROM File Interface

DESCRIPTION

The CMOS XC3000 Logic Cell™ Array (LCA™) family provides 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 IOBs, a core array of CLBs and resources for interconnection. The general structure of an LCA device is shown in Figure 1 on the next page. The XACT 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 LCA 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. Xilinx's companion XC1736 Serial Configuration PROM provides a very simple serial configuration program storage in a one-time programmable 8-pin DIP.

Basic Array	Logic Capacity (gates)	Config-urable Logic Blocks	Max User I/Os	No. of Pads	Program Data (bits)
XC3020	2000	64	64	74	14,779
XC3030	3000	100	80	98	22,176
XC3042	4200	144	96	118	30,784
XC3064	6400	224	120	140	46,064
XC3090	9000	320	144	166	64,160

The XC3000 Logic Cell Arrays are an enhanced family of Field Programmable Gate Arrays that provide a variety of logic capacities, package styles, temperature ranges and speed grades.

ARCHITECTURE

The perimeter of configurable I/O 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 blocks' 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 tran-

sistors. These LCA 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 LCA device at power-up and may be reloaded on command. The Logic Cell Array includes logic and control signals to implement automatic or passive configuration. Program data may be either bit serial or byte parallel. The XACT development system generates the configuration program bitstream used to configure the Logic Cell Array. The memory loading process is independent of the user logic functions.

Configuration Memory

The static memory cell used for the configuration memory in the Logic Cell Array has been designed specifically for high reliability and noise immunity. Integrity of the LCA configuration memory based on this design is assured even under adverse conditions. Compared with other programming alternatives, static memory provides the best combination of high density, high performance, high reliability and comprehensive testability. As shown in Figure 2, 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 re-written.

The memory cell outputs Q and $\overline{Q}$ use ground and Vcc 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

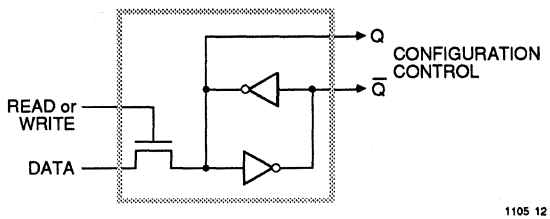


Figure 2. Static Configuration Memory Cell. It is loaded with one bit of configuration program and controls one program selection in the Logic Cell Array.

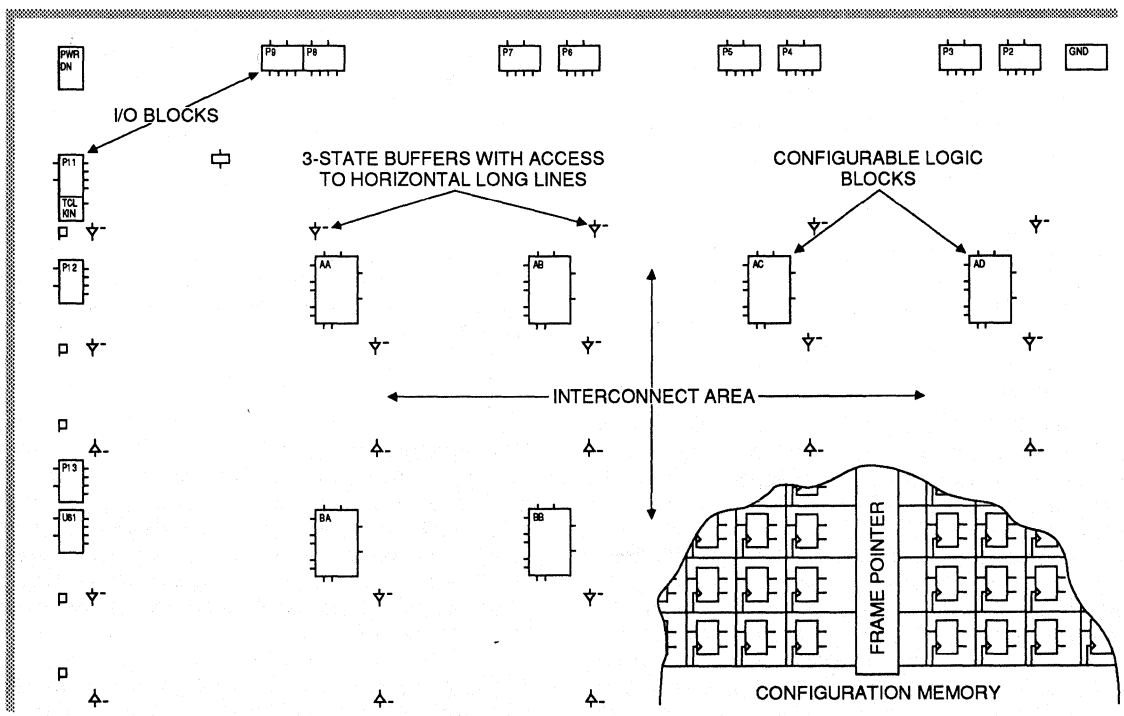


Figure 1. Logic Cell Array Structure. It consists of a perimeter of programmable I/O blocks, a core of configurable logic blocks and their interconnect resources. These are all controlled by the distributed array of configuration program memory cells.

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 XACT development system, to direct memory-cell loading. The serial-data framing and length-count preamble provide programming compatibility for mixes of various LCAs in a synchronous, serial, daisy-chain fashion.

I/O Block

Each user-configurable IOB shown in Figure 3, 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 electro-static protection, and circuits to inhibit latch-up produced by input currents.

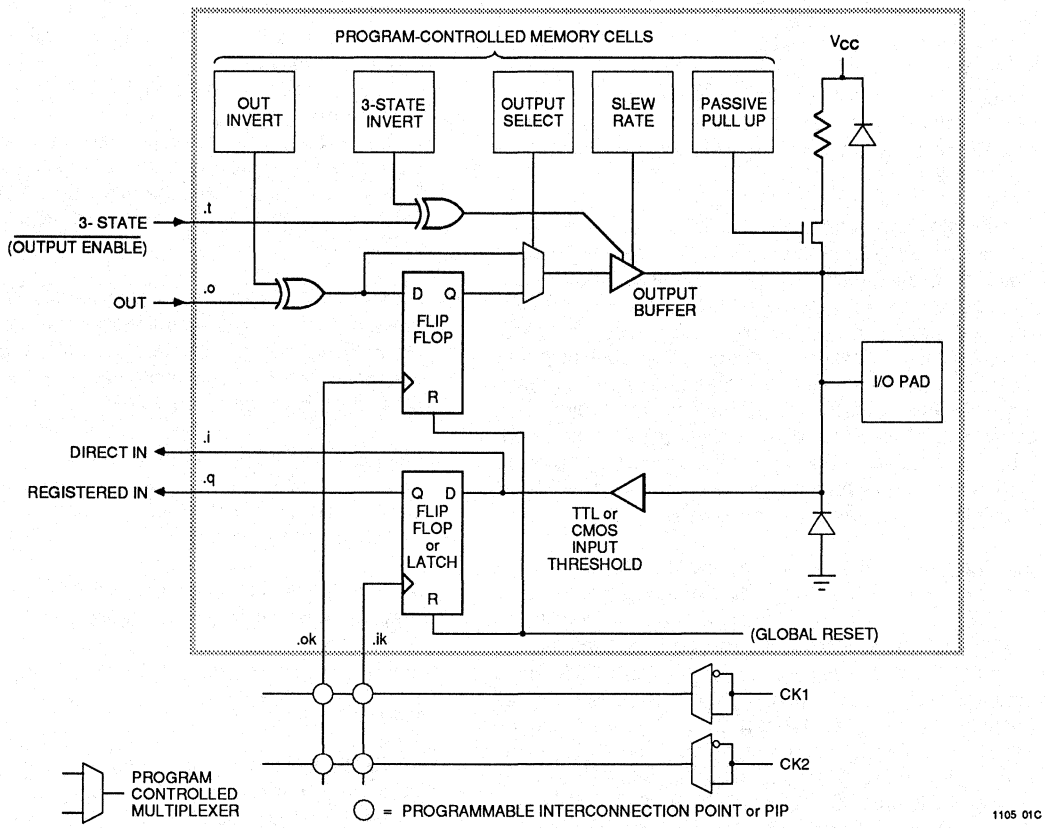


Figure 3. Input/Output Block. Each IOB includes input and output storage elements and I/O options selected by configuration memory cells. A choice of two clocks is available on each die edge. The polarity of each clock line (not each flip-flop or latch) is programmable. A clock line that triggers the 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 $\overline{\text{RESET}}$ 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 Logic Cell 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 about 3 ns. This short delay provides 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 Logic Cell 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. 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-type 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, logical signal inversion, and 3-state and slew-rate control of the output.

The program-controlled memory cells of Figure 3 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 which 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.

- A high-impedance **pull-up resistor** may be used to prevent unused inputs from floating.

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 Configurable Logic Blocks (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. The XC3020 has 64 such blocks arranged in 8 rows and 8 columns. The XACT development system is used to compile the configuration data which are to be loaded into the internal configuration memory to define the operation and interconnection of each block. User definition of configurable logic blocks 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 configurable logic block has a combinatorial logic section, two flip-flops, and an internal control section. See Figure 4. 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, data-in [*.di*]. Both flip-flops in each CLB share the asynchronous reset [*.rd*] which, when enabled and High, is dominant over clocked inputs. All flip-flops are

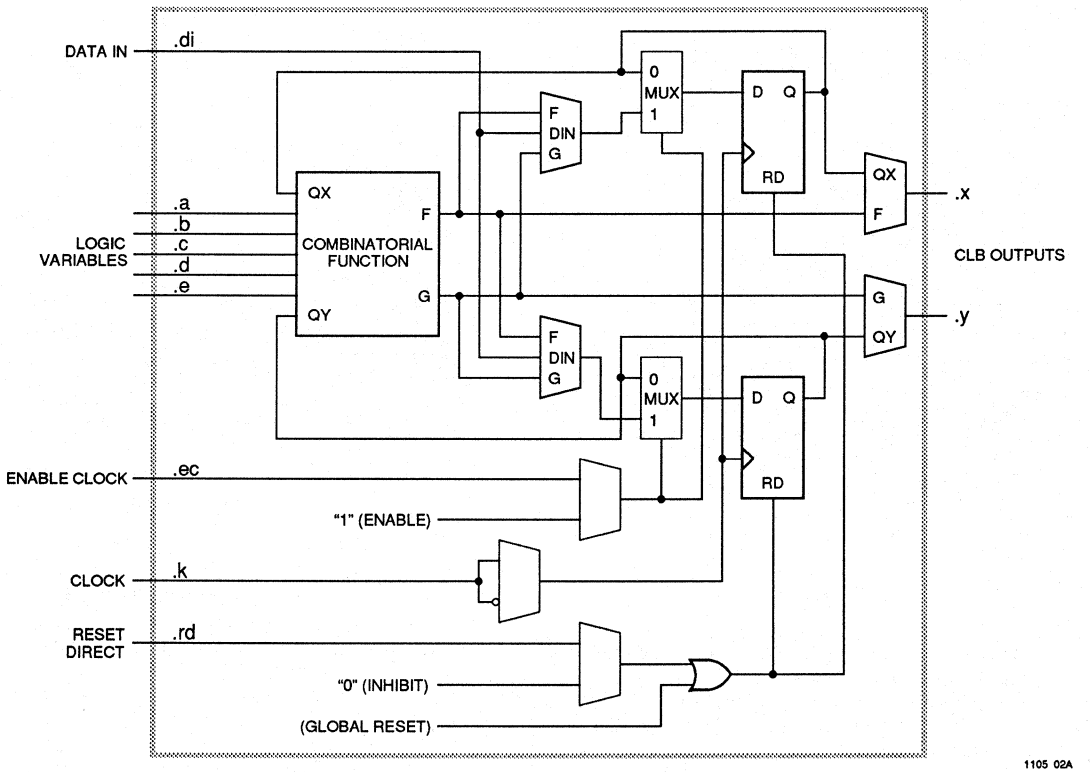


Figure 4. Configurable Logic Block. Each CLB includes a combinatorial logic section, two flip-flops and a program memory controlled multiplexer selection of function.

It has: 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 reset .rd
two outputs .x and .y

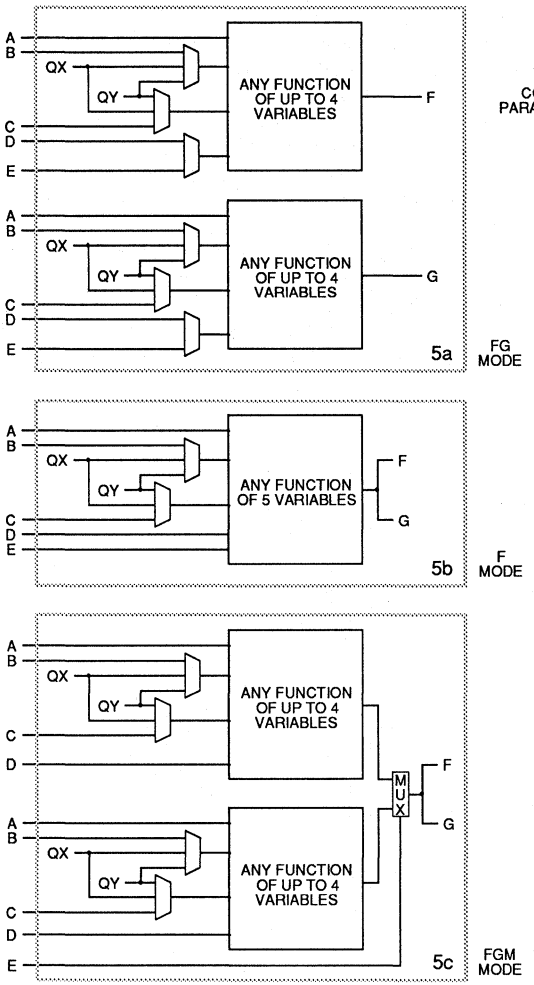


Figure 5

5a. 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 of B, C, Qx and Qy. The fourth variable can be any choice of D or E.

5b. Combinatorial Logic Option F generates any function of five variables: A, D, E and and two choices out of B, C, Qx, Qy.

5c. 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.

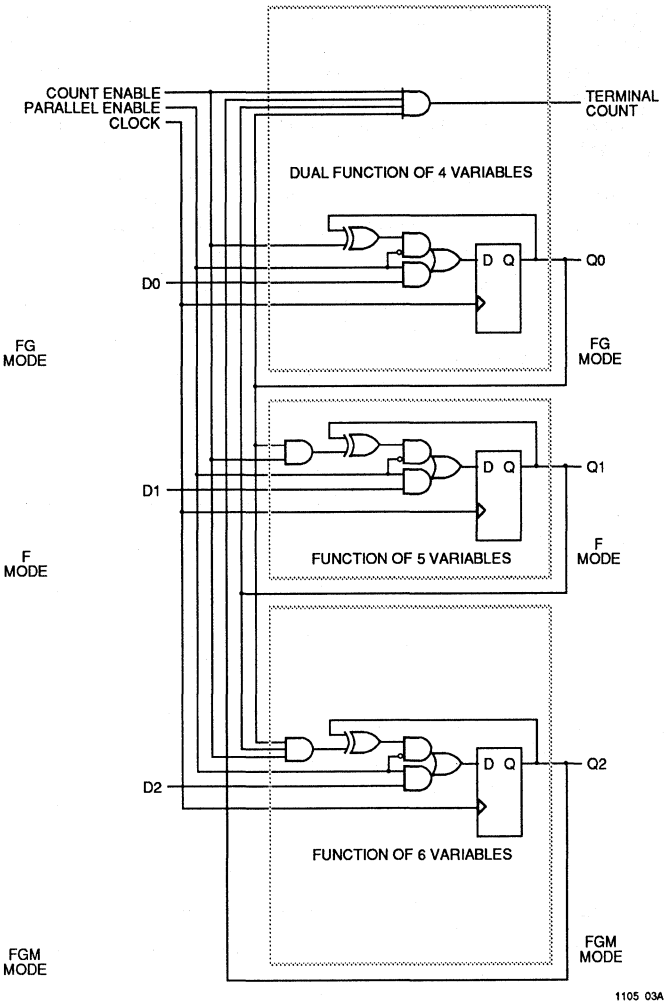


Figure 6. C8BCP Macro. TheC8BCP macro (modulo-8 binary counter with parallel enable and clock enable) uses one combinatorial logic block of each option.

reset by the active Low chip input, $\overline{\text{RESET}}$, 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 logic block. This programmable inversion eliminates the need to route both phases of a clock signal throughout the device. Flexible routing allows use of common or individual CLB clocking.

The combinatorial-logic portion of the logic block 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 5a, or a single function of five variables as shown in Figure 5b, or some functions of seven variables as shown in Figure 5c. Figure 6 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 logic blocks and IOBs.

PROGRAMMABLE INTERCONNECT

Programmable-interconnection resources in the Logic Cell Array provide routing paths to connect inputs and outputs of the I/O and logic blocks into logic networks. Interconnections between blocks are composed from 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 7 is an example of a routed net. The XACT development system provides automatic routing of these interconnections. Interactive routing (Editnet) is also available for design optimization. The inputs of the logic or IOBs are multiplexers which can be programmed to select an input network from the adjacent interconnect segments. ***As the switch connections to block inputs are unidirectional (as are block outputs) they are usable only for block input connection and not routing.*** Figure 8 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
- Long Lines (multiplexed busses and wide AND gates)

General Purpose Interconnect

General purpose interconnect, as shown in Figure 9, 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 using Editnet to select the desired pairs of matrix pins to be connected or disconnected. The legitimate switching matrix combinations for each pin are indicated in Figure 10 and may be highlighted by the use of the Show-Matrix command in XACT.

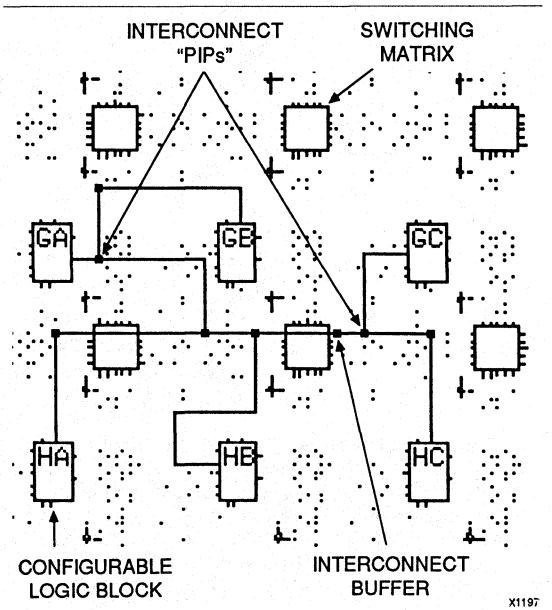


Figure 7. An XACT view of routing resources used to form a typical interconnection network from CLB GA.

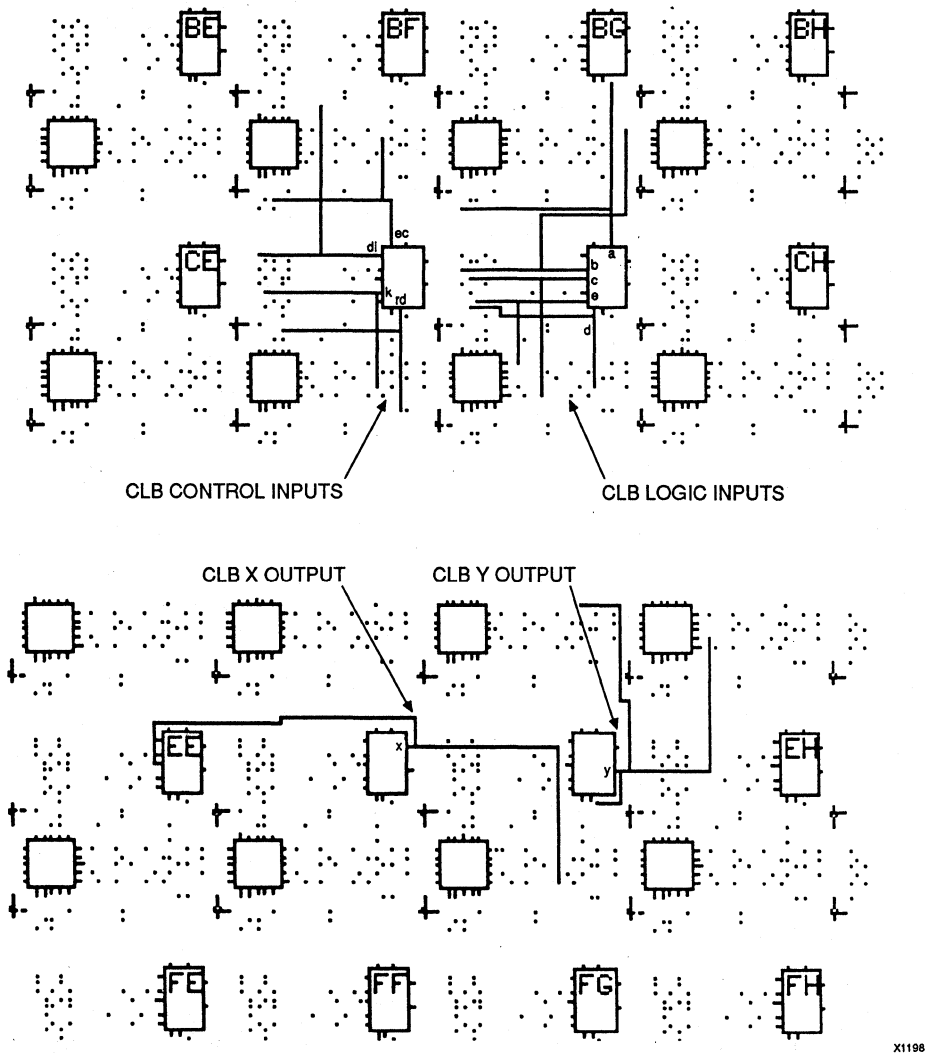


Figure 8. XACT Development System Locations of interconnect access, CLB control inputs, logic inputs and outputs. The dot pattern represents the available programmable interconnection points (PIPs).

Some of the interconnect PIPs are directional. This is indicated on the XACT design editor status line:

ND is a nondirectional interconnection.
D:H->V is a PIP which drives from a horizontal to a vertical line.
D:V->H is a PIP which drives from a vertical to a horizontal line.
D:C->T is a "T" PIP which drives from a cross of a T to the tail.
D:CW is a corner PIP which drives in the clockwise direction.
P0 indicates the PIP is non-conducting , P1 is "on."

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 and to the right and may be highlighted by the use of the "Show BIDI" command in XACT. The other PIPs adjacent to the matrices are access to or from long lines. The development system automatically defines the buffer direction based on the location of the interconnection network source. The delay calculator of the XACT 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 by an XACT option.

Direct Interconnect

Direct interconnect, shown in Figure 11, provides the most efficient implementation of networks between adjacent logic 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 intercon-

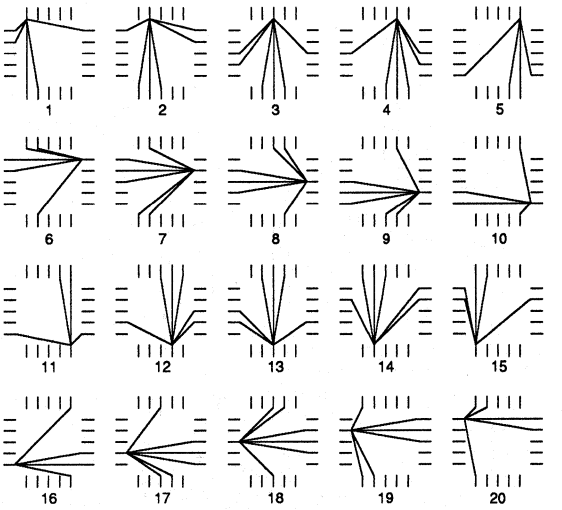


Figure 10. Switch Matrix Interconnection Options for Each Pin. Switch matrices on the edges are different. Use Show Matrix menu option in XACT

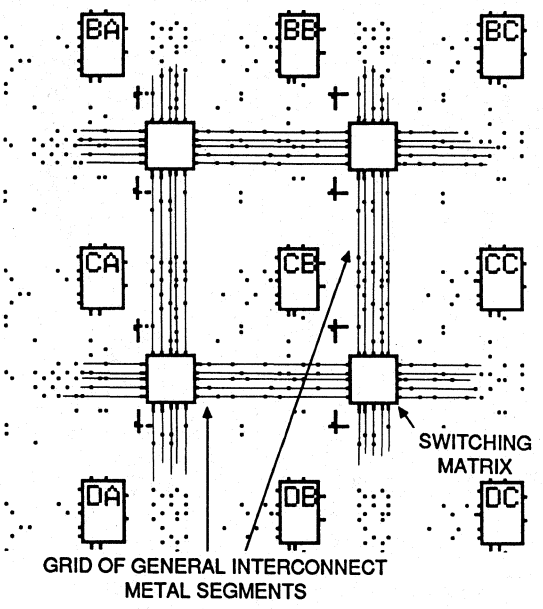


Figure 9. LCA General-Purpose Interconnect. Composed of a grid of metal segments which may be interconnected through switch matrices to form networks for CLB and IOB inputs and outputs.

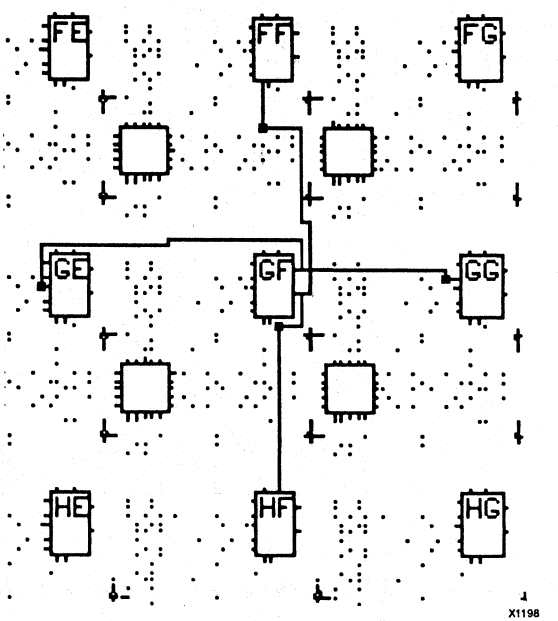


Figure 11. CLB .X and .Y Outputs. The .x and .y outputs of each CLB have single contact, direct access to inputs of adjacent CLBs.

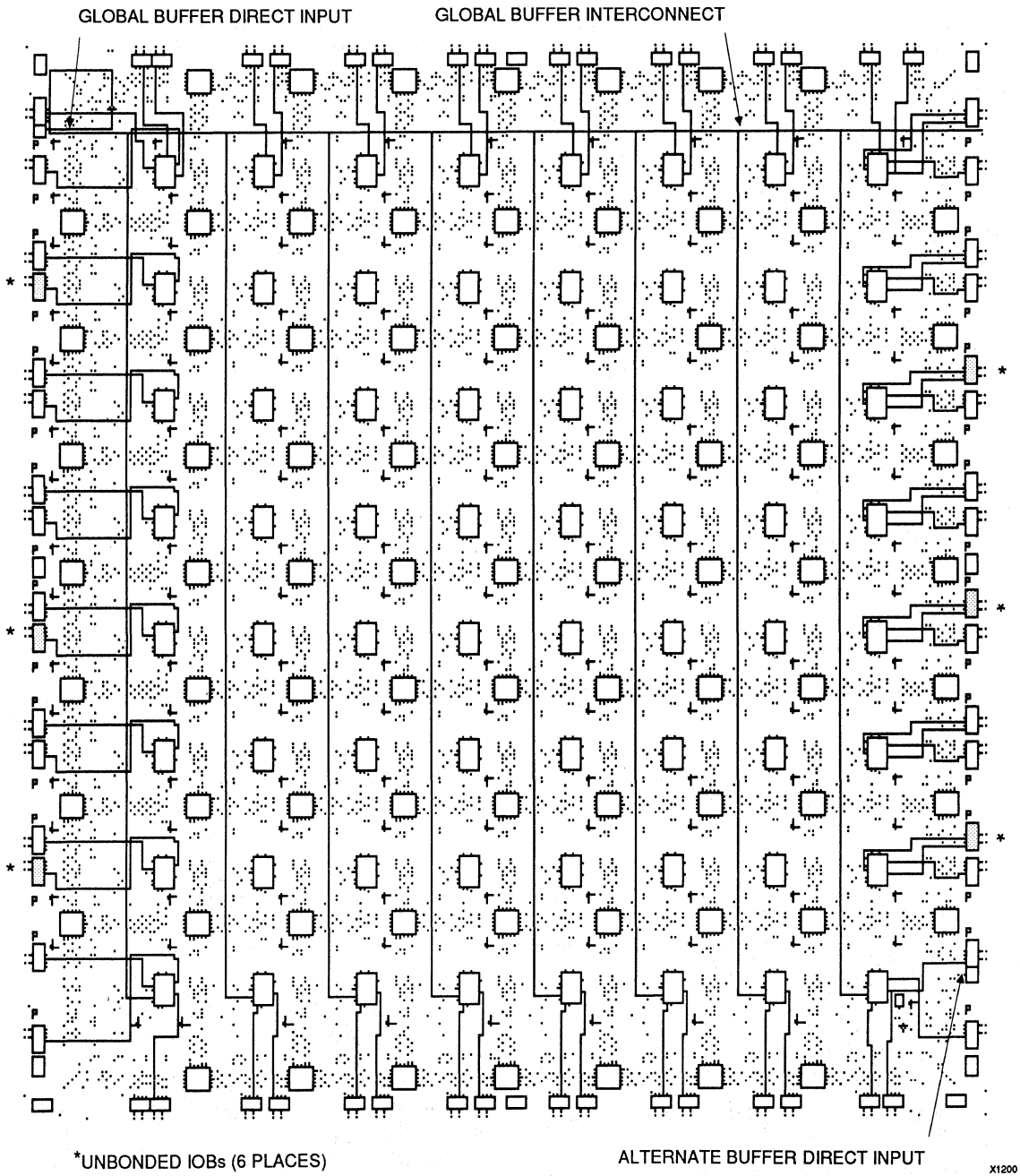


Figure 12. X3020 Die-Edge IOBs. The X3020 die-edge IOBs are provided with direct access to adjacent CLBs.

nect 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 12.

Long Lines

The long lines bypass the switch matrices and are intended primarily for signals that must travel a long distance, or must have minimum skew among multiple destinations. Long lines, shown in Figure 13, run vertically and horizontally the height or width of the interconnect area. Each interconnection column has three vertical long lines, and each interconnection row has two horizontal long lines. Two additional long lines are located adjacent to the outer sets of switching matrices. In devices larger than the XC3020, two vertical long lines in each column are connectable half-length lines. On the XC3020, only the outer long lines are connectable half-length lines.

Long lines 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 long lines are shown in Figure 14. Isolation buffers are provided at each input to a long line and are enabled automatically by the development system when a connection is made.

A buffer in the upper left corner of the LCA 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 I/O and logic blocks. 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 long line that can drive programmed connections to a vertical long line in each interconnection column. This alternate buffer also has low skew and high fan-out. The network formed by this alternate buffer's long lines can be selected to drive the *.k* inputs of the logic blocks. 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 long lines. 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 15a. The user is required to avoid contention

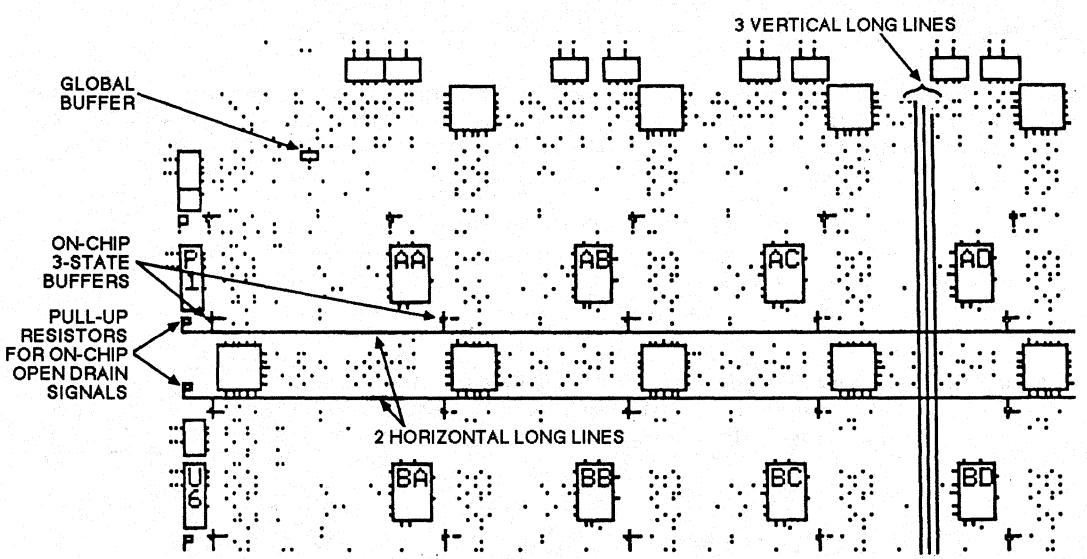


Figure 13. Horizontal and Vertical Long Lines. These long lines provide high fan-out, low-skew signal distribution in each row and column. The global buffer in the upper left die corner drives a common line throughout the LCA.

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 long line Low. See Figure 15b. Pull-up resistors are available at each end of the long line 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

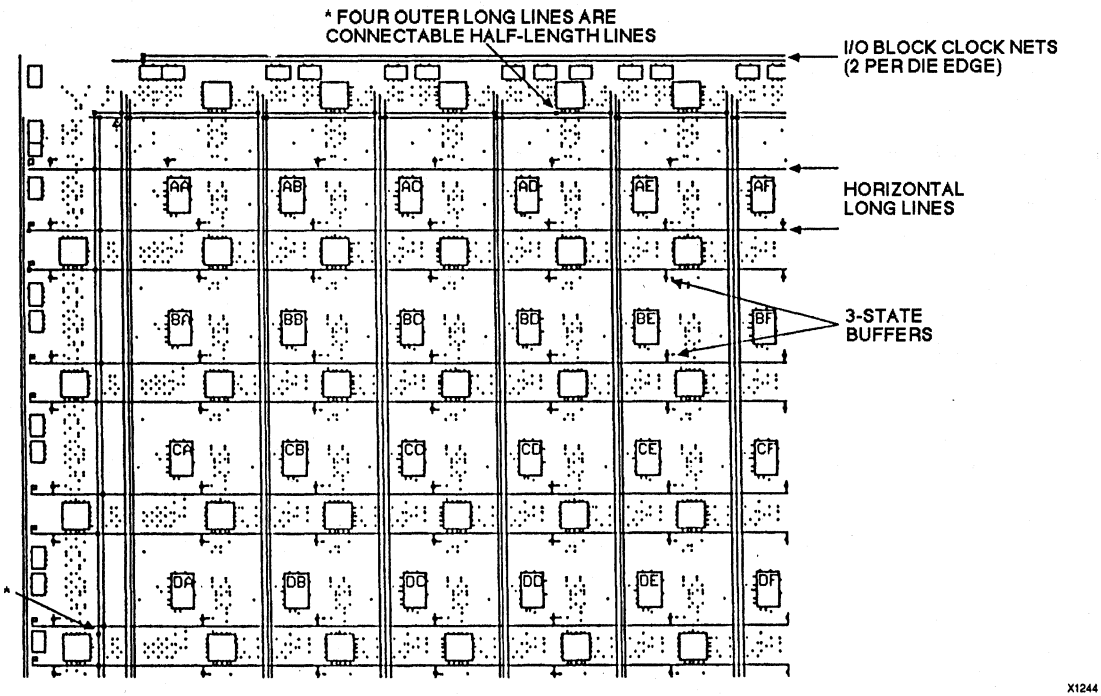


Figure 14. Programmable Interconnection of Long Lines. This is provided at the edges of the routing area. Three-state buffers allow the use of horizontal long lines to form on-chip wired-AND and multiplexed buses. The left two vertical long lines per column (except 3020) and the outer perimeter long lines may be programmed as connectible half-length.

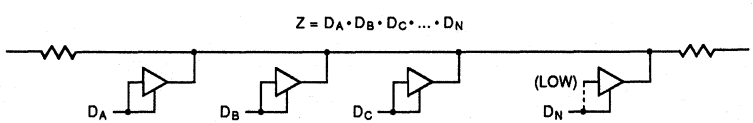
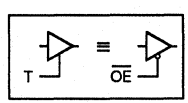


Figure 15a. **3-State Buffers Implement a Wired-AND Function.** When all the buffer 3-state lines are High, (high impedance), the pull-up resistor(s) provide the High output. The buffer inputs are driven by the control signals or a Low.



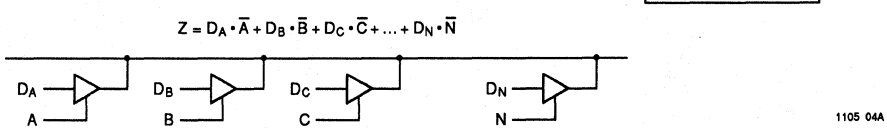


Figure 15b. 3-State Buffers Implement a Multiplexer.
The selection is accomplished by the buffer 3-state signal.

to prevent contention through multiple active buffers of conflicting levels on a common line. Figure 16 shows 3-state buffers, long lines and pull-up resistors.

CRYSTAL OSCILLATOR

Figure 16 also shows the location of an internal high speed inverting amplifier which 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 by MAKEBITS 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 17. A divide by two option is available to assure symmetry. The oscillator circuit becomes active before configuration is complete in order to allow the oscillator to stabilize. Actual internal connection is delayed until completion of configuration. In Figure 17 the feedback resistor R1, between the output and input, biases the amplifier at threshold. The value should be as large as practical to minimize loading of the crystal. 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 one-

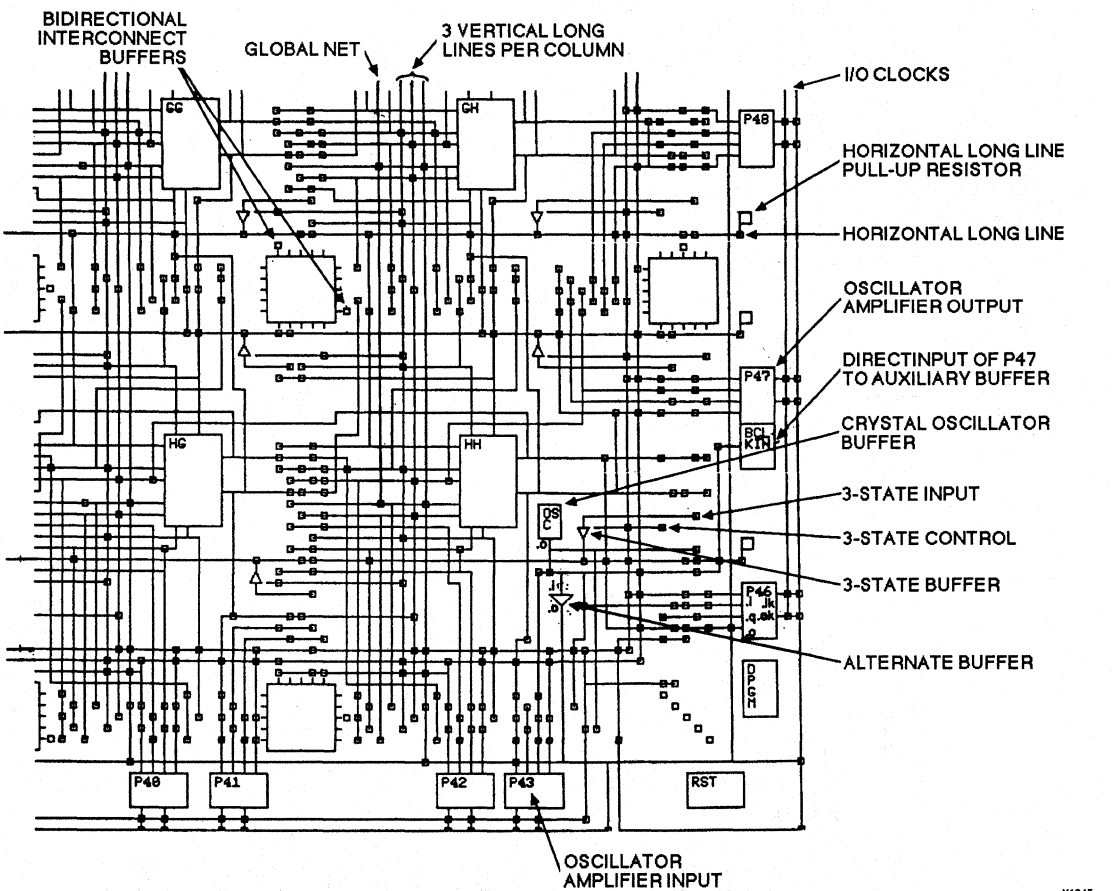


Figure 16. XACT Development System. An extra large view of possible interconnections in the lower right corner of the XC3020.

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 the R-C networks. When the oscillator inverter is not used, these IOBs and their package pins are available for general user I/O.

PROGRAMMING

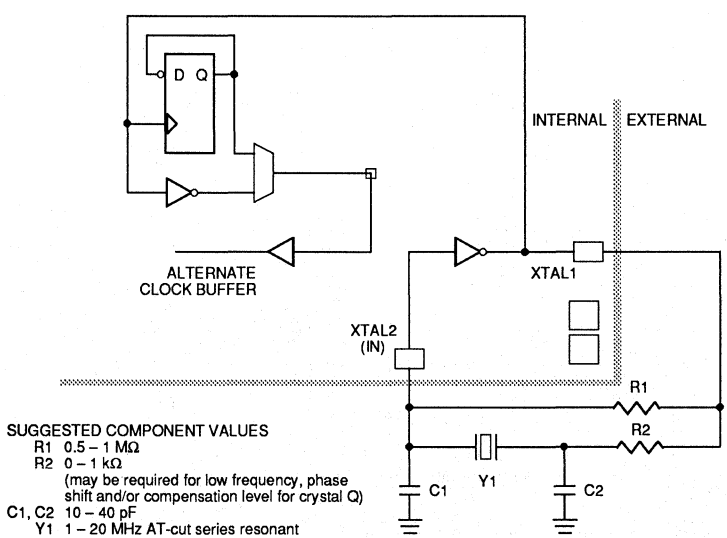
Initialization Phase

An internal power-on-reset circuit is triggered when power is applied. When Vcc reaches the voltage at which portions of the LCA begin to operate (nominally 2.5 to 3 V), the programmable I/O output buffers are disabled 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.

In Master configuration modes, the LCA 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 LCA 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)

Table 1

M0	M1	M2	Clock	Mode	Data
0	0	0	active	Master	Bit Serial
0	0	1	active	Master	Byte Wide Addr. = 0000 up
0	1	0	—	reserved	—
0	1	1	active	Master	Byte Wide Addr. = FFFF down
1	0	0	—	reserved	—
1	0	1	passive	Peripheral	Byte Wide
1	1	0	—	reserved	—
1	1	1	passive	Slave	Bit Serial



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

1105 14C

Figure 17. Crystal Oscillator Inverter. When activated in the MAKEBITS program 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.

very slow. Figure 18 shows the state sequences. At the end of Initialization the LCA enters the Clear state where it clears the configuration memory. The active Low, open-drain initialization signal $\overline{INIT}$ indicates when the Initialization and Clear states are complete. The LCA tests for the absence of an external active Low $\overline{RESET}$ before it makes a final sample of the mode lines and enters the Configuration state. An external wired-AND of one or more $\overline{INIT}$ pins can be used to control configuration by the assertion of the active low $\overline{RESET}$ of a master mode device or to signal a processor that the LCAs are not yet initialized.

If a configuration has begun, a re-assertion of $\overline{RESET}$ for a minimum of three internal timer cycles will be recognized and the LCA will initiate an abort, returning to the Clear state to clear the partially loaded configuration memory words. The LCA will then re-sample $\overline{RESET}$ and the mode lines before re-entering the Configuration state. A reprogram is initiated when a configured LCA senses a High to Low transition on the DONE/$\overline{PROG}$ package pin. The LCA 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 Logic Cell Arrays, of assorted sizes, to begin operation in a synchronized fashion. The configuration program generated by the MakePROM program of the XACT 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 19. All LCAs connected in series read and shift preamble and length count in on positive and out on negative configuration clock edges. An LCA 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 LCA is full and the length count does not compare, the LCA shifts any additional data through, as it did for preamble and length count.

When the LCA configuration memory is full and the length count compares, the LCA will execute a synchronous start-up sequence and become operational. See Figure 20. Three CCLK cycles after the completion of loading configuration data the user I/O pins are enabled as configured. As selected in MAKEBITS, 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/$\overline{PROG}$ output signal. DONE/$\overline{PROG}$ 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 when an LCA is in its Initialization, Clear or Configure states. They and DONE/$\overline{PROG}$ 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

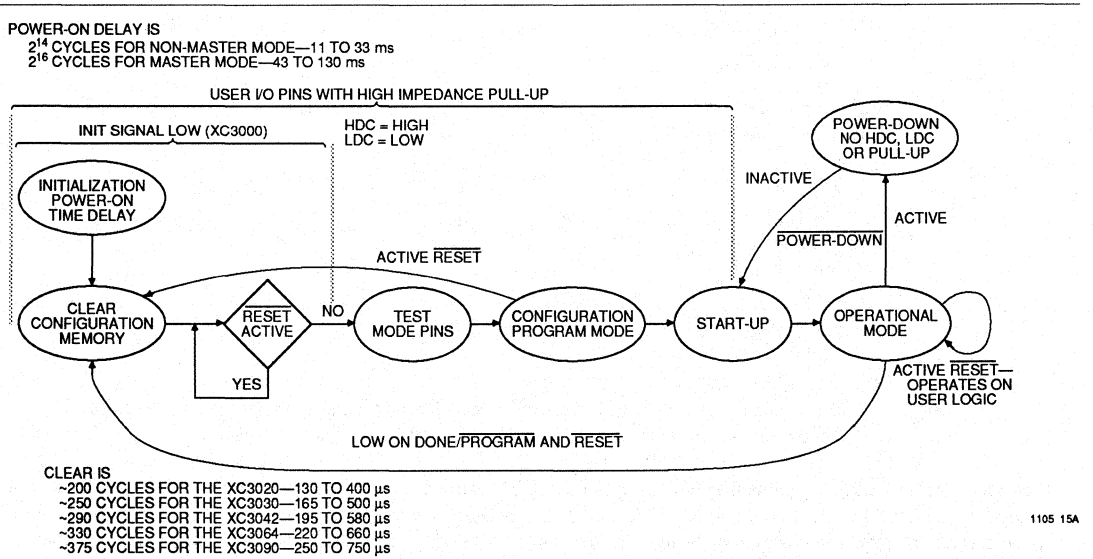


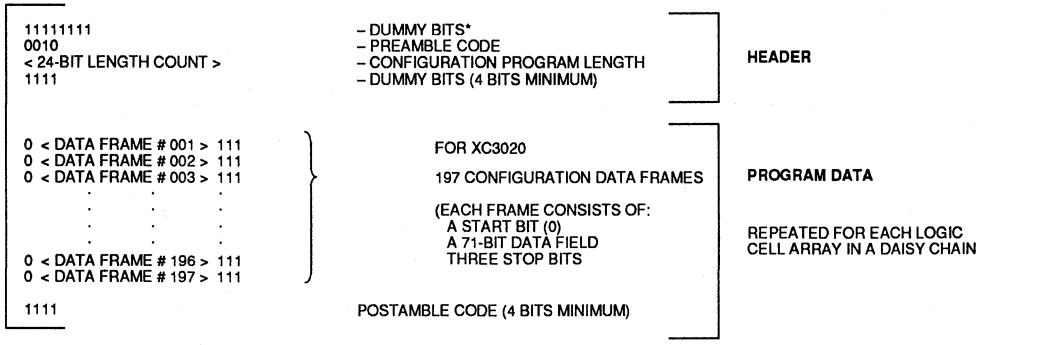
Figure 18. A State Diagram of the Configuration Process for Power-up and Reprogram.

have TTL thresholds and can change to CMOS thresholds at the completion of configuration if the user has selected CMOS thresholds. The threshold of $\overline{\text{PWRDWN}}$ 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 Logic Cell Array are loaded from an external storage at power-up and on a re-program signal. Several methods of automatic and controlled loading of the required data are available. Logic levels applied to mode



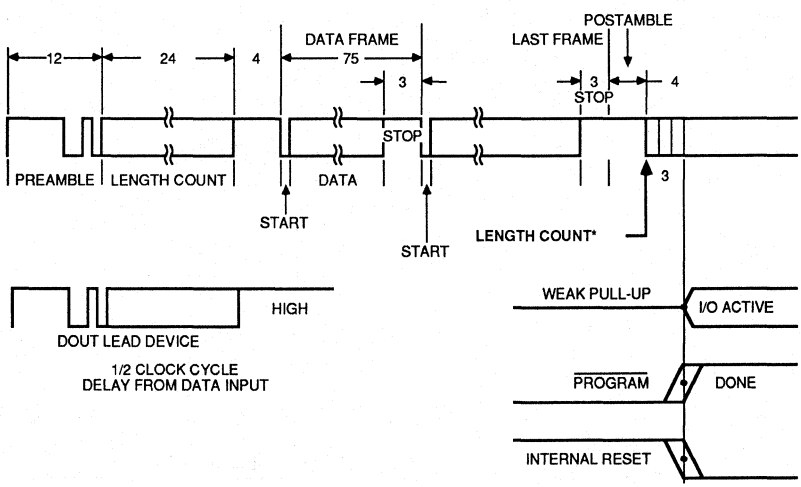
Device	XC3020	XC3030	XC3042	XC3064	XC3090
Gates	2000	3000	4200	6400	9000
CLBs Row x Col	64 (8 x 8)	100 (10 x 10)	144 (12 x 12)	224 (16 x 14)	320 (20 x 16)
IOBs	64	80	96	120	144
Flip-flops	256	360	480	688	928
Horizontal Long Lines	16	20	24	32	40
TBUFs/Horizontal LL	9	11	13	15	17
Bits per Frame (including1 start and 3 stop bits)	75	92	108	140	172
Frames	197	241	285	329	373
Program Data = Bits x Frames + 4 bits (excludes header)	14779	22176	30784	46064	64160
PROM size (bits) = Program Data + 40-bit Header	14819	22216	30824	46104	64200

Figure 19. Internal Configuration Data Structure for an LCA. This shows the preamble, length count and data frames which are generated by the XACT Development System.

The Length Count produced by the MAKEBIT program = [(40-bit preamble + sum of program data + 1 per daisy chain device) rounded up to multiple of 8] – ($2 \le K \le 4$) 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.

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. Various Xilinx Field Programmable Gate Arrays have different sizes and numbers of data frames. To maintain compatibility between various device types, the Xilinx 2000 and 3000 product families use compatible configuration formats. For the XC3020, configuration requires 14779 bits for each device, arranged in 197 data frames. An additional 40 bits are used in the header. See Figure 20. The specific data format for each device is produced by the MAKEBITS command of 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 'MAKE PROM' command of the XACT development system. A compatibility exception precludes the use of a 2000-series device as the master for 3000-series devices if their DONE or RESET are programmed to occur after their outputs become active. The "tie" option of the MAKEBITS program 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 FLAGNET command of EDITLCA can be used to indicate nets which must not be used to drive the remaining unused routing, as that might affect timing of user nets. NORESTORE will retain the results of TIE for timing analysis with QUERYNET before RESTORE returns the design to the untied condition. TIE can be omitted for quick breadboard iterations where a few additional milliamps of Icc are acceptable.

The configuration bitstream begins with High preamble bits, a 4-bit preamble code and a 24-bit length count. When configuration is initiated, a counter in the LCA 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 LCA, it is internally assembled into a data word. As each data word is completely assembled, it is 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.



* The configuration data consists of a composite 40-bit preamble/length count, followed by one or more concatenated LCA 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.

1105 06B

Figure 20. Configuration and Start-up of One or More LCAs.

Two user-programmable pins are defined in the unconfigured Logic Cell array. High During Configuration (HDC) and Low During Configuration ($\overline{LDC}$) as well as DONE/$\overline{PROG}$ may be used as external control signals during configuration. In Master mode configurations it is convenient to use $\overline{LDC}$ 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 in the MAKEBITS program 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/$\overline{PROG}$ output can be AND-tied with multiple LCAs 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 18 illustrates the configuration process.

Master Mode

In Master mode, the LCA device 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. Serial Master 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 21. Parallel Master Low and Master High modes automatically use parallel data sup-

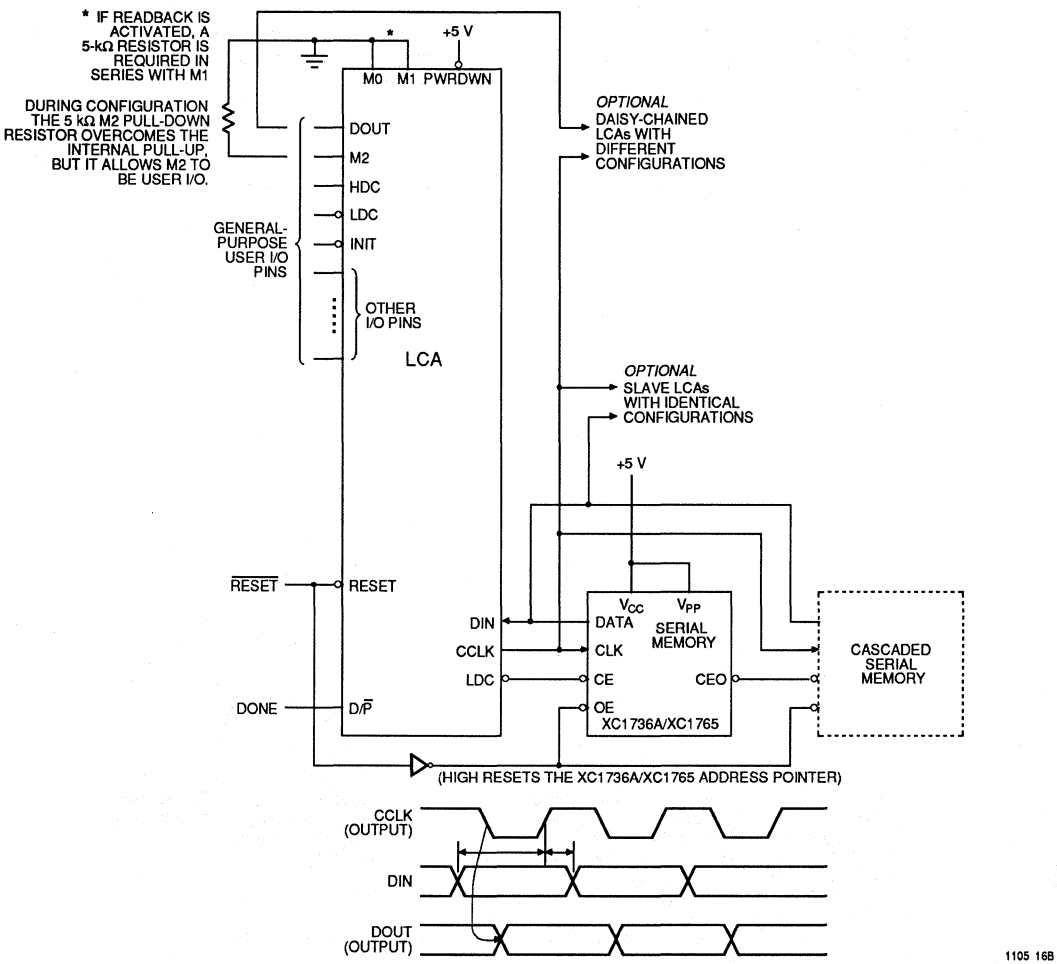


Figure 21. Master Serial Mode. The one-time-programmable XC1736A/XC1765 Serial Configuration PROM supports automatic loading of configuration programs up to 36K bits. Multiple devices can be cascaded to support additional LCAs. An early DONE inhibits the XC1736A data output a CCLK cycle before the LCA I/Os become active.

plied to the D0–D7 pins in response to the 16-bit address generated by the LCA. Figure 22 shows an example of the parallel Master mode connections required. The LCA 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. For Master High or Low, data bytes are read in parallel by each Read Clock (RCLK) and internally serialized by the Configuration Clock. As each data byte is read, the least significant bit of the next byte, D0, becomes the next bit in the internal serial configuration word. One Maste-mode LCA can be used to interface the configuration program-store and pass additional concatenated configuration data to additional LCAs in a serial daisy-chain fashion. CCLK is provided for the slaved devices and their serialized data is supplied from DOUT to DIN - DOUT to DIN etc.

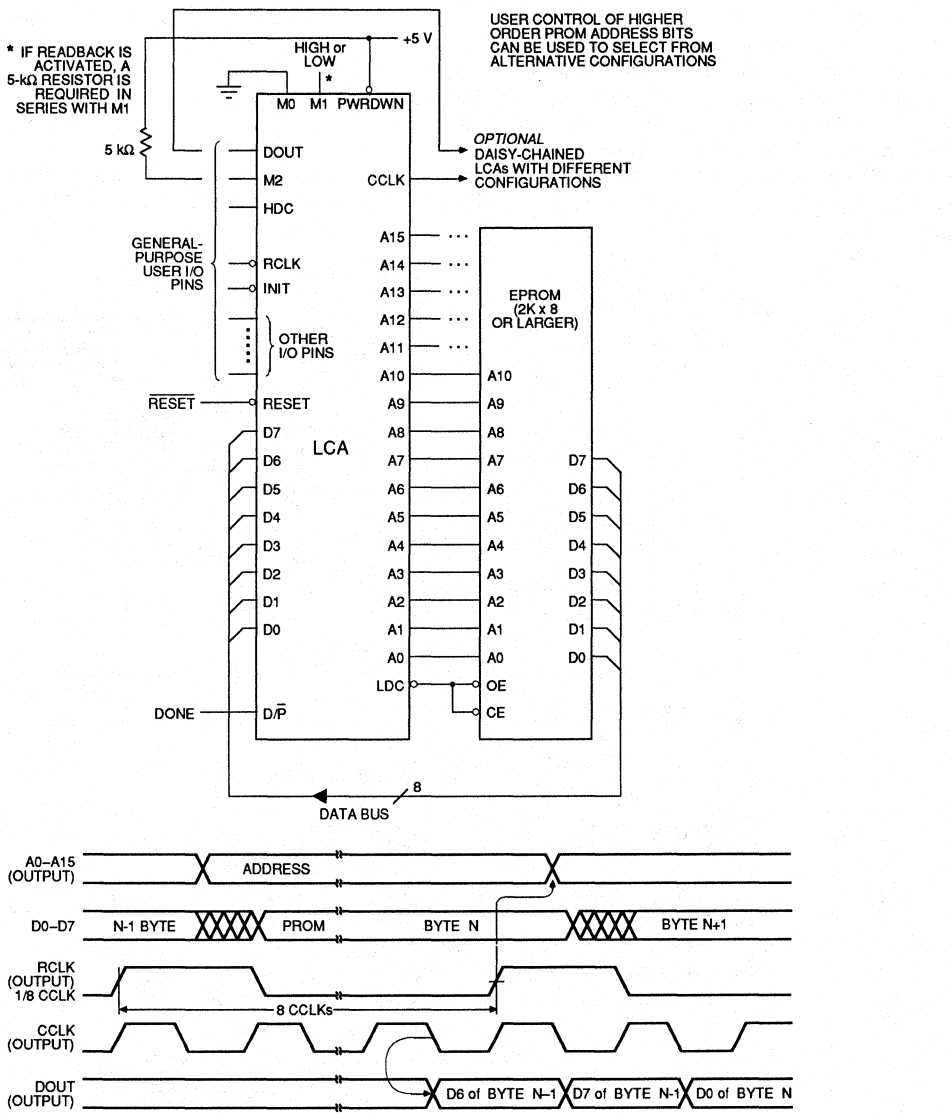


Figure 22. Master Parallel Mode. Configuration data are loaded automatically from an external byte wide PROM. An early DONE inhibits the PROM outputs a CCLK before the LCA I/O become active.

Peripheral Mode

Peripheral mode provides a simplified interface through which the device may be loaded byte-wide, as a processor peripheral. Figure 23 shows the peripheral mode connections. Processor write cycles are decoded from the common assertion of the active low Write Strobe ($\overline{WS}$), and two active low and one active high Chip Selects ($\overline{CS0}$, $\overline{CS1}$, CS2). If all these signals are not available, the unused inputs should be driven to their respective active levels. The Logic Cell Array will accept one byte of configuration data on the D0–D7 inputs for each selected processor Write cycle. Each byte of data is loaded into a buffer register. The LCA 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/$\overline{BUSY}$ 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 Mode

Slave mode provides a simple interface for loading the Logic Cell Array configuration as shown in Figure 24. Serial data are supplied in conjunction with a synchronizing input clock. Most Slave mode applications are in daisy-chain configurations in which the data input are supplied by the previous Logic Cell Array'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 XACT development system is used to create a composite configuration for selected LCAs including: a pre-

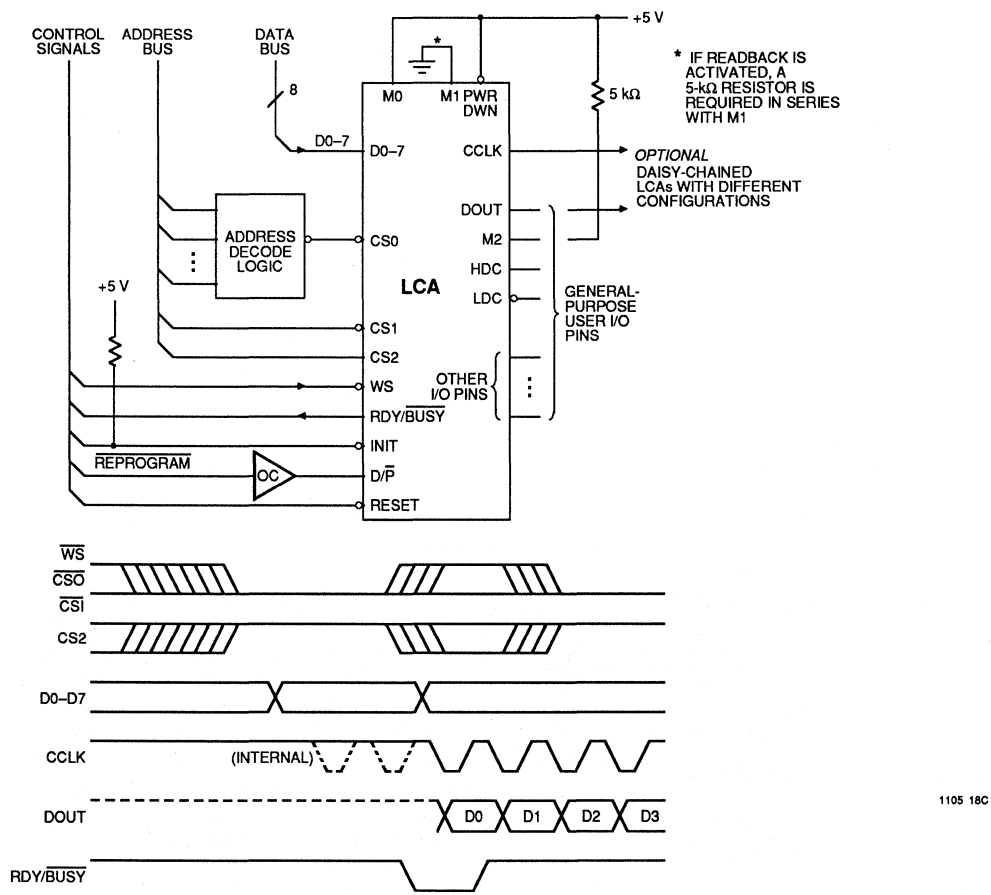


Figure 23. Peripheral Mode. Configuration data are loaded using a byte-wide data bus from a microprocessor.

amble, 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 are passed through the lead device and appear 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 LCAs. Data are 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 include 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 XACT development system bitstream generation process.

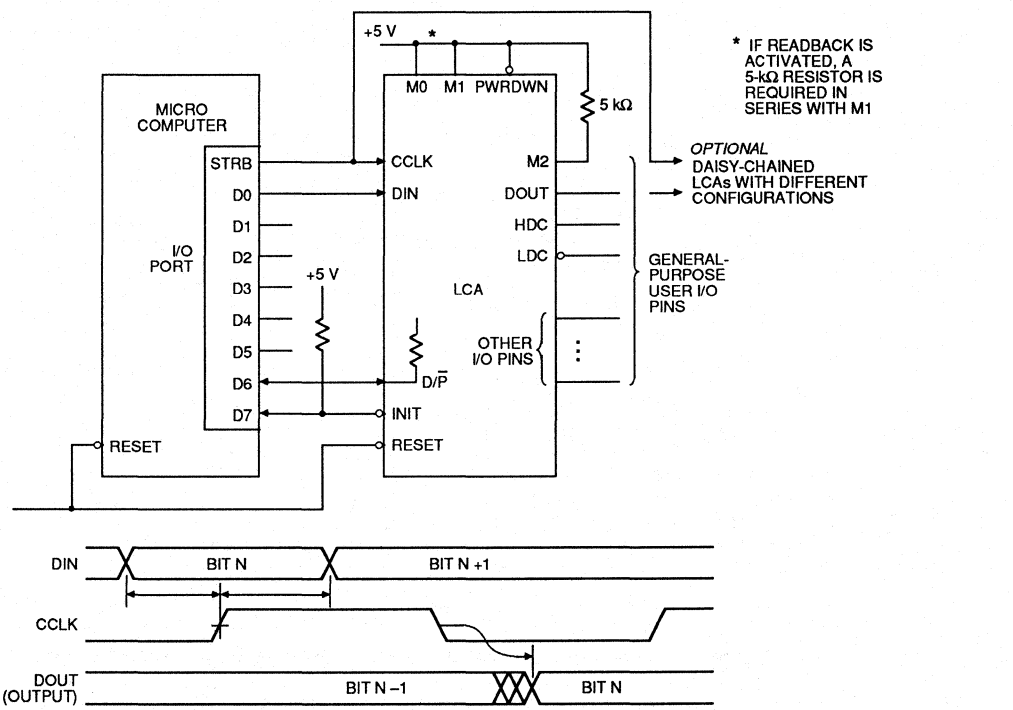


Figure 24. Slave Mode. Bit-serial configuration data are read at rising edge of the CCLK. Data on DOUT are provided on the falling edge of CCLK.

Input Thresholds

Prior to the completion of configuration all LCA 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 $\overline{\text{PWRDWN}}$ 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 Logic Cell 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 with the XACTOR In-Circuit debugger. There are three options in generating the configuration bitstream:

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

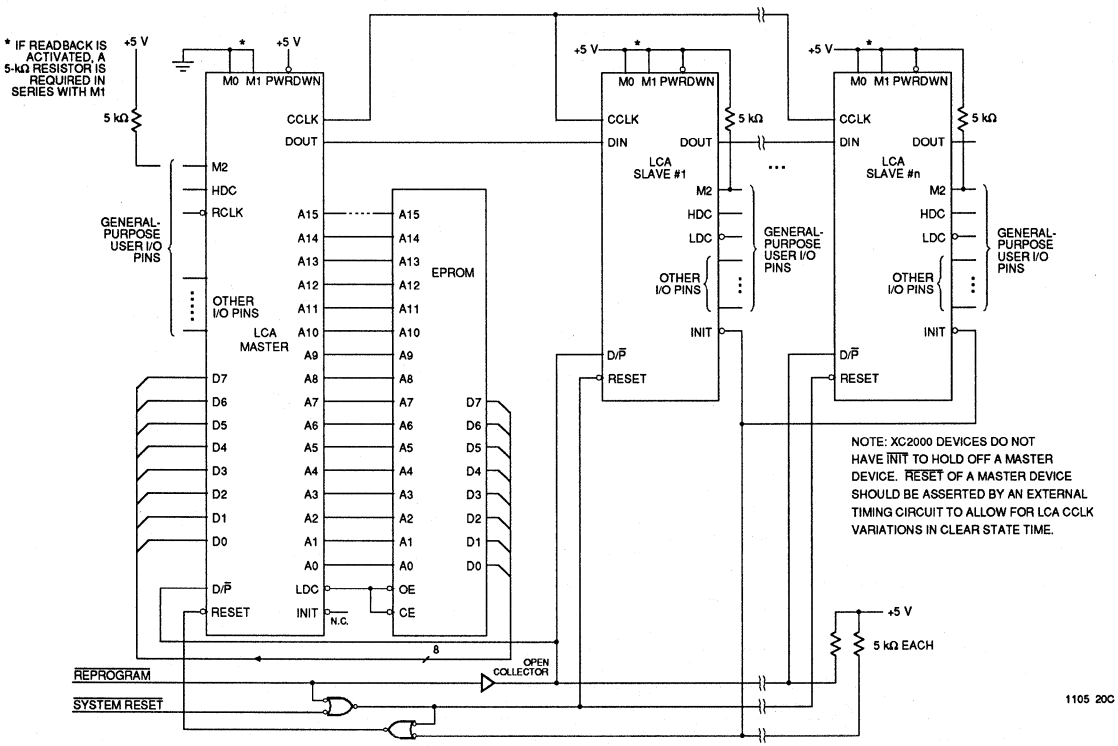


Figure 25. Master Mode Configuration with Daisy Chained Slave Mode Devices.
All are configured from the common EPROM source. The Slave mode device $\overline{\text{INIT}}$ signals delay the Master device configuration until they are initialized. A well defined termination of SYSTEM RESET is needed when controlling multiple LCAs.

Any XC3000 slave driven by an XC2000 master mode device must use "early DONE and early internal RESET". (The XC2000 master will not supply the extra clock required by a "late" programmed XC3000.)

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. Once the Readback command has been given, the input CCLK is driven by external logic to read back each data bit in a format similar to loading. After two dummy bits, the first data frame is shifted out, in inverted sense, on the M1/$\overline{\text{RDATA}}$ (Read Data) pin. All data frames must be read back to complete the process and return the Mode Select and CCLK pins to their normal functions.

The Readback data includes the current state of each internal logic block storage element, and the state of the [*.i* and *.ri*] connection pins on each IOB. These data are imbedded into unused configuration bit positions during Readback. This state information is used by the XACT 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.

Re-program

The LCA configuration memory can be re-written while the device is operating in the user's system. 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 LCA internal timing generator. When re-program 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 are 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 it recognizes a stable request, the Logic Cell Array will hold a Low until the new configuration has been completed. Even if the re-program request is externally held Low beyond the configuration period, the Logic Cell Array will begin operation upon completion of configuration.

DONE Pull-up

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

DONE Timing

The timing of the DONE status signal can be controlled by a selection in the MAKEBITS program to occur a CCLK cycle before, or after, the timing of outputs being activated. See Figure 20. This facilitates control of external functions such as a PROM enable or holding a system in a wait state.

RESET Timing

As with DONE timing, the timing of the release of the internal RESET can be controlled by a selection in the MAKEBITS program to occur a CCLK cycle before, or after, the timing of outputs being enabled. See Figure 20. This reset maintains all user programmable flip-flops and latches in a zero state during configuration.

Crystal Oscillator Division

A selection in the MAKEBITS program allows the user to incorporate a dedicated divide-by-two flip-flop in the crystal oscillator function. This provides higher assurance of a symmetrical timing signal. Although the frequency stability of crystal oscillators is high, the symmetry of the waveform can be affected by bias or feedback drive.

PERFORMANCE

Device Performance

The LCA high performance is due in part to the manufacturing process, which is similar to that used for high-speed CMOS static memories. Performance can be measured in terms of minimum propagation times for logic elements. Traditionally, the toggle frequency of a flip-flop has been used to describe the overall performance of a gate array. The configuration for determining the toggle performance of the Logic Cell Array is shown in Figure 26. The flip-flop output Q is fed back through the combinatorial logic as $\overline{Q}$ to form the toggle flip-flop.

Actual LCA performance is determined by the timing of critical paths, including both the fixed timing for the logic and storage elements in that path, and the timing associated with the routing of the network. Internal worst-case timing values are included in the performance data to allow the user to make the best use of the capabilities of the device. The XACT development system timing calculator or XACT generated simulation models should be used to calculate worst case paths by using actual impedance and loading information. Figure 27 shows a variety of elements which are involved in determining system performance. Actual measurement of internal timing is not practical and often only the sum of component timing is relevant as in the case of input to output. The relationship between input and output timing is arbitrary and only the total determines performance. Timing components of internal functions may be determined by measurement of differences at the pins of the package. A synchronous logic function which involves a clock to block-output, and a block-input to clock set-up is capable of higher speed operation than a logic configuration of two synchronous blocks with an extra combinatorial block level between them. System clock rates to 60% of the toggle frequency are practical for logic in which an extra combinatorial level is located between synchronized blocks. This allows implementation of functions of up to 25 variables. The use of the wired-AND is also available for wide, high-speed functions.

Logic Block Performance

Logic block performance is expressed as the propagation time from the interconnect point at the input of the combinatorial logic to the output of the block in the interconnect area. Combinatorial performance is independent of the specific logic function because of the table look-up based implementation. Timing is difierent when the combinatorial logic is used in conjunction with the storage element. For the combinatorial logic function driving the data input of the storage element, the critical timing is data set-up relative to the clock edge provided to the flip-flop element. The delay from the clock source to the output of the logic block is critical in the timing of signals produced by storage elements. Loading of a logic-block output is limited only by the resulting propagation delay of the larger interconnect network. Speed performance of the logic block is a function of supply voltage and temperature. See Figure 29.

Interconnect Performance

Interconnect performance depends on the routing resource used to implement the signal path. As discussed earlier, direct interconnect from block to block provides a fast path for a signal. The single metal segment used for long lines exhibits low resistance from end to end, but relatively high capacitance. Signals driven through a programmable switch will have the additional impedance of the switch added to their normal drive impedance.

General-purpose interconnect performance depends on the number of switches and segments used, the presence of the bidirectional repowering buffers and the overall loading on the signal path at all points along the path. In calculating the worst-case timing for a general interconnect path the timing-calculator portion of the XACT development system accounts for all of these elements. As an approximation, interconnect timing is proportional to the summation of totals of local metal segments beyond each programmable switch. In effect, the time is a sum of R-C time each approximated by an R times the total C it drives. The R of the switch and the C of the interconnect is a function of the particular device performance grade. For a string of three local interconnects, the approximate time at the first segment, after the first switch resistance would be three units; an additional two units after the next switch plus an additional unit after the last switch in the chain. The interconnect R-C chain terminates at each re-powering buffer. The capacitance of the actual block inputs is not significant; the capacitance is in the interconnect metal and switches. See Figure 28.

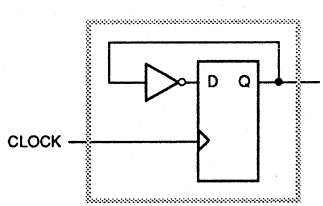
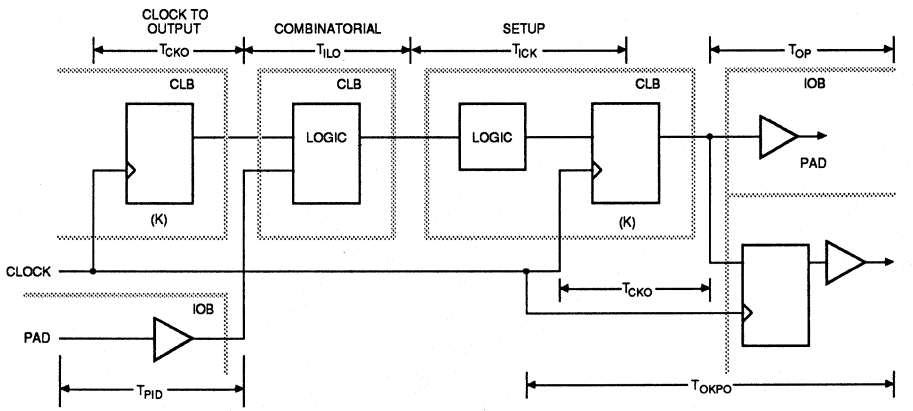


Figure 26. Toggle Flip-Flop. This is used to characterize device performance.



		Speed Grade	-70		-100		-125		Units
	Description	Symbol	Min	Max	Min	Max	Min	Max	
Logic input to Output	Combinatorial	T_{ILO}		9		7		5.5	ns
K Clock	To output	T_{CKO}		8		7		6	ns
	Logic-input setup	T_{ICK}	8		7		6		ns
	Logic-input hold	T_{CKI}	0		0		0		ns
Input/Output	Pad to input (direct)	T_{PID}		6		4		3	ns
	Output to pad (fast)	T_{OPF}		9		6		5	ns
	I/O clock to pad (fast)	T_{OKPO}		13		10		9	ns

Figure 27. Examples of Primary Block Speed Factors.
Actual timing is a function of various block factors combined with routing factors.
Overall performance can be evaluated with the XACT timing calculator or by an optional simulation.

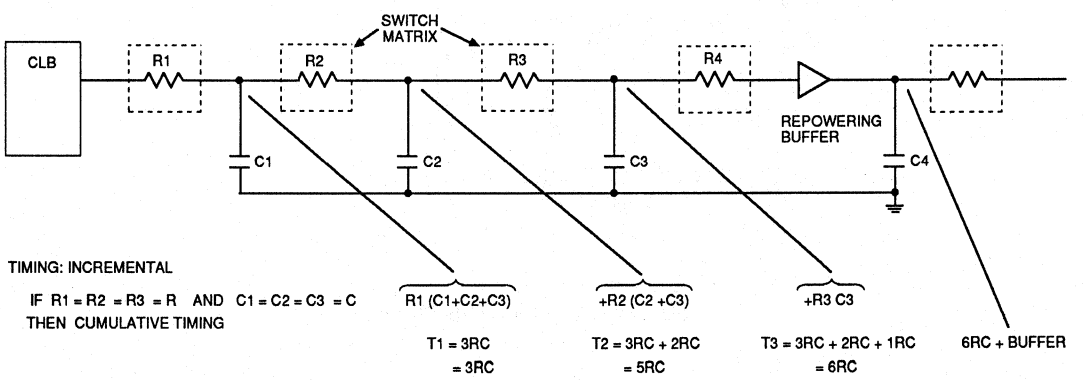


Figure 28. Interconnection Timing Example. Use of the XACT timing calculator or XACT-generated simulation model provides actual worst-case performance information.

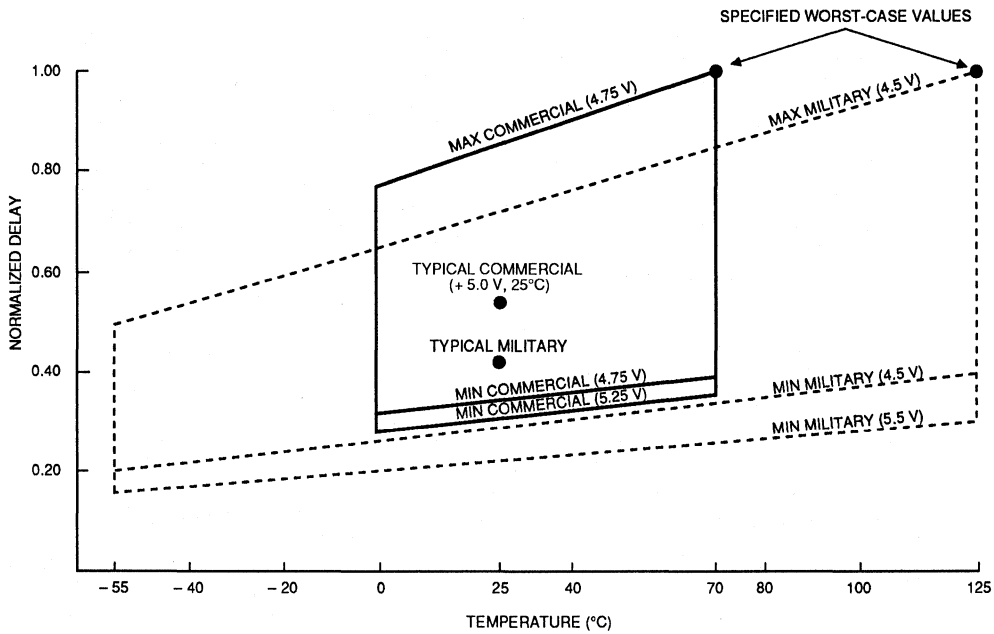


Figure 29. Relative Delay as a Function of Temperature, Supply Voltage and Processing Variations.

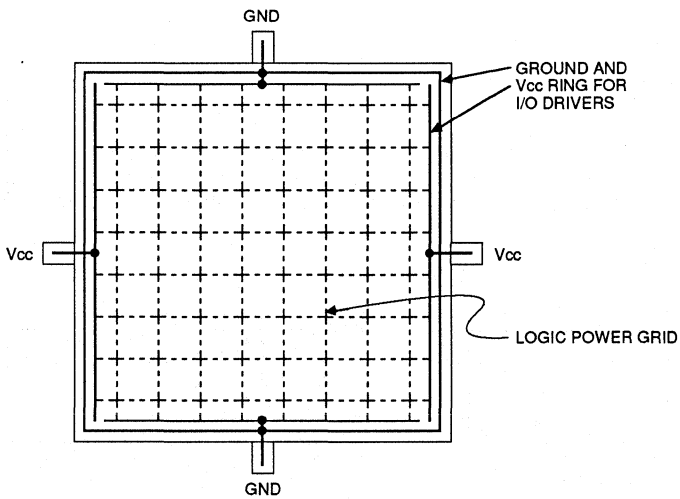


Figure 30. LCA Power Distribution.

POWER

Power Distribution

Power for the LCA is distributed through a grid to achieve high noise immunity and isolation between logic and I/O. Inside the LCA, a dedicated V_{CC} and ground ring surrounding the logic array provides power to the I/O drivers. See Figure 30. An independent matrix of V_{CC} and ground lines supplies the interior logic of the device. This power distribution grid provides a stable supply and ground for all internal logic, providing the external package power pins are all connected and appropriately decoupled. Typically a 0.1-µF capacitor connected near the VCC and ground pins will provide adequate decoupling.

Output buffers capable of driving the specified 4-mA loads under worst-case conditions may be capable of driving 25 to 30 times that current in a best case. Noise can be reduced by minimizing external load capacitance and reducing simultaneous output transitions in the same direction. It may also be beneficial to locate heavily loaded output buffers near the ground pads. The I/O Block output buffers have a slew-limited mode which should be used where output rise and fall times are not speed critical. Slew-limited outputs maintain their dc drive capability, but generate less external reflections and internal noise. A maximum total external capacitive load for simultaneous fast mode switching in the same direction is 500 pF per power/ground pin pair. Four slew-rate limited outputs this total is four times larger.

Power Consumption

The Logic Cell Array exhibits the low power consumption characteristic of CMOS ICs. For any design, Figure 31 can be used to calculate the total power requirement based on the sum of the capacitive and dc loads both external and internal. The configuration option of TTL chip input threshold requires power for the threshold reference. The power required by the static memory cells that hold the configuration data is very low and may be maintained in a power-down mode.

Typically, most of power dissipation is produced by external capacitive loads on the output buffers. This load and frequency dependent power is 25 µW/pF/MHz per output. Another component of I/O power is the dc loading on each output pin by devices driven by the Logic Cell Array.

Internal power dissipation is a function of the number and size of the nodes, and the frequency at which they change. In an LCA, the fraction of nodes changing on a given clock is typically low (10–20%). For example, in a large binary counter, the average clock cycle produces changes equal to one CLB output at the clock frequency. Typical global clock-buffer power is between 1.7 mW/MHz for the XC3020 and 3.6 mW/MHz for the XC3090. The internal capacitive load is more a function of interconnect than fan-out. With a typical load of three general interconnect segments, each CLB output requires about 0.4 mW per MHz of its output frequency.

$$\text{Total Power} = V_{CC} \cdot I_{CCO} + \text{external (dc + capacitive)} + \text{internal (CLB + IOB + long line + pull-up)}$$

Because the control storage of the Logic Cell Array is CMOS static memory, its cells require a very low standby current for data retention. In some systems, this low data retention current characteristic can be used as a method of preserving configurations in the event of a primary power loss. The Logic Cell Array has built in power-down logic which, when activated, will disable normal operation of the device and retain only the configuration data. All internal operation is suspended and output buffers are placed in their high-impedance state with no pull-ups. Power-down data retention is possible with a simple battery-backup circuit because the power requirement is extremely low. For retention at 2.4 V, the required current can be as low as 10 µA at room temperature.

To force the Logic Cell Array into the Power-Down state, the user must pull the $\overline{\text{PWRDWN}}$ pin Low and continue to supply a retention voltage to the VCC pins. When normal power is restored, VCC is elevated to its normal operating voltage and $\overline{\text{PWRDWN}}$ is returned to a High. The Logic Cell Array resumes operation with the same internal sequence that occurs at the conclusion of configuration. Internal-I/O and logic-block storage elements will be reset, the outputs will become enabled and the DONE/$\overline{\text{PROG}}$ pin will be released. No configuration programming is involved.

When the power supply is removed from a CMOS device, it is possible to supply some power from an input signal. The conventional electro-static input protection is implemented with diodes to the supply and ground. A positive voltage applied to an input (or output) will cause the positive protection diode to conduct and drive the V_{CC} connection. This condition can produce invalid power conditions and should be avoided. A large series resistor might be used to limit the current or a bipolar buffer may be used to isolate the input signal.

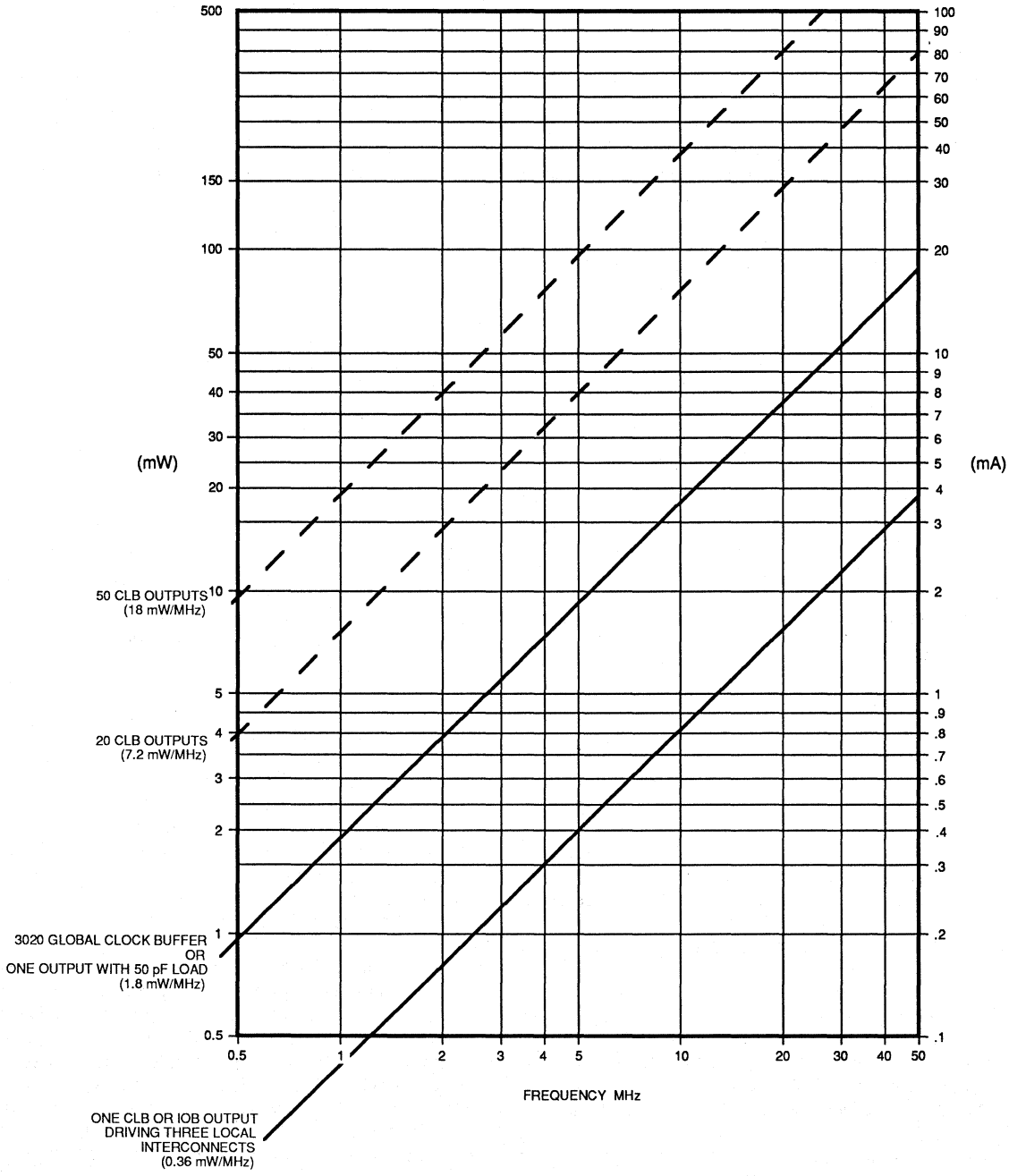


Figure 31. LCA Power Consumption by Element. Total chip power is the sum of Vcc•Icco plus effective internal and external values of frequency dependent capacitive charging currents and duty factor dependent resistive loads.

PIN DESCRIPTIONS

Permanently Dedicated Pins.

V_{CC}
Two to eight (depending on package type) connections to the nominal +5 V supply voltage. All must be connected.

GND
Two to eight (depending on package type) connections to ground. All must be connected.

$\overline{PWRDWN}$
A Low on this CMOS-compatible input stops all internal activity, but retains configuration. All flip-flops and latches are reset, all outputs are 3-stated, and all inputs are interpreted as High, independent of their actual level. While $\overline{PWRDWN}$ is Low, V_{CC} may be reduced to any value >2.3 V. When $\overline{PWDWN}$ returns High, the LCA becomes operational with DONE Low for two cycles of the internal 1-MHz clock. During configuration, $\overline{PWRDWN}$ must be High. If not used, $\overline{PWRDWN}$ must be tied to V_{CC}.

$\overline{RESET}$
This is an active Low input which has three functions.

Prior to the start of configuration, a Low input will delay the start of the configuration process. An internal circuit senses the application of power and begins a minimal time-out cycle. When the time-out and $\overline{RESET}$ are complete, the levels of the M lines are sampled and configuration begins.

If $\overline{RESET}$ is asserted during a configuration, the LCA device is re-initialized and restarts the configuration at the termination of $\overline{RESET}$.

If $\overline{RESET}$ is asserted after configuration is complete, it provides a global asynchronous reset of all IOB and CLB storage elements of the LCA device.

CCLK
During configuration, Configuration Clock is an output of an LCA in Master mode or Peripheral mode, but an input in Slave mode. During Readback, CCLK is a clock input for shifting configuration data out of the LCA

CCLK drives dynamic circuitry inside the LCA. The Low time may, therefore, not exceed a few microseconds. When used as an input, CCLK must be "parked High". An internal pull-up resistor maintains High when the pin is not being driven.

DONE
DONE is an open-drain output, configurable with or without an internal pull-up resistor. At the completion of configuration, the LCA circuitry becomes active in a synchronous order, and DONE is programmed to go active High either one cycle before or after the outputs go active.

$\overline{PROG}$
Once configuration is done, a High-to-Low transition of this pin will cause an initialization of the LCA and start a reconfiguration.

M0
As Mode 0, this input and M1, M2 are sampled before the start of configuration to establish the configuration mode to be used.

RTRIG
A Low-to-High input transition, after configuration is complete, acts as a Read Trigger and initiates a Readback of configuration and storage-element data clocked by CCLK. By selecting the appropriate Readback option when generating the bitstream, this operation may be limited to a single Readback, or be inhibited altogether.

M1
As Mode 1, this input and M0, M2 are sampled before the start of configuration to establish the configuration mode to be used. If Readback is never used, M1 can be tied directly to ground or V_{CC}. If Readback is ever used, M1 must use a 5-kΩ resistor to ground or V_{CC}, to accommodate the $\overline{RDATA}$ output.

$\overline{RDATA}$
As an active Low Read Data, after configuration is complete, this pin is the output of the Readback data.

User I/O Pins that can have special functions.

M2
During configuration, this input has a weak pull-up resistor. Together with M0 and M1, it is sampled before the start of configuration to establish the configuration mode to be used. After configuration, this pin is a user-programmable I/O pin.

HDC
During configuration, this output is held at a High level to indicate that configuration is not yet complete. After configuration, this pin is a user-programmable I/O pin.

$\overline{\text{LDC}}$
During Configuration, this output is held at a Low level to indicate that the configuration is not yet complete. After configuration, this pin is a user-programmable I/O pin. $\overline{\text{LDC}}$ is particularly useful in Master mode as a Low enable for an EPROM, but it must then be programmed as a High after configuration.

$\overline{\text{INIT}}$
This is an active Low open-drain output which is held Low during the power stabilization and internal clearing of the configuration memory. It can be used to indicate status to a configuring microprocessor or, as a wired AND of several slave mode devices, a hold-off signal for a master mode device. After configuration this pin becomes a user programmable I/O pin.

BCLKIN
This is a direct CMOS level input to the alternate clock buffer (Auxiliary Buffer) in the lower right corner.

XTL1
This user I/O pin can be used to operate as the output of an amplifier driving an external crystal and bias circuitry.

XTL2
This user I/O pin can be used as the input of an amplifier connected to an external crystal and bias circuitry. The I/O Block is left unconfigured. The oscillator configuration is activated by routing a net from the oscillator buffer symbol output and by the MAKEBITS program.

$\overline{\text{CS0}}$, $\overline{\text{CS1}}$, CS2, $\overline{\text{WS}}$
These four inputs represent a set of signals, three active Low and one active High, that are used to control configuration-data entry in the Peripheral mode. Simultaneous assertion of all four inputs generates a Write to the internal data buffer. The removal of any assertion clocks in the D0-D7 data. In Master-Parallel mode, $\overline{\text{WS}}$ and CS2 are the A0 and A1 outputs. After configuration, these pins are user-programmable I/O pins.

$\overline{\text{RCLK}}$
During Master parallel mode configuration $\overline{\text{RCLK}}$ represents a "read" of an external dynamic memory device (normally not used).

RDY/$\overline{\text{BUSY}}$
During Peripheral parallel mode configuration this pin indicates when the chip is ready for another byte of data to be written to it. After configuration is complete, this pin becomes a user programmed I/O pin.

D0–D7
This set of eight pins represents the parallel configuration byte for the parallel Master and Peripheral modes. After configuration is complete they are user programmed I/O pins.

A0–A15
During Master Parallel mode, these 16 pins present an address output for a configuration EPROM. After configuration, they are user-programmable I/O pins.

DIN
During Slave or Master Serial configuration, this pin is used as a serial-data input. In the Master or Peripheral configuration, this is the Data 0 input.

DOUT
During configuration this pin is used to output serial-configuration data to the DIN pin of a daisy-chained slave.

TCLKIN
This is a direct CMOS level input to the global clock buffer.

Unrestricted User I/O Pins.

I/O
An I/O pin may be programmed by the user to be an Input or an Output pin following configuration. All unrestricted I/O pins, plus the special pins mentioned on the following page, have a weak pull-up resistor of 40 to 100 kΩ that becomes active as soon as the device powers up, and stays active until the end of configuration.

XC3000 Family Configuration Pin Assignments

CONFIGURATION MODE: <M2:M1:M0>					***		**								USER
SLAVE <1:1:1>	MASTER-SER <0:0:0>	PERIPHERAL <1:0:1>	MASTER-HIGH <1:1:0>	MASTER-LOW <1:0:0>	44 PLCC	68 PLCC	84 PLCC	84 PGA	100 PQFP	100 CQFP	132 PGA	160 PQFP	164 CQFP	175 PGA	OPERATION
PWR DWN (I)	PWR DWN (I)	PWR DWN (I)	PWR DWN (I)	PWR DWN (I)	7	10	12	B2	29	14	A1	159	20	B2	PWR DWN (I)
VCC	VCC	VCC	VCC	VCC	12	18	22	F3	41	26	C8	20	42	D9	VCC
M1 (HIGH) (I)	M1 (LOW) (I)	M1 (LOW) (I)	M1 (HIGH) (I)	M1 (LOW) (I)	16	25	31	J2	52	37	B13	40	62	B14	RDATA
M0 (HIGH) (I)	M0 (LOW) (I)	M0 (HIGH) (I)	M0 (LOW) (I)	M0 (LOW) (I)	17	26	32	L1	54	39	A14	42	64	B15	RTRIG (I)
M2 (HIGH) (I)	M2 (LOW) (I)	M2 (HIGH) (I)	M2 (HIGH) (I)	M2 (HIGH) (I)	18	27	33	K2	56	41	C13	44	66	C15	I/O
HDC (HIGH)	HDC (HIGH)	HDC (HIGH)	HDC (HIGH)	HDC (HIGH)	19	28	34	K3	57	42	B14	45	67	E14	I/O
LDC (LOW)	LDC (LOW)	LDC (LOW)	LDC (LOW)	LDC (LOW)	20	30	36	L3	59	44	D14	49	71	D16	I/O
INIT *	INIT *	INIT *	INIT *	INIT *	22	34	42	K6	65	50	G14	59	81	H15	I/O
GND	GND	GND	GND	GND	23	35	43	J6	66	51	H12	19	83	J14	GND
					26	43	53	L11	76	61	M13	76	99	P15	XTL2 OR I/O
RESET (I)	RESET (I)	RESET (I)	RESET (I)	RESET (I)	27	44	54	K10	78	63	P14	78	101	R15	RESET (I)
DONE	DONE	DONE	DONE	DONE	28	45	55	J10	80	65	N13	80	103	R14	PROGRAM (I)
		DATA 7 (I)	DATA 7 (I)	DATA 7 (I)		46	56	K11	81	66	M12	81	104	N13	I/O
					30	47	57	J11	82	67	P13	82	105	T14	XTL1 OR I/O
		DATA 6 (I)	DATA 6 (I)	DATA 6 (I)		48	58	H10	83	68	N11	86	109	P12	I/O
		DATA 5 (I)	DATA 5 (I)	DATA 5 (I)		49	60	F10	87	72	M9	92	115	T11	I/O
		CS0 (I)				50	61	G10	88	73	N9	93	116	R10	I/O
		DATA 4 (I)	DATA 4 (I)	DATA 4 (I)		51	62	G11	89	74	N8	98	121	R9	I/O
VCC	VCC	VCC	VCC	VCC	34	52	64	F9	91	76	M8	100	123	N9	VCC
		DATA 3 (I)	DATA 3 (I)	DATA 3 (I)		53	65	F11	92	77	N7	102	125	P8	I/O
		CS1 (I)				54	66	E11	93	78	P6	103	126	R8	I/O
		DATA 2 (I)	DATA 2 (I)	DATA 2 (I)		55	67	E10	94	79	M6	108	131	R7	I/O
		DATA 1 (I)	DATA 1 (I)	DATA 1 (I)		56	70	D10	98	83	M5	114	137	R5	I/O
		RDY/BUSY	RCLK	RCLK		57	71	C11	99	84	N4	115	138	P5	I/O
DIN (I)	DIN (I)	DATA 0 (I)	DATA 0 (I)	DATA 0 (I)	38	58	72	B11	100	85	N2	119	143	R3	I/O
DOUT	DOUT	DOUT	DOUT	DOUT	39	59	73	C10	1	86	M3	120	144	N4	I/O
CCLK (I)	CCLK	CCLK	CCLK	CCLK	40	60	74	A11	2	87	P1	121	145	R2	CCLK (I)
		WS (I)	A0	A0		61	75	B10	5	90	M2	124	148	P2	I/O
		CS2 (I)	A1	A1		62	76	B9	6	91	N1	125	149	M3	I/O
			A2	A2		63	77	A10	8	93	L2	128	152	P1	I/O
			A3	A3		64	78	A9	9	94	L1	129	153	N1	I/O
			A15	A15		65	81	B6	12	97	K1	132	156	M1	I/O
			A4	A4		66	82	B7	13	98	J2	133	157	L2	I/O
			A14	A14		67	83	A7	14	99	H1	136	160	K2	I/O
			A5	A5		68	84	C7	15	100	H2	137	161	K1	I/O
GND	GND	GND	GND	GND	1	1	1	C6	16	1	H3	139	164	J3	GND
			A13	A13		2	2	A6	17	2	G2	141	2	H2	I/O
			A6	A6		3	3	A5	18	3	G1	142	3	H1	I/O
			A12	A12		4	4	B5	19	4	F2	147	8	F2	I/O
			A7	A7		5	5	C5	20	5	E1	148	9	E1	I/O
			A11	A11		6	8	A3	23	8	D1	151	12	D1	I/O
			A8	A8		7	9	A2	24	9	D2	152	13	C1	I/O
			A10	A10		8	10	B3	25	10	B1	155	16	E3	I/O
			A9	A9		9	11	A1	26	11	C2	156	17	C2	I/O
						X	X	X	X	X					XC3020
					X	X	X	X	X	X					XC3030
							X	X	X	X	X				XC3042
							X**				X				XC3064
							X**					X	X	X	XC3090

AVAILABLE PACKAGES

REPRESENTS A 50-kΩ TO 100-kΩ PULL-UP

* INIT IS AN OPEN DRAIN OUTPUT DURING CONFIGURATION

(I) REPRESENTS AN INPUT

** PIN ASSIGNMENTS FOR THE XC3064/XC3090 DIFFER FROM THOSE SHOWN. SEE PAGE 2-35.

*** PERIPHERAL MODE AND MASTER PARALLEL MODE ARE NOT SUPPORTED IN THE PC44 PACKAGE. SEE PAGE 2-33.

1105 25D

Note: Pin assignments of "PGA Footprint" PLCC sockets and PGA packages are not electrically identical. Generic I/O pins are not shown.

XC3000 FAMILY PIN ASSIGNMENTS

Xilinx offers the five different devices of the XC3000 family in a variety of surface-mount and through-hole package types, with pin counts from 44 to 175.

Each chip is offered in several package types to accomodate the available pc board space and manufacturing technology. Most package types are also offered with different chips to accomodate design changes without the need for pc board changes.

Note that there is no perfect match between the number of bonding pads on the chip and the number of pins on a package. In some cases, the chip has more pads than there are pins on the package, as indicated by the information ("unused" pads) below the line in the following table. The IOBs of the unconnected pads can still be used as storage elements if the specified propagation delays and set-up times are acceptable.

In other cases, the chip has fewer pads than there are pins on the package; therefore, some package pins are not connected (n.c.), as shown above the line in the following table.

		Number of Package Pins						
Device	**Pads**	**44**	**68**	**84**	**100**	**132**	**164**	**175**
XC3020	74	—	6 unused	10 n.c.	26 n.c.	—	—	—
XC3030	98	54 unused	30 unused	14 unused	2 n.c.	—	—	—
XC3042	118	—	—	34 unused	18 unused	14 n.c.	—	—
XC3064	142	—	—	58 unused	—	10 unused	—	—
XC3090	166	—	—	82 unused	—	—	2 unused	9 n.c.

XC3000 Family 44-Pin PLCC Pinouts

Pin No.	XC3030
1	GND
2	I/O
3	I/O
4	I/O
5	I/O
6	I/O
7	PWRDWN
8	TCLKIN-I/O
9	I/O
10	I/O
11	I/O
12	VCC
13	I/O
14	I/O
15	I/O
16	M1-RDATA
17	M0-RTRIG
18	M2-I/O
19	HDC-I/O
20	LDC-I/O
21	I/O
22	INIT-I/O

Pin No.	XC3030
23	GND
24	I/O
25	I/O
26	XTL2(IN)-I/O
27	RESET
28	DONE-PGM
29	I/O
30	XTL1(OUT)-BCLK-I/O
31	I/O
32	I/O
33	I/O
34	VCC
35	I/O
36	I/O
37	I/O
38	DIN-I/O
39	DOUT-I/O
40	CCLK
41	I/O
42	I/O
43	I/O
44	I/O

Peripheral mode and Master Parallel mode are not supported in the PC44 package

XC3000 Family 68-Pin PLCC, 84-Pin PLCC and PGA Pinouts

68 PLCC	XC-3020* XC-3030, XC-3042	84 PLCC	84 PGA
10	PWRDN	12	B2
11	TCLKIN-I/O	13	C2
	I/O*	14	B1
12	I/O	15	C1
13	I/O	16	D2
—	I/O	17	D1
14	I/O	18	E3
15	I/O	19	E2
16	I/O	20	E1
17	I/O	21	F2
18	VCC	22	F3
19	I/O	23	G3
—	I/O	24	G1
20	I/O	25	G2
21	I/O	26	F1
22	I/O	27	H1
—	I/O	28	H2
23	I/O	29	J1
24	I/O	30	K1
25	M1-RDATA	31	J2
26	M0-RTRIG	32	L1
27	M2-I/O	33	K2
28	HDC-I/O	34	K3
29	I/O	35	L2
30	LDC-I/O	36	L3
31	I/O	37	K4
	I/O*	38	L4
32	I/O	39	J5
33	I/O	40	K5
	I/O*	41	L5
34	INIT-I/O	42	K6
35	GND	43	J6
36	I/O	44	J7
37	I/O	45	L7
38	I/O	46	K7
39	I/O	47	L6
40	I/O	48	L8
41	I/O	49	K8
	I/O*	50	L9
	I/O*	51	L10
42	I/O	52	K9
43	XTL2(IN)-I/O	53	L11

68 PLCC	XC-3020* XC-3030, XC-3042	84 PLCC	84 PGA
44	RESET	54	K10
45	DONE-PG	55	J10
46	D7-I/O	56	K11
47	XTL1(OUT)-BCLKIN-I/O	57	J11
48	D6-I/O	58	H10
—	I/O	59	H11
49	D5-I/O	60	F10
50	CS0-I/O	61	G10
51	D4-I/O	62	G11
—	I/O	63	G9
52	VCC	64	F9
53	D3-I/O	65	F11
54	CS1-I/O	66	E11
55	D2-I/O	67	E10
—	I/O	68	E9
	I/O*	69	D11
56	D1-I/O	70	D10
57	RDY/BUSY-RCLK-I/O	71	C11
58	D0-DIN-I/O	72	B11
59	DOUT-I/O	73	C10
60	CCLK	74	A11
61	A0-WS-I/O	75	B10
62	A1-CS2-I/O	76	B9
63	A2-I/O	77	A10
64	A3-I/O	78	A9
	I/O*	79	B8
	I/O*	80	A8
65	A15-I/O	81	B6
66	A4-I/O	82	B7
67	A14-I/O	83	A7
68	A5-I/O	84	C7
1	GND	1	C6
2	A13-I/O	2	A6
3	A6-I/O	3	A5
4	A12-I/O	4	B5
5	A7-I/O	5	C5
	I/O*	6	A4
	I/O*	7	B4
6	A11-I/O	8	A3
7	A8-I/O	9	A2
8	A10-I/O	10	B3
9	A9-I/O	11	A1

Unprogrammed IOBs have a default pull-up. This prevents an undefined pad level for unbonded or unused IOBs. Programmed outputs are default slew-rate limited.

* This table describes the pinouts of three different chips in two different packages. The second column lists 84 of the 118 pads on the XC3042 (and 84 of the 98 pads on the XC3030) that are connected to the 84 package pins. Ten pads indicated by an asterisk, do not exist on the XC3020, which has 74 pads; therefore the corresponding pins on the 84-pin packages have no connections. Six pads, indicated by a dash (–) in the 68 PLCC column, have no connections in the 68 PLCC package, but are connected in the 84-pin package. (See table on page 2-32.)

XC3064/XC3090 84-Pin PLCC Pinouts

PLCC Pin Number	XC3064, XC3090
12	PWRDN
13	TCLKIN-I/O
14	I/O
15	I/O
16	I/O
17	I/O
18	I/O
19	I/O
20	I/O
21	GND*
22	VCC
23	I/O
24	I/O
25	I/O
26	I/O
27	I/O
28	I/O
29	I/O
30	I/O
31	M1-RDATA
32	M0-RTRIG
33	M2-I/O
34	HDC-I/O
35	I/O
36	LDC-I/O
37	I/O
38	I/O
39	I/O
40	I/O
41	INIT/I/O*
42	VCC*
43	GND
44	I/O
45	I/O
46	I/O
47	I/O
48	I/O
49	I/O
50	I/O
51	I/O
52	I/O
53	XTL2(IN)-I/O

PLCC Pin Number	XC3064, XC3090
54	RESET
55	DONE-PG
56	D7-I/O
57	XTL1(OUT)-BCLKIN-I/O
58	D6-I/O
59	I/O
61	D5-I/O
61	CS0-I/O
62	D4-I/O
63	I/O
64	VCC
65	GND*
66	D3-I/O*
67	CS1-I/O*
68	D2-I/O*
69	I/O
70	D1-I/O
71	RDY/BUSY-RCLK-I/O
72	D0-DIN-I/O
73	DOUT-I/O
74	CCLK
75	A0-WS-I/O
76	A1-CS2-I/O
77	A2-I/O
78	A3-I/O
79	I/O
80	I/O
81	A15-I/O
82	A4-I/O
83	A14-I/O
84	A5-I/O
1	GND
2	VCC*
3	A13-I/O*
4	A6-I/O*
5	A12-I/O*
6	A7-I/O*
7	I/O
8	A11-I/O
9	A8-I/O
10	A10-I/O
11	A9-I/O

Unprogrammed IOBs have a default pull-up. This prevents an undfined pad level for unbonded or unused IOBs. Programmed ouptuts are default slew-rate limited. DEVICE POWER MUST BE LESS THAN 1 WATT.

* Different pin definition than 3020/3030/3042 PC84 package

XC3000 Family 100-Pin QFP Pinouts

Pin No.		XC3020 XC3030 XC3042
CQFP	PQFP	
1	16	**GND**
2	17	A13-I/O
3	18	A6-I/O
4	19	A12-I/O
5	20	A7-I/O
6	21	**I/O***
7	22	**I/O***
8	23	A11-I/O
9	24	A8-I/O
10	25	A10-I/O
11	26	A9-I/O
12	27	**VCC***
13	28	**GND***
14	29	$\overline{\text{PWRDN}}$
15	30	TCLKIN-I/O
16	31	**I/O****
17	32	**I/O***
18	33	**I/O***
19	34	I/O
20	35	I/O
21	36	I/O
22	37	I/O
23	38	I/O
24	39	I/O
25	40	I/O
26	41	**VCC**
27	42	I/O
28	43	I/O
29	44	I/O
30	45	I/O
31	46	I/O
32	47	I/O
33	48	I/O
34	49	I/O

Pin No.		XC3020 XC3030 XC3042
CQFP	PQFP	
35	50	**I/O***
36	51	**I/O***
37	52	M1-$\overline{\text{RD}}$
38	53	**GND***
39	54	MO-RT
40	55	**VCC***
41	56	M2-I/O
42	57	HDC-I/O
43	58	I/O
44	59	$\overline{\text{LDC}}$-I/O
45	60	**I/O***
46	61	I/O*
47	62	I/O
48	63	I/O
49	64	I/O
50	65	$\overline{\text{INIT}}$-I/O
51	66	**GND**
52	67	I/O
53	68	I/O
54	69	I/O
55	70	I/O
56	71	I/O
57	72	I/O
58	73	I/O
59	74	**I/O***
60	75	**I/O***
61	76	XTAL2-I/O
62	77	**GND***
63	78	$\overline{\text{RESET}}$
64	79	**VCC***
65	80	DONE-$\overline{\text{PG}}$
66	81	D7-I/O
67	82	BCLKIN-XTAL1-I/O
68	83	D6-I/O

Pin No.		XC3020 XC3030 XC3042
CQFP	PQFP	
69	84	**I/O***
70	85	**I/O***
71	86	I/O
72	87	D5-I/O
73	88	$\overline{\text{CS0}}$-I/O
74	89	D4-I/O
75	90	I/O
76	91	**VCC**
77	92	D3-I/O
78	93	$\overline{\text{CS1}}$-I/O
79	94	D2-I/O
80	95	I/O
81	96	**I/O***
82	97	**I/O***
83	98	D1-I/O
84	99	$\overline{\text{RCLK}}$-$\overline{\text{BUSY}}$/RDY-I/O
85	100	DO-DIN-I/O
86	1	DOUT-I/O
87	2	CCLK
88	3	**VCC***
89	4	**GND***
90	5	AO-$\overline{\text{WS}}$-I/O
91	6	A1-CS2-I/O
92	7	**I/O****
93	8	A2-I/O
94	9	A3-I/O
95	10	**I/O***
96	11	**I/O***
97	12	A15-I/O
98	13	A4-I/O
99	14	A14-I/O
100	15	A5-I/O

Unprogrammed IOBs have a default pull-up.
This prevents an undefined pad level for unbonded or unused IOBs.
Programmed outputs are default slew-rate limited.

* This table describes the pinouts of three different chips in two different packges. The third column lists 100 of the 118 pads on the XC3042 that are connected to the 100 package pins. Two pads, indicated by double asterisks, do not exist on the XC3030, which has 98 pads; therefore the corresponding pins have no connections. Twenty-six pads, indicated by single or double asterisks, do not exist on the XC3020, which has 74 pads; therefore, the corresponding pins have no connections. (See table on page 2-33.)

XC3000 Family 132-Pin Ceramic and Plastic PGA Pinouts

PGA Pin Number	XC-3042 XC-3064
C4	**GND**
A1	PWRDN
C3	I/O-TCLKIN
B2	I/O
B3	I/O
A2	**I/O***
B4	I/O
C5	I/O
A3	**I/O***
A4	I/O
B5	I/O
C6	I/O
A5	I/O
B6	I/O
A6	I/O
B7	I/O
C7	**GND**
C8	**VCC**
A7	I/O
B8	I/O
A8	I/O
A9	I/O
B9	I/O
C9	I/O
A10	I/O
B10	I/O
A11	**I/O***
C10	I/O
B11	I/O
A12	**I/O***
B12	I/O
A13	**I/O***
C12	I/O

PGA Pin Number	XC-3042 XC-3064
B13	M1-RD
C11	**GND**
A14	M0-RT
D12	**VCC**
C13	M2-I/O
B14	HDC-I/O
C14	I/O
E12	I/O
D13	I/O
D14	LDC-I/O
E13	**I/O***
F12	I/O
E14	I/O
F13	I/O
F14	I/O
G13	I/O
G14	INIT-I/O
G12	**VCC**
H12	**GND**
H14	I/O
H13	I/O
J14	I/O
J13	I/O
K14	I/O
J12	I/O
K13	I/O
L14	**I/O***
L13	I/O
K12	I/O
M14	I/O
N14	I/O
M13	XTAL2(IN)-I/O
L12	**GND**

PGA Pin Number	XC-3042 XC-3064
P14	RESET
M11	**VCC**
N13	DONE-PG
M12	D7-I/O
P13	XTAL1-I/O-BCLKIN
N12	I/O
P12	I/O
N11	D6-I/O
M10	I/O
P11	**I/O***
N10	I/O
P10	I/O
M9	D5-I/O
N9	CS0-I/O
P9	**I/O***
P8	**I/O***
N8	D4-I/O
P7	I/O
M8	**VCC**
M7	**GND**
N7	D3-I/O
P6	CS1-I/O
N6	**I/O***
P5	**I/O***
M6	D2-I/O
N5	I/O
P4	I/O
P3	I/O
M5	D1-I/O
N4	RCLK-BUSY/RDY-I/O
P2	I/O
N3	I/O
N2	D0-DIN-I/O

PGA Pin Number	XC-3042 XC-3064
M3	DOUT-I/O
P1	CCLK
M4	**VCC**
L3	**GND**
M2	A0-WS-I/O
N1	A1-CS2-I/O
M1	I/O
K3	I/O
L2	A2-I/O
L1	A3-I/O
K2	I/O
J3	I/O
K1	A15-I/O
J2	A4-I/O
J1	**I/O***
H1	A14-I/O
H2	A5-I/O
H3	**GND**
G3	**VCC**
G2	A13-I/O
G1	A6-I/O
F1	**I/O***
F2	A12-I/O
E1	A7-I/O
F3	I/O
E2	I/O
D1	A11-I/O
D2	A8-I/O
E3	I/O
C1	I/O
B1	A10-I/O
C2	A9-I/O
D3	**VCC**

Unprogrammed IOBs have a default pull-up. This prevents an undefined pad level for unbonded or unused IOBs. Programmed outputs are default slew-rate limited.

* Indicates unconnected package pins (14) for the XC3042.

XC3000 Family 160-Pin PQFP Pinouts

PLCC Pin Number	XC3090
1	I/O
2	I/O
3	I/O
4	I/O
5	I/O
6	I/O
7	I/O
8	I/O
9	I/O
10	I/O
11	I/O
12	I/O
13	I/O
14	I/O
15	I/O
16	I/O
17	I/O
18	I/O
19	GND
20	VCC
21	I/O
22	I/O
23	I/O
24	I/O
25	I/O
26	I/O
27	I/O
28	I/O
29	I/O
30	I/O
31	I/O
32	I/O
33	I/O
34	I/O
35	I/O
36	I/O
37	I/O
38	I/O
39	I/O
40	M1-$\overline{\text{RDATA}}$

PLCC Pin Number	XC3090
41	GND
42	M0–RTRIG
43	VCC
44	M2-I/O
45	HDC-I/O
46	I/O
47	I/O
48	I/O
49	$\overline{\text{LDC}}$-I/O
50	I/O
51	I/O
52	I/O
53	I/O
54	I/O
55	I/O
56	I/O
57	I/O
58	I/O
59	$\overline{\text{INIT}}$-I/O
60	VCC
61	GND
62	I/O
63	I/O
64	I/O
65	I/O
66	I/O
67	I/O
68	I/O
69	I/O
70	I/O
71	I/O
72	I/O
73	I/O
74	I/O
75	I/O
76	XTAL2-I/O
77	GND
78	$\overline{\text{RESET}}$
79	VCC
80	DONE/$\overline{\text{PG}}$

PLCC Pin Number	XC3090
81	D7-I/O
82	XTAL1-I/O-BCLKIN
83	I/O
84	I/O
85	I/O
86	D6-I/O
87	I/O
88	I/O
89	I/O
90	I/O
91	I/O
92	D5-I/O
93	$\overline{\text{CS0}}$-I/O
94	I/O
95	I/O
96	I/O
97	I/O
98	D4-I/O
99	I/O
100	VSS
101	GND
102	D3-I/O
103	$\overline{\text{CS1}}$-I/O
104	I/O
105	I/O
106	I/O
107	I/O
108	D2-I/O
109	I/O
110	I/O
111	I/O
112	I/O
113	I/O
114	D1-I/O
115	RDY-$\overline{\text{BSY}}$/RCLK-I/O
116	I/O
117	I/O
118	I/O
119	D0-DIN-I/O
120	DOUT

PLCC Pin Number	XC3090
121	CCLK
122	VCC
123	GND
124	A0-$\overline{\text{WS}}$-I/O
125	A1-CS2-I/O
126	I/O
127	I/O
128	A2-I/O
129	A3-I/O
130	I/O
131	I/O
132	A15-I/O
133	A4-I/O
134	I/O
135	I/O
136	A14-I/O
137	A5-I/O
138	I/O
139	GND
140	VCC
141	A13-I/O
142	A6-I/O
143	I/O
144	I/O
145	I/O
146	I/O
147	A12-I/O
148	A7-I/O
149	I/O
150	I/O
151	A11-I/O
152	A8-I/O
153	I/O
154	I/O
155	A10-I/O
156	A9-I/O
157	VCC
158	GND
159	$\overline{\text{PWRDWN}}$
160	TCLKIN-I/O

Unprogrammed IOBs have a default pull-up. This prevents an undefined pad level for unbonded or unused IOBs.
Programmed IOBs are default slew-rate limited.

XC3000 Family 164-Pin PQFP Pinouts

CQFP Pin Number	XC-3090
20	PWRDN
21	TCLKIN-I/O
22	I/O
23	I/O
24	I/O
25	I/O
26	I/O
27	I/O
28	I/O
29	I/O
30	I/O
31	I/O
32	I/O
33	I/O
34	I/O
35	I/O
36	I/O
37	I/O
38	I/O
39	I/O
40	I/O
41	GND
42	VCC
43	I/O
44	I/O
45	I/O
46	I/O
47	I/O
48	I/O
49	I/O
50	I/O
51	I/O
52	I/O
53	I/O
54	I/O
55	I/O
56	I/O
57	I/O
58	I/O
59	I/O
60	I/O

CQFP Pin Number	XC-3090
61	I/O
62	M1-RDATA
63	GND
64	M0-RTRIG
65	VCC
66	M2-I/O
67	HDC-I/O
68	I/O
69	I/O
70	I/O
71	LDC-I/O
72	I/O
73	I/O
74	I/O
75	I/O
76	I/O
77	I/O
78	I/O
79	I/O
80	I/O
81	INIT-I/O
82	VCC
83	GND
84	I/O
85	I/O
86	I/O
87	I/O
88	I/O
89	I/O
90	I/O
91	I/O
92	I/O
93	I/O
94	I/O
95	I/O
96	I/O
97	I/O
98	I/O
99	XTAL2(IN)-I/O
100	GND
101	RESET
102	VCC

CQFP Pin Number	XC-3090
103	DONE-PG
104	D7-I/O
105	XTAL1(OUT)-BCLKIN-I/O
106	I/O
107	I/O
108	I/O
109	D6-I/O
110	I/O
111	I/O
112	I/O
113	I/O
114	I/O
115	D5-I/O
116	CS0-I/O
117	I/O
118	I/O
119	I/O
120	I/O
121	D4-I/O
122	I/O
123	VCC
124	GND
125	D3-I/O
126	CS1-I/O
127	I/O
128	I/O
129	I/O
130	I/O
131	D2-I/O
132	I/O
133	I/O
134	I/O
135	I/O
136	I/O
137	D1-I/O
138	RDY/BUSY-RCLK-I/O
139	I/O
140	I/O
141	I/O
142	I/O

CQFP Pin Number	XC-3090
143	D0-DIN-I/O
144	DOUT-I/O
145	CCLK
146	VCC
147	GND
148	A0-WS-I/O
149	A1-CS2-I/O
150	I/O
151	I/O
152	A2-I/O
153	A3-I/O
154	I/O
155	I/O
156	A15-I/O
157	A4-I/O
158	I/O
159	I/O
160	A14-I/O
161	A5-I/O
162	I/O
163	I/O
164	GND
1	VCC
2	A13-I/O
3	A6-I/O
4	I/O
5	I/O
6	I/O
7	I/O
8	A12-I/O
9	A7-I/O
10	I/O
11	I/O
12	A11-I/O
13	A8-I/O
14	I/O
15	I/O
16	A10-I/O
17	A9-I/O
18	VCC
19	GND

Unprogrammed IOBs have a default pull-up.
This Prevents an undefined pad level for unbonded or unused IOBs.
Programmed outputs are default slew-rate limited.

XC3000 Family 175-Pin Ceramic and Plastic PGA Pinouts

PGA Pin Number	XC-3090
B2	PWRDN
D4	TCLKIN-I/O
B3	I/O
C4	I/O
B4	I/O
A4	I/O
D5	I/O
C5	I/O
B5	I/O
A5	I/O
C6	I/O
D6	I/O
B6	I/O
A6	I/O
B7	I/O
C7	I/O
D7	I/O
A7	I/O
A8	I/O
B8	I/O
C8	I/O
D8	**GND**
D9	**VCC**
C9	I/O
B9	I/O
A9	I/O
A10	I/O
D10	I/O
C10	I/O
B10	I/O
A11	I/O
B11	I/O
D11	I/O
C11	I/O
A12	I/O
B12	I/O
C12	I/O
D12	I/O
A13	I/O
B13	I/O
C13	I/O
A14	I/O

PGA Pin Number	XC-3090
D13	I/O
B14	M1-RDATA
C14	**GND**
B15	M0-RTRIG
D14	**VCC**
C15	M2-I/O
E14	HDC-I/O
B16	I/O
D15	I/O
C16	I/O
D16	LDC-I/O
F14	I/O
E15	I/O
E16	I/O
F15	I/O
F16	I/O
G14	I/O
G15	I/O
G16	I/O
H16	I/O
H15	INIT-I/O
H14	**VCC**
J14	**GND**
J15	I/O
J16	I/O
K16	I/O
K15	I/O
K14	I/O
L16	I/O
L15	I/O
M16	I/O
M15	I/O
L14	I/O
N16	I/O
P16	I/O
N15	I/O
R16	I/O
M14	I/O
P15	XTAL2(IN)-I/O
N14	**GND**
R15	RESET
P14	**VCC**

PGA Pin Number	XC-3090
R14	DONE-PG
N13	D7-I/O
T14	XTAL1(OUT)-BCLKIN-I/O
P13	I/O
R13	I/O
T13	I/O
N12	I/O
P12	D6-I/O
R12	I/O
T12	I/O
P11	I/O
N11	I/O
R11	I/O
T11	D5-I/O
R10	CS0-I/O
P10	I/O
N10	I/O
T10	I/O
T9	I/O
R9	D4-I/O
P9	I/O
N9	**VCC**
N8	**GND**
P8	D3-I/O
R8	CS1-I/O
T8	I/O
T7	I/O
N7	I/O
P7	I/O
R7	D2-I/O
T6	I/O
R6	I/O
N6	I/O
P6	I/O
T5	I/O
R5	D1-I/O
P5	RDY/BUSY-RCLK-I/O
N5	I/O
T4	I/O
R4	I/O
P4	I/O

PGA Pin Number	XC-3090
R3	D0-DIN-I/O
N4	DOUT-I/O
R2	CCLK
P3	**VCC**
N3	**GND**
P2	A0-WS-I/O
M3	A1-CS2-I/O
R1	I/O
N2	I/O
P1	A2-I/O
N1	A3-I/O
L3	I/O
M2	I/O
M1	A15-I/O
L2	A4-I/O
L1	I/O
K3	I/O
K2	A14-I/O
K1	A5-I/O
J1	I/O
J2	I/O
J3	**GND**
H3	**VCC**
H2	A13-I/O
H1	A6-I/O
G1	I/O
G2	I/O
G3	I/O
F1	I/O
F2	A12-I/O
E1	A7-I/O
E2	I/O
F3	I/O
D1	A11-I/O
C1	A8-I/O
D2	I/O
B1	I/O
E3	A10-I/O
C2	A9-I/O
D3	**VCC**
C3	**GND**

Unprogrammed IOBs have a default pull-up. This prevents an undefined pad level for unbonded or unused IOBs. Programmed outputs are default slew-rate limited.

Pins A2, A3, A15, A16, T1, T2, T3, T15 and T16 are not connected.
Pin A1 does not exist.

Xilinx maintains test specifications for each product as controlled documents. To insure the use of the most recently released device performance parameters, please request a copy of the current test-specification revision.

ABSOLUTE MAXIMUM RATINGS

Symbol	Description		Units
V_{CC}	Supply voltage relative to GND	–0.5 to +7.0	V
V_{IN}	Input voltage with respect to GND	–0.5 to Vcc +0.5	V
V_{TS}	Voltage applied to 3-state output	–0.5 to Vcc +0.5	V
T_{STG}	Storage temperature (ambient)	–65 to +150	°C
T_{SOL}	Maximum soldering temperature (10 s @ 1/16 in.)	+260	°C
T_J	Junction temperature plastic	+125	°C
	Junction temperature ceramic	+150	°C

Note: Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those listed under Recommended Operating Conditions is not implied. Exposure to Absolute Maximum Ratings conditions for extended periods of time may affect device reliability.

OPERATING CONDITIONS

Symbol	Description	Min	Max	Units
V_{CC}	Supply voltage relative to GND Commercial 0°C to +70°C	4.75	5.25	V
	Supply voltage relative to GND Industrial –40°C to +85°C	4.5	5.5	V
	Supply voltage relative to GND Military –55°C to +125°C	4.5	5.5	V
V_{IHT}	High-level input voltage — TTL configuration	2.0	Vcc	V
V_{ILT}	Low-level input voltage — TTL configuration	0	0.8	V
V_{IHC}	High-level input voltage — CMOS configuration	70%	100%	V_{CC}
V_{ILC}	Low-level input voltage — CMOS configuration	0	20%	V_{CC}
T_{IN}	Input signal transition time		250	ns

DC CHARACTERISTICS OVER OPERATING CONDITIONS

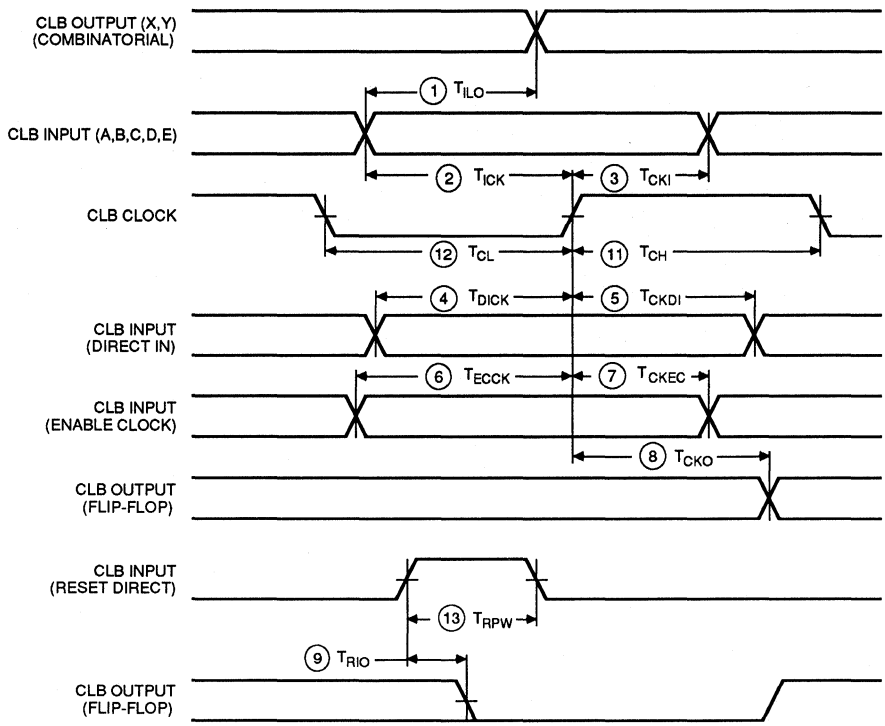
Symbol	Description		Min	Max	Units
V_{OH}	High-level output voltage (@ I_{OH} = –4.0 mA, V_{CC} min)	Commercial	3.86		V
V_{OL}	Low-level output voltage (@ I_{OL} = 4.0 mA, V_{CC} max)			0.32	V
V_{OH}	High-level output voltage (@ I_{OH} = –4.0 mA, V_{CC} min)	Industrial Military	3.76		V
V_{OL}	Low-level output voltage (@ I_{OL} = 4.0 mA, V_{CC} max)			0.37	V
V_{CCPD}	Power-down supply voltage ($\overline{PWRDWN}$ must be Low)		2.3		V
I_{CCPD}	Power-down supply current ($V_{CC\ (MAX)}$ @ T_{MAX})[1]	XC3020		50	µA
		XC3030		80	µA
		XC3042		120	µA
		XC3064		170	µA
		XC3090		250	µA
I_{CCO}	Quiescent LCA supply current in addition to I_{CCPD}[2]				
	Chip thresholds programmed as CMOS levels			500	µA
	Chip thresholds programmed as TTL levels			10	mA
I_{IL}	Input Leakage Current		–10	+10	µA
C_{IN}	Input capacitance, all packages except PGA 175 (sample tested)				
	All Pins except XTL1 and XTL2			10	pF
	XTL1 and XTL2			15	pF
	Input capacitance, PGA 175 (sample tested)				
	All Pins except XTL1 and XTL2			15	pF
	XTL1 and XTL2			20	pF
I_{RIN}	Pad pull-up (when selected) @ V_{IN} = 0V (sample tested)		0.02	0.17	mA
I_{RLL}	Horizontal long line pull-up (when selected) @ logic Low		0.2	2.5	mA

Note: 1. Devices with much lower I_{CCPD} tested and guaranteed at Vcc = 3.2 V, T = 25°C can be ordered with a Special Product Code.

XC3020: SPC0110 I_{CCPD} = 1 µA
XC3030: SPC0104 I_{CCPD} = 2 µA
XC3042: SPC0107 I_{CCPD} = 3 µA
XC3064: SPC0108 I_{CCPD} = 4 µA
XC3090: SPC0109 I_{CCPD} = 5 µA

2. With no output current loads, no active input or long line pull-up resistors, all package pins at Vcc or GND, and the LCA configured with a MAKEBITS "tie" option. See LCA power chart for additional activity-dependent operating component.

CLB SWITCHING CHARACTERISTIC GUIDELINES



1105 26

BUFFER (Internal) SWITCHING CHARACTERISTIC GUIDELINES

Description	Symbol	-70 Max	-100 Max	-125 Max	Units
Global and Alternate Clock Distribution					
Either: **Normal** IOB input pad to clock buffer input	T_{PID}	6	4	3	ns
Or: **Fast** (CMOS only) input pad to clock buffer input	T_{PIDC}	3	2	2	ns
Plus: Clock buffer input to any CLB/IOB clock					
on XC3020		3.2/4.5	2.9/3.9	2.8/3.5	ns
on XC3030		3.4/5.1	3.1/4.3	2.9/3.9	ns
on XC3042		3.7/5.7	3.3/4.9	3.1/4.4	ns
on XC3064		4.1/6.6	3.6/5.5	3.4/5.0	ns
on XC3090		4.6/7.9	4.0/6.4	3.8/5.8	ns
TBUF driving a Horizontal Long line (L.L.)**					
I to L.L. while T is Low (buffer active)	T_{ID}	5	4	3	ns
T↓ to L.L. active and valid	T_{ON}	9	7	6	ns
T↑ to L.L. (inactive) with single pull-up resistor	T_{PUS}	22	14	13	ns
with pair of pull-up resistors	T_{PUF}	11	7	6.5	ns
BIDI					
Bidirectional buffer delay	T_{BIDI}	4	3	2.5	ns

** Timing is based on the XC3020, for other devices see XACT timing calculator.

CLB SWITCHING CHARACTERISTIC GUIDELINES (Continued)

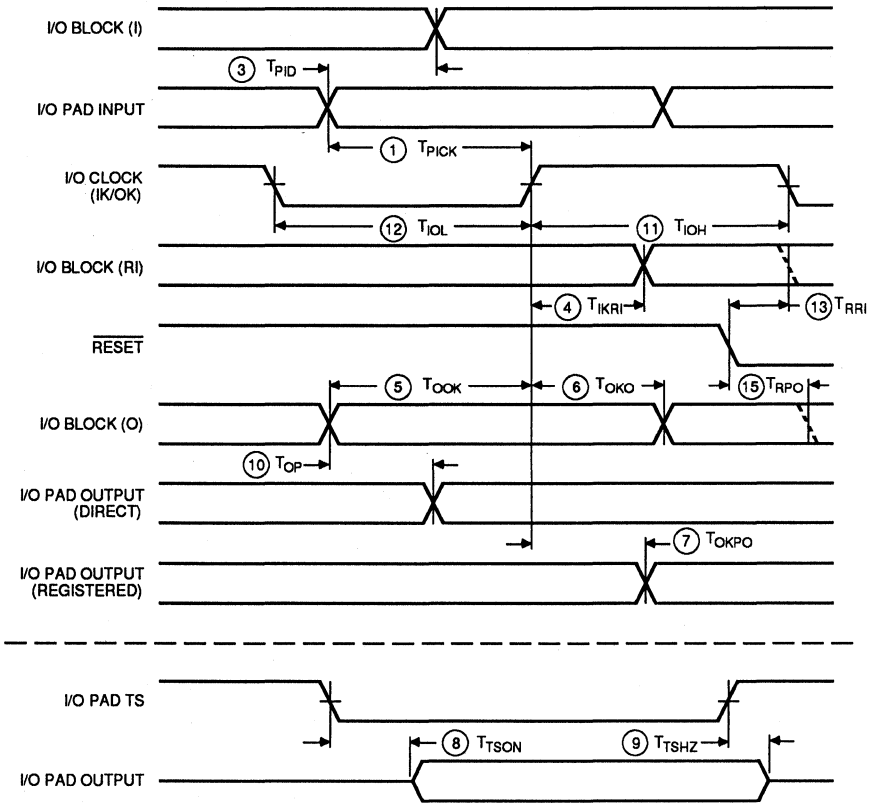
Testing of the switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% functionally tested. Since many internal timing parameters cannot be measured directly, they are derived from benchmark timing patterns. The following guidelines reflect worst-case values over the recommended operating conditions. For more detailed, more precise, and more up-to-date timing information, use the values provided by the XACT timing calculator and used in the simulator.

	Speed Grade		-70		-100		-125		Units
Description	Symbol		Min	Max	Min	Max	Min	Max	
Combinatorial Delay									
Logic Variables a, b, c, d, e, to outputs x, y	1	TILO		9		7		5.5	ns
Sequential delay									
Clock k to outputs x, y	8	TCKO		8		7		6	ns
Clock k to outputs x,y when Q is returned through function generators F or G to drive x, y				15		12		10	ns
Set-up time before clock K									
Logic Variables a, b, c, d, e	2	TICK	8		7		6		ns
Data In di	4	TDICK	5		4		3		ns
Enable Clock ec	6	TECCK	7		5		4.5		ns
Reset Direct inactive rd			1		1		1		ns
Hold Time after clock k									
Logic Variables a, b, c, d, e	3	TCKI	0		0		0		ns
Data In di	5	TCKDI	4		2		1.5		ns
Enable Clock ec	7	TCKEC	0		0		0		ns
Clock									
Clock High time*	11	TCH	7		5		4		ns
Clock Low time*	12	TCL	7		5		4		ns
Max. flip-flop toggle rate*		FCLK	70		100		125		MHz
Reset Direct (rd)									
rd width	13	TRPW	8		7		6		ns
delay from rd to outputs x, y	9	TRIO		8		7		6	ns
Global Reset ($\overline{\text{RESET}}$ Pad)									
$\overline{\text{RESET}}$ width (Low)		TMRW	25		21		20		ns
delay from $\overline{\text{RESET}}$ pad to outputs x, y		TMRQ		20		17		16	ns

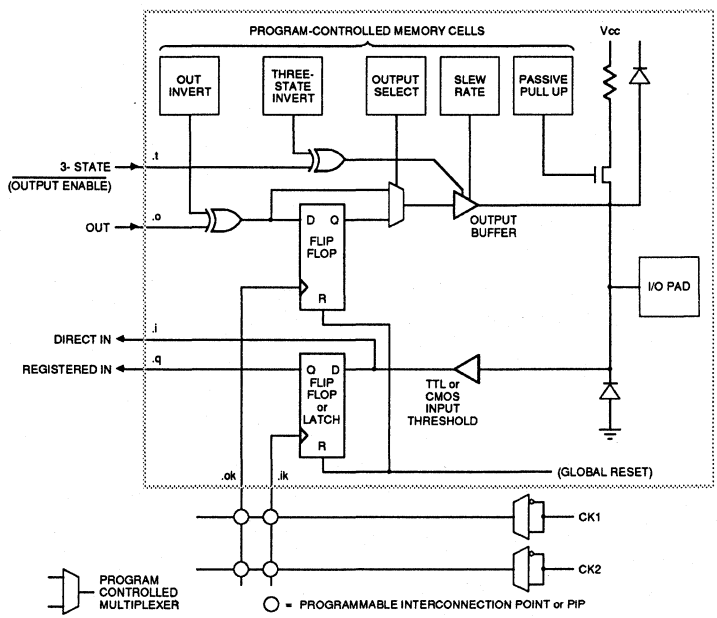
* These timing limits are based on calculations.

Note: The CLB K to Q output delay (TCKO, #8) of any CLB, plus the shortest possible interconnect delay, is always longer than the Data In hold time requirement (TCKDI, #5) of any CLB on the same die.

IOB SWITCHING CHARACTERISTIC GUIDELINES



1105 27C



1105 01A

IOB SWITCHING CHARACTERISTIC GUIDELINES (Continued)

Testing of the switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% functionally tested. Since many internal timing parameters cannot be measured directly, they are derived from benchmark timing patterns. The following guidelines reflect worst-case values over the recommended operating conditions. For more detailed, more precise, and more up-to-date timing information, use the values provided by the XACT timing calculator and used in the simulator.

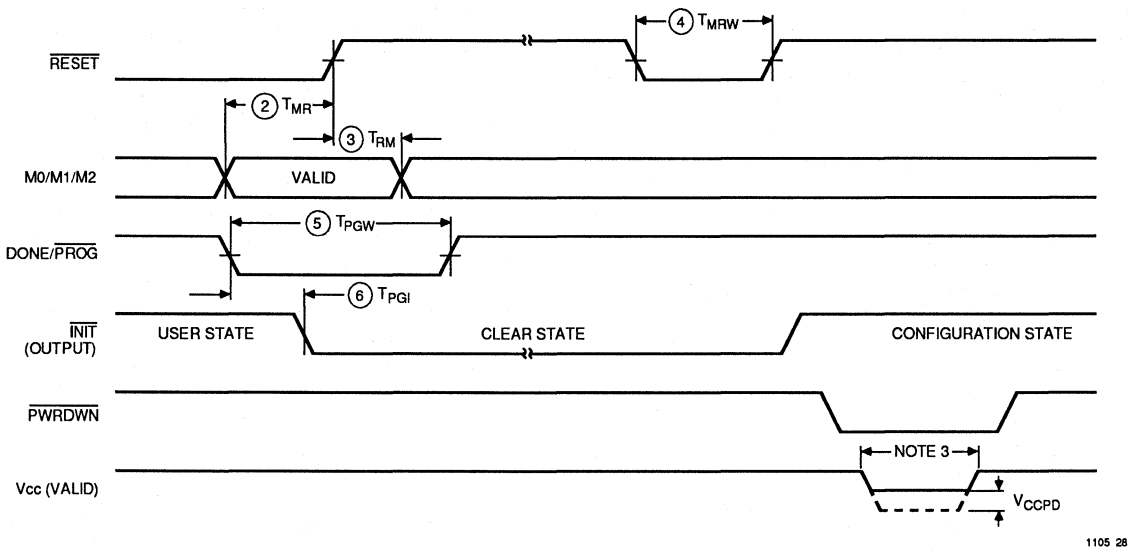
			-70		-100		-125		Units
Description	Symbol		Min	Max	Min	Max	Min	Max	
Propagation Delays (Input)									
Pad to Direct In (i)	3	T_{PID}		6		4		3	ns
Pad to Registered In (q) with latch transparent		T_{PTG}		21		17		16	ns
Clock (ik) to Registered In (q)	4	T_{IKRI}		7		6		5	ns
Set-up Time (Input)									
Pad to Clock (ik) set-up time	1	T_{PICK}	20		17		16		ns
Propagation Delays (Output)									
Clock (ok) to Pad (fast)	7	T_{OKPO}		13		10		9	ns
same (slew rate limited)	7	T_{OKPO}		33		27		24	ns
Output (o) to Pad (fast)	10	T_{OPF}		9		6		5	ns
same (slew-rate limited)	10	T_{OPS}		29		23		20	ns
3-state to Pad begin hi-Z (fast)	9	T_{TSHZ}		8		8		7	ns
same (slew-rate limited)	9	T_{TSHZ}		28		25		24	ns
3-state to Pad active and valid (fast)	8	T_{TSON}		14		12		11	ns
same (slew -rate limited)	8	T_{TSON}		34		29		27	ns
Set-up and Hold Times (Output)									
Output (o) to clock (ok) set-up time	5	T_{OOK}	10		9		8		ns
Output (o) to clock (ok) hold time	6	T_{OKO}	0		0		0		ns
Clock									
Clock High time	11	T_{IOH}	7		5		4		ns
Clock Low time	12	T_{IOL}	7		5		4		ns
Max. flip-flop toggle rate		F_{CLK}	70		100		125		MHz
Global Reset Delays									
$\overline{\text{RESET}}$ Pad to Registered In (q)	13	T_{RRI}		23		20		19	ns
$\overline{\text{RESET}}$ Pad to output pad (fast)	15	T_{RPO}		33		28		26	ns
(slew-rate limited)	15	T_{RPO}		53		45		42	ns

Notes: 1. Timing is measured at pin threshold, with 50 pF external capacitive loads (incl. test fixture).
For larger capacitive loads, see page 6-9.
Typical slew rate limited output rise/fall times are approximately four times longer.
A maximum total external capacitive load for simultaneous fast mode switching in the same direction is 500 pF per power/ground pin pair. For slew-rate limited outputs this total is four times larger.

2. Voltage levels of unused (bonded and unbonded) pads must be valid logic levels. Each can be configured with the internal pull-up resistor or alternatively configured as a driven output or driven from an external source.

3. Input pad set-up time is specified with respect to the internal clock (.ik)
In order to calculate system set-up time, subtract clock delay (pad to ik) from the input pad set-up time value.
Input pad holdtime with respect to the internal clock (ik) is negative. This means that pad level changes immediately before the internal clock edge (ik) will not be recognized.
For a more detailed description see the discussion on "LCA Performance" in the Applications Section.

GENERAL LCA SWITCHING CHARACTERISTICS



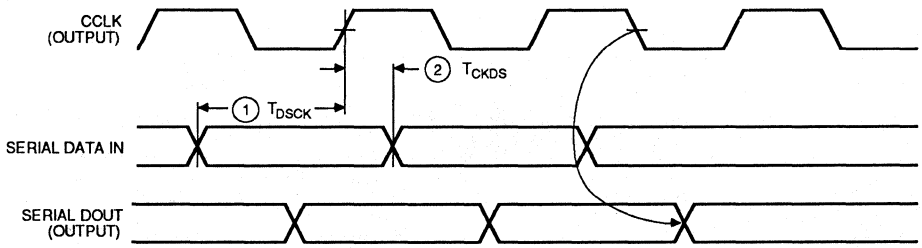
				-70		-100		-125		Units
	Description	Symbol		Min	Max	Min	Max	Min	Max	
RESET (2)	M0, M1, M2 setup time required	2	T_{MR}	1	0	1	0	0	0	µs
	M0, M1, M2 hold time required	3	T_{RM}	1		1		1		µs
	RESET Width (Low) req. for Abort	4	T_{MRW}	6		6		6		µs
DONE/PROG	Width (Low) required for Re-config.	5	T_{PGW}	6		6		6		µs
	INIT response after D/P is pulled Low	6	T_{PGI}		7		7		7	µs
PWRDWN (3)	Power Down Vcc		V_{CCPD}	2.3		2.3		2.3		V

Notes: 1. At power-up, Vcc must rise from 2.0 V to Vcc min in less than 25 ms. If this is not possible, configuration can be delayed by holding RESET Low until Vcc has reached 4.0 V. A very long Vcc rise time of >100 ms, or a non-monotonically rising Vcc may require a RESET pulse (High-to-Low-to-High) of >6 µs duration after Vcc has reached 4.0 V.

2. RESET timing relative to valid mode lines (M0, M1, M2) is relevant when RESET is used to delay configuration.

3. PWRDWN transitions must occur while Vcc >4.0 V.

MASTER SERIAL MODE PROGRAMMING SWITCHING CHARACTERISTICS



1105 29

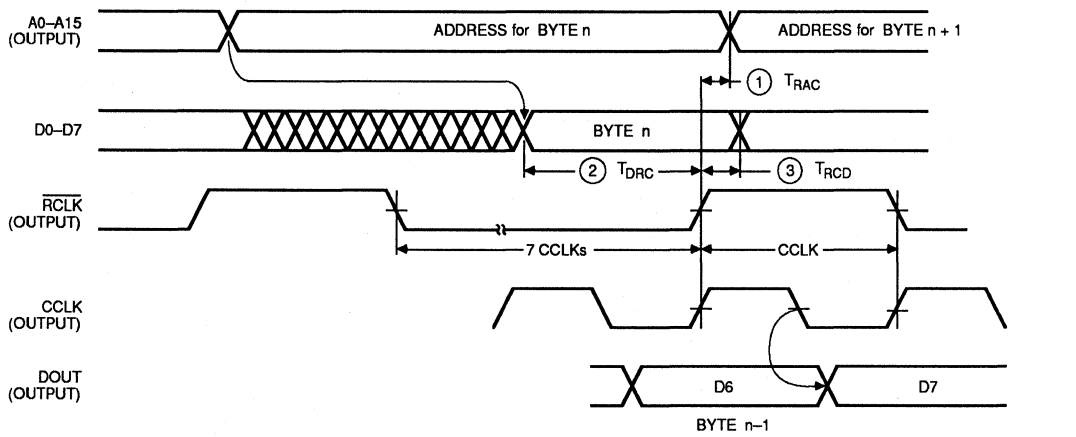
Speed Grade				-70		-100		-125		Units
	Description	Symbol		Min	Max	Min	Max	Min	Max	
CCLK[3]	Data In setup	1	T_{DSCK}	60		60		60		ns
	Data In hold	2	T_{CKDS}	0		0		0		ns

Notes: 1. At power-up, Vcc must rise from 2.0 V to Vcc min in less than 25 ms. If this is not possible, configuration can be delayed by holding $\overline{\text{RESET}}$ Low until Vcc has reached 4.0 V. A very long Vcc rise time of >100 ms, or a non-monotonically rising Vcc may require a $\overline{\text{RESET}}$ pulse (High-to-Low-to-High) of >6 µs duration after Vcc has reached 4.0 V.

2. Configuration can be controled by holding $\overline{\text{RESET}}$ Low with or until after the $\overline{\text{INIT}}$ of all daisy-chain slave-mode devices is High.

3. Master-serial-mode timing is based on slave-mode testing.

MASTER PARALLEL MODE PROGRAMMING SWITCHING CHARACTERISTICS



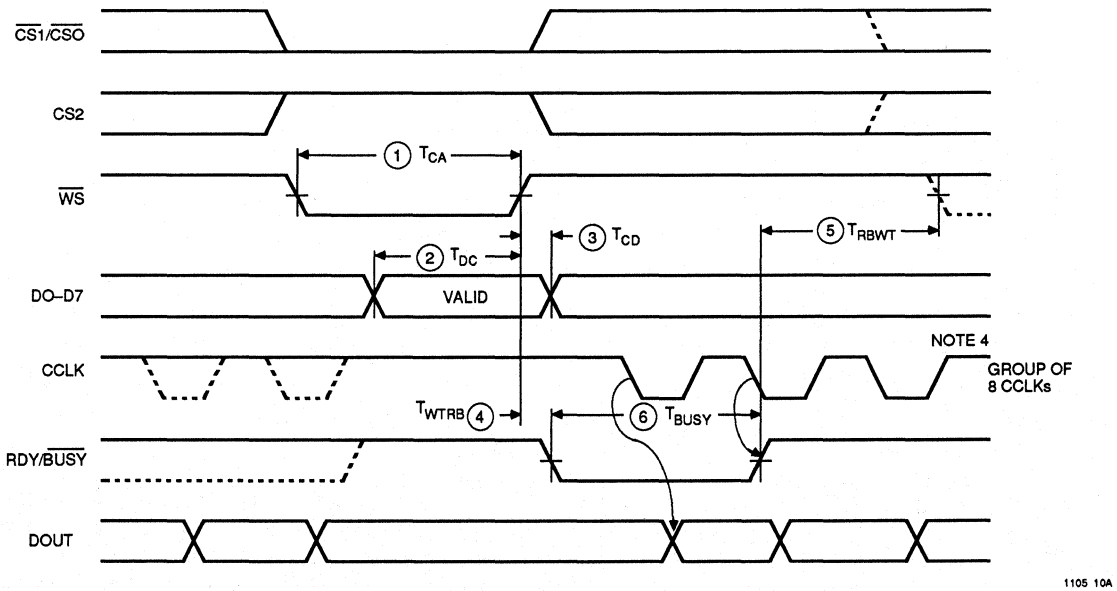
				-70		-100		-125		Units
	Description	Symbol		Min	Max	Min	Max	Min	Max	
RCLK	To address valid	1	T_{RAC}	0	200	0	200	0	200	ns
	To data setup	2	T_{DRC}	60		60		60		ns
	To data hold	3	T_{RCD}	0		0		0		ns
	RCLK high		T_{RCH}	600		600		600		ns
	RCLK low		T_{RCL}	4.0		4.0		4.0		µs

Notes: 1. At power-up, Vcc must rise from 2.0 V to Vcc min in less than 25 ms. If this is not possible, configuration can be delayed by holding $\overline{\text{RESET}}$ Low until Vcc has reached 4.0 V. A very long Vcc rise time of >100 ms, or a non-monotonically rising Vcc may require a $\overline{\text{RESET}}$ pulse (High-to-Low-to-High) of >6 µs duration after Vcc has reached 4.0 V.

2. Configuration can be controlled by holding $\overline{\text{RESET}}$ Low with or until after the $\overline{\text{INIT}}$ of all daisy-chain slave-mode devices is High.

This timing diagram shows that the EPROM requirements are extremely relaxed:
EPROM access time can be longer than 4000 ns. EPROM data output has no hold time requirements.

PERIPHERAL MODE PROGRAMMING SWITCHING CHARACTERISTICS

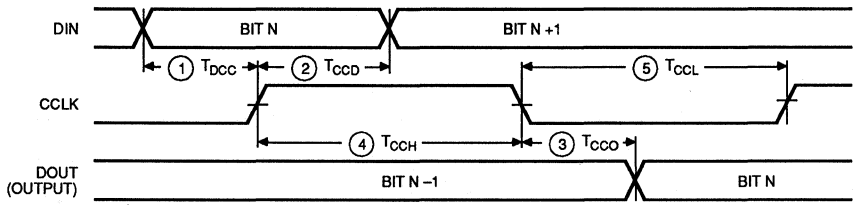


				-70		-100		-125		Units
	Description	Symbol		Min	Max	Min	Max	Min	Max	
Write	Effective Write time required ($\overline{CS0} \cdot \overline{CS1} \cdot CS2 \cdot \overline{WS}$)	1	T_{CA}	100		100		100		ns
	DIN Setup time required	2	T_{DC}	60		60		60		ns
	DIN Hold time required	3	T_{CD}	0		0		0		ns
	RDY/$\overline{BUSY}$ delay after end of $\overline{WS}$	4	T_{WTRB}		60		60		60	ns
RDY	Earliest next $\overline{WS}$ after end of $\overline{BUSY}$	5	T_{RBWT}	0		0		0		ns
	$\overline{BUSY}$ Low time generated	6	T_{BUSY}	2	9	2	9	2	9	CCLK Periods

Notes:
1. At power-up, Vcc must rise from 2.0 V to Vcc min in less than 25 ms. If this is not possible, configuration can be delayed by holding $\overline{RESET}$ Low until Vcc has reached 4.0 V. A very long Vcc rise time of >100 ms, or a non-monotonically rising Vcc may require a $\overline{RESET}$ pulse (High-to-Low-to-High) of >6 µs duration after Vcc has reached 4.0 V.
2. Configuration must be delayed until the $\overline{INIT}$ of all LCAs is High.
3. Time from end of $\overline{WS}$ 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.
4. CCLK and DOUT timing is tested in slave mode.

This timing diagram shows very relaxed requirements:
Data need not be held beyond the rising edge of $\overline{WS}$. $\overline{BUSY}$ will go active within 60 ns after the end of $\overline{WS}$. $\overline{BUSY}$ will stay active for several microseconds. $\overline{WS}$ may be asserted immediately after the end of $\overline{BUSY}$.

SLAVE MODE PROGRAMMING SWITCHING CHARACTERISTICS



1105 31

				-70		-100		-125		
	Description	Symbol		Min	Max	Min	Max	Min	Max	Units
CCLK	To DOUT	3	T_{CCO}		100		100		100	ns
	DIN setup	1	T_{DCC}	60		60		60		ns
	DIN hold	2	T_{CCD}	0		0		0		ns
	High time	4	T_{CCH}	0.5		0.5		0.5		µs
	Low time (Note 1)	5	T_{CCL}	0.5	5.0	0.5	5.0	0.5	5.0	µs
	Frequency		F_{CC}		1		1		1	MHz

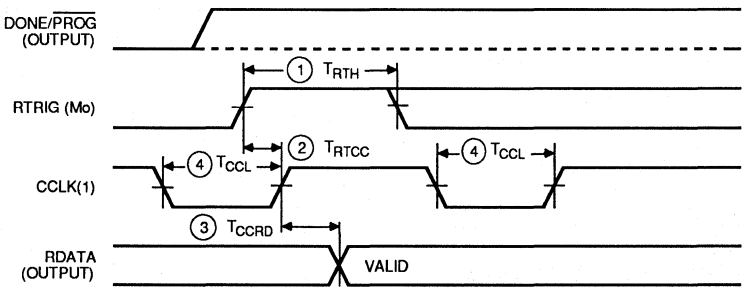
Notes: 1. The max limit of CCLK Low time is caused by dynamic circuitry inside the LCA device.

2. Configurationmust be delayed until the $\overline{INIT}$ of all LCAs is High.

3. At power-up, Vcc must rise from 2.0 V to Vcc min in less than 25 ms. If this is not possible, configuration can be delayed by holding $\overline{RESET}$ Low until Vcc has reached 4.0 V. A very long Vcc rise time of >100 ms, or a non-monotonically rising Vcc may require a $\overline{RESET}$ pulse (High-to-Low-to-High) of >6 µs duration after Vcc has reached 4.0 V.

4. For configuration (not Readback), CCLK frequency can be increased to 5 MHz and TCCH and TCCL min reduced to 100 ns, worst case over temperature and supply voltage. This high-speed CCLK frequency will be tested, documented and guaranteed some time in 1991. For further information on running CCLK faster than 1 MHz, contact Xilinx Product Marketing.

PROGRAM READBACK SWITCHING CHARACTERISTICS



1105 32A

				-70		-100		-125		
	Description	Symbol		Min	Max	Min	Max	Min	Max	Units
RTRIG	RTRIG high	1	T_{RTH}	250		250		250		ns
CCLK	RTRIG setup	2	T_{RTCC}	10		10		10		ns
	RDATA delay	3	T_{CCRD}		100		100		100	ns
	Low time	4	T_{CCL}	0.5	5	0.5	5	0.5	5	µs

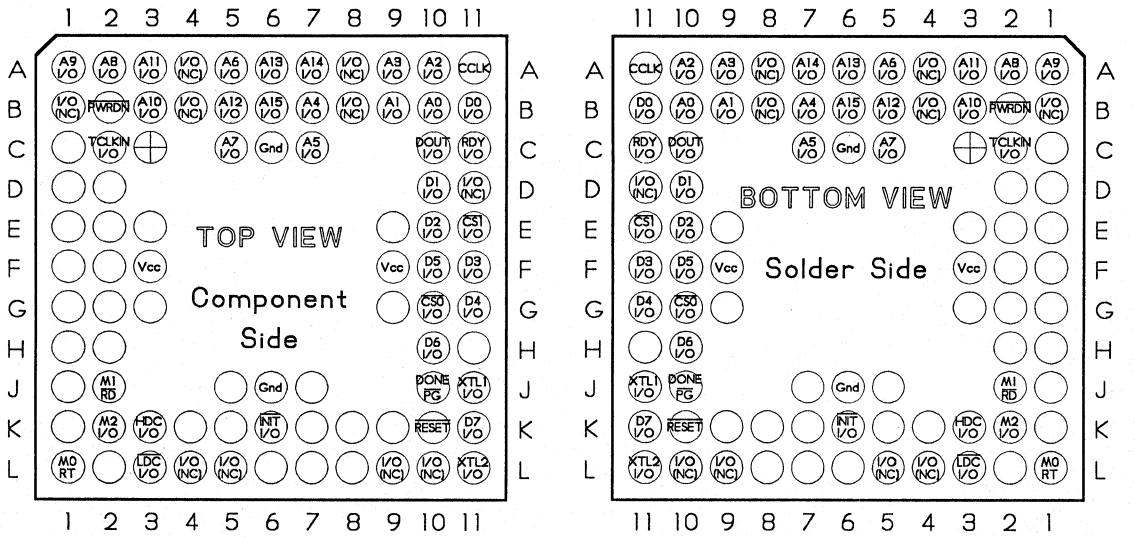
Notes: 1. At power-up, Vcc must rise from 2.0 V to Vcc min in less than 25 ms. If this is not possible, configuration can be delayed by holding $\overline{RESET}$ Low until Vcc has reached 4.0 V. A very long Vcc rise time of >100 ms, or a non-monotonically rising Vcc may require a $\overline{RESET}$ pulse (High-to-Low-to-High) of >6 µs duration after Vcc has reached 4.0 V.

2. CCLK and DOUT timing are the same as for slave mode.

3. RETRIG (M0 positive transition) shall not be done until after one clock following active I/O pins.

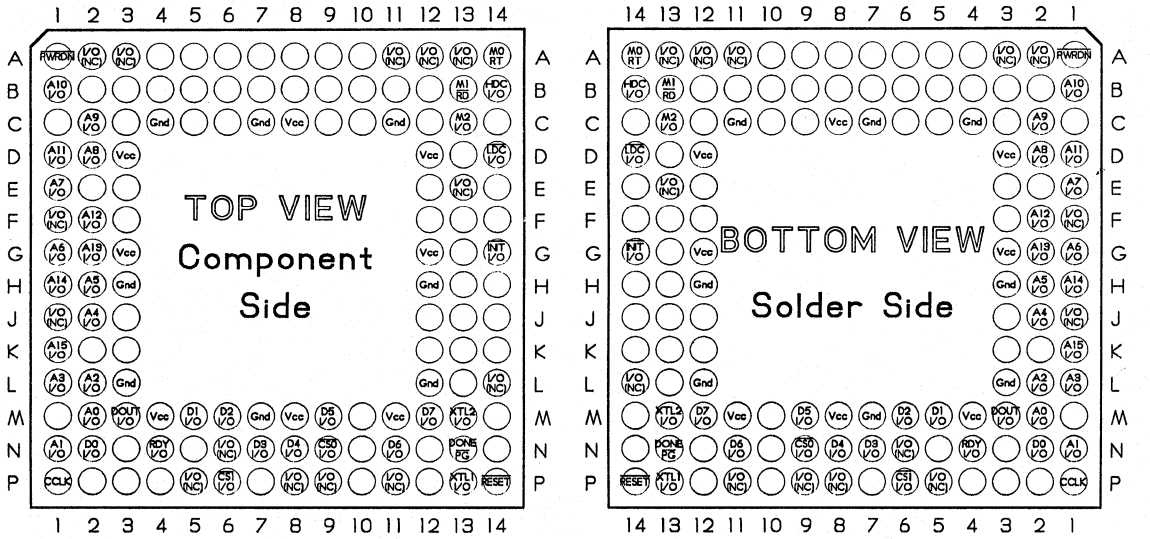
4. Readback should not be initiated until configuration is complete.

PGA PIN-OUTS



⊕ = Index pin which may or may not be electrically connected to pin C2
(NC) = Pin Not Connected for XC3020, unlabeled pin = unrestricted I/O pin

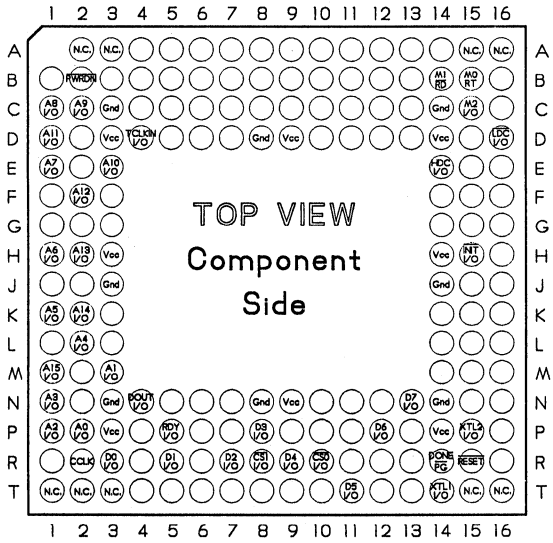
PG84 Pin-outs–XC3042, XC3030, XC3020

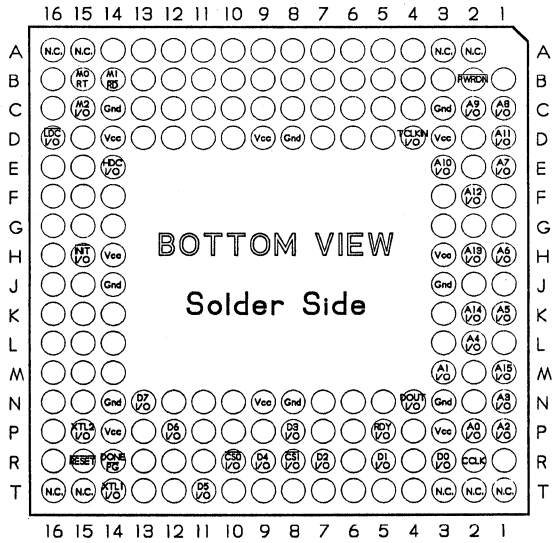


(NC) = Pin Not Connected for XC3042, unlabeled pin = unrestricted I/O pin

PG132 Pin-outs–XC3064, XC3042-PG, -PP

PGA PIN-OUTS (cont'd)

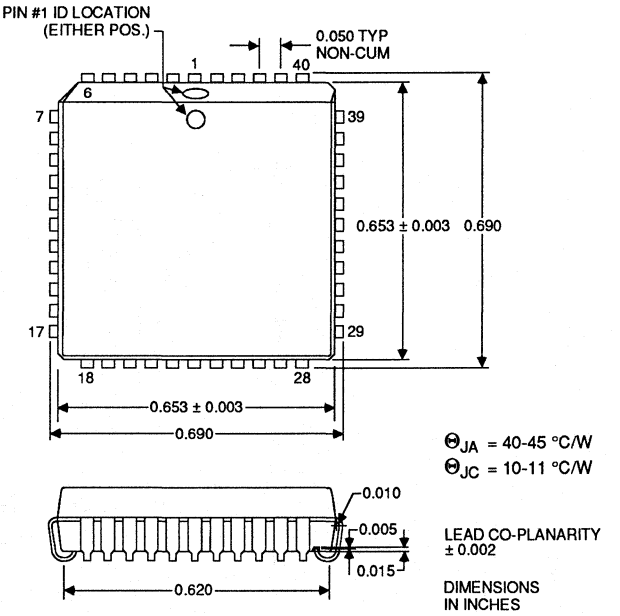




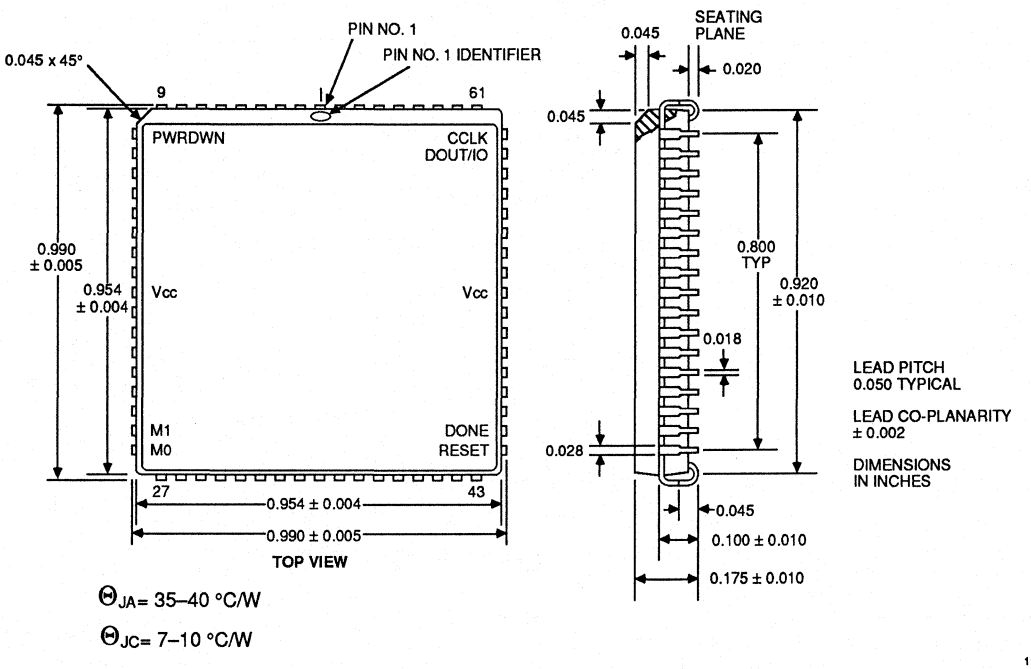
(NC) = Pin Not Connected , unlabeled pin = unrestricted I/O pin

PG175 Pin-outs–XC3090-PG, -PP

PHYSICAL DIMENSIONS

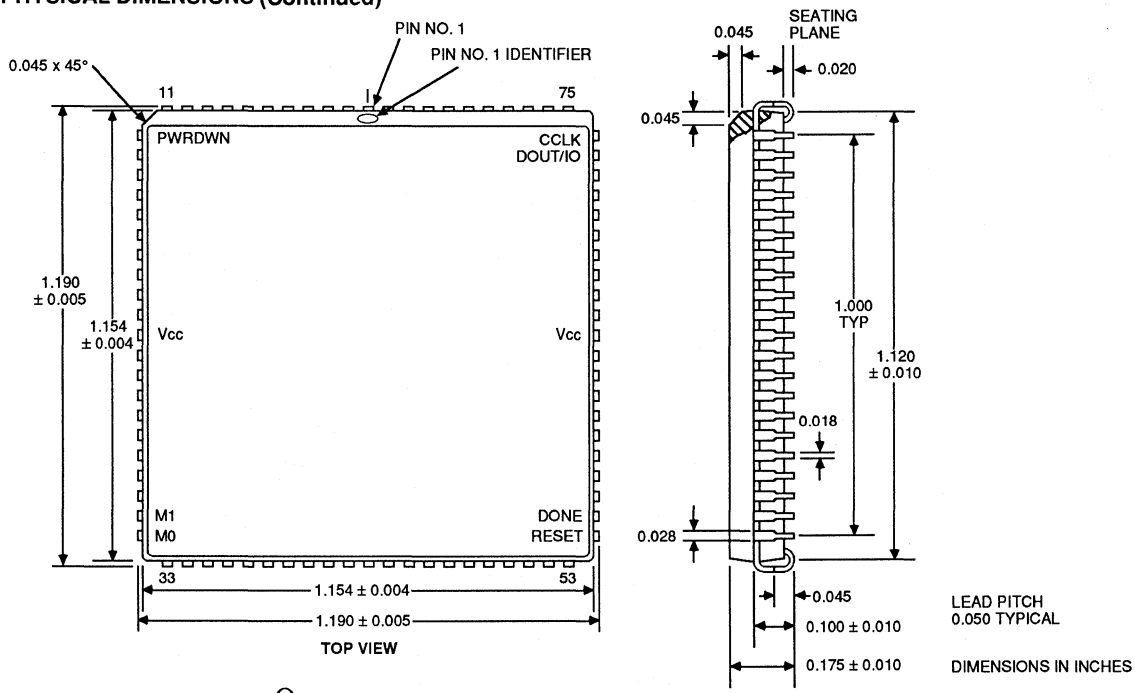


44-Pin PLCC Package



68-Pin PLCC Package

PHYSICAL DIMENSIONS (Continued)

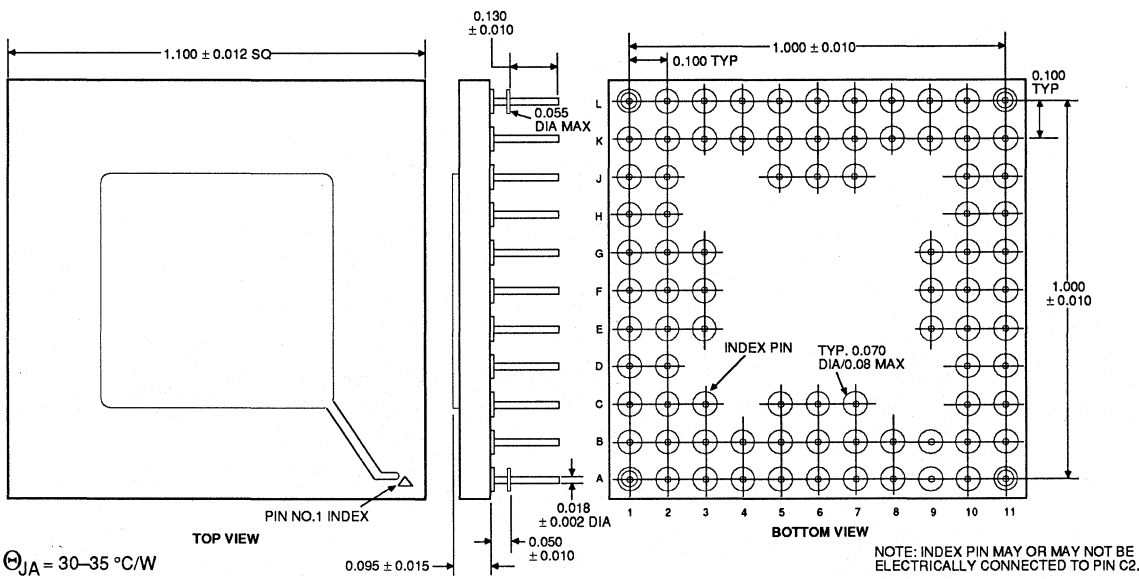


Θ_{JA}= 30–35 °C/W

Θ_{JC}= 3–7 °C/W

1105 36C

84-Pin PLCC Package



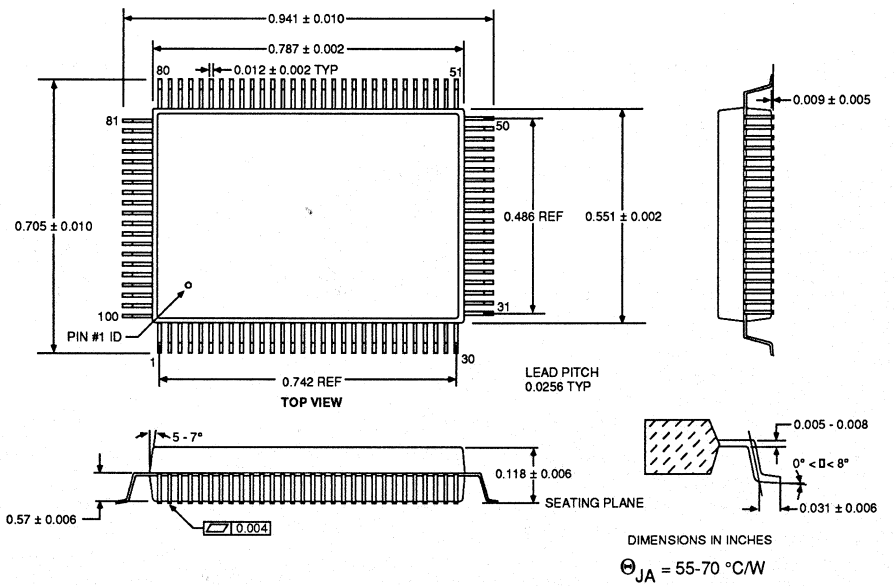
Θ_{JA} = 30–35 °C/W

Θ_{JC} = 4–7 °C/W

1105 35C

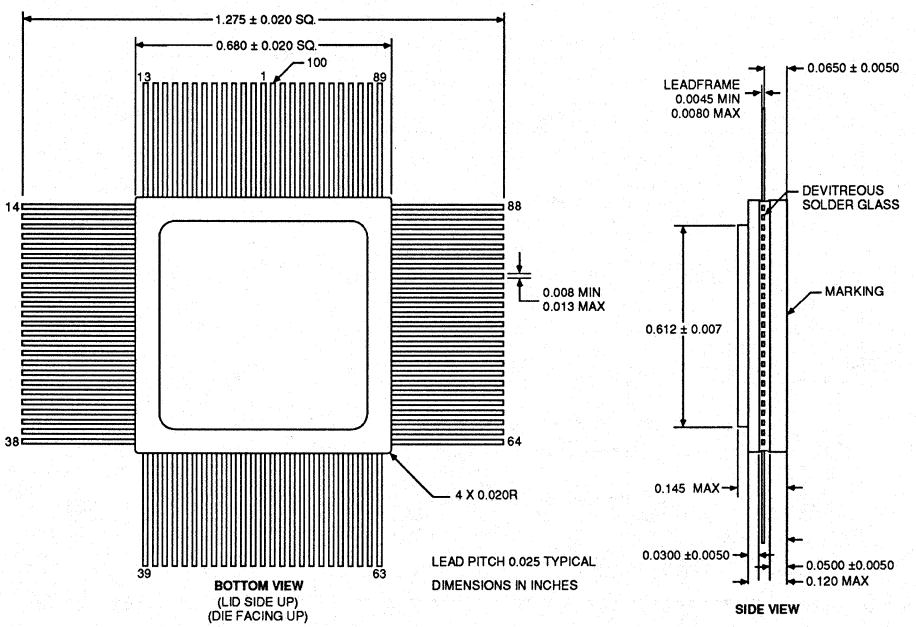
84-Pin PGA Package

PHYSICAL DIMENSIONS (Continued)



1105 39C

100-Pin PQFP Package



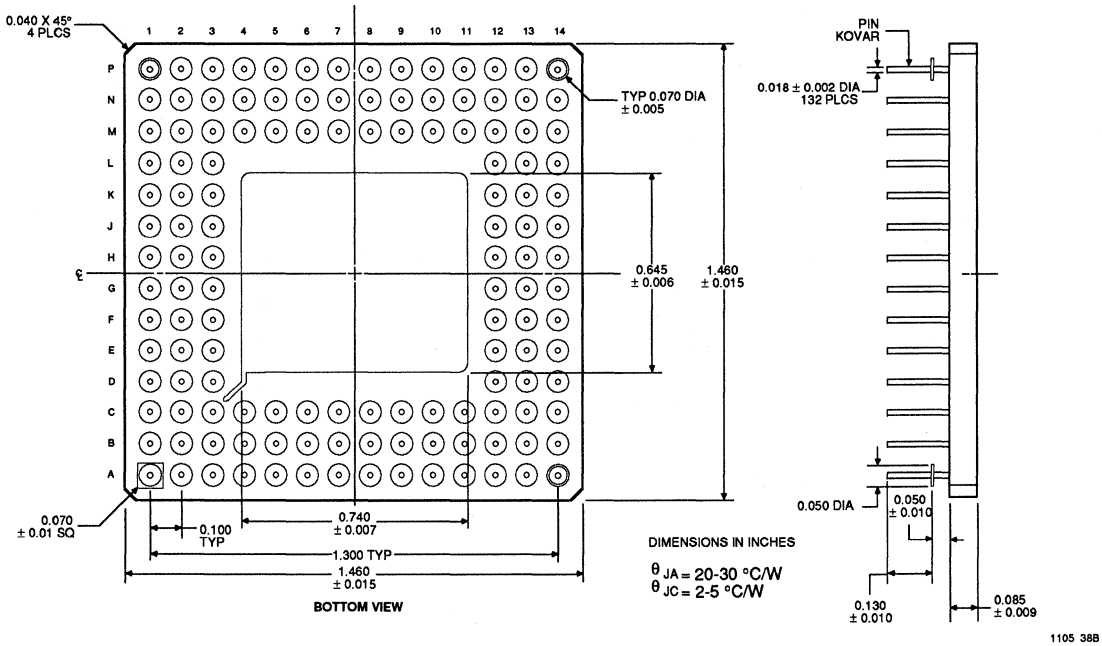
Θ_{JA} = 40-50° C/W
Θ_{JC} = 5-8° C/W

NOTES:
1. LEADS ARE SHIPPED UNFORMED IN CARRIERS IN TRAYS, TOPSIDE UP
2. FORMING TOOL INFORMATION:
 – FANCORT INDUSTRIES – (201) 575-0610 WEST CALDWELL NJ.
 – RISI INDUSTRIES (619) 425-3970 CHULA VISTA, CA.

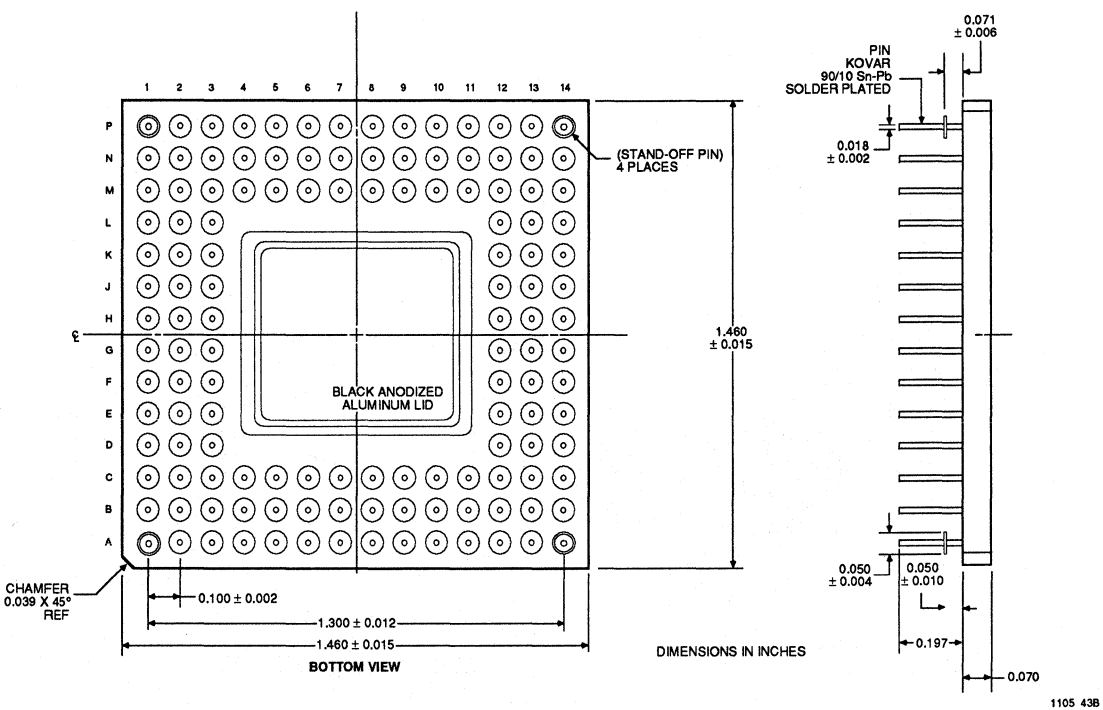
1105 40C

100-Pin CQFP Package

PHYSICAL DIMENSIONS (Continued)

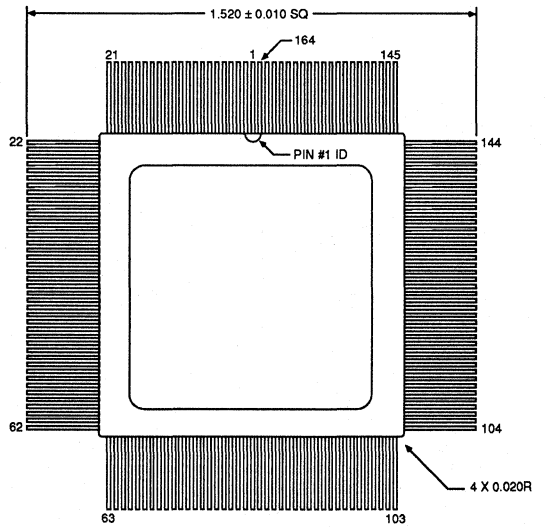


132-Pin PGA Package

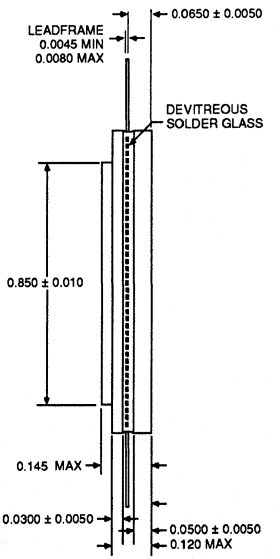


132-Pin PPGA Package

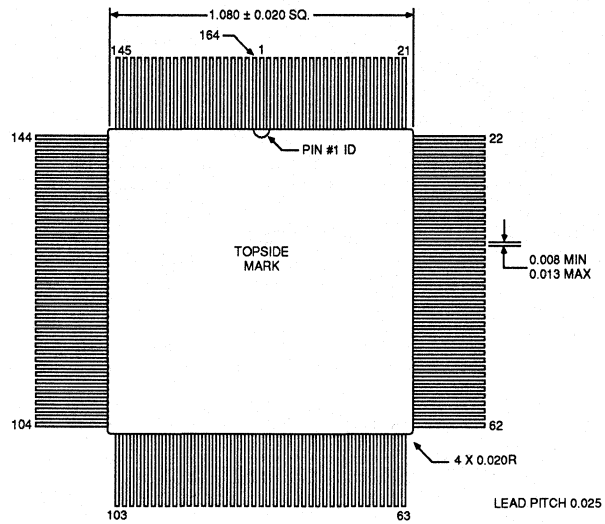
PHYSICAL DIMENSIONS (Continued)



BOTTOM VIEW
(LID SIDE UP)
(DIE FACING UP)



SIDE VIEW



TOP VIEW
(DIE FACING DOWN)

LEAD PITCH 0.025 TYPICAL

DIMENSIONS IN INCHES

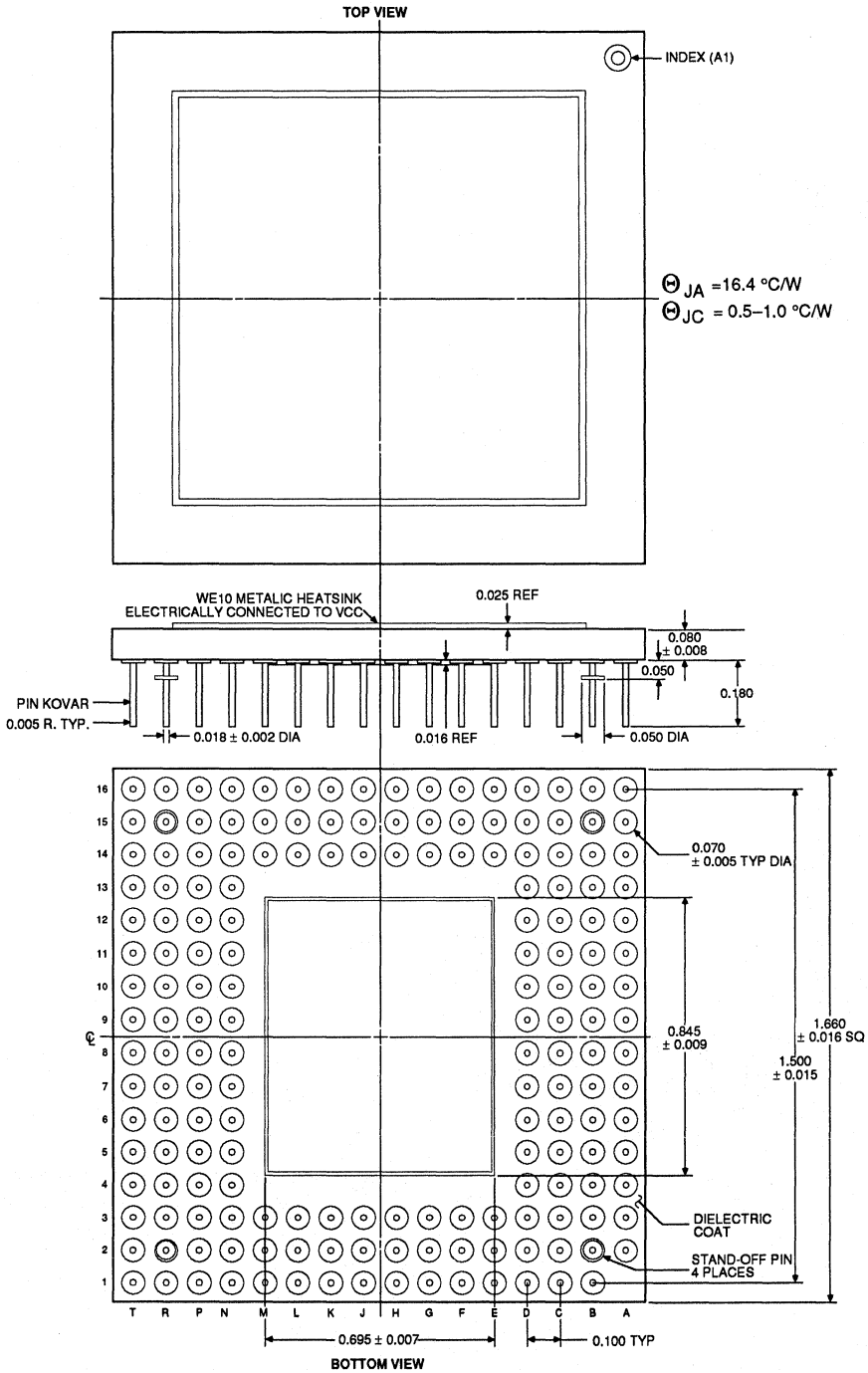
NOTES:
1. LEADS ARE SHIPPED UNFORMED IN CARRIERS IN TRAYS, TOPSIDE UP
2. FORMING TOOL INFORMATION:
   – FANCORT INDUSTRIES – (201) 575-0610 WEST CALDWELL NJ.
   – RISI INDUSTRIES INC. (619) 425-3970 CHULA VISTA, CA.

Θ_{JA} = 35-45° C/W
Θ_{JC} = 3-5° C/W

1105 41D

164-Pin CQFP Package

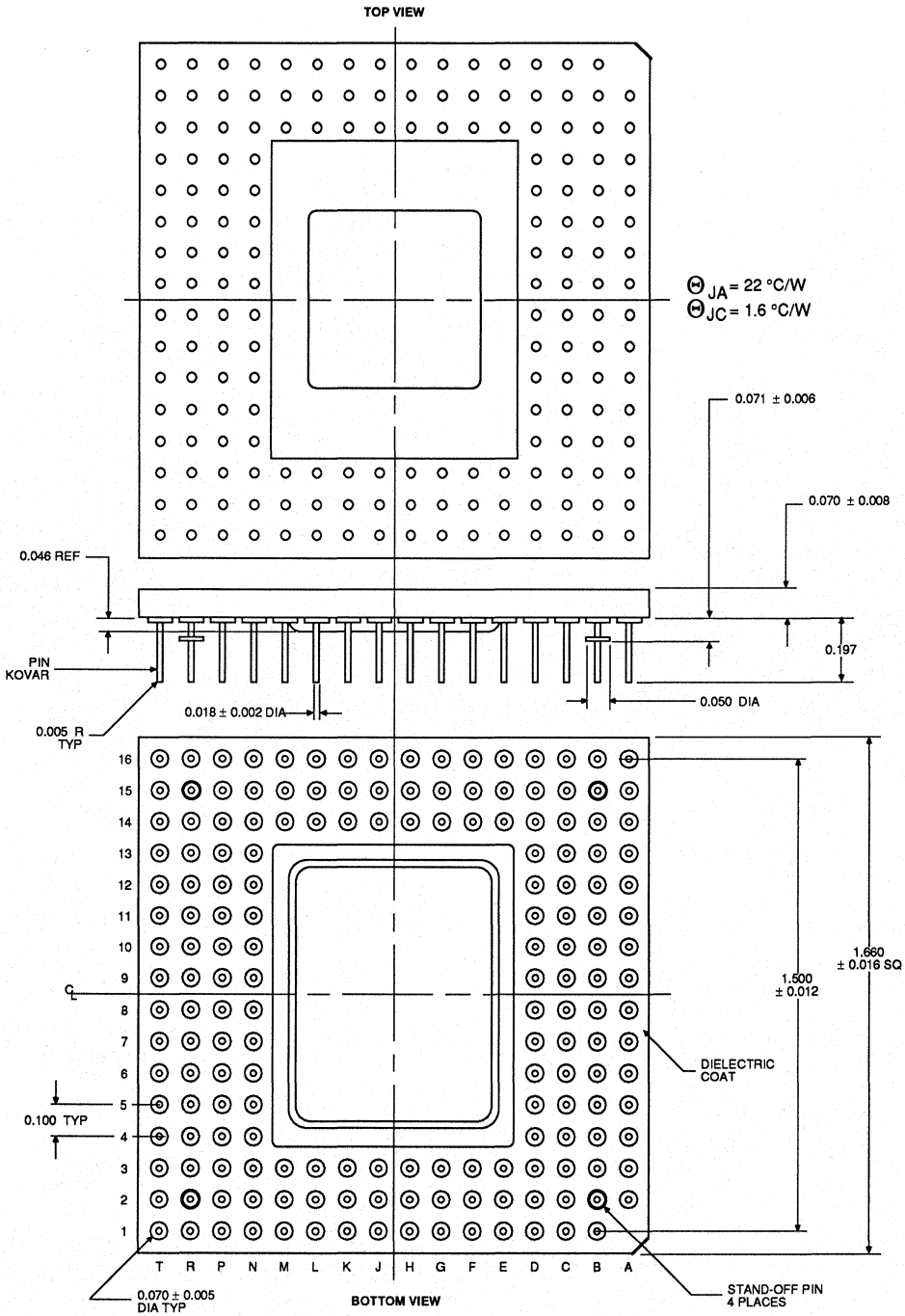
PHYSICAL DIMENSIONS (Continued)



1105 37C

175-Pin PGA Package (Ceramic)

PHYSICAL DIMENSIONS (Continued)



1991 01B

175-Pin PPGA Package (Plastic)

COMPONENT AVAILABILITY (11/90)

		44 PIN	48 PIN		68 PIN		84 PIN		100 PIN		132 PIN		160 PIN	164 PIN	175 PIN	
		PLASTIC PLCC	PLASTIC DIP	CERAMIC DIP	PLASTIC PLCC	CERAMIC PGA	PLASTIC PCC	CERAMIC PGA	PLASTIC PQFP	CERAMIC CQFP	PLASTIC PGA	CERAMIC PGA	PLASTIC PQFP	CERAMIC CQFP	PLASTIC PGA	CERAMIC PGA
		PC44	PD48	CD48	PC68	PG68	PC84	PG84	PQ100	CQ100	PP132	PG132	PQ160	CQ164	PP175	PG175
XC2064	-50		C	I	CI	CIM										
	-70				CI	CIM										
	-100				C	C										
XC2018	-33							MB								
	-50				CI		CI	CIMB								
	-70				CI		CI	CIMB								
	-100				C		C	C								
XC3020	-50				CI		CI	CIMB	CI	CIMB						
	-70				CI		CI	CIMB	CI	CIMB						
	-100				C		C	C	C	C						
XC3030	-50	CI			CI		CI	CIM	C							
	-70	CI			CI		CI	CIM	CI							
	-100	C			C		C	C	C							
XC3042	-50						CI	CIMB	C	CIMB	CI	CIMB				
	-70						CI	CIMB	CI	CIMB	CI	CI				
	-100						C	C	C	C	C	C				
XC3064	-50						C				CI	CIM				
	-70						C				CI	CIM				
	-100						C				C	C				
XC3090	-50						C						CI*	CIMB	CI	CIMB
	-70						C						CI*	CIMB	CI	CIMB
	-100						C						C*	C	C	C

X1104

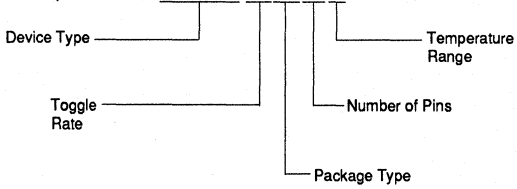
XC1736A/XC1765-PD8C Plastic 8-Pin Mini-DIP
–40°C to 85°C
XC1736A/XC1765-CD8M Ceramic 8-Pin Mini-DIP
–55°C to 125°C

LCA Temperature Options

Symbol	Description	Temperature
C	Commercial	0°C to 70°C
I	Industrial	–40°C to 85°C
M	Mil Temp	–55°C to 125°C
B	Military	MIL-STD-883, Class B

ORDERING INFORMATION

Example: XC3020-70PC68C



COMPATIBLE PACKAGE OPTIONS

A range of LCA devices is available in identical packages with identical pin-outs. A design can thus be started with one device, then migrated to a larger or smaller chip while retaining the original footprint and PC-board layout.

Examples:
PC 68: 2064-2018-3020-3030
PC 84: 2018-3020-3030-3042-3064-3090
PG 84: 2018-3020-3030-3042
PQ 100: 3020-3030-3042
PG 132: 3042-3064

Note, however, that the XC2000 and XC3000 families differ in the position of XTL1 as well as three parallel address bits (6, 7 and 11) and most of the data pins used in parallel master mode.

XC2018 and XC3020 are not available in PGA68, since the PGA84 is the same size and offers more I/O.

Note that a PLCC in a socket with PGA footprint generates a printed circuit board pin-out **different** from a PGA device.

XC2064/XC2018 Logic Cell™ Array

Product Specification

FEATURES

- Fully Field-Programmable:
  - I/O functions
  - Digital logic functions
  - Interconnections
- General-purpose array architecture
- Complete user control of design cycle
- Compatible arrays with logic cell complexity equivalent to 1200 and 1800 gates
- Standard product availability
- 100% factory-tested
- Selectable configuration modes
- Low-power, CMOS, static-memory technology
- Performance equivalent to TTL SSI/MSI
- TTL or CMOS input thresholds
- Complete development system support
  - XACT Design Editor
  - Schematic Entry
  - XACTOR In-Circuit Emulator
  - Macro Library
  - Timing Calculator
  - Logic and Timing Simulator
  - Auto Place / Route

DESCRIPTION

The Logic Cell™ Array (LCA™) is a high density CMOS integrated circuit. Its user-programmable array architecture is made up of three types of configurable elements: Input/Output Blocks, logic blocks and Interconnect. The designer can define individual I/O blocks for interface to external circuitry, define logic blocks to implement logic functions and define interconnection networks to compose larger scale logic functions. The XACT™ Development System provides interactive graphic design capture and automatic routing. Both logic simulation and in-circuit emulation are available for design verification.

The Logic Cell Array is available in a variety of logic capacities, package styles, temperature ranges and speed grades.

Part Number	Logic Capacity (gates)	Configurable Logic Blocks	User I/Os	Configuration Program (bits)
XC2064	1200	64	58	12038
XC2018	1800	100	74	17878

The LCA logic functions and interconnections are determined by data stored in internal static-memory cells. On-chip logic provides for automatic loading of configuration data at power-up. The program data can reside in an EEPROM, EPROM or ROM on the circuit board or on a floppy disk or hard disk. The program can be loaded in a number of modes to accommodate various system requirements.

ARCHITECTURE

The general structure of a Logic Cell Array is shown in Figure 1. The elements of the array include three categories of user programmable elements: I/O Blocks (IOBs), Configurable Logic Blocks (CLBs) and Programmable Interconnections. The I/OBs provide an interface between the logic array and the device package pins. The CLBs perform user-specified logic functions, and the interconnect resources are programmed to form networks that carry logic signals among the blocks.

LCA configuration is established through a distributed array of memory cells.The XACT development system generates the program used to configure the Logic Cell Array which includes logic to implement automatic configuration.

Configuration Memory

The configuration of the Logic Cell Array is established by programming memory cells which determine the logic functions and interconnections. The memory loading process is independent of the user logic functions.

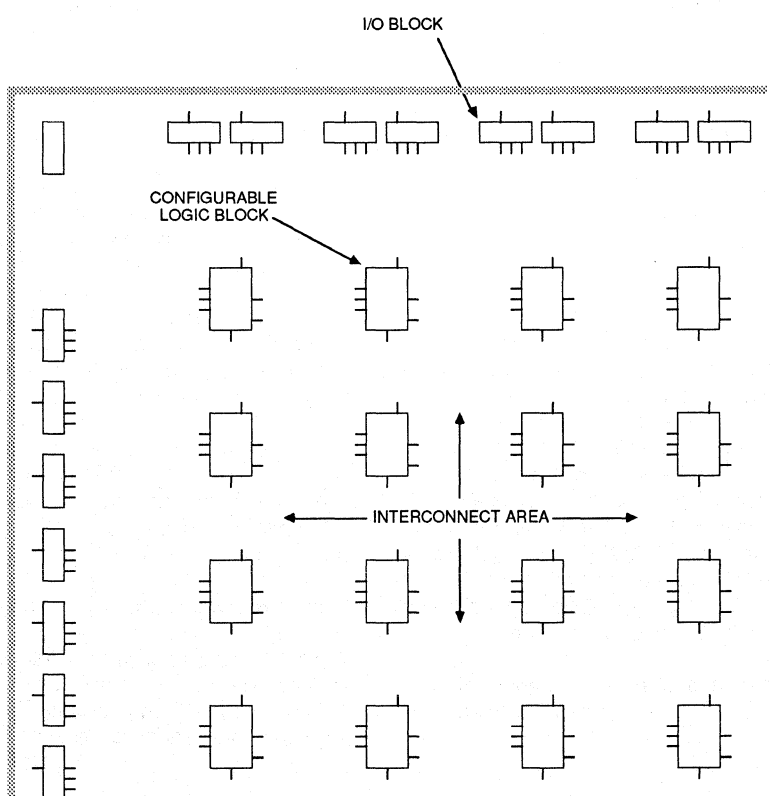
The static memory cell used for the configuration memory in the Logic Cell Array has been designed specifically for high reliability and noise immunity. Based on this design, which has been patented, integrity of the LCA configuration memory is assured even under adverse conditions. Compared with other programming alternatives, static memory provides the best combination of high density, high performance, high reliability and comprehensive testability. As shown in Figure 2, the basic memory cell consists of two CMOS inverters plus a pass transistor used for writing data to the cell. The cell is only written during configuration and only read during readback. During normal operation the pass transistor is "off" and does not affect the stability of the cell. This is quite different from the normal operation of conventional memory devices, in which the cells are continuously read and rewritten.

The outputs Q and $\overline{Q}$ control pass-transistor gates directly. The absence of sense amplifiers and the output capacitive load provide additional 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.

Input/Output Block

Each user-configurable I/O block (IOB) provides an interface between the external package pin of the device and the internal logic. Each I/O block includes a programmable input path and a programmable output buffer. It also provides input clamping diodes to provide protection from electro-static damage, and circuits to protect the LCA from latch-up due to input currents. Figure 3 shows the general structure of the I/O block.

The input buffer portion of each I/O block provides threshold detection to translate external signals applied to the package pin to internal logic levels. The input buffer threshold of the I/O blocks can be programmed to be compatible with either TTL (1.4 V) or CMOS (2.2 V) levels.



1104 01

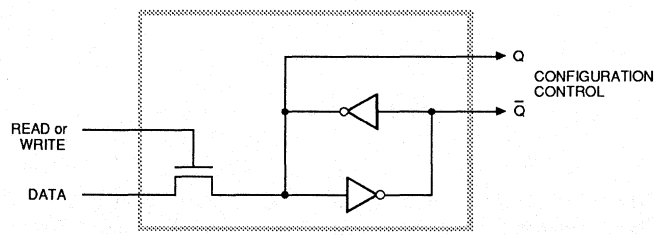
Figure 1. Logic Cell Array Structure

The buffered input signal drives both the data input of an edge-triggered D flip-flop and one input of a two-input multiplexer. The output of the flip-flop provides the other input to the multiplexer. The user can select either the direct input path or the registered input, based on the content of the memory cell controlling the multiplexer. The I/O Blocks along each edge of the die share common clocks. The flip-flops are reset during configuration as well as by the active-low chip $\overline{\text{RESET}}$ input.

Output buffers in the I/O blocks provide 4-mA drive for high fan-out CMOS or TTL-compatible signal levels. The output data (driving I/O block pin O) is the data source for the I/O block output buffer. Each I/O block output buffer is controlled by the contents of two configuration memory cells which turn the buffer ON or OFF or select 3-state buffer control. The user may also select the output buffer 3-state control (I/O block pin TS). When this I/O block output control signal is High (a logic "1"), the buffer is disabled and the package pin is high-impedance.

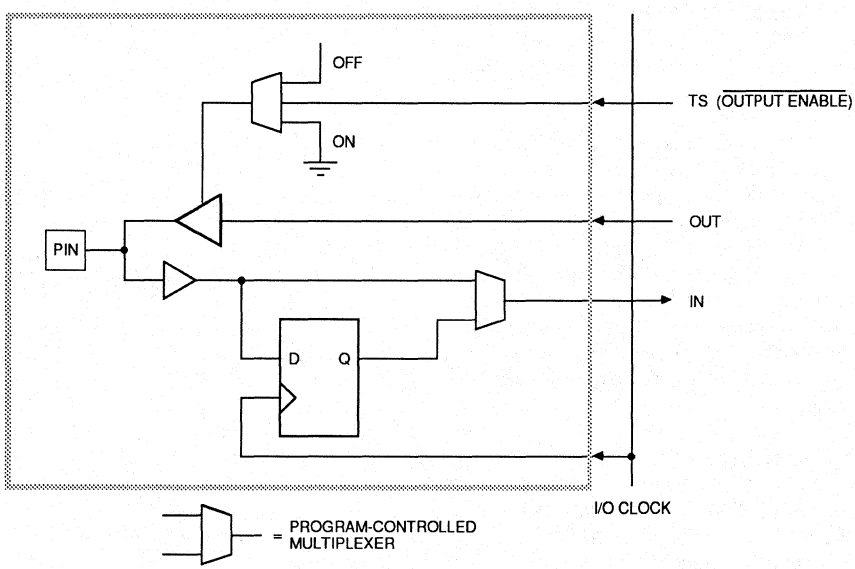
Configurable Logic Block

An array of Configurable Logic Blocks (CLBs) provides the functional elements from which the user's logic is constructed. The logic blocks are arranged in a matrix in the



1105 12

Figure 2. Configuration Memory Cell



1104 03

Figure 3. I/O Block

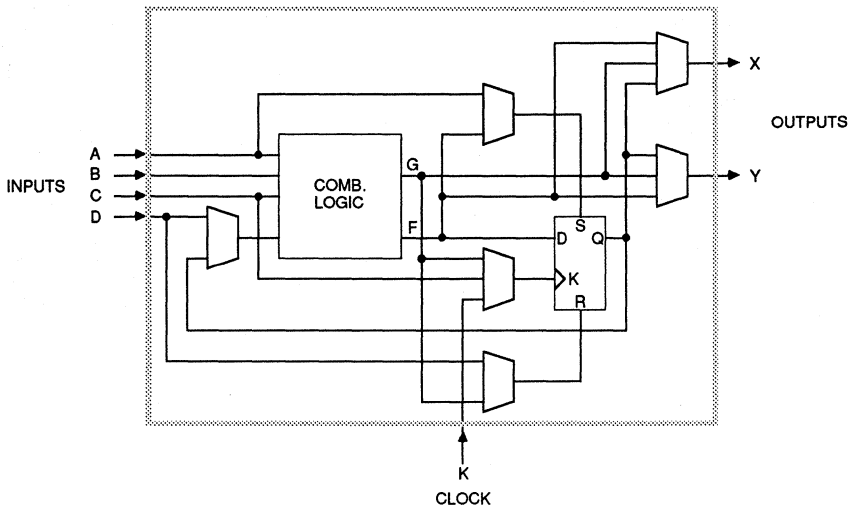


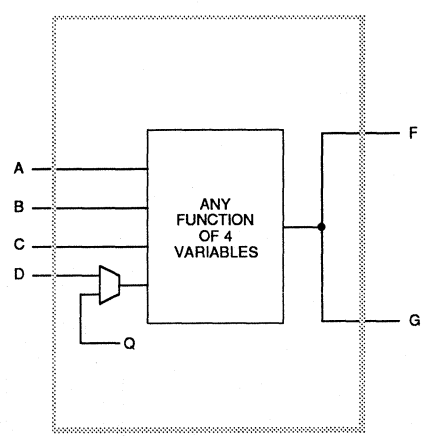
Figure 4. Configurable Logic Block

center of the device. The XC2064 has 64 such blocks arranged in an 8-row by 8-column matrix. The XC2018 has 100 logic blocks arranged in a 10 by 10 matrix.

Each logic block has a combinatorial logic section, a storage element, and an internal routing and control section. Each CLB has four general-purpose inputs: A, B, C and D; and a special clock input (K), which may be driven from the interconnect adjacent to the block. Each CLB also has two outputs, X and Y, which may drive interconnect networks. Figure 4 shows the resources of a Configurable Logic Block.

The logic block combinatorial logic uses a table look-up memory to implement Boolean functions. This tech-nique can generate any logic function of up to four variables with a high speed sixteen-bit memory. The propagation delay through the combinatorial network is independent of the function generated. Each block can perform any function of four variables or any two functions of three variables each. The variables may be selected from among the four inputs and the block's storage element output "Q". Figure 5 shows various options which may be specified for the combinatorial logic.

If the single 4-variable configuration is selected (Option 1), the F and G outputs are identical. If the 2-function alternative is selected (Option 2), logic functions F and G may be independent functions of three variables each. The three variables can be selected from among the four logic block inputs and the storage element output "Q". A third form of the combinatorial logic (Option 3) is a special case of the 2-function form in which the B input dynamically selects between the two function tables providing a single



OPTION 1

1 FUNCTION OF 4 VARIABLES

merged logic function output. This dynamic selection allows some 5-variable functions to be generated from the four block inputs and storage element Q. Combinatorial functions are restricted in that one may not use both its storage element output Q and the input variable of the logic block pin "D" in the same function.

If used, the storage element in each Configurable Logic Block (Figure 6) can be programmed to be either an edge-sensitive "D" type flip-flop or a level-sensitive "D" latch. The clock or enable for each storage element can be selected from:

- The special-purpose clock input K
- The general-purpose input C
- The combinatorial function G

The user may also select the clock active sense within each logic block. This programmable inversion eliminates the need to route both phases of a clock signal throughout the device.

The storage element data input is supplied from the function F output of the combinatorial logic. Asynchronous SET and RESET controls are provided for each storage element. The user may enable these controls independently and select their source. They are active High inputs and the asynchronous reset is dominant. The storage elements are reset by the active-Low chip $\overline{\text{RESET}}$ pin as well as by the initialization phase preceding configuration. If the storage element is not used, it is disabled.

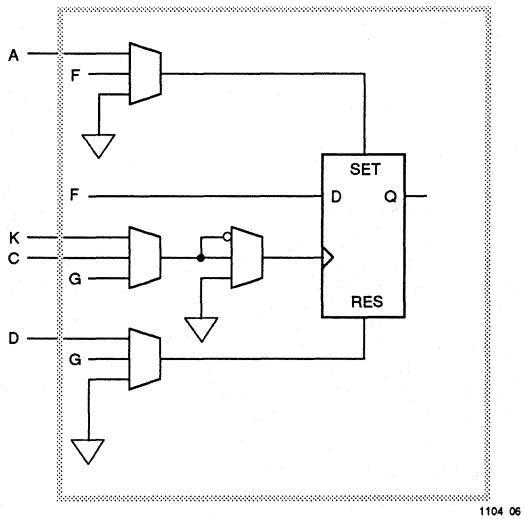


Figure 6. CLB Storage Elememt

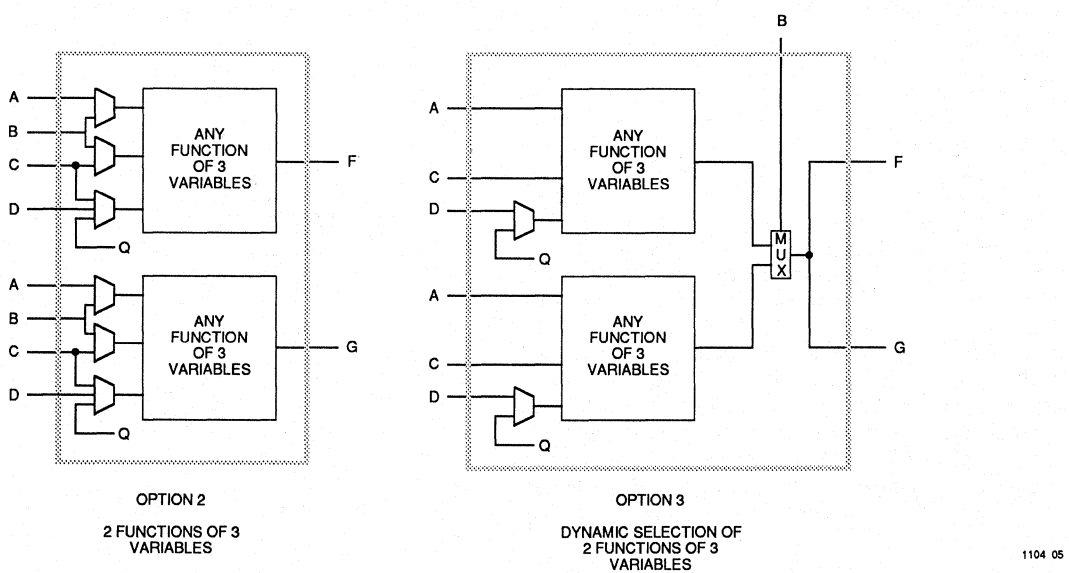


Figure 5. CLB Combinatorial Logic Options
Note: Variables D and Q can not be used in the same function.

The two block outputs, X and Y, can be driven by either the combinatorial functions, F or G, or the storage element output Q (Figure 4). Selection of the outputs is completely interchangeable and may be made to optimize routing efficiencies of the networks interconnecting the logic blocks and I/O blocks.

PROGRAMMABLE INTERCONNECT

Programmable interconnection resources in the Logic Cell Array provide routing paths to connect inputs and outputs of the I/O and logic blocks into desired networks. All interconnections are composed of metal segments, with programmable switching points provided to implement the necessary routing. Three types of resources accommodate different types of networks:

- General purpose interconnect
- Long lines
- Direct connection

General-Purpose Interconnect

General-purpose interconnect, as shown in Figure 7a, is composed of four horizontal metal segments between the rows and five vertical metal segments between the columns of logic and I/O blocks. Each segment is only the "height" or "width" of a logic block. Where these segments would cross at the intersections of rows and columns, switching matrices are provided to allow interconnections of metal segments from the adjoining rows and columns. Switches in the switch matrices and on block outputs are specially designed transistors, each controlled by a configuration bit.

Logic-block output switches provide contacts to adjacent general interconnect segments and therefore to the switching matrix at each end of those segments. A switch matrix can connect an interconnect segment to other segments to form a network. Figure 7a shows the general interconnect used to route a signal from one logic block to three other logic blocks. As shown, combinations of closed switches in a switch matrix allow multiple branches for each network. The inputs of the logic or I/O blocks are multiplexers that can be program-med with configuration bits 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 input connection. The development system software provides automatic routing of these interconnections. Interactive routing is also available for design optimization. This is accomplished by selecting a network and then toggling the states of the interconnect points by selecting them with the "mouse". In this mode, the connections through the switch matrix may be established by selecting pairs of matrix pins. The switching matrix combinations are indicated in Figure 7b.

Special buffers within the interconnect area provide periodic signal isolation and restoration for higher general interconnect fan-out and better performance. The repowering buffers are bidirectional, since signals must be able to propagate in either direction on a general interconnect segment. Direction controls are automatically established by the Logic Cell Array development system software. Repowering buffers are provided only for the general-purpose interconnect since the direct and long line resources do not exhibit the same R-C delay accumulation. The Logic Cell Array is divided into nine sections with buffers automatically provided for general interconnect at the boundaries of these sections. These boundaries can be viewed with the development system. For routing within a section, no buffers are used. The delay calculator of the XACT development system automatically calculates and displays the block, interconnect and buffer delays for any selected paths.

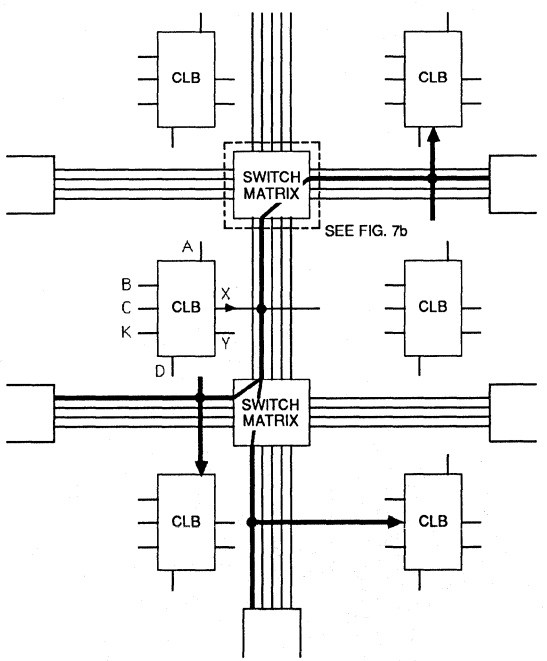


Figure 7a. General-Purpose Interconnect

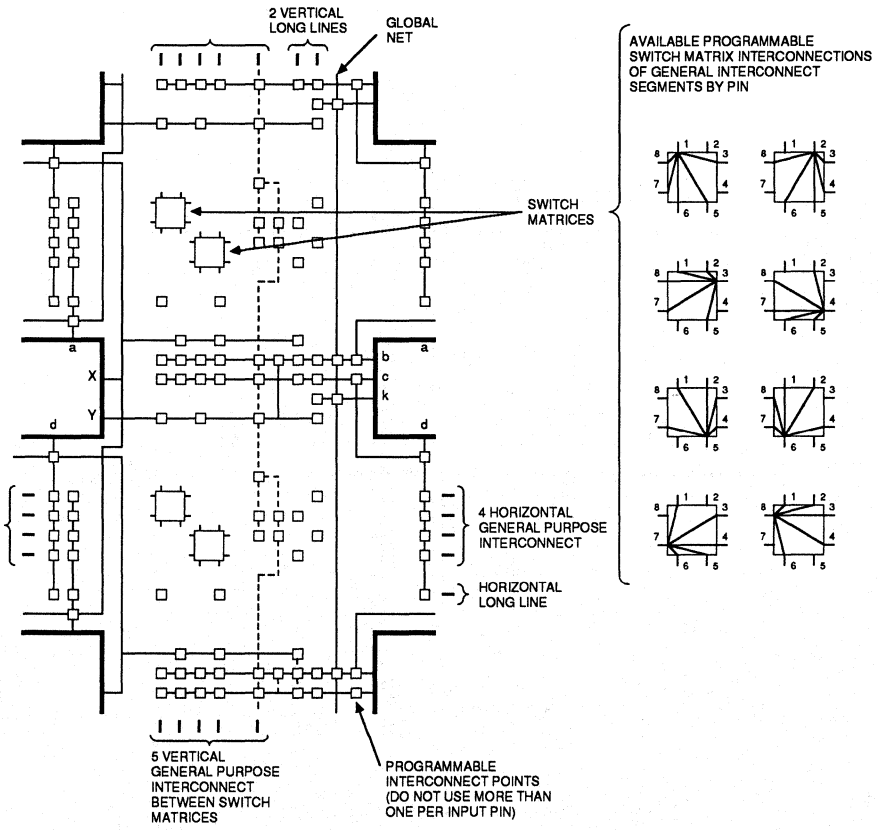


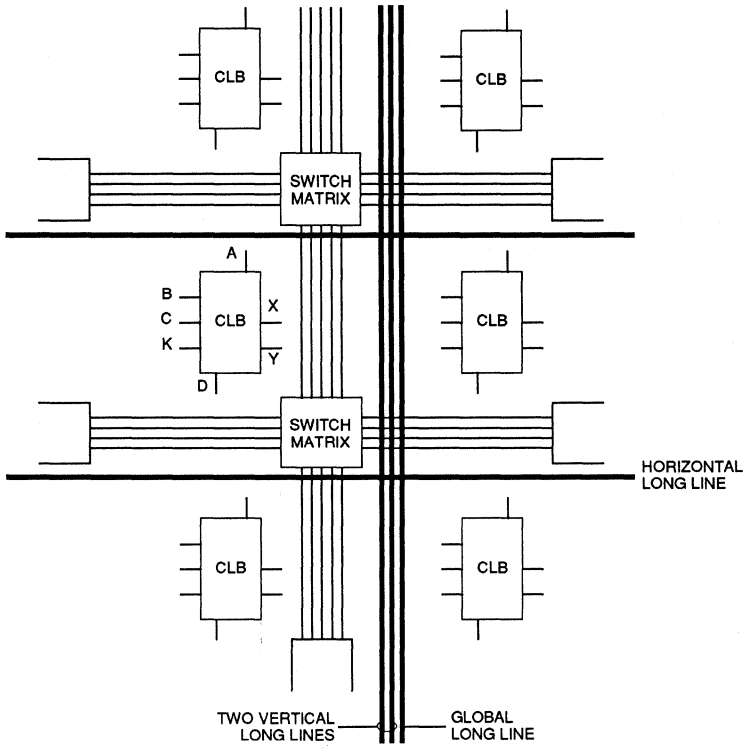
Figure 7b. Routing and Switch Matrix Connections

Long Lines

Long-lines, shown in Figure 8a, run both vertically and horizontally the height or width of the interconnect area. Each vertical interconnection column has two long lines; each horizontal row has one, with an additional long line adjacent to each set of I/O blocks. The long lines bypass the switch matrices and are intended primarily for signals that must travel a long distance or must have minimum skew among multiple destinations.

A global buffer in the Logic Cell Array is available to drive a single signal to all B and K inputs of logic blocks. Using the global buffer for a clock provides a very low skew, high fan-out synchronized clock for use at any or all of the logic blocks. At each block, a configuration bit for the K input to the block can select this global line as the storage element clock signal. Alternatively, other clock sources can be used.

A second buffer below the bottom row of the array drives a horizontal long line which, in turn, can drive a vertical long line in each interconnection column. This alternate buffer also has low skew and high fan-out capability. The network formed by this alternate buffer's long lines can be selected to drive the B, C or K inputs of the logic blocks.



1104 09

Figure 8a. Long Line Interconnect

Alternatively, these long lines can be driven by a logic or I/O block on a column by column basis. This capability provides a common, low-skew clock or control line within each column of logic blocks. Interconnections of these long lines are shown in Figure 8b.

Direct Interconnect

Direct interconnect, shown in Figure 9, provides the most efficient implementation of networks between adjacent logic or I/O blocks. Signals routed from block to block by means of direct interconnect exhibit minimum interconnect propagation and use minimum interconnect resources. For each Configurable Logic Block, the X output may be connected directly to the C or D inputs of the CLB above and to the A or B inputs of the CLB below it. The Y output can use direct interconnect to drive the B input of the block immediately to its right. Where logic blocks are adjacent to I/O blocks, direct connect is provided to the I/O block input (I) on the left edge of the die, the output (O) on the right edge, or both on I/O blocks at the top and bottom of the die. Direct interconnections of I/O blocks with CLBs are shown in Figure 8b.

CRYSTAL OSCILLATOR

An internal high speed inverting amplifier is available to implement an on-chip crystal oscillator. It is associated with the auxiliary clock buffer in the lower right corner of the die. When configured to drive the auxiliary clock buffer, two special adjacent user I/O blocks are also configured to connect the oscillator amplifier with external crystal oscillator components, as shown in Figure 10. This circuit becomes active before configuration is complete in order to allow the oscillator to stabilize. Actual internal connection is delayed until completion of configuration. The feedback resistor R1 between output and input, biases the amplifier at threshold. It should be as large a value as practical to minimize loading of the crystal. The inversion of the amplifier, together with the R-C networks and crystal, produce the 360-degree phase shift of the Pierce oscillator. A series resistor R2 may be included to add to

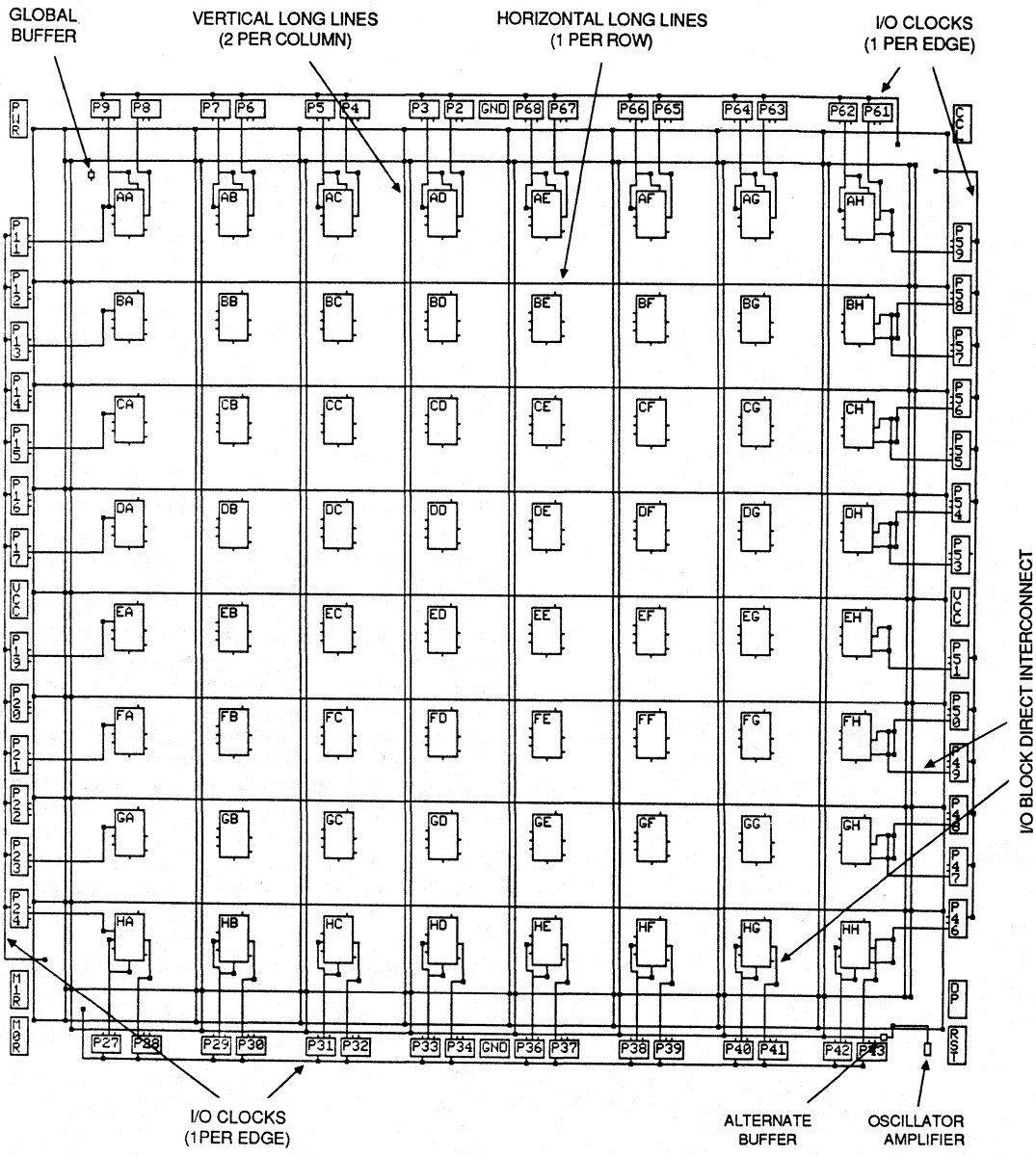


Figure 8b. XC2064 Long Lines, I/O Clocks, I/O Direct Interconnect

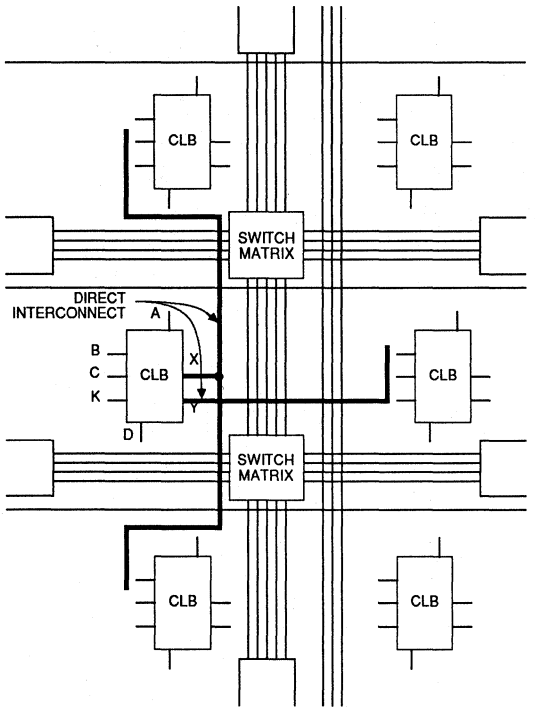
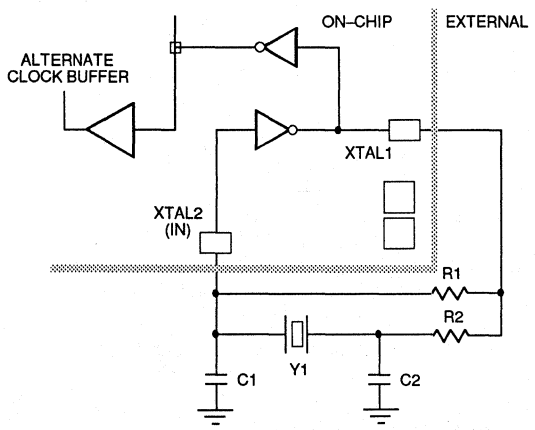


Figure 9. Direct Interconnect



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 series resonant

	XTAL1	XTAL2
48 DIP	33	30
68 PLCC	46	43
68 PGA	J10	L10
84 PLCC	56	53
84 PGA	K11	L11

Figure 10. Crystal Oscillator

the amplifier output impedance when needed for phase-shift control or crystal resistance matching or to limit the amplifier input swing to control clipping at large amplitudes. Excess feedback voltage may be adjusted by the ratio of C2/C1. The amplifier is designed to be used over the range from 1 MHz up to one-half the specified CLB toggle frequency. Use at frequencies below 1 MHz may require individual characterization with respect to a series resistance. Operation at frequencies above 20 MHz generally requires a crystal to operate in a third overtone mode, in which the fundamental frequency must be suppressed by the R-C networks. When the amplifier does not drive the auxiliary buffer, these I/O blocks and their package pins are available for general user I/O.

POWER

Power Distribution

Power for the LCA is distributed through a grid to achieve high noise immunity and isolation between logic and I/O. For packages having more than 48 pins, two Vcc pins and two ground pins are provided (see Figure 11). Inside the LCA, a dedicated Vcc and ground ring surrounding the logic array provides power to the I/O drivers. An independent matrix of Vcc and ground lines supplies the interior logic of the device. This power distribution grid provides a stable supply and ground for all internal logic, providing the external package power pins are appropriately decoupled. Typically a 0.1 µF capacitor connected between the Vcc and ground pins near the package will provide adequate decoupling.

Output buffers capable of driving the specified 4 mA loads under worst-case conditions may be capable of driving 25 to 30 times that current in a best case. Noise can be reduced by minimizing external load capacitance and reducing simultaneous output transitions in the same direction. It may also be beneficial to locate heavily loaded output buffers near the ground pads. Multiple Vcc and ground pin connections are required for package types which provide them.

Power Consumption

The Logic Cell Array exhibits the low power consumption characteristic of CMOS ICs. Only quiescent power is required for the LCA configured for CMOS input levels. The TTL input level configuration option requires additional power for level shifting. The power required by the static memory cells which hold the configuration data is very low and may be maintained in a power-down mode.

Typically most of power dissipation is produced by capacitive loads on the output buffers, since the power per output is 25 µW / pF / MHz . Another component of I/O power is the DC loading on each output pin. For any given system, the user can calculate the I/O power requirement based on the sum of capacitive and resistive loading of the devices driven by the Logic Cell Array.

Internal power supply dissipation is a function of clock frequency and the number of nodes changing on each clock. In an LCA the fraction of nodes changing on a given clock is typically low (10–20%). For example, in a 16-bit binary counter, the average clock produces a change in slightly less than 2 of the 16 bits. In a 4-input AND gate there will be 2 transitions in 16 states. Typical global clock buffer power is about 3 mW / MHz for the XC2064 and 4 mW / MHz for the XC2018. With a "typical" load of three general interconnect segments, each Configurable Logic Block output requires about 0.4 mW / MHz of its output frequency. Graphs of power versus operating frequency are shown in Table 1 on page 2-83.

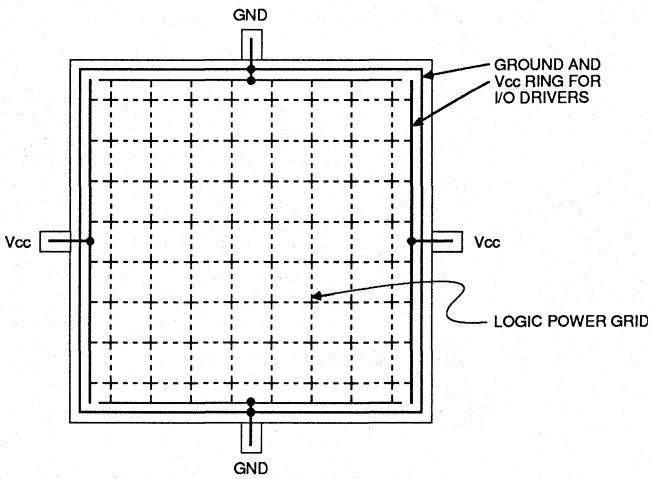


Figure 11. LCA Power Distribution

PROGRAMMING

Configuration data to define the function and interconnection within a Logic Cell Array are loaded automatically at power-up or upon command. Several methods of automatically loading the required data are designed into the Logic Cell Array and are determined by logic levels applied to mode selection pins at configuration time. The form of the data may be either serial or parallel, depending on the configuration mode. The programming data are independent of the configuration mode selected. The state diagram of Figure 12 illustrates the configuration process.

Input thresholds for user I/O pins can be selected to be either TTL-compatible or CMOS-compatible. At power-up, all inputs are TTL-compatible and remain in that state until the LCA begins operation. If the user has selected CMOS compatibility, the input thresholds are changed to CMOS levels during configuration.

Figure 13 shows the specific data arrangement for the XC2064 device. Future products will use the same data format to maintain compatibility between different devices of the Xilinx product line, but they will have different sizes and numbers of data frames. For the XC2064,

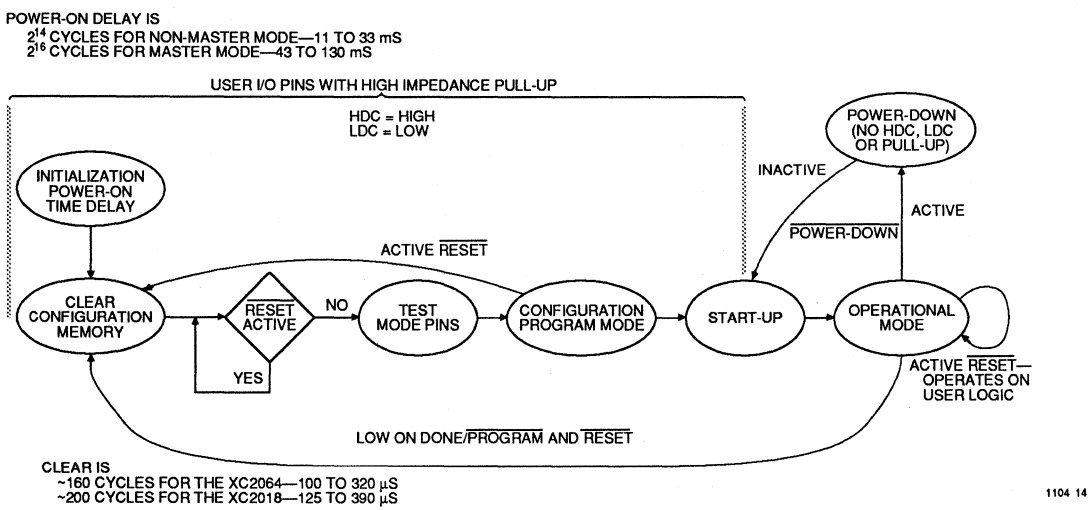


Figure 12. A State Diagram of the Configuration Process for Power-up and Re-program

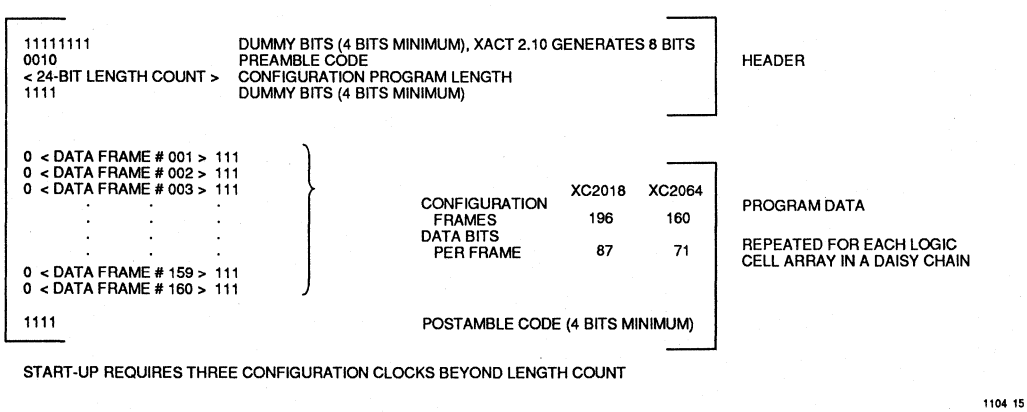


Figure 13. XC2064 Internal Configuration Data Arrangement

configuration requires 12,038 bits for each device. For the XC2018, the configuration of each device requires 17,878 bits. The XC2064 uses 160 configuration data frames and the XC2018 uses 197.

The configuration bit stream begins with preamble bits, a preamble code and a length count. The length count is loaded into the control logic of the Logic Cell Array and is used to determine the completion of the configuration process. When configuration is initiated, a 24-bit length counter is set to 0 and begins to count the total number of configuration clock cycles applied to the device. When the current length count equals the loaded length count, the configuration process is complete. Two clocks before completion, the internal logic becomes active and is reset. On the next clock, the inputs and outputs become active as configured and consideration should be given to avoid configuration signal contention. *(Attention must be paid to avoid contention on pins which are used as inputs during configuration and become outputs in operation.)* On the last configuration clock, the completion of configuration is signalled by the release of the DONE / $\overline{\text{PROG}}$ pin of the device as the device begins operation. This open-drain output can be AND-tied with multiple Logic Cell Arrays and used as an active-High READY or active-Low , $\overline{\text{RESET}}$, to other portions of the system. High during configuration (HDC) and low during configuration ($\overline{\text{LDC}}$), are released one CCLK cycle before DONE is asserted. In master mode configurations, it is convenient to use $\overline{\text{LDC}}$ as an active-Low EPROM chip enable.

As each data bit is supplied to the LCA, it is internally assembled into a data word. As each data word is completely assembled, it is loaded in parallel into one word of the internal configuration memory array. The last word must be loaded before the current length count compare is true. If the configuration data are in error, e.g., PROM address lines swapped, the LCA will not be ready at the length count and the counter will cycle through an additional complete count prior to configuration being "done".

Figure 14 shows the selection of the configuration mode based on the state of the mode pins M0 and M1. These package pins are sampled prior to the start of the configuration process to determine the mode to be used. Once configuration is DONE and subsequent operation has begun, the mode pins may be used to perform data readback, as discussed later. An additional mode pin, M2, must be defined at the start of configuration. This package pin is a user-configurable I/O after configuration is complete.

MODE PIN			MODE SELECTED
M0	M1	M2	
0	0	0	MASTER SERIAL
0	0	1	MASTER LOW MODE
0	1	1	MASTER HIGH MODE
1	0	1	PERIPHERAL MODE
1	1	1	SLAVE MODE

MASTER LOW ADDRESSES BEGIN AT 0000 AND INCREMENT
MASTER HIGH ADDRESSES BEGIN AT FFFF AND DECREMENT

1104 13

Figure 14. Configuration Mode Selection

Initialization Phase

When power is applied, an internal power-on-reset circuit is triggered. When Vcc reaches the voltage at which the LCA begins to operate (nominally 2.5 to 3 V), the chip is initialized, outputs are made high-impedance and a time-out is initiated to allow time for power to stabilize. This time-out (11 to 33 ms) is determined by a counter driven by a self-generated, internal sampling clock that drives the configuration clock (CCLK) in master configuration mode. This internal sampling clock will vary with process, temperature and power supply over the range of 0.5 to 1.5 MHz. LCAs with mode lines set for master mode will time-out of their initialization using a longer counter (43 to 130 ms) to assure that all devices, which it may be driving in a daisy chain, will be ready. Configuration using peripheral or slave modes must be delayed long enough for this initialization to be completed.

The initialization phase may be extended by asserting the active-Low external $\overline{\text{RESET}}$. If a configuration has begun, an assertion of $\overline{\text{RESET}}$ will initiate an abort, including an orderly clearing of partially loaded configuration memory bits. After about three clock cycles for synchronization, initialization will require about 160 additional cycles of the internal sampling clock (197 for the XC2018) to clear the internal memory before another configuration may begin. Reprogramming is initialized by a High-to-Low transition on $\overline{\text{RESET}}$ (after $\overline{\text{RESET}}$ has been High for at least 6 µs) followed by a Low level (for at least 6 µs) on both the $\overline{\text{RESET}}$ and the open-drain DONE/$\overline{\text{PROG}}$ pins. This returns the LCA to the CLEAR state, as shown in Fig. 12.

Master Mode

In Master mode, the Logic Cell Array automatically loads the configuration program from an external memory device. Figure 15a shows an example of the Master mode connections required. The Logic Cell Array provides 16 address outputs and the control signals $\overline{RCLK}$ (Read Clock), HDC (High during configuration) and $\overline{LDC}$ (Low during configuration) to execute Read cycles from the external memory. Parallel 8-bit data words are read and internally serialized. As each data word is read, the least significant bit of each byte, normally D0, is the next bit in the serial stream.

Addresses supplied by the Logic Cell Array can be selected by the mode lines to begin at address 0 and incremented to reach the memory (master Low mode), or they can begin at address FFFF Hex and be decremented (master High mode). This capability is provided to allow the Logic Cell Array to share external memory with another device, such as a microprocessor. For example, if the processor begins its execution from Low memory, the

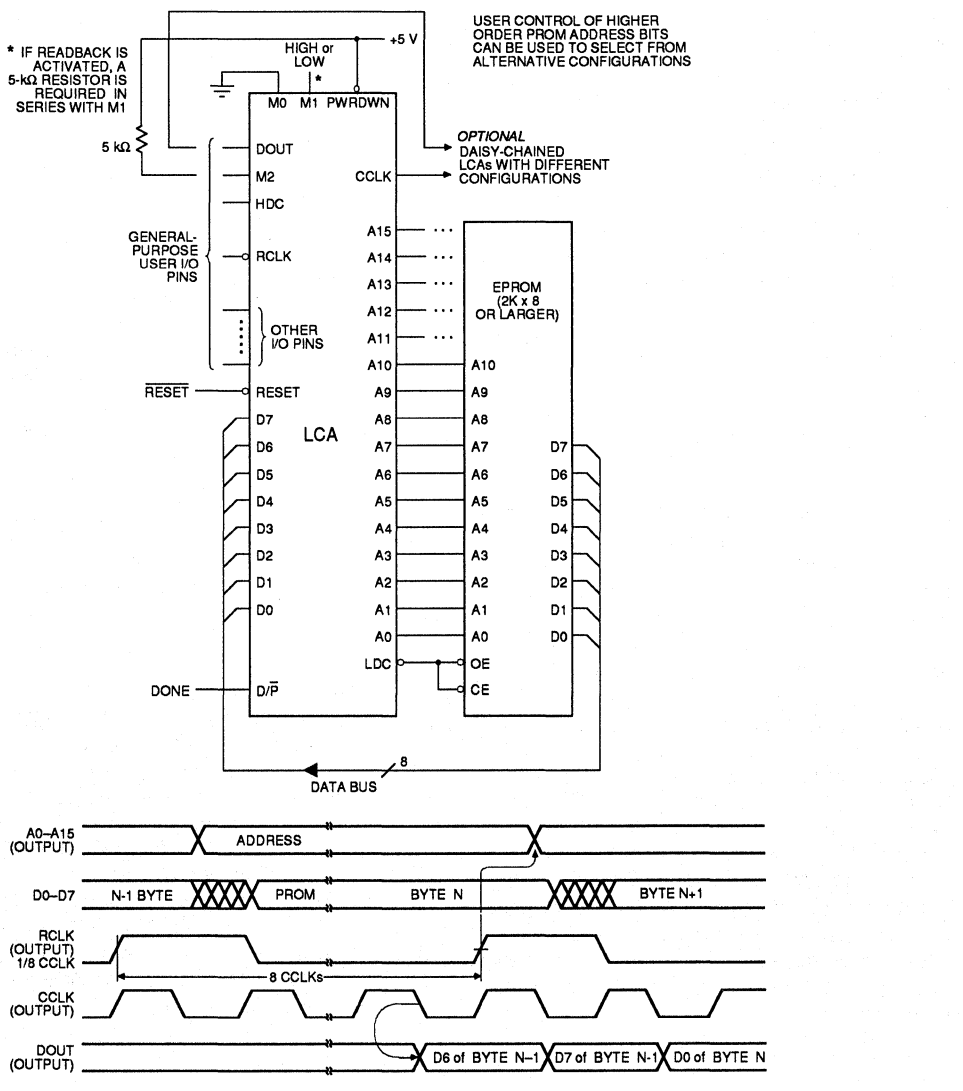


Figure 15a. Master Parallel Mode. Configuration data are loaded automaticaly from an external byte wide PROM. An XC2000 LDC signal can provide a PROM inhibit as the user I/Os become active.

Logic Cell Array can load itself from High memory and enable the processor to begin execution once configuration is completed. The Done/$\overline{PROG}$ output pin can be used to hold the processor in a Reset state until the Logic Cell Array has completed the configuration process

The Master Serial mode uses serial configuration data, synchronized by the rising edge of CCLK, as shown in Figure 15b.

Peripheral Mode (Bit Serial)

Peripheral mode provides a simplified interface through which the device may be loaded as a processor peripheral.

Figure 16 shows the peripheral mode connections. Processor Write cycles are decoded from the common assertion of the active-Low write strobe ($\overline{IOWRT}$), and two active-Low and of the active-High chip selects ($\overline{CS0}$ $\overline{CS1}$ CS2). If all these signals are not available, the unused inputs should be driven to their respective active levels. The Logic Cell Array will accept one bit of the configuration program on the data input (DIN) pin for each processor Write cycle. Data is supplied in the serial sequence described earlier.

Since only a single bit from the processor data bus is loaded per cycle, the loading process involves the processor reading a byte or word of data, writing a bit of the data to the Logic cell Array, shifting the word and writing a

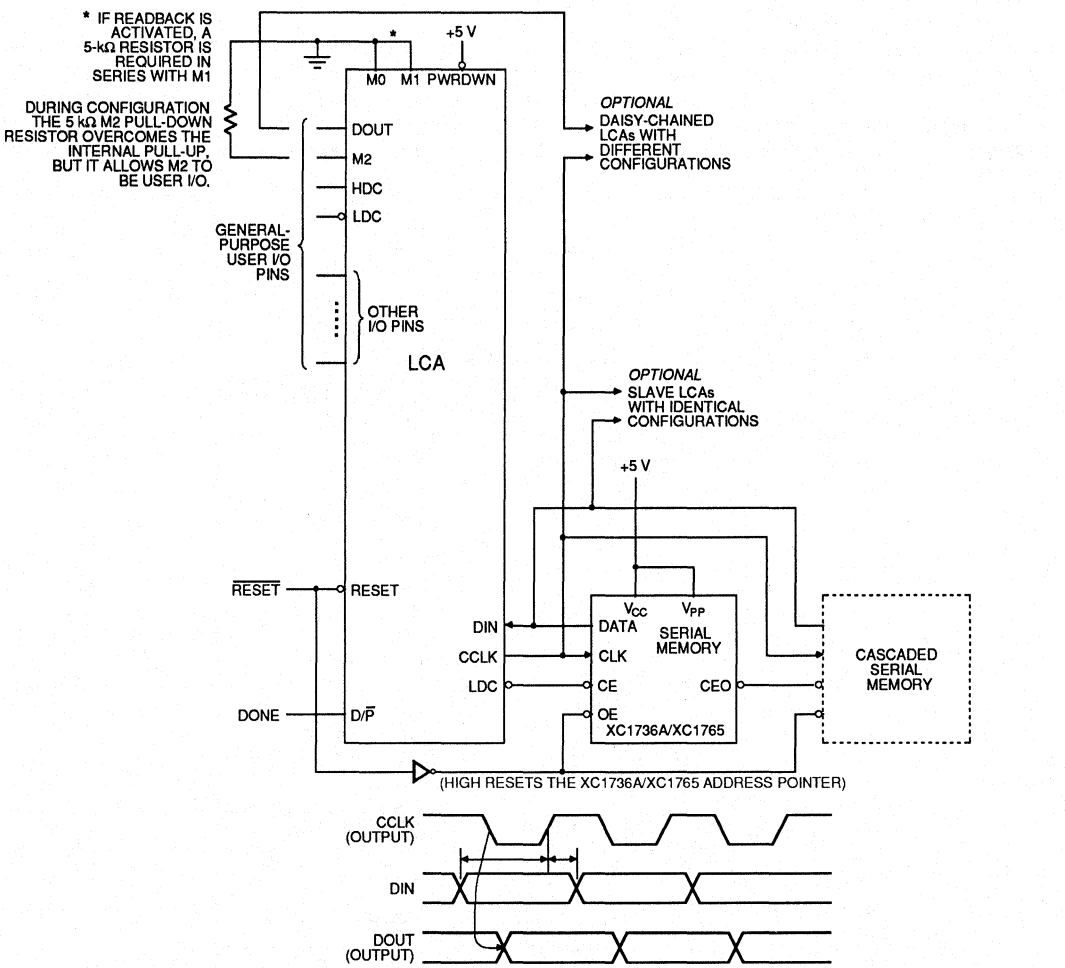


Figure 15b. Master Serial Mode. The one time programmable XC1736A Serial Configuration PROM supports automatic loading of configuration programs up to 36 Kbits. Multiple XC1736As can be cascaded to support additional LCAs. An XC2000 LDC signal can provide an XC1736A inhibit as the user I/Os become active.

bit until all bits of the word are written, then continuing in the same fashion with the next word, etc. After the configuration program has been loaded, an additional three clocks (a total of three more than the length count) must be supplied in order to complete the configuration process. When more than one device is being used in the system, each device can be assigned a different bit in the processor data bus, and multiple devices can be loaded on each processor write cycle. This "broadside" loading method provides a very easy and time-efficient method of loading several devices.

Slave Mode

Slave mode, Figure 17, provides the simplest interface for loading the Logic Cell Array configuration. Data is supplied in conjunction with a synchronizing clock. For each Low-to-High input transition of configuration clock (CCLK), the data present on the data input (DIN) pin is loaded into the internal shift register. Data may be supplied by a processor or by other special circuits. Slave mode is used for downstream devices in a daisy-chain configuration. The data for each slave LCA are supplied by the preceding LCA in the chain, and the clock is supplied by the lead device, which is configured in master or peripheral mode. After the configuration program has been loaded, an additional three clocks (a total of three more than the length count) must be supplied in order to complete the configuration process.

Daisy Chain

The daisy-chain programming mode is supported by Logic Cell Arrays in all programming modes. In master mode and peripheral mode, the LCA can act as a source of data and control for slave devices. For example, Figure 18 shows a single device in master mode, with 2 devices in slave mode. The master mode device reads the external memory and begins the configuration loading process for all of the devices.

The data begin with a preamble and a length count which are supplied to all devices at the beginning of the configuration. The length count represents the total number of cycles required to load all of the devices in the daisy chain. After loading the length count, the lead device will load its

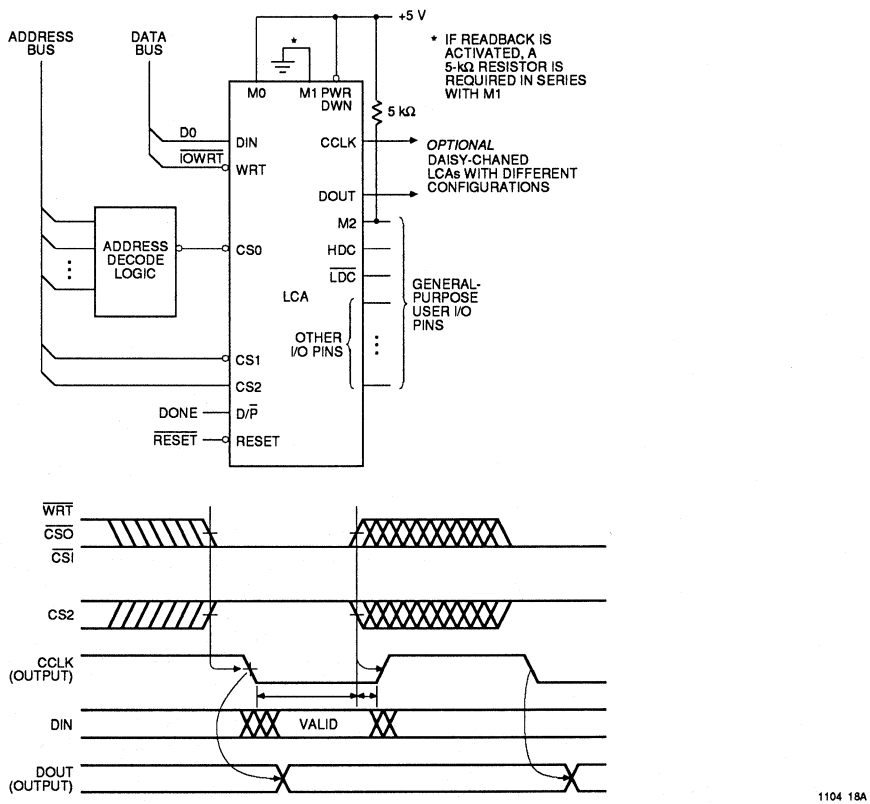


Figure 16. Peripheral Mode. Configuration data are loaded using serialized data from a microprocessor.

configuration data while providing a High DOUT to downstream devices. When the lead device has been loaded and the current length count has not reached the full value, memory access continues. Data bytes are read and serialized by the lead device. The data are passed through the lead device and appear on the data out (DOUT) pin in serial form. The lead device also generates the configuration clock (CCLK) to synchronize the serial output data. A master mode device generates an internal CCLK of 8 times the EPROM address rate, while a peripheral mode device produces CCLK from the chip select and write strobe timing.

Operation

When all of the devices have been loaded and the length count is complete, a synchronous start-up of operation is performed. On the clock cycle following the end of loading, the internal logic begins functioning in the reset state. On the next CCLK, the configured output buffers become active to allow signals to stabilize. The next CCLK cycle produces the DONE condition. The length count control of operation allows a system of multiple Logic Cell Arrays to begin operation in a synchronized fashion. 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.

SPECIAL CONFIGURATION FUNCTIONS

In addition to the normal user logic functions and interconnect, the configuration data include control for several special functions:

- Input thresholds
- Readback disable
- Reprogram
- DONE pull-up resistor

Each of these functions is controlled by a portion of the configuration program generated by the XACT Development System.

Input Thresholds

During configuration, all input thresholds are TTL level. During configuration input thresholds are established as specified, either TTL or CMOS. The $\overline{\text{PWRDWN}}$ input threshold is an exception; it is always a CMOS level input. The TTL threshold option requires additional power for threshold shifting.

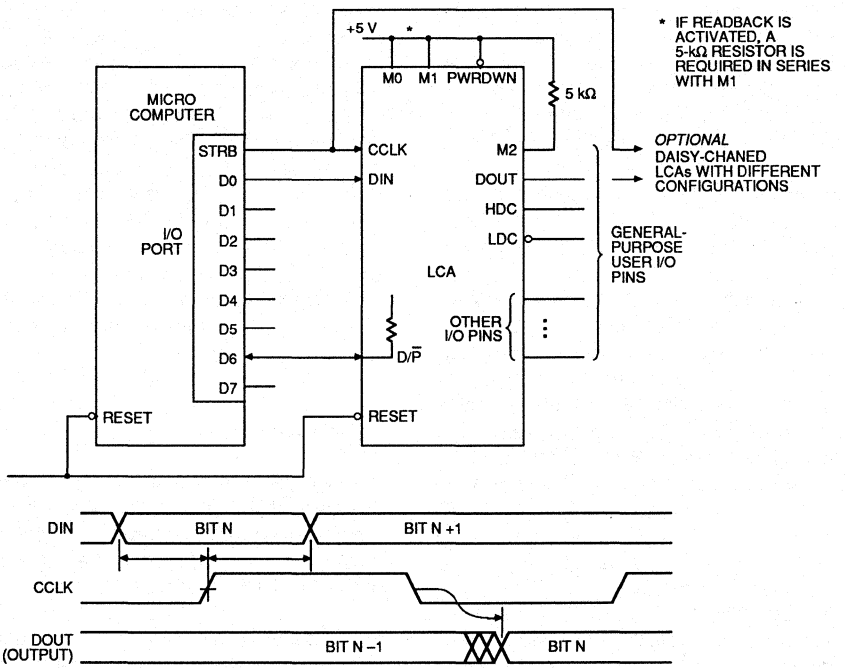


Figure 17. Slave Mode. Bit-serial configuration data are read at rising edge of the CCLK. Data on DOUT are provided on the falling edge of CCLK. Identically configured non-master mode LCAs can be configured in parallel by connecting DINs and CCLKs.

Readback

After a Logic Cell Array has been programmed, the configuration program may be read back from the device. Readback may be used for verification of configuration, and as a method of determining the state of internal logic nodes during debugging. Three readback options are provided: on command, only once, and never.

An initiation of readback is produced by a Low-to-High transition of the M0 / RTRIG (read trigger) pin. Once the readback command has been given, CCLK is cycled to read back each data bit in a format similar to loading. After two dummy bits, the first data frame is shifted out, in inverted sense, on the M1 / $\overline{\text{RDATA}}$ (read data) pin. 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 state of all internal storage elements. This information is used by the Logic Cell Array development system In-Circuit Debugger to provide visibility into the internal operation of the logic while the system is operating. To read back a uniform time sample of all storage elements, it may be necessary to inhibit the system clock.

Re-program

The Logic Cell Array configuration memory may be re-written while the device is operating in the user's system. The LCA returns to the Clear state where the configuration memory is cleared, I/O pins disabled, and mode lines re-sampled. Re-program control is often implemented using an external open collector driver which pulls DONE/$\overline{\text{PROG}}$ LOW. Once it recognizes a stable request, the Logic Cell Array will hold DONE/$\overline{\text{PROG}}$ LOW until the new configuration has been completed. Even if the DONE/$\overline{\text{PROG}}$ pin is externally held LOW beyond the configuration period, the Logic Cell Array will begin operation upon completion of configuration. To reduce sensitivity to noise, these re-program signals are filtered for 2–3 cycles of the LCA's internal timing generator (2 to 6 µs). Note that the Clear time out for a Master mode re-program or abort does not have the 4 times delay of the Initialization state. If a daisy chain is used, an external $\overline{\text{RESET}}$ is required, long enough to guarantee clearing all non-master mode devices. For XC2000 series LCAs this is accomplished with an external time delay.

In some applications the system power supply might have momentary failures which can leave the LCA's control logic in an invalid state. There are two metods to recover from this state. The first is to cycle the Vcc supply to less than 0.1 Volt and reapply valid Vcc. The second is to provide the LCA with simultaneous Low levels of at least 6 µs on $\overline{\text{RESET}}$ and DONE/$\overline{\text{PROG}}$ pins after the $\overline{\text{RESET}}$ pin has been High following a return to valid Vcc. This guarantees that the LCA will return to the Clear state. Either of these methods may be needed in the event of an incomplete voltage interruption. They are not needed for a normal application of power from an off condition.

DONE Pull-up

The DONE / $\overline{\text{PROG}}$ pin is an open drain I/O that indicates programming status. As an input, it initiates a reprogram operation. An optional internal pull-up resistor may be enabled.

Battery Backup

Because the control store of the Logic Cell Array is a CMOS static memory, its cells require only a very low standby current for data retention. In some systems, this low data retention current characteristic facilitates preserving configurations in the event of a primary power loss. The Logic Cell Array has built in power-down logic which, when activated, will disable normal operation of the device and retain only the configuration data. All internal operation is suspended and all output buffers are placed in their high impedance state.

Power-down data retention is possible with a simple battery-backup circuit because the power requirement is extremely low. For retention at 2.0 V, the required current is typically on the order of 500 nA. Screening to this parameter is available. To force the Logic Cell Array into the power-down state, the user must pull the $\overline{\text{PWRDWN}}$ pin Low and continue to supply a retention voltage to the Vcc pins of the package. When normal power is restored, Vcc is elevated to its normal operating voltage and $\overline{\text{PWRDWN}}$ is returned to a High. The Logic Cell Array resumes operation with the same internal sequence that occurs at the conclusion of configuration. Internal I/O and logic block storage elements will be reset, the outputs will become enabled and then the DONE/$\overline{\text{PROG}}$ pin will be released. No configuration programming is involved.

PERFORMANCE

The high performance of the Logic Cell Array results from its patented architectural features and from the use of an advanced high-speed CMOS manufacturing process. Performance may be measured in terms of minimum propagation times for logic elements.

Flip-flop loop delays for the I/O block and logic block flip-flops are about 3 ns. This short delay provides very good performance under asynchronous clock and data conditions. Short loop delays minimize the probability of a metastable condition which can result from assertion of

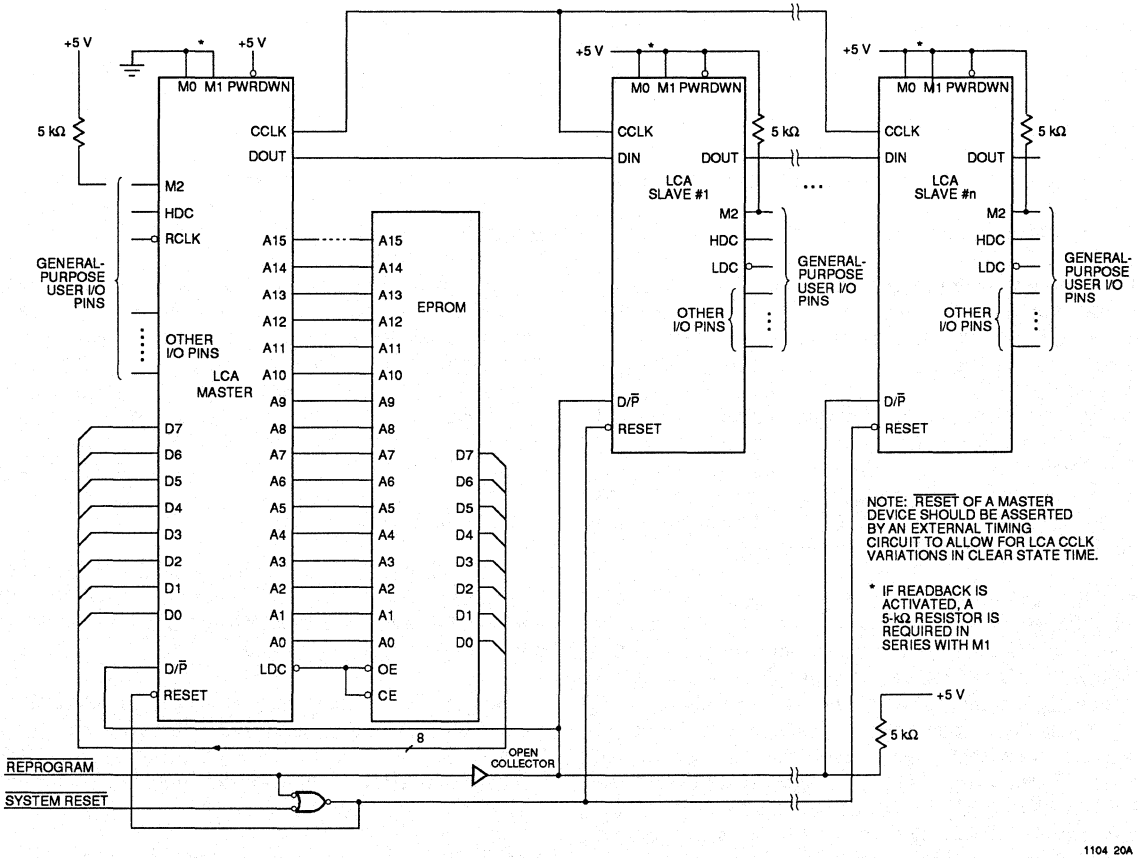


Figure 18. Master Mode Configuration with Daisy Chained Slave Mode Devices.
All are configured from the common EPROM source. A well defined termination of SYSTEM RESET is needed when controlling multiple LCAs.

Any XC3000 slave driven by an XC2000 master mode device must use "early DONE and early internal reset". (The XC2000 master will not supply the extra clock required by a "late" programmed XC3000.)

the clock during data transitions. Because of the short loop delay characteristic in the LCA device, the I/O block flip-flops can be used very effectively to synchronize external signals applied to the device. Once synchronized in the I/O block, 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.

Device Performance

The single parameter which most accurately describes the overall performance of the Logic Cell Array is the maximum toggle rate for a logic block storage element configured as a toggle flip-flop. The configuration for determining the toggle performance of the Logic Cell Array is shown in Figure 19. The clock for the storage element is provided by the global clock buffer and the flip-flop output Q is fed back through the combinatorial logic to form the data input for the next clock edge. Using this arrangement, flip-flops in the Logic Cell Array can be toggled at clock rates from 33–70 MHz, depending on the speed grade used.

Actual Logic Cell Array performance is determined by the critical path speed, including both the speed of the logic and storage elements in that path, and the speed of the particular network routing. Figure 20 shows a typical system logic configuration of two flip-flops with an extra combinatorial level between them. Depending on speed grade, system clock rates to 35 MHz are practical for this logic. To allow the user to make the best use of the capabilities of the device, the delay calculator in the XACT Development System determines worst-case path delays using actual impedance and loading information.

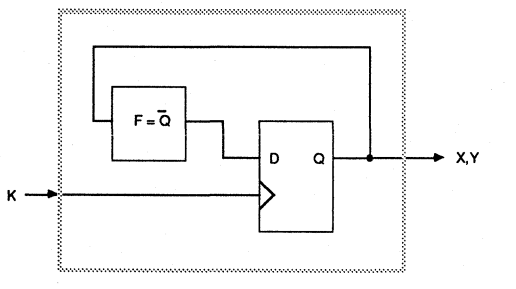


Figure 19. Logic Block Configuration for Toggle Rate Measurement

Logic Block Performance

Logic block propagation times are measured from the interconnect point at the input of the combinatorial logic to the output of the block in the interconnect area. Combinatorial performance is independent of logic function because of the table look-up based implementation. Timing is different when the combinatorial logic is used in conjunction with the storage element. For the combinatorial logic function driving the data input of the storage element, the critical timing is data set-up relative to the clock edge provided to the storage element. The delay from the clock source to the output of the logic block is critical in the timing of signals produced by storage elements. The loading on a logic block output is limited only by the additional propagation delay of the interconnect network. Performance of the logic block is a function of supply voltage and temperature, as shown in Figure 22 .

Interconnect Performance

Interconnect performance depends on the routing resource used to implement the signal path. As discussed earlier, direct interconnect from block to block provides a minimum delay path for a signal.

The single metal segment used for long lines exhibits low resistance from end to end, but relatively high capacitance. Signals driven through a programmable switch will have the additional impedance of the switch added to their normal drive impedance.

General-purpose interconnect performance depends on the number of switches and segments used, the pre-sence of the bidirectional repowering buffers and the overall loading on the signal path at all points along the path. In calculating the worst-case delay for a general interconnect path, the delay calculator portion of the XACT development system accounts for all of these elements. As an approximation, interconnect delay is proportional to the summation of totals of local metal segments beyond each programmable switch. In effect, the delay is a sum of R-C delays each approximated by an R times the total C it drives. The R of the switch and the C of the interconnect are functions of the particular device performance grade. For a string of three local interconnects, the approximate delay at the first segment, after the first switch resistance, would be three units; an additional two delay units after the next switch plus an additional delay after the last switch in the chain. The interconnect R-C chain terminates at each repowering buffer. Nearly all of the capacitance is in the interconnect metal and switches; the capacitance of the block inputs is not significant. Figure 21 shows an estimation of this delay.

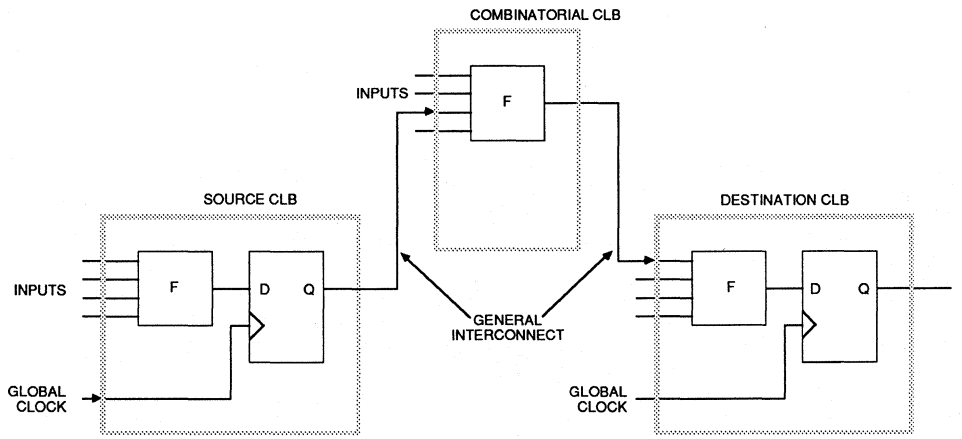


Figure 20. Typical Logic Path

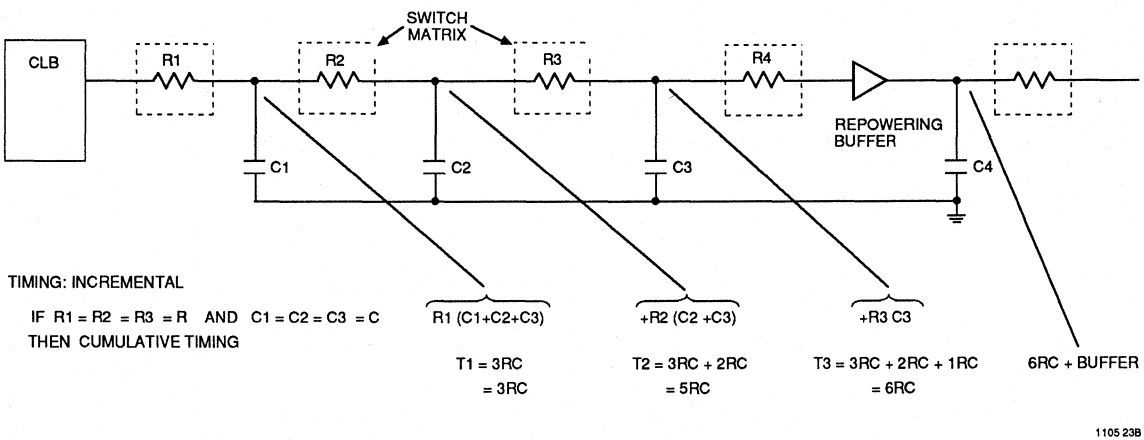


Figure 21. Interconnection Timing Example. Use of the XACT timing calculator or XACT-generated simulation model provides actual worst-case performance information.

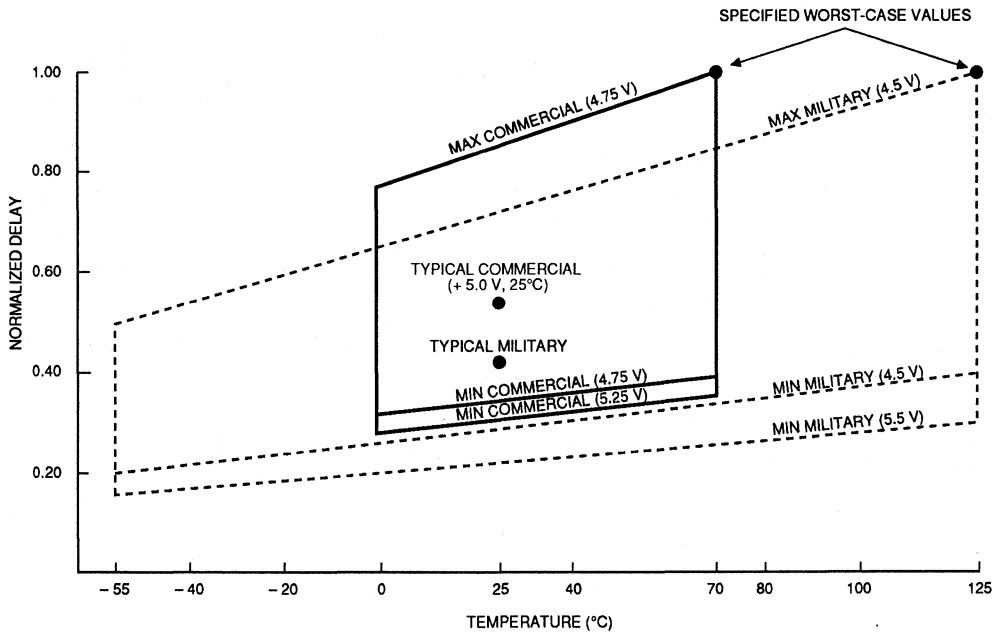


Figure 22. Relative Delay as a function of Temperature, Supply Voltage and Processing Variations.

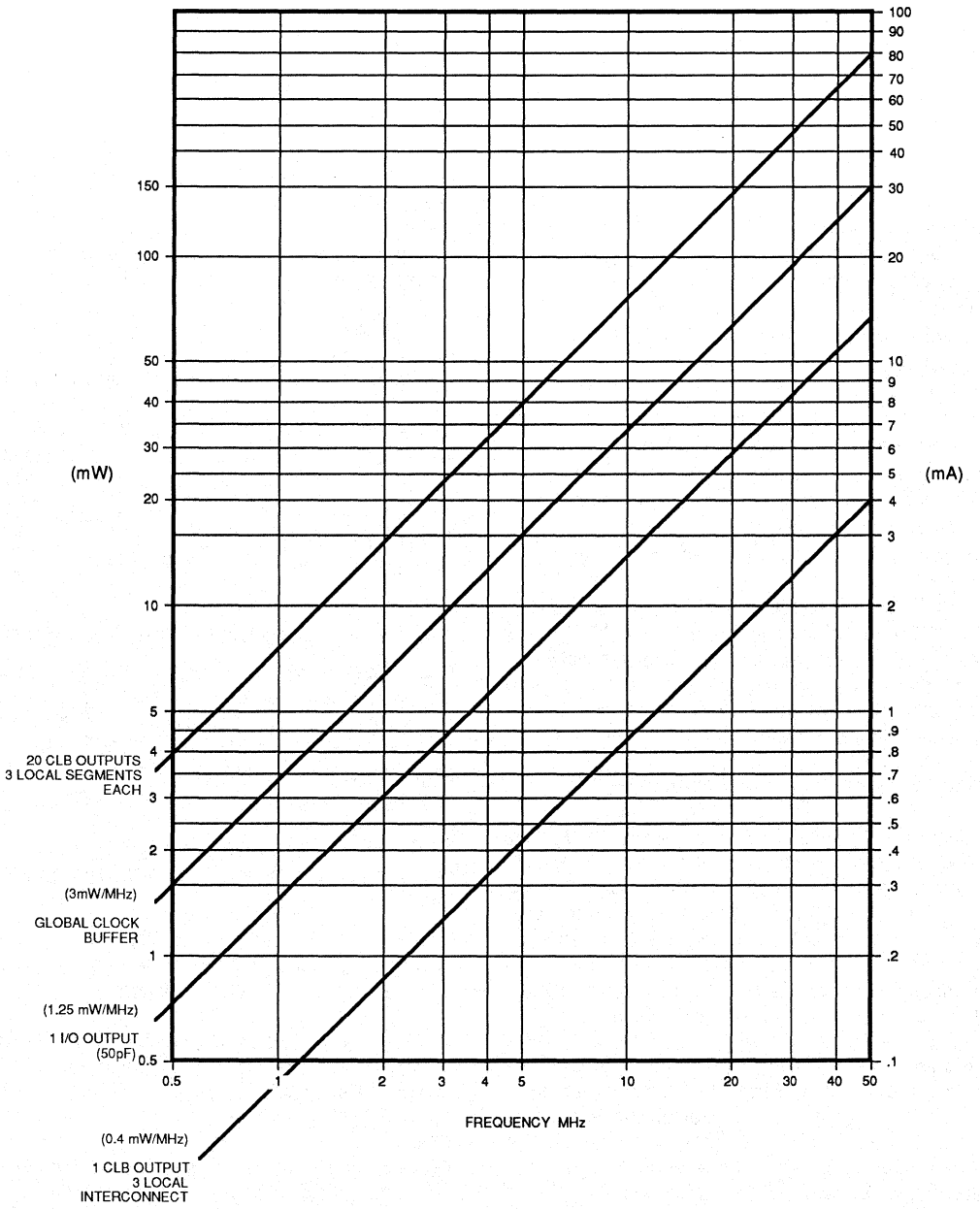


Table 1. Typical LCA Power Consumption By Element

DEVELOPMENT SYSTEMS

To accomplish hardware development support for the Logic Cell Array, Xilinx provides a development system with several options to support added capabilities. The XACT system provides the following:

- Schematic entry
- Automatic place and route
- Interactive design editing for optimization
- Interactive timing calculations
- Macro library support, both for standard Xilinx supplied functions and user defined functions
- Design entry checking for consistency and completeness
- Automatic design documentation generation
- PROM programmer format output capabilities
- Simulation interface support including automatic netlist (circuit description) and timing extraction
- Logic and timing simulation
- In-circuit design verification for multiple devices

The host system on which the XACT system operates is an IBM PC/AT or compatible system with DOS 3.0 or higher. The system requires 640K bytes of internal RAM, 3 Mbyte of Extended Memory, color graphics and a mouse. A complete system requires one parallel I/O port and two serial ports for the mouse and in-circuit emulation.

Designing with the XACT Development System

Designing with the Logic Cell Array is similar to using conventional MSI elements or gate array cells. A range of supported packages, including FutureNet and VIEWlogic, provide schematic capture with elements from a macro library. The XACT development system then translates the schematic description into partitioned Logic Blocks and I/O Blocks, based on shared input variables or efficient use of flip-flop and combinatorial logic. Design entry can also be implemented directly with the XACT development system using an interactive graphic design editor. The design information includes both the functional specifications for each block and a definition of the interconnection networks. Automatic placement and routing is available for either method of design entry. After routing the interconnections, various checking stages and processing of that data are performed to insure that the design is correct. Design changes may be implemented in minutes. The design file is used to generate the programming data which can be down loaded directly into an LCA in the user's target system and operated. The program information may be used to program PROM, EPROM or ROM devices, or stored in some other media as needed by the final system. Design verification may be accomplished by using the XACTOR In-Circuit Design Verifier directly in the target system and/or the P-SILOS logic simulator.

PIN DESCRIPTIONS

Permanently Dedicated Pins.

V_{CC}
One or two (depending on package type) connections to the nominal +5 V supply voltage. All must be connected.

GND
One or two (depending on package type) connections to ground. All must be connected.

$\overline{PWRDWN}$
A Low on this CMOS-compatible input stops all internal activity, but retains configuration. All flip-flops and latches are reset, all outputs are 3-stated, and all inputs are interpreted as High, independent of their actual level. While $\overline{PWRDWN}$ is Low, V_{CC} may be reduced to any value >2.3 V. When $\overline{PWDWN}$ returns High, the LCA becomes operational with DONE Low for two cycles of the internal 1-MHz clock. During configuration, $\overline{PWRDWN}$ must be High. If not used, $\overline{PWRDWN}$ must be tied to V_{CC}.

$\overline{RESET}$
This is an active Low input which has three functions.

Prior to the start of configuration, a Low input will delay the start of the configuration process. An internal circuit senses the application of power and begins a minimal time-out cycle. When the time-out and $\overline{RESET}$ are complete, the levels of the M lines are sampled and configuration begins.

If $\overline{RESET}$ is asserted during a configuration, the LCA device is re-initialized and restarts the configuration at the termination of $\overline{RESET}$.

If $\overline{RESET}$ is asserted after configuration is complete, it provides a global asynchronous reset of all IOB and CLB storage elements of the LCA device.

$\overline{RESET}$ can also be used to recover from partial power failure. See section on Re-program under "Special Configuration Functions."

CCLK
During configuration, Configuration Clock is an output of an LCA in Master mode or Peripheral mode, but an input in Slave mode. During a Readback, CCLK is a clock input for shifting configuration data out of the LCA

CCLK drives dynamic circuitry inside the LCA. The Low time may, therefore, not exceed a few microseconds. When used as an input, CCLK must be "parked High". An internal pull-up resistor maintains High when the pin is not being driven.

DONE
DONE is an open-drain output, configurable with or without an internal pull-up resistor. At the completion of configuration, the LCA circuitry becomes active in a synchronous order, and DONE is programmed to go active High either one cycle before or after the outputs go active.

$\overline{\text{PROG}}$
Once configuration is done, a High-to-Low transition of this pin will cause an initialization of the LCA and start a reconfiguration.

M0
As Mode 0, this input and M1, M2 are sampled before the start of configuration to establish the configuration mode to be used.

RTRIG
A Low-to-High input transition, after configuration is complete, acts as a Read Trigger and initiates a Readback of configuration and storage-element data clocked by CCLK. By selecting the appropriate Readback option when generating the bitstream, this operation may be limited to a single Readback, or be inhibited altogether.

M1
As Mode 1, this input and M0, M2 are sampled before the start of configuration to establish the configuration mode to be used. If Readback is never used, M1 can be tied directly to ground or V_{CC}. If Readback is ever used, M1 must use a 5-kΩ resistor to ground or V_{CC}, to accommodate the $\overline{\text{RDATA}}$ output.

$\overline{\text{RDATA}}$
As an active Low Read Data, after configuration is complete, this pin is the output of the Readback data.

User I/O Pins that can have special functions.

M2
During configuration, this input has a weak pull-up resistor. Together with M0 and M1, it is sampled before the start of configuration to establish the configuration mode to be used. After configuration, this pin is a user-programmable I/O pin.

HDC
During configuration, this output is held at a High level to indicate that configuration is not yet complete. After configuration, this pin is a user-programmable I/O pin.

$\overline{\text{LDC}}$
During Configuration, this output is held at a Low level to indicate that the configuration is not yet complete. After configuration, this pin is a user-programmable I/O pin. $\overline{\text{LDC}}$ is particularly useful in Master mode as a Low enable for an EPROM, but it must then be programmed as a High after configuration.

XTL1
This user I/O pin can be used to operate as the output of an amplifier driving an external crystal and bias circuitry.

XTL2
This user I/O pin can be used as the input of an amplifier connected to an external crystal and bias circuitry. The I/O Block is left unconfigured. The oscillator configuration is activated by routing a net from the oscillator buffer symbol output and by the MAKEBITS program.

$\overline{\text{CS0}}$, $\overline{\text{CS1}}$, CS2, $\overline{\text{WRT}}$
These four inputs represent a set of signals, three active Low and one active High, that are used to control configuration-data entry in the Peripheral mode. Simultaneous assertion of all four inputs generates a Write to the internal data buffer. The removal of any assertion clocks in the D0-D7 data. In Master mode,these pins become part of the parallel configuration byte, D4, D3, D2, D1. After configuration, these pins are user-programmable I/O pins.

$\overline{\text{RCLK}}$
During Master parallel mode configuration $\overline{\text{RCLK}}$ represents a "read" of an external dynamic memory device (normally not used).

D0–D7
This set of eight pins represents the parallel configuration byte for the parallel Master mode. After configuration is complete they are user programmed I/O pins.

A0–A15
During Master Parallel mode, these 16 pins present an address output for a configuration EPROM. After configuration, they are user-programmable I/O pins.

DIN
During Slave or Master Serial configuration, this pin is used as a serial-data input. In the Master or Peripheral configuration, this is the Data 0 input.

DOUT
During configuration this pin is used to output serial-configuration data to the DIN pin of a daisy-chained slave.

TCLKIN
This is a direct CMOS level input to the global clock buffer.

Unrestricted User I/O Pins.
I/O
An I/O pin may be programmed by the user to be an Input or an Output pin following configuration. All unrestricted I/O pins, plus the special pins mentioned on the following page, have a weak pull-up resistor of 40 to 100 kΩ that becomes active as soon as the device powers up, and stays active until the end of configuration.

CONFIGURATION MODE: <M2:M1:M0>					48 DIP	68 PLCC	68 PGA	USER OPERATION
MASTER-SER <0:0:0>	SLAVE <1:1:1>	PERIPHERAL <1:0:1>	MASTER-HIGH <1:1:0>	MASTER-LOW <1:0:0>				
GND						1	B6	GND
<<HIGH>>			A13 (O)			2	A6	I/O
			A6 (O)		1	3	B5	
			A12 (O)			4	A5	
			A7 (O)		2	5	B4	
			A11 (O)		3	6	A4	
			A8 (O)		4	7	B3	
			A10 (O)		5	8	A3	
			A9 (O)		6	9	A2	
PWRDWN (I)					7	10	B2	PWR DWN
<<HIGH>>					8	11	B1	I/O
						12	C2	
					9	13	C1	
						14	D2	
					10	15	D1	
						16	E2	
					11	17	E1	
VCC					12	18	F2	VCC
<<HIGH>>					13	19	F1	I/O
						20	G2	
					14	21	G1	
						22	H2	
					15	23	H1	
					16	24	J2	
M1 (LOW)	M1 (HIGH)	M1 (LOW)	M1 (HIGH)	M1 (LOW)	17	25	J1	RDATA (O)
M0 (LOW)	M0 (HIGH)	M0 (HIGH)	M0 (LOW)	M0 (LOW)	18	26	K1	RTRIG (I)
M2 (LOW)	M2 (HIGH)				19	27	K2	I/O
HDC (HIGH)					20	28	L2	
<<HIGH>>						29	K3	
LDC (LOW)					21	30	L3	
<<HIGH>>						31	K4	
					22	32	L4	
						33	K5	
					23	34	L5	
GND					24	35	K6	GND
<<HIGH>>						36	L6	I/O
					25	37	K7	
						38	L7	
					26	39	K8	
					27	40	L8	
			D7 (I)		28	41	K9	
			D6 (I)		29	42	L9	
					30	43	L10	XTL2 OR I/O
RESET (I)					31	44	K10	RESET
DONE (O)					32	45	K11	PROG (I)
<<HIGH>>					33	46	J10	XTL1 OR I/O
						47	J11	I/O
			D5 (I)		34	48	H10	
						49	H11	
		CSO (I)	D4 (I)		35	50	G10	
		CS1 (I)	D3 (I)		36	51	G11	
VCC						52	F10	VCC
						53	F11	I/O
<<HIGH>>		CS2 (I)	D2 (I)		37	54	E10	
						55	E11	
		WRT (I)	D1 (I)		38	56	D10	
RCLK			RCLK		39	57	D11	
DIN (I)			D0 (I)		40	58	C10	
DOUT (O)					41	59	C11	
CCLK (O)	CCLK (I)	CCLK (O)			42	60	B11	CCLK (I)
<<HIGH>>			A0 (O)		43	61	B10	I/O
			A1 (O)		44	62	A10	
			A2 (O)		45	63	B9	
			A3 (O)		46	64	A9	
			A15 (O)			65	B8	
			A4 (O)		47	66	A8	
			A14 (O)			67	B7	
			A5 (O)		48	68	A7	

<<HIGH>> IS HIGH IMPEDANCE WITH A 20–50 kΩ INTERNAL PULL-UP DURING CONFIGURATION

1104 28A

Table 2a. XC2064 Pin Assignments

A PLCC in a "PGA-Footprint" socket has a **different** signal pinout than a PGA device.

CONFIGURATION MODE: <M2:M1:M0>					68 PLCC	84 PLCC	84 PGA	USER OPERATION
MASTER-SER <0:0:0>	SLAVE <1:1:1>	PERIPHERAL <1:0:1>	MASTER-HIGH <1:1:0>	MASTER-LOW <1:0:0>				
GND					1	1	C6	GND
<<HIGH>>			A13 (O)		2	2	A6	I/O
						3	A5	
						4	B5	
			A6 (O)		3	5	C5	
			A12 (O)		4	6	A4	
			A7 (O)		5	7	B4	
			A11 (O)		6	8	A3	
			A8 (O)		7	9	A2	
			A10 (O)		8	10	B3	
			A9 (O)		9	11	A1	
PWRDWN (I)					10	12	B2	PWR DWN
<<HIGH>>					11	13	C2	I/O
					12	14	B1	
					13	15	C1	
					14	16	D2	
					15	17	D1	
						18	E3	
					16	19	E2	
						20	E1	
					17	21	F2	
VCC					18	22	F3	VCC
<<HIGH>>					19	23	G3	I/O
						24	G1	
					20	25	G2	
						26	F1	
					21	27	H1	
					22	28	H2	
					23	29	J1	
					24	30	K1	
M1 (LOW)	M1 (HIGH)	M1 (LOW)	M1 (HIGH)	M1 (LOW)	25	31	J2	ADATA (O)
M0	M0 (HIGH)	M0 (HIGH)	M0 (LOW)	M0 (LOW)	26	32	L1	RTRIG (I)
M2 (LOW)	M2 (HIGH)				27	33	K2	I/O
HDC (HIGH)					28	34	K3	
<<HIGH>>					29	35	L2	
LDC (LOW)					30	36	L3	
<<HIGH>>					31	37	K4	
					32	38	L4	
						39	J5	
					33	40	K5	
					34	41	L5	
						42	K6	
GND					35	43	J6	GND
<<HIGH>>						44	J7	I/O
					36	45	L7	
					37	46	K7	
					38	47	L6	
						48	L8	
					39	49	K8	
					40	50	L9	
			D7 (I)		41	51	L10	
			D6 (I)		42	52	K9	
					43	53	L11	XTL2 OR I/O
RESET (I)					44	54	K10	RESET
DONE (O)					45	55	J10	PROG (I)
<<HIGH>>					46	56	K11	XTL1 OR I/O
					47	57	J11	I/O
			D5 (I)		48	58	H10	
						59	H11	
					49	60	F10	
						61	G10	
		CSO (I)	D4 (I)		50	62	G11	
		CS1 (I)	D3 (I)		51	63	G9	
VCC					52	64	F9	VCC
<<HIGH>>					53	65	F11	I/O
		CS2 (I)	D2 (I)		54	66	E11	
						67	E10	
					55	68	E9	
						69	D11	
		WRT (I)	D1 (I)		56	70	D10	
RCLK			RCLK		57	71	C11	
DIN (I)			D0 (I)		58	72	B11	
DOUT (O)					59	73	C10	
CCLK (O)	CCLK (I)	CCLK (O)			60	74	A11	CCLK (I)
<<HIGH>>			A0 (O)		61	75	B10	I/O
			A1 (O)		62	76	B9	
			A2 (O)		63	77	A10	
			A3 (O)		64	78	A9	
			A15 (O)		65	79	B8	
			A4 (O)		66	80	A8	
			A14 (O)		67	81	B6	
						82	B7	
						83	A7	
			A5 (O)		68	84	C7	

<<HIGH>> IS HIGH IMPEDANCE WITH A 20–50 kΩ INTERNAL PULL-UP DURING CONFIGURATION

1104 29A

Table 2b. XC2018 Pin Assignments

Xilinx maintains test specifications for each product as controlled documents. To insure the use of the most recently released device performance parameters, please request a copy of the current test-specification revision.

ABSOLUTE MAXIMUM RATINGS

Symbol	Description		Units
V_{CC}	Supply voltage relative to GND	–0.5 to +7.0	V
V_{IN}	Input voltage with respect to GND	–0.5 to V_{CC} +0.5	V
V_{TS}	Voltage applied to 3-state output	–0.5 to V_{CC} +0.5	V
T_{STG}	Storage temperature (ambient)	–65 to +150	°C
T_{SOL}	Maximum soldering temperature (10 s @ 1/16 in.)	+260	°C

Note: Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those listed under Recommended Operating Conditions is not implied. Exposure to Absolute Maximum Ratings conditions for extended periods of time may affect device reliability.

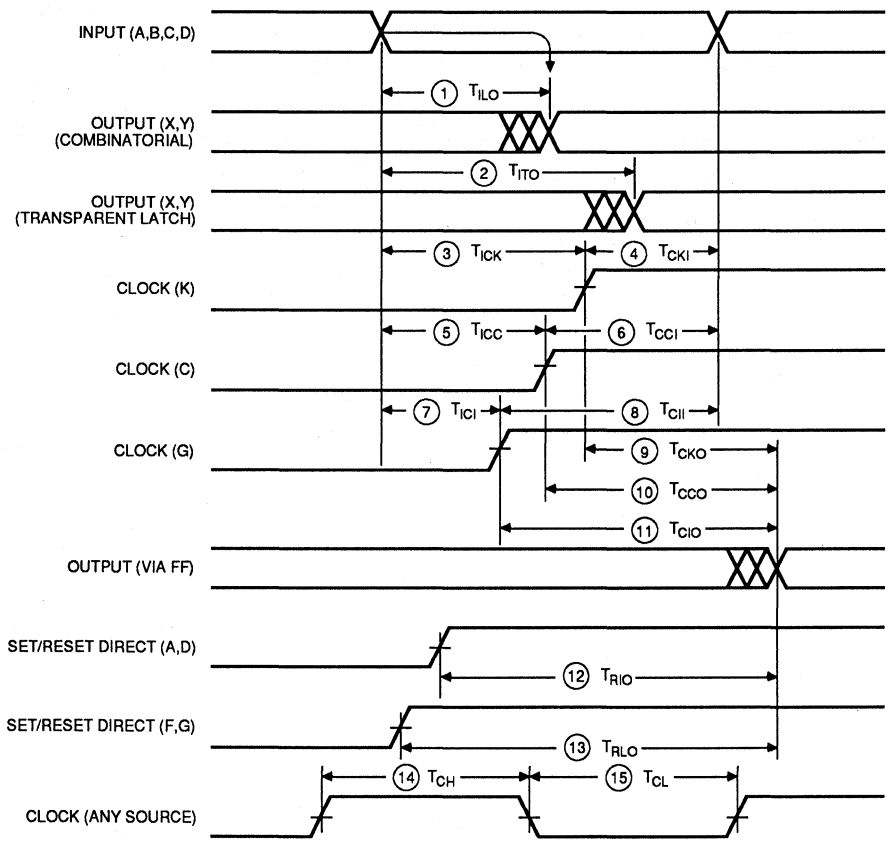
OPERATING CONDITIONS

Symbol	Description	Min	Max	Units
V_{CC}	Supply voltage relative to GND Commercial 0°C to +70°C	4.75	5.25	V
	Supply voltage relative to GND Industrial –40°C to +85°C	4.5	5.5	V
	Supply voltage relative to GND Military –55°C to +125°C	4.5	5.5	V
V_{IHT}	High-level input voltage — TTL configuration	2.0	Vcc	V
V_{ILT}	Low-level input voltage — TTL configuration	0	0.8	V
V_{IHC}	High-level input voltage — CMOS configuration	70%	100%	V_{CC}
V_{ILC}	Low-level input voltage — CMOS configuration	0	20%	V_{CC}
T_{IN}	Input signal transition time		250	ns

DC CHARACTERISTICS OVER OPERATING CONDITIONS

Symbol	Description		Min	Max	Units
V_{OH}	High-level output voltage (@ I_{OH} = –4.0 ma V_{CC} min)	Commercial	3.86		V
V_{OL}	Low-level output voltage (@ I_{OL} = 4.0 ma V_{CC} max)			0.32	V
V_{OH}	High-level output voltage (@ I_{OH} = –4.0 ma V_{CC} min)	Industrial Military	3.76		V
V_{OL}	Low-level output voltage (@ I_{OL} = 4.0 ma V_{CC} max)			0.37	V
V_{CCPD}	Power-down supply voltage ($\overline{PWRDWN}$ must be Low)		2.3		V
I_{CCO}	Quiescent operating power supply current				
	CMOS thresholds (@ V_{CC} Max)			5	mA
	TTL thresholds (@ V_{CC} Max)			12	mA
I_{CCPD}	Power-down supply current ($V_{CC(MAX)}$ @ T_{MAX})			500	µA
I_{IL}	Input Leakage Current		–10	+10	µA
C_{IN}	Input capacitance (sample tested) All Pins except XTL1 and XTL2 XTL1 and XTL2			10 15	pF pF

CLB SWITCHING CHARACTERISTIC GUIDELINES



1104 30

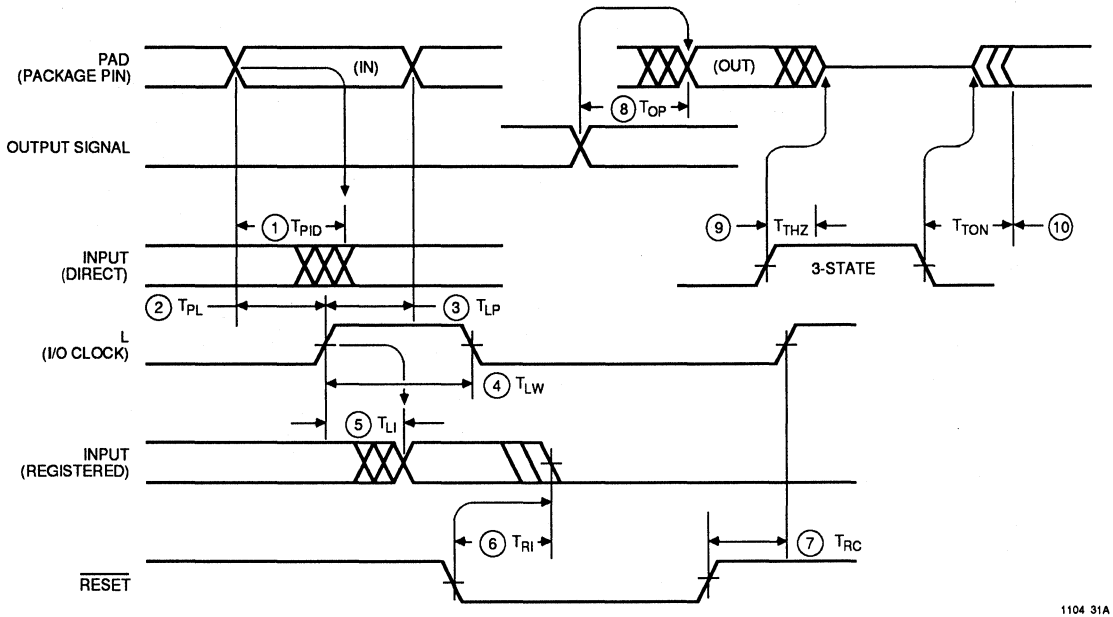
CLB SWITCHING CHARACTERISTIC GUIDELINES (Continued)

Speed Grade				-50		-70		-100		Units
	Description	Symbol		Min	Max	Min	Max	Min	Max	
Logic Input to Output	Combinatorial	1	T_{ILO}		15		10		7.5	ns
	Transparent latch	2	T_{ITO}		20		14		10	ns
	Additional for Q through F or G to out		T_{QLO}		8		6		6	ns
K Clock	To output	9	T_{CKO}		15		10		7	ns
	Logic-input setup	3	T_{ICK}	9		7		6		ns
	Logic-input hold	4	T_{CKI}	0		0		0		ns
C Clock	To output	10	T_{CCO}		19		13	9		ns
	Logic-input setup	5	T_{ICC}	8		6		5		ns
	Logic-input hold	6	T_{CCI}	0		0		0		ns
Logic Input to G Clock	To output	11	T_{CIO}		27		20		13	ns
	Logic-input setup	7	T_{ICI}	4		3		2		ns
	Logic-input hold	8	T_{CII}	5		4		3		ns
Set/Reset direct	Input A or D to output x, y	12	T_{RIO}		22		16		10	ns
	Through F or G to output	13	T_{RLO}		28		21		14	ns
	Reset pad to output x, y		T_{MRQ}		25		20		17	ns
	Separation of set/reset		T_{RS}	9		7		6		ns
	Set/Reset pulse-width		T_{RPW}	9		7		6		ns
Flip-flop Toggle rate	Q through F to flip-flop		F_{CLK}	50		70		100*		MHz
Clock	Clock High	14	T_{CH}	8		7		5*		ns
	Clock Low	15	T_{CL}	8		7		5*		ns

Notes: 1. All switching characteristics apply to all valid combinations of process, temperature and supply with a nominal chip power dissipation of 250 mW.

* These parameters are for clock pulses generated within a CLB. For an externally generated pulse, derate these parameters by 20%.

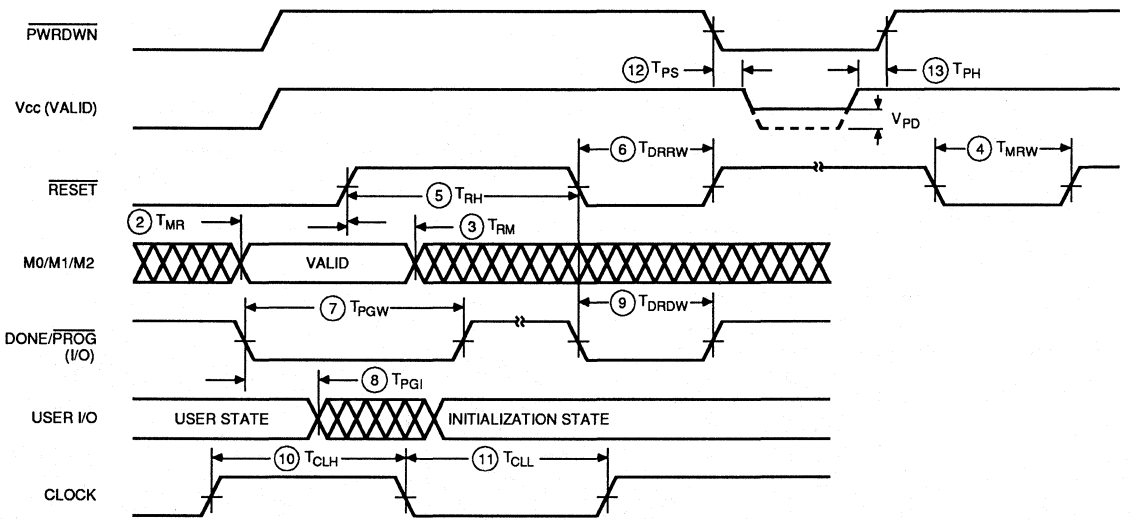
IOB SWITCHING GUIDELINES



		Speed Grade		-50		-70		-100		Units
	Description	Symbol		Min	Max	Min	Max	Min	Max	
Pad (package pin)	To input (direct)	1	T_{PID}		8		6		4	ns
I/O Clock	To input (storage)	5	T_{LI}		15		11		8	ns
	To pad-input setup	2	T_{PL}	8		6		4		ns
	To pad-input hold	3	T_{LP}	0		0		0		ns
	Pulse width	4	T_{LW}		9		7		5*	ns
	Frequency			50		70		100*		MHz
Output	To pad (output enabled)	8	T_{OP}		12		9		7	ns
Three-state	To pad begin hi-Z	9	T_{THZ}		20		15		11	ns
	To pad end hi-Z	10	T_{TON}		20		15		13	ns
$\overline{RESET}$	To input (storage)	6	T_{RI}		30		25		17	ns
	To input clock	7	T_{RC}	25		20		14		ns

Note: Timing is measured at 0.5 Vcc levels with 50 pF output load.
*These parameters are for clock pulses generated within an LCA. For an externally applied clock, derate these parameters by 20%.

GENERAL LCA SWITCHING CHARACTERISTIC



1104 32

	Speed Grade			-50		-70		-100		Units
	Description	Symbol		Min	Max	Min	Max	Min	Max	
RESET[2]	M2, M1, M0 setup	2	T_{MR}	60		60		60		ns
	M2, M1, M0 hold	3	T_{RM}	60		60		60		ns
	Width—FF Reset	4	T_{MRW}	150		150		150		ns
	High before RESET[4]	5	T_{RH}	6		6		6		µs
	Device Reset[5]	6	T_{DRRW}	6		6		6		µs
DONE/PROG	Progam width (Low)	7	T_{PGW}	6		6		6		µs
	Initialization	8	T_{PGI}		7		7		7	µs
	Device Reset [4]	9	T_{DRDW}	6		6		6		µs
CLOCK	Clock (High)	10	T_{CLH}	8		7		5		ns
	Clock (Low)	11	T_{CLL}	8		7		5		ns

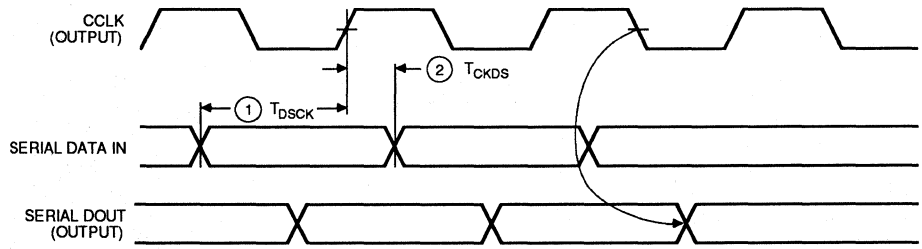
Notes: 1. At power-up, Vcc must rise from 2.0 Volts to Vcc min in less than 25 ms. If this is not possible, configuration can be delayed by holding RESET Low until Vcc has reached 4.0 V. A very long Vcc rise time of >100 ms, or a non-monotonically rising Vcc may require a RESET pulse (High-to-Low-to-High) of >6 µs duration **after** Vcc has reached 4.0 V.

2. RESET timing relative to power-on and valid mode lines (M0, M1, M2) is relevant only when RESET is used to delay configuration.

3. Minimum CLOCK widths for the auxillary buffer are 1.25 times the TCLH, TCLL.

4. After RESET is High, RESET = D/P = Low for 6 µs will abort to CLEAR.

MASTER SERIAL MODE PROGRAMMING SWITCHING CHARACTERISTICS



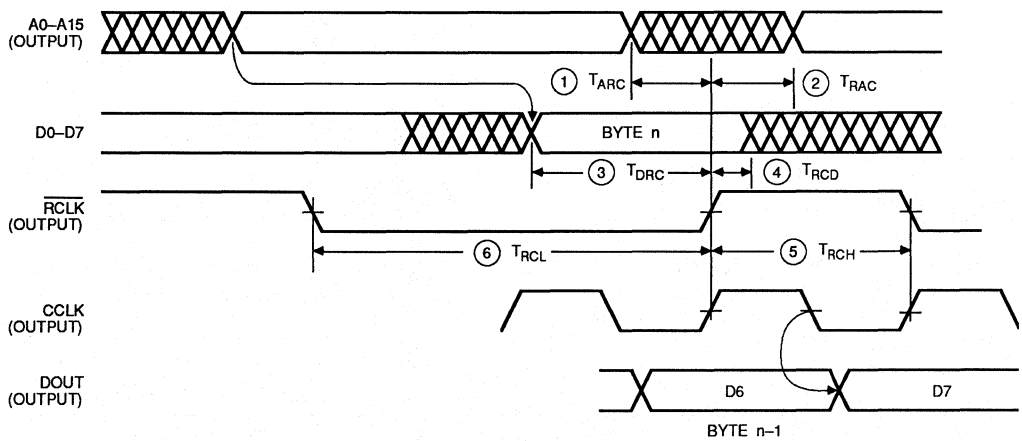
1105 29

Speed Grade				-50		-70		-100		Units
	Description	Symbol		Min	Max	Min	Max	Min	Max	
CCLK[2]	Data In setup	1	T_{DSCK}	60		60		60		ns
	Data In hold	2	T_{CKDS}	0		0		0		ns

Notes: 1. At power-up, Vcc must rise from 2.0 Volts to Vcc min in less than 25 ms. If this is not possible, configuration can be delayed by holding $\overline{\text{RESET}}$ Low until Vcc has reached 4.0 V. A very long Vcc rise time of >100 ms, or a non-monotonically rising Vcc may require a $\overline{\text{RESET}}$ pulse (High-to-Low-to-High) of >6 µs duration **after** Vcc has reached 4.0 V.

2. Master-serial-mode timing is based on slave-mode testing.

MASTER PARALLEL MODE PROGRAMMING SWITCHING CHARACTERISTICS



1104 33

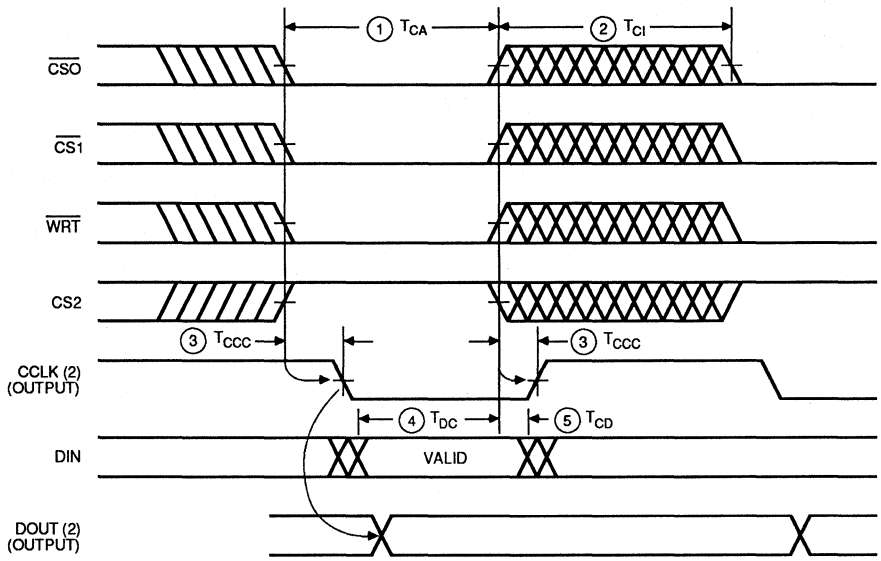
	Speed Grade			-50		-70		-100		Units
	Description	Symbol			Min	Max	Min	Max	Min	Max
RCLK	From address invalid	1	T_{ARC}		0		0		0	ns
	To address valid	2	T_{RAC}		200		200		200	ns
	To data setup	3	T_{DRC}	60		60		60		ns
	To data hold	4	T_{RCD}	0		0		0		ns
	$\overline{RCLK}$ high	5	T_{RCH}	600		600		600		ns
	$\overline{RCLK}$ low	6	T_{RCL}	4.0		4.0				µs

Note: 1. CCLK and DOUT timing are the same as for slave mode.

2. At power-up, Vcc must rise from 2.0 Volts to Vcc min in less than 25 ms. If this is not possible, configuration can be delayed by holding $\overline{RESET}$ Low until Vcc has reached 4.0 V. A very long Vcc rise time of >100 ms, or a non-monotonically rising Vcc may require a $\overline{RESET}$ pulse (High-to-Low-to-High) of >6 µs duration **after** Vcc has reached 4.0 V.

This timing diagram shows that the EPROM requirements are extremely relaxed: EPROM access time can be longer than 4000 ns, EPROM data output has no hold time requirement

PERIPHERAL MODE PROGRAMMING SWITCHING CHARACTERISTICS



1104 34

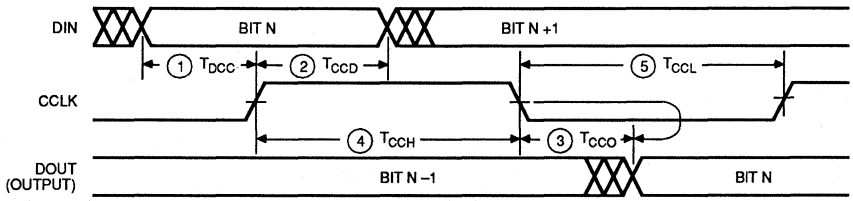
	Speed Grade			-50		-70		-100		Units
	Description	Symbol			Min	Max	Min	Max	Min	Max
Controls[1] (CS0, CS1, CS2, WRT)	Active (last active input to first inactive)	1	T_{CA}	0.25	5.0	0.25	5.0	0.25	5.0	µs
	Inactive (first inactive input to last active)	2	T_{CI}	0.25		0.25		0.25		µs
	CCLK[2]	3	T_{CCC}		75		75		75	ns
	DIN setup	4	T_{DC}	50		50		50		ns
	DIN hold	5	T_{CD}	0		0		0		ns

Notes: 1. Peripheral mode timing determined from last control signal of the logical AND of (CS0, CS1, CS2, WRT) to transition to active or inactive state.

2. CCLK and DOUT timing are the same as for slave mode.

3. At power-up, Vcc must rise from 2.0 Volts to Vcc min in less than 25 ms. If this is not possible, configuration can be delayed by holding RESET Low until Vcc has reached 4.0 V. A very long Vcc rise time of >100 ms, or a non-monotonically rising Vcc may require a RESET pulse (High-to-Low-to-High) of >6 µs duration **after** Vcc has reached 4.0 V.

SLAVE MODE PROGRAMMING SWITCHING CHARACTERISTICS

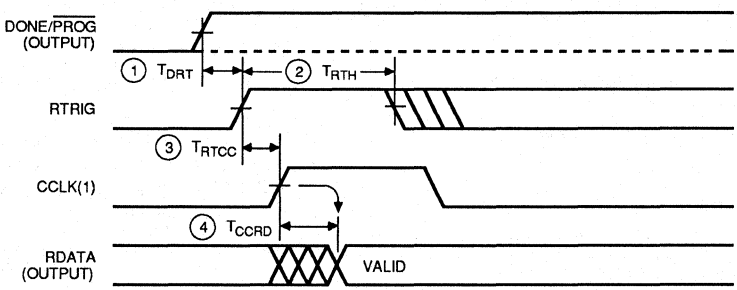


1104 35

	Speed Grade			-50		-70		-100		Units
	Description	Symbol		Min	Max	Min	Max	Min	Max	
CCLK	To DOUT	3	T_{CCO}		65		65		65	ns
	DIN setup	1	T_{DCC}	10		10		10		ns
	DIN hold	2	T_{CCD}	40		40		40		ns
	High time	4	T_{CCH}	0.25		0.25		0.25		µs
	Low time	5	T_{CCL}	0.25	5.0	0.25	5.0	0.25	5.0	µs
	Frequency		F_{CC}		2		2		2	MHz

Note: At power-up, Vcc must rise from 2.0 Volts to Vcc min in less than 25 ms. If this is not possible, configuration can be delayed by holding RESET Low until Vcc has reached 4.0 V. A very long Vcc rise time of >100 ms, or a non-monotonically rising Vcc may require a RESET pulse (High-to-Low-to-High) of >6 µs duration after Vcc has reached 4.0 V.

PROGRAM READBACK SWITCHING CHARACTERISTICS



1104 36

	Speed Grade			-50		-70		-100		Units
	Description	Symbol		Min	Max	Min	Max	Min	Max	
RTRIG	PROG setup	1	T_{DRT}	300		300		300		ns
	RTRIG high	2	T_{RTH}	250		250		250		ns
CCLK	RTRIG setup	3	T_{RTCC}	100		100		100		ns
	RDATA delay	4	T_{CCRD}		100		100		100	ns

Notes: 1. CCLK and DOUT timing are the same as for slave mode.
2. DONE/PROG output/input must be HIGH (device programmed) prior to a positive transition of RTRIG (M0).

Component Selection, Ordering Information, & Physical Dimensions

COMPONENT AVAILABILITY (11/90)

		44 PIN	48 PIN		68 PIN		84 PIN		100 PIN		132 PIN		160 PIN	164 PIN	175 PIN	
		PLASTIC PLCC	PLASTIC DIP	CERAMIC DIP	PLASTIC PLCC	CERAMIC PGA	PLASTIC PCC	CERAMIC PGA	PLASTIC PQFP	CERAMIC CQFP	PLASTIC PGA	CERAMIC PGA	PLASTIC PQFP	CERAMIC CQFP	PLASTIC PGA	CERAMIC PGA
		PC44	PD48	CD48	PC68	PG68	PC84	PG84	PQ100	CQ100	PP132	PG132	PQ160	CQ164	PP175	PG175
XC2064	-50		C	I	CI	CIM										
	-70				CI	CIM										
	-100				C	C										
XC2018	-33							MB								
	-50				CI		CI	CIMB								
	-70				CI		CI	CIMB								
	-100				C		C	C								
XC3020	-50				CI		CI	CIMB	CI	CIMB						
	-70				CI		CI	CIMB	CI	CIMB						
	-100				C		C	C	C	C						
XC3030	-50	CI			CI		CI	CIM	C							
	-70	CI			CI		CI	CIM	CI							
	-100	C			C		C	C	C							
XC3042	-50						CI	CIMB	C	CIMB	CI	CIMB				
	-70						CI	CIMB	CI	CIMB	CI	CI				
	-100						C	C	C	C	C	C				
XC3064	-50						C				CI	CIM				
	-70						C				CI	CIM				
	-100						C				C	C				
XC3090	-50						C						CI*	CIMB	CI	CIMB
	-70						C						CI*	CIMB	CI	CIMB
	-100						C						C*	C	C	C

X1104

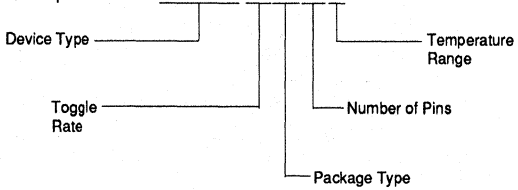
XC1736A/XC1765-PD8C Plastic 8-Pin Mini-DIP
–40°C to 85°C
XC1736A/XC1765-CD8M Ceramic 8-Pin Mini-DIP
–55°C to 125°C

LCA Temperature Options

Symbol	Description	Temperature
C	Commercial	0°C to 70°C
I	Industrial	–40°C to 85°C
M	Mil Temp	–55°C to 125°C
B	Military	MIL-STD-883, Class B

ORDERING INFORMATION

Example: XC2064-70PC68C



COMPATIBLE PACKAGE OPTIONS

A range of LCA devices is available in identical packages with identical pin-outs. A design can thus be started with one device, then migrated to a larger or smaller chip while retaining the original footprint and PC-board layout.

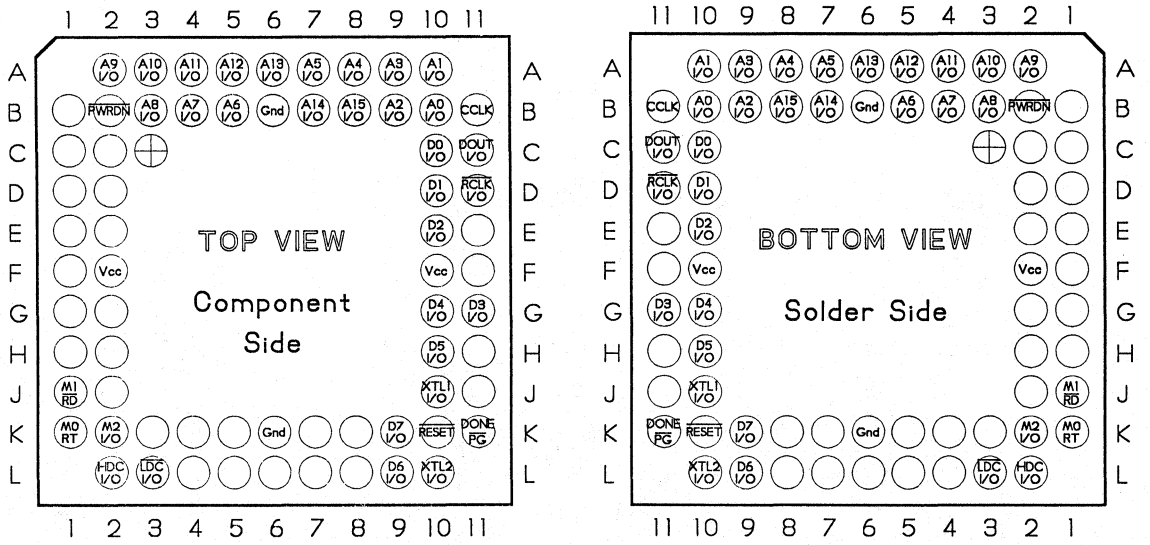
Examples:
PC 68: 2064-2018-3020-3030
PC 84: 2018-3020-3030-3042-3064-3090
PG 84: 2018-3020-3030-3042
PQ 100: 3020-3030-3042
PG 132: 3042-3064

Note, however, that the XC2000 and XC3000 families differ in the position of XTL1 as well as three parallel address bits (6, 7 and 11) and most of the data pins used in parallel master mode.

XC2018 and XC3020 are not available in PGA68, since the PGA84 is the same size and offers more I/O.

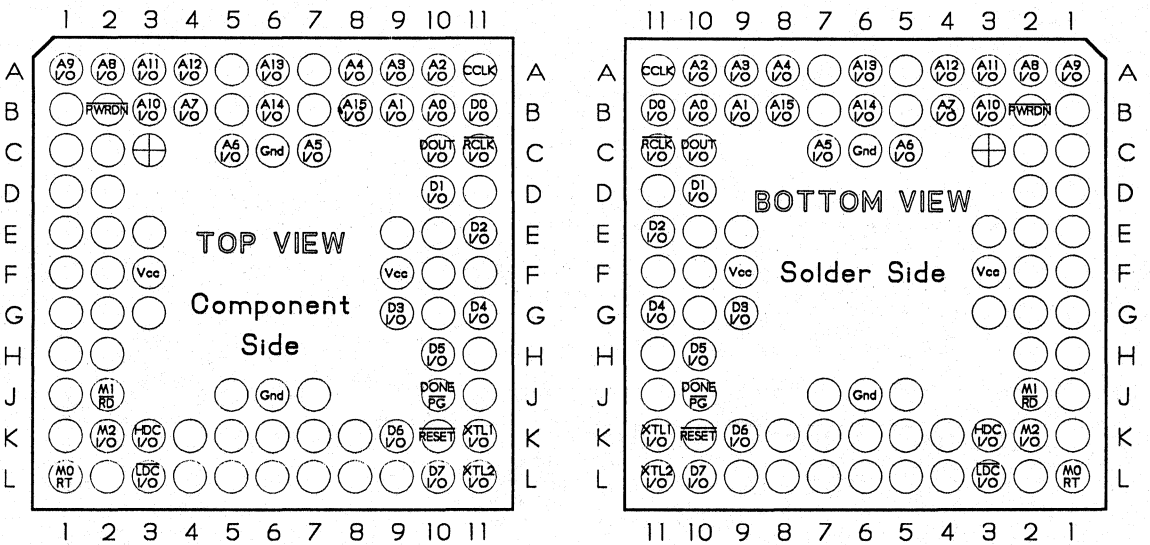
Note that a PLCC in a socket with PGA footprint generates a printed circuit board pin-out **different** from a PGA device.

PGA PIN-OUTS



⊕ = Index pin which may or may not be electrically connected to pin C2
unlabeled pin = unrestricted I/O pin

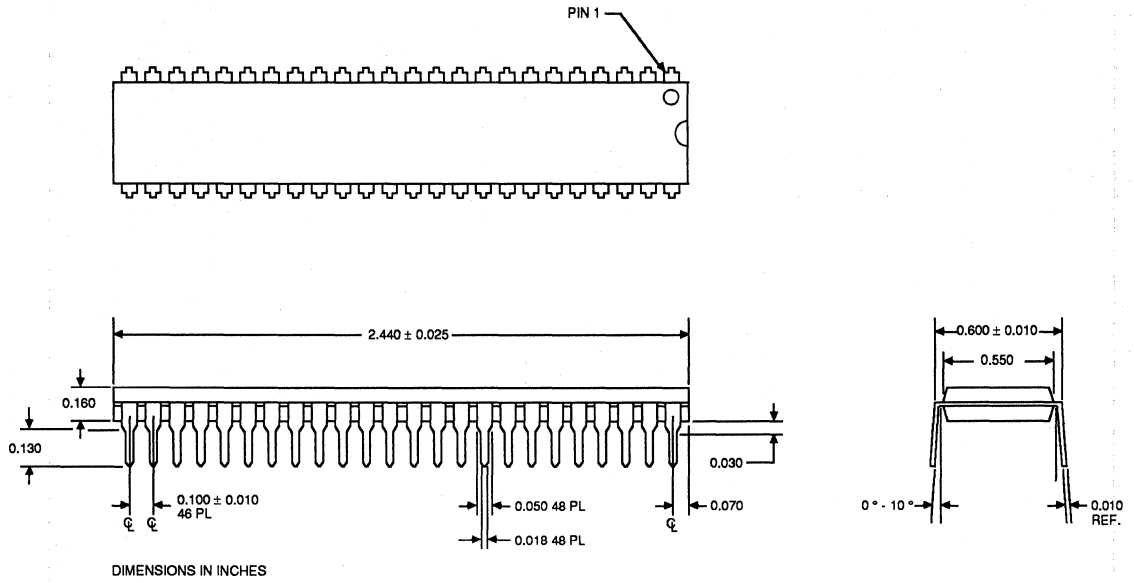
PG68 Pin-outs–XC2064



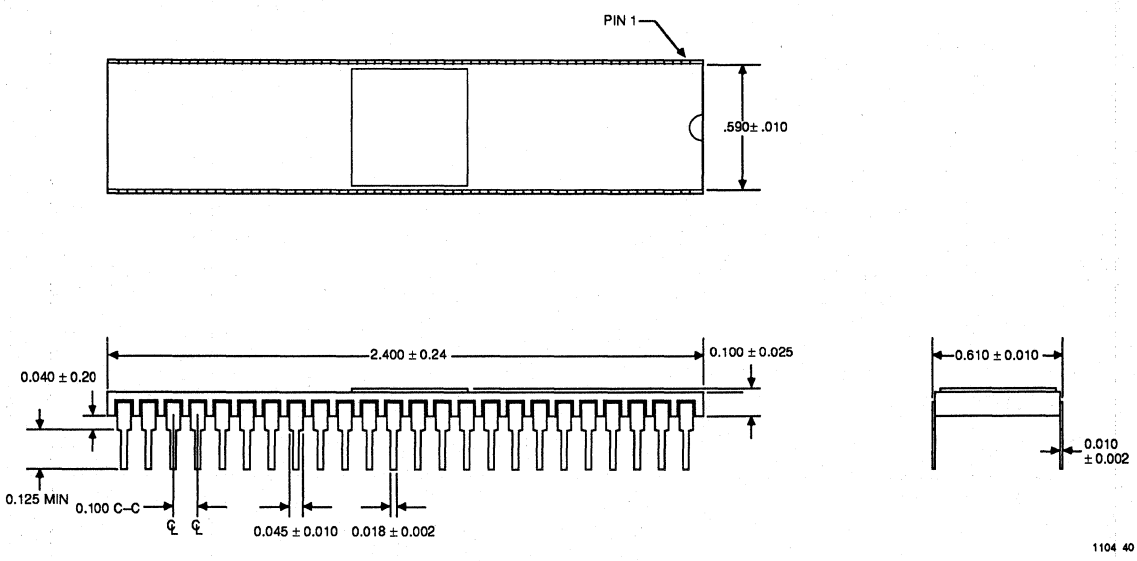
⊕ = Index pin which may or may not be electrically connected to pin C2
unlabeled pin = unrestricted I/O pin

PG84 Pin-outs–XC2018

PHYSICAL DIMENSIONS

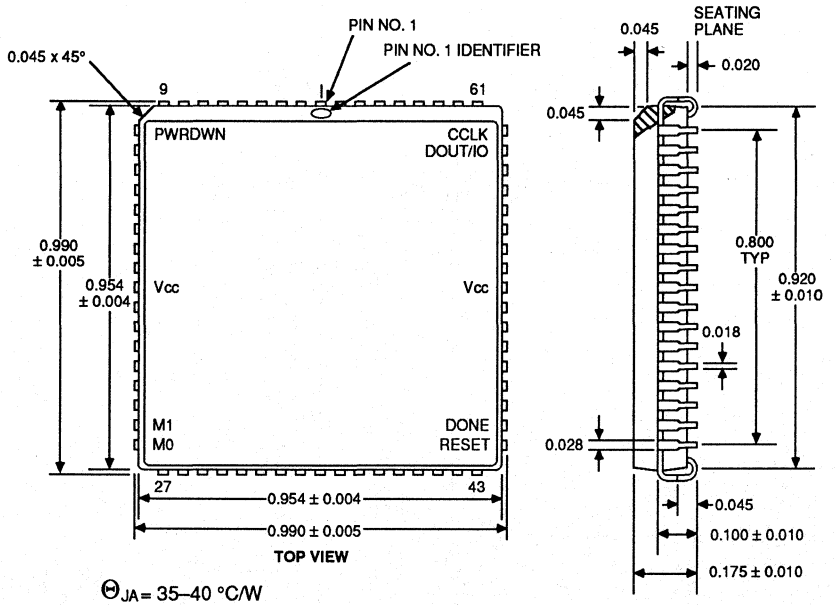


48-Pin Plastic DIP Package



48-Pin Ceramic DIP Package

PHYSICAL DIMENSIONS (Continued)

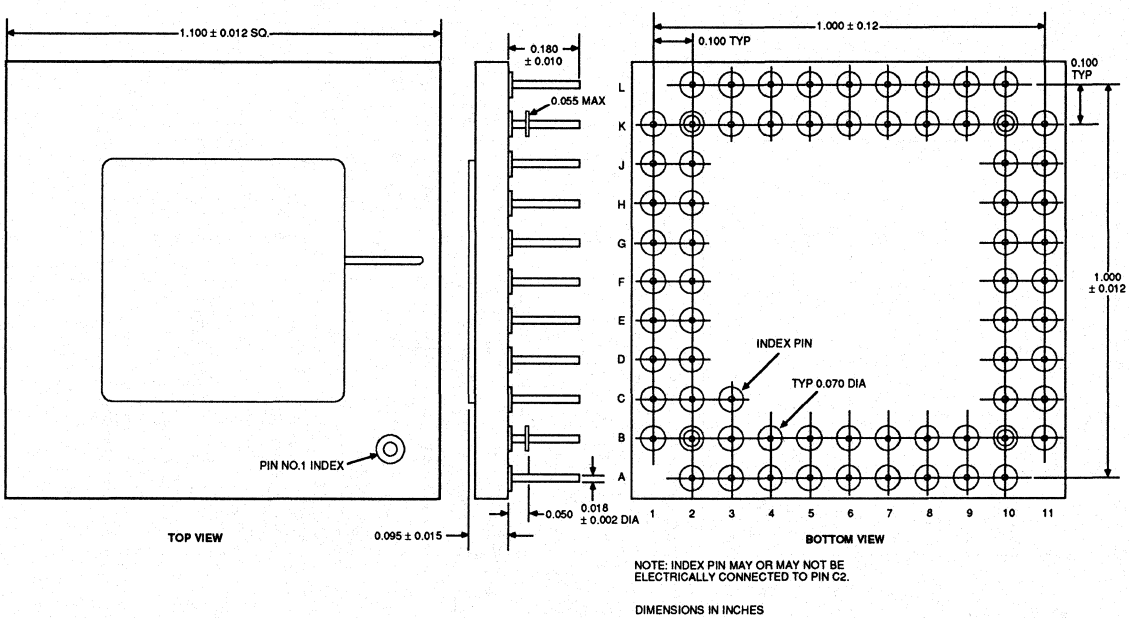


Θ_{JA} = 35–40 °C/W

Θ_{JC} = 7–10 °C/W

1105 34C

68-Pin PLCC Package



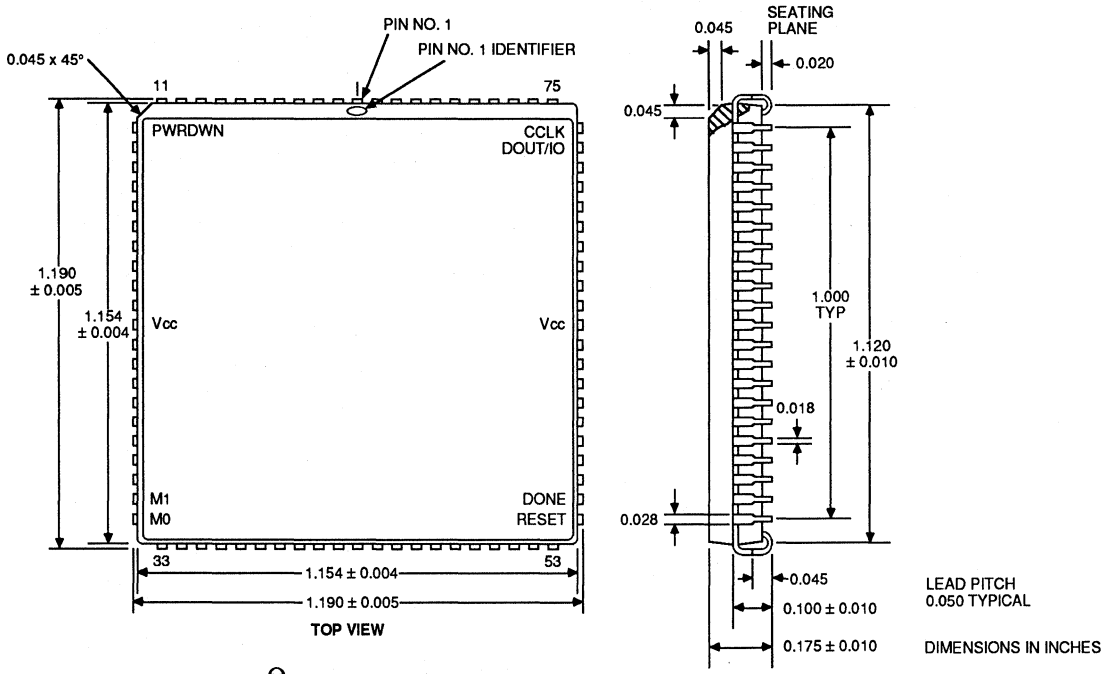
NOTE: INDEX PIN MAY OR MAY NOT BE ELECTRICALLY CONNECTED TO PIN C2.

DIMENSIONS IN INCHES

1104 42

68-Pin PGA Package

PHYSICAL DIMENSIONS (Continued)

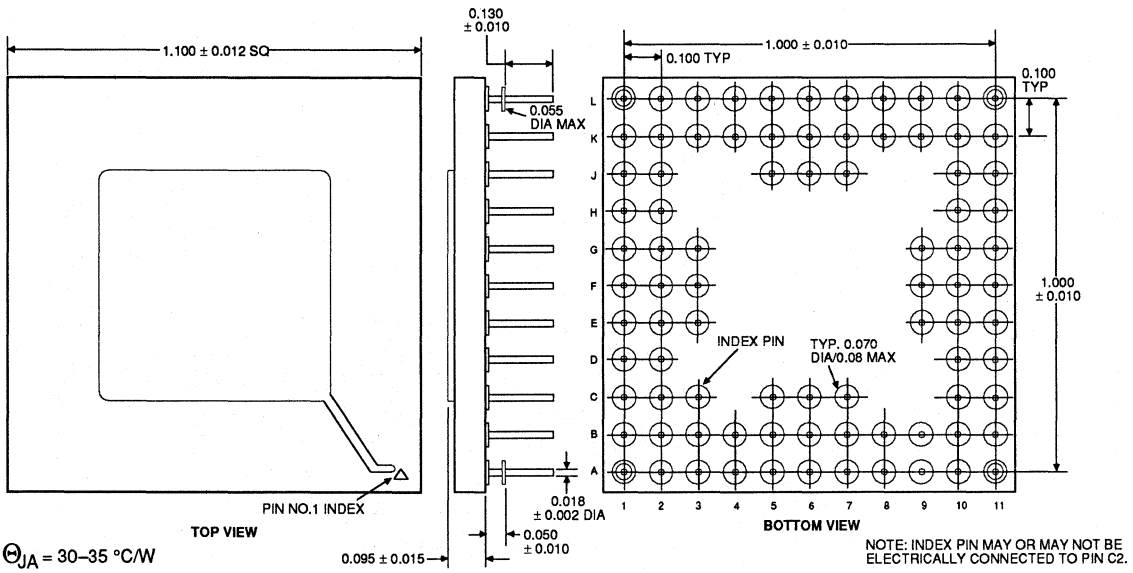


Θ_{JA}= 30–35 °C/W

Θ_{JC}= 3–7 °C/W

1105 36C

84-Pin PLCC Package



Θ_{JA} = 30–35 °C/W

Θ_{JC} = 4–7 °C/W

1105 35C

84-Pin PGA Package

Military Logic Cell™ Arrays

XC2018B, XC3020B, XC3042B, XC3090B

Product Specifications

INTRODUCTION

Xilinx introduced the first field programmable gate array (FPGA) in 1985. The development of the PGA was the result of a number of technical breakthroughs and truly represents the latest in advanced technology for microelectronic applications. Due to its density and the convenience of user programmability, the Logic Cell™ Array is an important new alternative in the ASIC market. Xilinx continues to concentrate its resources exclusively on expanding its growing family of programmable gate arrays and associated development systems. See the Xilinx Programmable Gate Array Data Book for a complete description of the architecture of both the 2000 and 3000 series arrays.

MIL-STD-883 CLASS B INTRODUCED

Xilinx continues its leadership in field programmable gate arrays (FPGA) by announcing the first military qualified FPGA's. These four devices meet all requirements of MIL-STD-883 paragraph 1.2.1

Device	Logic Capacity (gates)	Configurable Logic Blocks	User I/Os	Program Data (bits)
XC2018B	1800	100	74	17,878
XC3020B	2000	64	64	14,779
XC3042B	4200	144	96	30,784
XC3090B	9000	320	144	64,160

MILITARY PACKAGING

Xilinx offers two military packaging alternatives. In addition to the industry standard ceramic pin grid array (CPGA) packages we offer a ceramic quad flat package (CQFP) that meets the JEDEC standard outline drawing #MO-082. This CQFP has 25 mil pin-to-pin spacing. It is shipped with the leads unformed allowing selection of cavity up or cavity down and lead forming at the point of board assembly for better contact.

		Surface Mount		Through Hole	
Device	Total I/O	Ceramic QFP	User I/O	Ceramic PGA	User I/O
XC2018	74	—	—	CPGA 84	74
XC3020	64	CQFP 100	64	CPGA 84	64
XC3042	96	CQFP 100	82	CPGA 132	96
XC3090	144	CQFP 164	142	CPGA 175	144

STANDARD MILITARY DRAWINGS (SMD)

The Standard Military Drawing program (SMD) is a program initiated by the Federal government to simplify the procurement of Integrated Microcircuits (especially the more advanced technologies) by military contractors. The Defense Electronics Supply Center (DESC) issues the SMD that is consistent with the Xilinx military product specification and test conditions. DESC assigns an SMD specification number and releases the drawing. This drawing is then availble for use by all departments and agencies of the Department of Defense. The Xilinx device can then be easily procured by a military contractor by specifying the SMD# instead of the Xilinx part number. This eliminates the need for a separate Source Control Drawing (SCD) and greatly reduces paperwork.

DESC has assigned the XC2018B device SMD# 5962-88638, the XC3020B device SMD# 5962-89948, the XC3042B device SMD# 5962-89713 and the XC3090B device SMD# 5962-89823. Contact your Xilinx representative or DESC for more information.

LCA IDEAL FOR MILITARY APPLICATIONS

Field programmable gate arrays are taking market share from mask gate arrays in the commercial market are expected to be even more successful in the military market. Approximately 50% of all logic sales in the U.S. military market are ASIC's today. That number is expected to grow to 70% by 1993. FPGA's offer lower costs and more flexibility than mask gate arrays.

The LCA is especially suited to military ASIC applications. With a FPGA one specification can be written to cover multiple applications. Xilinx programmable gate arrays are "configured" by downloading software to the part - no fuses are blown. There is no requirement for post-programming testing for fault verification. The device is never obsolete because it can be reprogrammed many times.

Because Xilinx FPGA's are standard parts, they can be stocked in inventory at Xilinx, at Xilinx distributors or at the user site. One part can be stocked for multiple applications, minimizing inventory costs. Another benefit of being a standard product is the inherent high reliability of a high volume memory product rather than a low volume custom circuit. Non-recurring engineering costs (NRE) are never required for a FPGA thereby providing cost effective solutions in military volumes and allowing very inexpensive design iterations.

For maximum security the configuration data may be "down-loaded" from a remote site thereby eliminating the potential of tampering with the configuration data locally. The FPGA can be made non-volatile in this instance with the addition of a small battery backup.

One of the most effective advantages of the Xilinx FPGA is the ability to reconfigure some or all of the device while it remains in the circuit. This opens up entirely new possibilities allowing the same gates to be used by different functions at different times.

IMPORTANT BENEFITS FOR MILITARY DESIGNS

Cost Containment

- No NRE
  - — Very cost effective in military volumes
  - — Low cost design iterations
- Standard Product
  - — No overrun charges
  - — Simplified product qualification.
  - — No test vectors to write
  - — Simplified documentation (SMD)

Reliability

- Standard Product
  - — Reliability of hi-volume memory product
- Fully tested by Xilinx
  - — Fault coverage assured by vendor

Security

- No design information needed by manufacturer
  - — Secure design process. Design data held to vendor at user site.
- Remote configuration
  - — Ensures secure design data capability

Flexibility

- Standard product
  - — An ASIC where one spec can be used for multiple applications
  - — An ASIC stocked by distribution
- Reprogrammable
  - — Logic can be changed "on the fly"
- No FAB turnaround
  - — Design changes in minutes

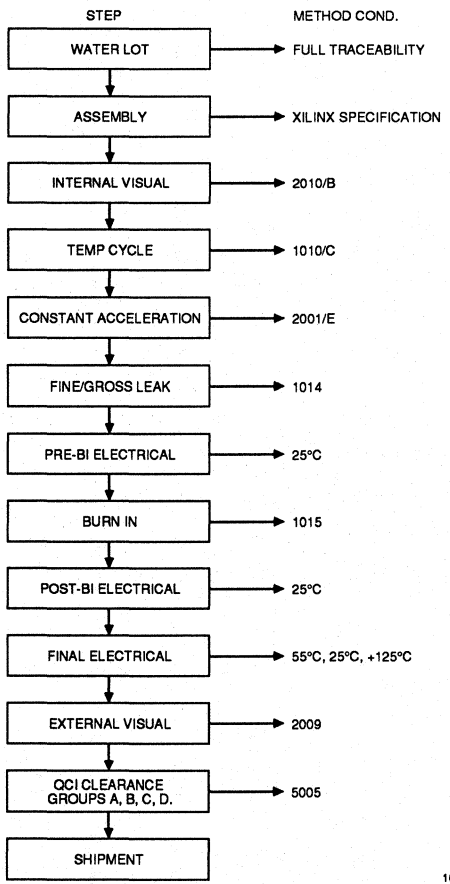
MIL-STD-883 CLASS B COMPLIANCE

Xilinx is now serving military customers in accordance with MIL-STD-883 Class B paragraph 1.2.1 together with the attendant requirements of MIL-M-38510. This includes full compliance with all processing requirements of Method 5004 and all Quality Conformance Inspection (QCI) requirements of Method 5005 (Groups A,B,C,D).

MIL-M-38510 (as invoked by MIL-STD-883)
Military Specification Microcircuits—General Specification (describes the design, processing and assembly workmanship guidelines)

MIL-STD-883
Military Standards—Test Methods and Procedures for Microelectronics (delineates the detailed testing and inspection methods for military integrated circuits)

MIL-STD-883 Class B—Method 5004 Processing Flow



1637 01

MIL-STD-883 CLASS B–METHOD 5005 QUALITY CONFORMANCE INSPECTION (QCI) TESTING

Every lot of devices shipped to the requirements of MIL-STD-883C is required to be qualified by four kinds of Quality Conformance Inspection (QCI) Tests. The QCI requirements specified by the Defense Electronics Supply Center (DESC) undergo regular revisions. Xilinx rigorously incorporates these revisions into our QCI testing in conformance with the requirements of MIL-STD-883C. These are:

Group A—Electrical tests done to data sheet limits at all three temperatures of the military temperature range, –55°C to +125°C. These are performed on a sample from the same lot being shipped.

Group B—Mechanical tests performed on a sample of devices of the same device/package type assembled within the same 6 week widow of the lot being shipped. This group consists of up to 8 subgroups including physical dimensions, mark permanency, solderability, internal visual/mechanical, bond strength, internal water vapor content, fine & gross leak, and ESD sensitivity.

Group C—Package related reliability tests performed on a sample of devices made with die from the same 1 year window. This group consists of up to 2 subgroups including (1) life testing (1000 hr at 125°C) and (2) temperature cycling, constant acceleration, fine & gross leak, and a visual examination.

Group D—Package related reliability tests performed on a sample of devices made in the same package within the same 1 year window. This group consists of up to 8 subgroups: physical dimensions; lead integrity and seal; thermal shock/temperature cycling/moisture resistance/seal/visual; mechanical shock vibration (variable frequency)/constant acceleration/seal/visual; salt atmosphere/seal/visual; internal water-vapor content; adhesion of lead finish; lid torque.

XC2018B
Military Logic Cell™ Array

Product Specification. See Note 1.

FEATURES

- MIL–STD–883 Class B Processing. Complies with paragraph 1.2.1
- Field-programmable gate array
- Low power CMOS static memory technology
- Standard product. Completely tested at factory
- Design changes made in minutes
- Complete user control for design cycle. Secure design process
- Complete PC or workstation based development system
  - Schematic entry
  - Auto Place/ Route (DS23)
  - Design Editor (DS21)
  - Logic & Timing Simulator (DS22)
  - XACTOR In-circuit Verifier (DS24)

Part Number	Logic Capacity (gates)	Configurable Logic Blocks	User I/Os	Configuration Program (bits)
XC2018	1800	100	74	17878

DESCRIPTION

The Logic Cell™ Array (LCA) is a high density CMOS programmable gate array. Its patented array architecture consists of three types of configurable elements: Input/ Output Blocks, Configurable Logic Blocks and Interconnect. The designer can define individual I/O blocks for interface to external circuitry, define logic blocks to implement logic functions and define interconnection networks to compose larger scale logic functions.

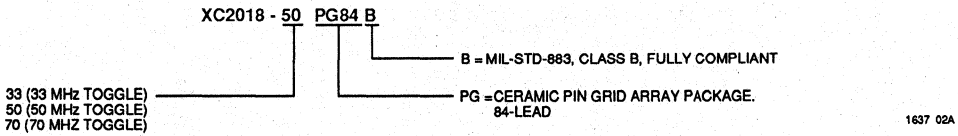
The Logic Cell Array's logic functions and interconnections are determined by the configuration program stored in internal static memory cells. On-chip logic provides for automatic loading of configuration data at power-up or on command. The program data can reside in an EEPROM, EPROM or ROM on the circuit board or on a floppy disk or hard disk.

Several methods of automatically loading the required data are designed into the Logic Cell Array and are determined by logic levels applied to mode selection pins at configuration time. The form of the data may be either serial or parallel, depending on the configuration mode. The programming data are independent of the configuration mode selected.

The XACT development system allows the user to define the logic functions of the device. Schematic capture is available for design entry, while logic and timing simulation, and in-circuit debugging are available for design verification. XACT is used to compile the data pattern which represents the configuration program. This data can then be converted to a PROM programmer format file to create the configuration program storage.

See the XC2018 Commercial data sheet for a full description.

ORDERING INFORMATION



TSC0026

PIN ASSIGNMENTS

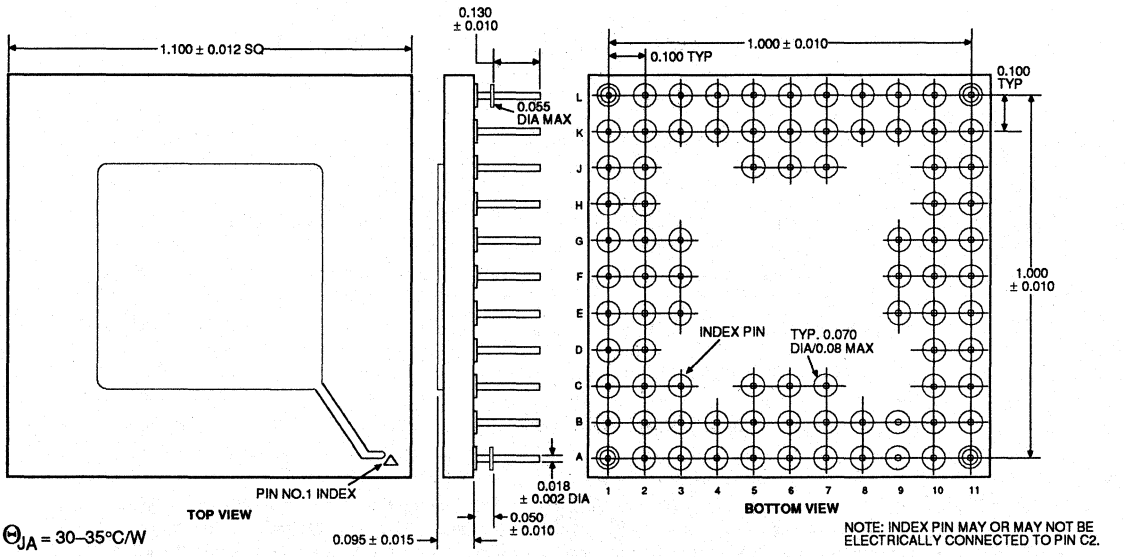
CONFIGURATION MODE: <M2:M1:M0>					84 PGA	USER OPERATION
MASTER-SER <0:0:0>	SLAVE <1:1:1>	PERIPHERAL <1:0:1>	MASTER-HIGH <1:1:0>	MASTER-LOW <1:0:0>		
GND					C6	GND
<<HIGH>>			A13 (O)		A6	I/O
					A5	
					B5	
			A6 (O)		C5	
			A12 (O)		A4	
			A7 (O)		B4	
			A11 (O)		A3	
			A8 (O)		A2	
			A10 (O)		B3	
			A9 (O)		A1	
PWRDWN (I)					B2	PWR DWN
<<HIGH>>					C2	I/O
					B1	
					C1	
					D2	
					D1	
					E3	
					E2	
					E1	
					F2	
VCC					F3	VCC
<<HIGH>>					G3	I/O
					G1	
					G2	
					F1	
					H1	
					H2	
					J1	
					K1	
M1 (LOW)	M1 (HIGH)	M1 (LOW)	M1 (HIGH)	M1 (LOW)	J2	RDATA (O)
M0 (LOW)	M0 (HIGH)	M0 (HIGH)	M0 (LOW)	M0 (LOW)	L1	RTRIG (I)
M2 (LOW)	M2 (HIGH)				K2	I/O
HDC (HIGH)					K3	
<<HIGH>>					L2	
LDC (LOW)					L3	
<<HIGH>>					K4	
					L4	
					J5	
					K5	
					L5	
					K6	
GND					J6	GND
<<HIGH>>					J7	
					L7	
					K7	
					L6	
					I/O	
					K8	
					L9	
			D7 (I)		L10	
			D6 (I)		K9	
					L11	XTL2 OR I/O
RESET (I)					K10	RESET
DONE (O)					J10	PROG (I)
<<HIGH>>					K11	XTL1 OR I/O
					J11	I/O
			D5 (I)		H10	
					H11	
					F10	
					G10	
		CSO (I)	D4 (I)		G11	
		CS1 (I)	D3 (I)		G9	
VCC					F9	VCC
<<HIGH>>					F11	I/O
		CS2 (I)	D2 (I)		E11	
					E10	
					E9	
					D11	
		WRT (I)	D1 (I)		D10	
RCLK			RCLK		C11	
DIN (I)			D0 (I)		B11	
DOUT (O)					C10	
CCLK (O)	CCLK (I)	CCLK (O)			A11	CCLK (I)
<<HIGH>>			A0 (O)		B10	I/O
			A1 (O)		B9	
			A2 (O)		A10	
			A3 (O)		A9	
			A15 (O)		B8	
			A4 (O)		A8	
			A14 (O)		B6	
					B7	
					A7	
			A5 (O)		C7	

TSC0026

<<HIGH>> IS HIGH IMPEDANCE WITH A 20–50 kΩ INTERNAL PULL-UP DURING CONFIGURATION

1637 03

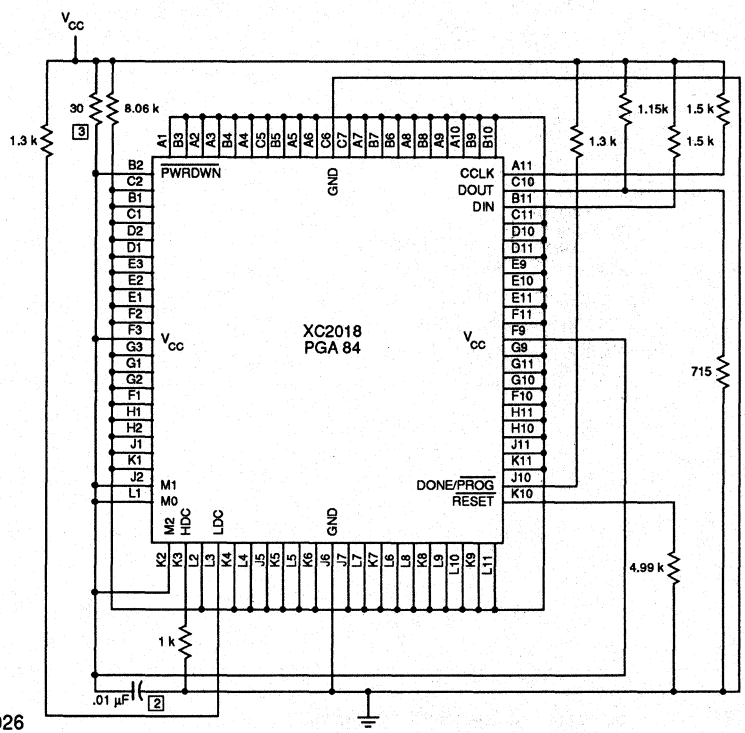
PHYSICAL DIMENSIONS - Conforms to MIL-M-38510 Appendix C, Case P-BC.



XC2018: 84-Pin PGA Package

1105 35B

STATIC BURN-IN CIRCUITS



NOTES:
1. UNLESS OTHERWISE SPECIFIED, ALL RESISTORS ARE METAL FILM AND ARE RATED FOR 1/8 WATT AT 150°C WITH A BUILD TOLERANCE OF 1% AND A 5% TOLERANCE OVER LIFE.
2. CAPACITOR HAS 10% TOLERANCE, 50 V RATING WITH AN X7R TEMPERATURE CHARACTERISTIC.
3. 30 Ω RESISTOR IS METAL OXIDE AND IS RATED FOR 1 W AT 150°C WITH A TOLERANCE OF 5%.

TSC0026

1637 05A

TEST SPECIFICATION

Xilinx maintains test specifications for each product as controlled documents. To insure the use of the most recently released device performance parameters, please request a copy of the current test-specification revision.

Absolute Maximum Ratings		Limits	Units
V_{CC}	Supply voltage relative to GND	–0.5 to +7.0	V
V_{IN}	Input voltage with respect to GND	–0.5 to VCC +0.5	V
V_{TS}	Voltage applied to three-state output	–0.5 to VCC +0.5	V
T_{STG}	Storage temperature (ambient)	–65 to +150	°C
T_{SOL}	Maximum soldering temperature (10 sec @ 1/16 in.)	+260	°C
T_J	Maximum junction temperature	+150	°C

Note: Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those listed under Recommended Operating Conditions is not implied. Exposure to Absolute Maximum Ratings conditions for extended periods of time may affect device reliability.

Test	Symbol	Conditions $-55°C \le T_C \le +125°C$ V_{CC} = 5.0 V ±10%	Group A Subgroups	Min	Max	Units
High Level Output Voltage	V_{OH}	V_{CC} = 4.5 V, I_{OH} = –4.0 mA	1,2,3	3.7		V
Low Level Output Voltage	V_{OL}	V_{CC} = 5.5 V, I_{OL} = 4.0 mA	1,2,3		0.4	V
Quiescent Operating	I_{CCO}	CMOS Inputs, $V_{in} = V_{CC}$ = 5.5 V	1,2,3		10	mA
Power Supply Current		TTL Inputs, $V_{in} = V_{CC}$ = 5.5 V	1,2,3		15	mA
Power-Down Supply Current	I_{CCPD}	$V_{in} = V_{CC}$ = 5.5 V, $\overline{PWR}$ $\overline{DWN}$ = 0 V	1,2,3		0.5	mA
Leakage Current	I_{IL}	V_{CC} = 5.5 V, V_{in} = V_{CC} and 0 V	1,2,3	–10	10	µA
Input High Level TTL	V_{IHT}	Guaranteed Input High	1,2,3	2.0		V
Input Low Level TTL	V_{ILT}	Guaranteed Input Low	1,2,3		0.8	V
Input High Level CMOS	V_{IHC}	Guaranteed Input High	1,2,3	.7 V_{CC}		V
Input Low Level CMOS	V_{ILC}	Guaranteed Input Low	1,2,3		.2 V_{CC}	V

Switching Characteristics, General LCA

DONE/$\overline{PROG}$							
Program Width (Low)	T_{PGW}	4	See Fig. 3	9,10,11	6		µs
Initialization	T_{PGI}	5		9,10,11		7	µs
$\overline{PWR}$ $\overline{DWN}$[2]							
Power Down Supply	V_{PD}			1,2,3	3.5		V

TSC0026

Table 1. Electrical Performance Characteristics

Test	Sym	Conditions -55°C ≤TC ≤+125°C V_{CC} = 5.0 V ±10%	Group A Subgroups	-33 Min	-33 Max	-50 Min	-50 Max	-70 Min	-70 Max	Units
Switching Characteristics, Peripheral Mode Programming										
Controls (CS, WRT)[3,4]										
Last Active Input to First Active	T_{CA}	1 See Figure 4	9, 10, 11	0.5	1.0	0.5	1.0	0.5	1.0	µs
First Inactive Input to Last Active	T_{CI}	2	9, 10, 11	0.5		0.5		0.5		µs
CCLK[5]	T_{CCC}	3	9, 10, 11		75		75		75	ns
DIN Setup	T_{DC}	4	9, 10, 11	50		50		50		ns
DIN Hold	T_{CD}	5	9, 10, 11	5		5		5		ns
Switching Characteristics, Program Readback[6]										
RTRIG Setup	T_{RTH}	1 See Fig. 7	9, 10, 11	250		250		250		ns
CCLK,										
RTRIG Setup	T_{RTCC}	2	9, 10, 11	100		100		100		ns
RDATA Delay	T_{CCRD}	3	9, 10, 11		100		100		100	ns
Benchmark Patterns[7]										
T_{PID} + interconnect + 10 (T_{ILO}) + T_{OP}. Measured on 10 cols.	T_{B1}		9, 10, 11		238		178		119	ns
T_{PID} + interconnect + 10 (T_{ITO}) + T_{OP}. Measured on 10 cols.	T_{B2}		9, 10, 11		288		288		159	ns
T_{PID} + interconnect + 10 (T_{QLO}) + 10 (T_{ITO}) +T_{OP}. Measured on 10 cols.	T_{B3}		9, 10, 11		410		302		217	ns
T_{CKO} + 2 (T_{ILO}) + T_{ICK} + interconnect	T_{B4}		9, 10, 11		85		62		42	ns
T_{CIO} + T_{ILO} + T_{ICI} + interconnect	T_{B5}		9, 10, 11		90		67		41	ns
T_{CCO} + 2 (T_{ILO}) + T_{ICC} + interconnect	T_{B6}		9, 10, 11		66		49		38	ns
T_{PID} + interconnect + 10 (T_{RIO}) + T_{OP}. Measured on 10 rows.	T_{B7}		9, 10, 11		318		269		183	ns
T_{LI} + 3 (T_{IPD}) + 4 (T_{OP}) + T_{PL} + interconnect. Tested on all IOBs.	T_{B8}		9, 10, 11		274		204		141	ns

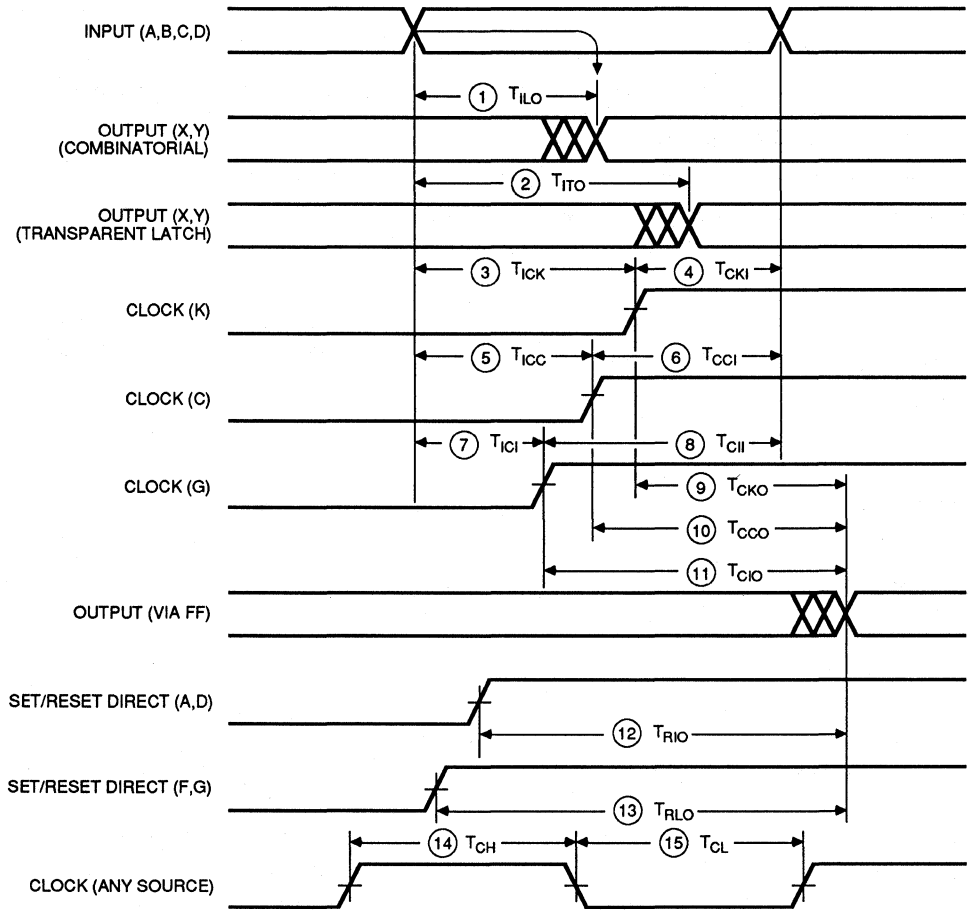
Table 1. Electrical Performance Characteristics (cont'd)

Test	Sym	Conditions -55°C ≤TC ≤+125°C V_{CC} = 5.0 V ±10%	Group A Subgroups	-33 Min	-33 Max	-50 Min	-50 Max	-70 Min	-70 Max	Units
Application Guidelines, Switching, CLB[7]										
Logic Input to Output, Combinatorial	T_{ILO}	1 See Fig. 1	N/A		20		15		10	ns
Transparent Latch	T_{ITO}	2	N/A		25		20		14	ns
Additional for Q Through F or G to Out	T_{QLO}		N/A		13		8		6	ns
K Clock,										
To Output	T_{CKO}	9	N/A		20		15		10	ns
Logic-Input Setup	T_{ICK}	3	N/A	12		8		7		ns
Logic-Input Hold	T_{CKI}	4	N/A	1		1		1		ns
C Clock,										
To Output	T_{CCO}	10	N/A		25		19		13	ns
Logic- Input Setup	T_{ICC}	5	N/A	12		9		6		ns
Logic-Input Hold	T_{CCI}	6	N/A	6		1		1		ns
Logic Input to G Clock,										
To Output	T_{CIO}	11	N/A		37		27		20	ns
Logic-Input Setup	T_{ICI}	7	N/A	6		4		3		ns
Logic-Input Hold	T_{CII}	8	N/A	9		5		4		ns
Set/Reset Direct,										
Input A or D to Out	T_{RIO}	12	N/A		25		22		16	ns
Through F or G to Out	T_{RLO}	13	N/A		37		28		21	ns
Master Reset Pin to Out	T_{MRQ}		N/A		55		45		35	ns
Separation of Set/Reset	T_{RS}		N/A	17		9		7		ns
Set/Reset Pulse-Width	T_{RPW}		N/A	12		9		7		ns
Flip-Flop Toggle Rate, Q Through F to Flip-Flop	F_{CLK}		N/A	33		50		70		MHz
Clock High[8]	T_{CH}	14	N/A	12		8		7		ns
Clock Low[8]	T_{CL}	15	N/A	12		8		7		ns

Table 1. Electrical Performance Characteristics (Continued)

Test	Sym	Conditions -55°C ≤TC ≤+125°C V_{CC} = 5.0 V ±10%	Group A Subgroups	-33 Min	-33 Max	-50 Min	-50 Max	-70 Min	-70 Max	Units
Application Guidelines, Switching, IOB[7]										
Pad (Package Pin) to Input (Direct)	T_{PID}	1 See Fig. 2	N/A		12		8		6	ns
I/O Clock										
To Input (Storage)	T_{LI}	5	N/A		20		15		11	ns
To Pad-Input Setup	T_{PL}	2	N/A	12		8		6		ns
To Pad Input Hold	T_{LP}	3	N/A	0		0		0		ns
Pulse Width	T_{LW}	4	N/A	12		9		7		ns
Frequency			N/A		33		50		70	MHz
Output,										
To Pad (Output Enable)	T_{OP}	8	N/A		15		12		9	ns
Three-State,										
To Pad Begin hi-Z	T_{THZ}	9	N/A		25		20		15	ns
To Pad End hi-Z	T_{TON}	10	N/A		25		20		15	ns
$\overline{RESET}$,										
To Input (Storage)	T_{RI}	6	N/A		40		30		25	ns
To Input Clock	T_{RC}	7	N/A		35		25		20	ns
Application Guidelines, Switching, Slave Mode Programming[7]										
CCLK,										
To DOUT	T_{CCO}	3 See Fig. 6	N/A		100		100		100	ns
DIN Setup	T_{DCC}	1	N/A	10		10		10		ns
DIN Hold	T_{CCD}	2	N/A	40		40		40		ns
High Time	T_{CCH}	4	N/A	0.5		0.5		0.5		µs
Low Time	T_{CCL}	5	N/A	0.5	1.0	0.5	1.0	0.5	1.0	µs
Frequency	F_{CC}		N/A		1		1		1	MHz
Application Guidelines, Switching, Master Mode Programming[7,9]										
$\overline{RCLK}$,										
From Address Invalid	T_{ARC}	1 See Fig. 5	N/A		0		0		0	ns
To Address Valid	T_{RAC}	2	N/A		200		200		200	ns
To Data Setup	T_{DRC}	3	N/A	60		60		60		ns
To Data Hold	T_{RCD}	4	N/A	0		0		0		ns
$\overline{RCLK}$ High	T_{RCH}	5	N/A	600		600		600		ns
$\overline{RCLK}$ Low	T_{RCL}	6	N/A	4.0		4.0		4.0		µs
Application Guidelines, Switching, General LCA[7]										
$\overline{RESET}$[10]										
M2, M1, M0 Setup	T_{MR}	1 See Fig. 3	N/A	1		1		1		µs
M2, M1, M0 Hold	T_{RM}	2	N/A	1		1		1		µs
Width (Low)	T_{MRW}	3	N/A	150		150		150		ns

TSC0026 **Table 1. Electrical Performance Characteristics (Continued)**



Timing is measured at 0.5 V_{CC} levels with 50 pF minimum output load.

Input signal conditioning: Rise and fall times ≤ 6 ns, Amplitude = 0 and 3V

1637 06

Figure 1. Switching Characteristics Waveforms, CLB

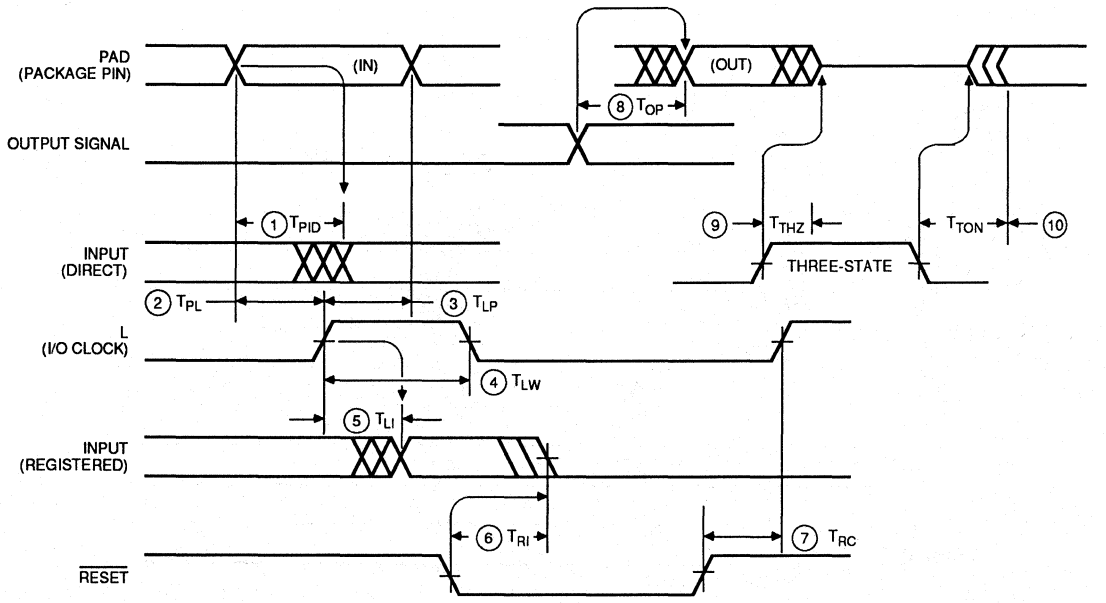


Figure 2. Switching Characteristics, IOB

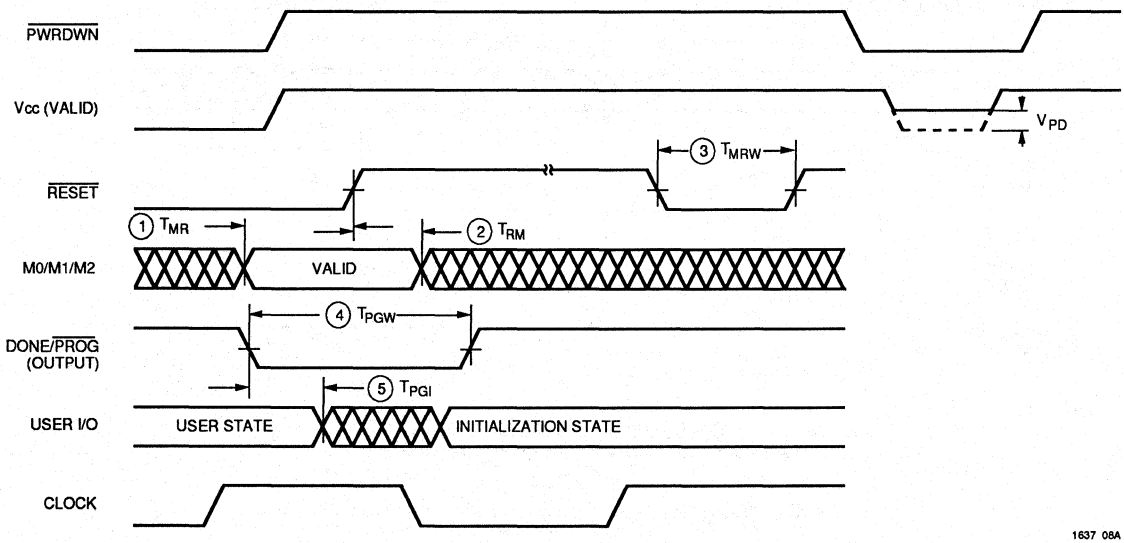


Figure 3. General LCA Switching Characteristics

TSC0026

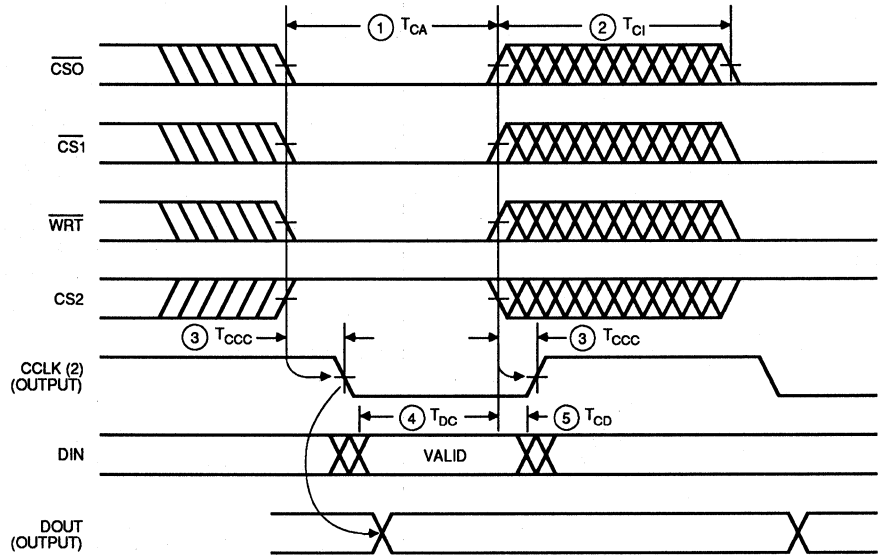
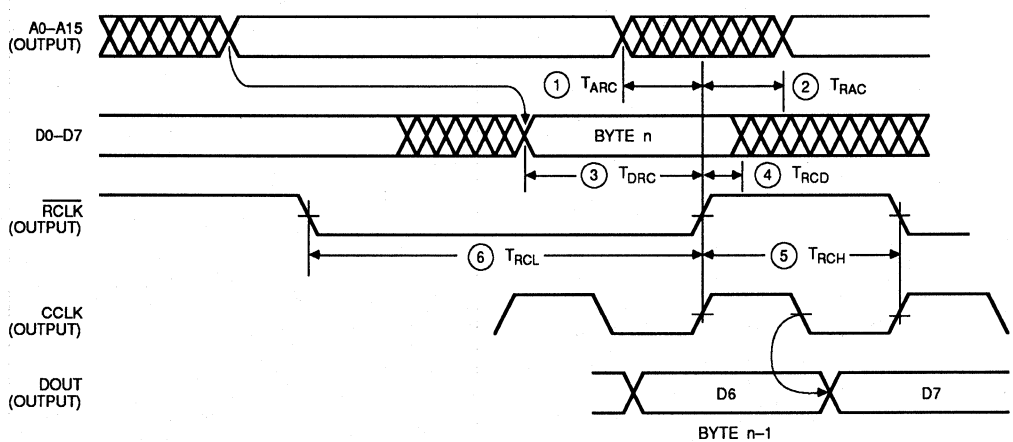


Figure 4. Peripheral Mode Programming Characteristics



CCLK and DOUT timing are the same as for slave mode.
At power–up, V_{CC} must rise from 2.0 V to V_{CC} min. in less than 10 ms.

1637 10

Figure 5. Master Mode Programming Switching Characteristics

TSC0026

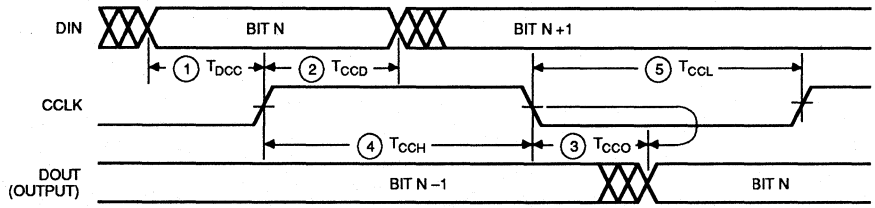


Figure 6. Slave Mode Programming Switching Characteristics

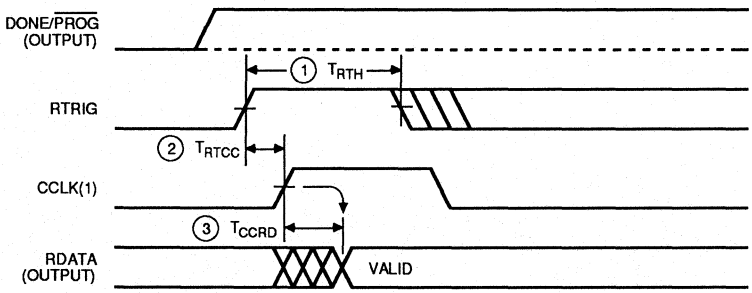
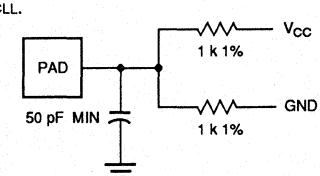


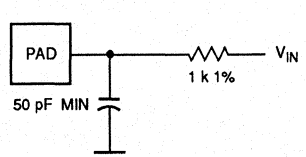
Figure 7. Program Readback Characteristics

XC2018B Data Sheet Notes

Notes:
1. Xilinx maintains this specification as a controlled document. To comply with the intent of MIL-STD-883, and to assure that you are using the most recently released device performance parameters, please request a copy of the current revision of this Test Specification from Xilinx.
2. $\overline{PWR\ DWN}$ must be active before Vcc goes below specified range, and inactive after Vcc reaches specified range.
3. Peripheral mode timing determined from last control signal of the logical AND of ($\overline{CS0}$, $\overline{CS1}$, CS2, $\overline{WRT}$) to transition to active or inactive state.
4. Configuration must be delayed at least 40 ms after Vcc min.
5. CCLK and DOUT timing are the same as for slave mode.
6. D/$\overline{P}$ must be high before RTRIG goes High.
7. Testing of the Applications Guidelines is modeled after testing specified by MIL-M-38510/605. Devices are first 100% functionally tested. Benchmark patterns are then used to measure the Application Guidelines. Characterization data are taken at initial device qualification, prior to introduction of significant changes, and at least twice yearly to monitor correlation between benchmark patterns, device performance, XACT software timings, and the data sheet.
8. Minimum CLOCK widths for the auxiliary buffer are 1.25 times the TCLH, TCLL.
9. Vcc must rise from 2.0 V to Vcc minimum in less than 10 ms for master mode.
10. $\overline{RESET}$ timing relative to power-on and valid mode lines (M0, M1, M2) is relevant only when $\overline{RESET}$ is used to delay configuration.
11. All timings except TTSHZ and TTSON are measured at 1.5 Vcc level with 50 pF minimum load output. For input signals, rise and fall times are less than 6 ns, with low amplitude = 0 V, and high = 3 V. TTHZ is determined when the output shifts 10% (of the output voltage swing) from VOL level or VOH level. The following circuit is used:

    TTON is measured at 0.5 Vcc level with VIN = 0 for 3-State to active High, and VIN = Vcc for 3-State to active Low. The following load circuit is used:





TSC0026

TSC0026

XC3020B
Military Logic Cell™ Array

Product Specification. See Note 1.

FEATURES

- MIL–STD–883 Class B Processing. Complies with paragraph 1.2.1
- Field-programmable gate array
- Low power CMOS static memory technology
- Standard product. Completely tested at factory
- Design changes made in minutes
- Complete user control for design cycle. Secure design process
- Complete PC or workstation based development system
  - Schematic entry
  - Auto Place/ Route (DS23)
  - Design Editor (DS21)
  - Logic & Timing Simulator (DS22)
  - XACTOR In-circuit Verifier (DS24)

Part Number	Logic Capacity (gates)	Configurable Logic Blocks	User I/Os	Configuration Program (bits)
XC3020	2000	64	64	14779

DESCRIPTION

The Logic Cell™ Array (LCA) is a high density CMOS programmable gate array. Its patented array architecture consists of three types of configurable elements: Input/ Output Blocks, Configurable Logic Blocks and Interconnect. The designer can define individual I/O blocks for interface to external circuitry, define logic blocks to implement logic functions and define interconnection networks to compose larger scale logic functions.

The Logic Cell Array's logic functions and interconnections are determined by the configuration program stored in internal static memory cells. On-chip logic provides for automatic loading of configuration data at power-up or on command. The program data can reside in an EEPROM, EPROM or ROM on the circuit board or on a floppy disk or hard disk.

Several methods of automatically loading the required data are designed into the Logic Cell Array and are determined by logic levels applied to mode selection pins at configuration time. The form of the data may be either serial or parallel, depending on the configuration mode. The programming data are independent of the configuration mode selected.

The XACT development system allows the user to define the logic functions of the device. Schematic capture is available for design entry, while logic and timing simulation, and in-circuit debugging are available for design verification. XACT is used to compile the data pattern which represents the configuration program. This data can then be converted to a PROM programmer format file to create the configuration program storage.

See the XC3000 Commercial data sheet for a full description.

ORDERING INFORMATION

XC3020 - 50 PG84 B

- B = MIL-STD-883, CLASS B, FULLY COMPLIANT
- 50 (50 MHz TOGGLE)
  70 (70 MHz TOGGLE)
- PG84 = CERAMIC PIN GRID ARRAY PACKAGE. 84-LEAD
  CQ100 = CERAMIC QUAD FLAT PACKAGE. 100 LEAD

TSC0085

1637 15B

XC3020 CONFIGURATION PIN ASSIGNMENTS

	CONFIGURATION MODE: <M2:M1:M0>					84 PGA	100 CQFP	USER OPERATION
	SLAVE <1:1:1>	MASTER-SER <0:0:0>	PERIPHERAL <1:0:1>	MASTER-HIGH <1:1:0>	MASTER-LOW <1:0:0>			
	PWR DWN (I)	PWR DWN (I)	PWR DWN (I)	PWR DWN (I)	PWR DWN (I)	B2	14	PWR DWN (I)
	VCC	VCC	VCC	VCC	VCC	F3	26	VCC
	M1 (HIGH) (I)	M1 (LOW) (I)	M1 (LOW) (I)	M1 (HIGH) (I)	M1 (LOW) (I)	J2	37	RDATA
	M0 (HIGH) (I)	M0 (LOW) (I)	M0 (HIGH) (I)	M0 (LOW) (I)	M0 (LOW) (I)	L1	39	RTRIG (I)
	M2 (HIGH) (I)	M2 (LOW) (I)	M2 (HIGH) (I)	M2 (HIGH) (I)	M2 (HIGH) (I)	K2	41	I/O
	HDC (HIGH)	HDC (HIGH)	HDC (HIGH)	HDC (HIGH)	HDC (HIGH)	K3	42	I/O
	LDC (LOW)	LDC (LOW)	LDC (LOW)	LDC (LOW)	LDC (LOW)	L3	44	I/O
*	INIT	INIT	INIT	INIT	INIT	K6	50	I/O
	GND	GND	GND	GND	GND	J6	51	GND
						L11	61	XTL2 OR I/O
	RESET (I)	RESET (I)	RESET (I)	RESET (I)	RESET (I)	K10	63	RESET (I)
	DONE	DONE	DONE	DONE	DONE	J10	65	PROGRAM (I)
			DATA 7 (I)	DATA 7 (I)	DATA 7 (I)	K11	66	I/O
						J11	67	XTL1 OR I/O
			DATA 6 (I)	DATA 6 (I)	DATA 6 (I)	H10	68	I/O
			DATA 5 (I)	DATA 5 (I)	DATA 5 (I)	F10	72	I/O
			CS0 (I)			G10	73	I/O
			DATA 4 (I)	DATA 4 (I)	DATA 4 (I)	G11	74	I/O
	VCC	VCC	VCC	VCC	VCC	F9	76	VCC
			DATA 3 (I)	DATA 3 (I)	DATA 3 (I)	F11	77	I/O
			CS1 (I)			E11	78	I/O
			DATA 2 (I)	DATA 2 (I)	DATA 2 (I)	E10	79	I/O
			DATA 1 (I)	DATA 1 (I)	DATA 1 (I)	D10	83	I/O
			RDY/BUSY	RCLK	RCLK	C11	84	I/O
	DIN (I)	DIN (I)	DATA 0 (I)	DATA 0 (I)	DATA 0 (I)	B11	85	I/O
	DOUT	DOUT	DOUT	DOUT	DOUT	C10	86	I/O
	CCLK (I)	CCLK	CCLK	CCLK	CCLK	A11	87	CCLK (I)
			WS (I)	A0	A0	B10	90	I/O
			CS2 (I)	A1	A1	B9	91	I/O
				A2	A2	A10	93	I/O
				A3	A3	A9	94	I/O
				A15	A15	B6	97	I/O
				A4	A4	B7	98	I/O
				A14	A14	A7	99	I/O
				A5	A5	C7	100	I/O
	GND	GND	GND	GND	GND	C6	1	GND
				A13	A13	A6	2	I/O
				A6	A6	A5	3	I/O
				A12	A12	B5	4	I/O
				A7	A7	C5	5	I/O
				A11	A11	A3	8	I/O
				A8	A8	A2	9	I/O
				A10	A10	B3	10	I/O
				A9	A9	A1	11	I/O

REPRESENTS A 50KΩ TO 100KΩ PULL-UP

* INIT IS AN OPEN DRAIN OUTPUT DURING CONFIGURATION

(I) REPRESENTS AN INPUT

1637 16B

PIN ASSIGNMENTS (Continued)

PGA Pin Number	XC3020	PGA Pin Number	XC3020
B2	PWRDN	K10	RESET
C2	TCLKIN-I/O	J10	DONE-PG
B1	NC	K11	D7-I/O
C1	I/O	J11	XTL1(OUT)-BLCKIN-I/O
D2	I/O	H10	D6-I/O
D1	I/O	H11	I/O
E3	I/O	F10	D5-I/O
E2	I/O	G10	CS0-I/O
E1	I/O	G11	D4-I/O
F2	I/O	G9	I/O
F3	**VCC**	F9	**VCC**
G3	I/O	F11	D3-I/O
G1	I/O	E11	CS1-I/O
G2	I/O	E10	D2-I/O
F1	I/O	E9	I/O
H1	I/O	D11	NC
H2	I/O	D10	D1-I/O
J1	I/O	C11	RDY/BUSY-RCLK-I/O
K1	I/O	B11	D0-DIN-I/O
J2	M1-RDATA	C10	DOUT-I/O
L1	M0-RTRIG	A11	CCLK
K2	M2-I/O	B10	A0-WS-I/O
K3	HDC-I/O	B9	A1-CS2-I/O
L2	I/O	A10	A2-I/O
L3	LDC-I/O	A9	A3-I/O
K4	I/O	B8	NC
L4	NC	A8	NC
J5	I/O	B6	A15-I/O
K5	I/O	B7	A4-I/O
L5	NC	A7	A14-I/O
K6	INIT-I/O	C7	A5-I/O
J6	**GND**	C6	**GND**
J7	I/O	A6	A13-I/O
L7	I/O	A5	A6-I/O
K7	I/O	B5	A12-I/O
L6	I/O	C5	A7-I/O
L8	I/O	A4	NC
K8	I/O	B4	NC
L9	NC	A3	A11-I/O
L10	NC	A2	A8-I/O
K9	I/O	B3	A10-I/O
L11	XTL2(IN)-I/O	A1	A9-I/O

Unprogrammed IOBs have a default pull-up. This prevents an undefined pad level for unbonded or unused IOBs. Programmed outputs are default slew-rate limited.

TSC0085

XC3020 100-PIN QFP PINOUTS

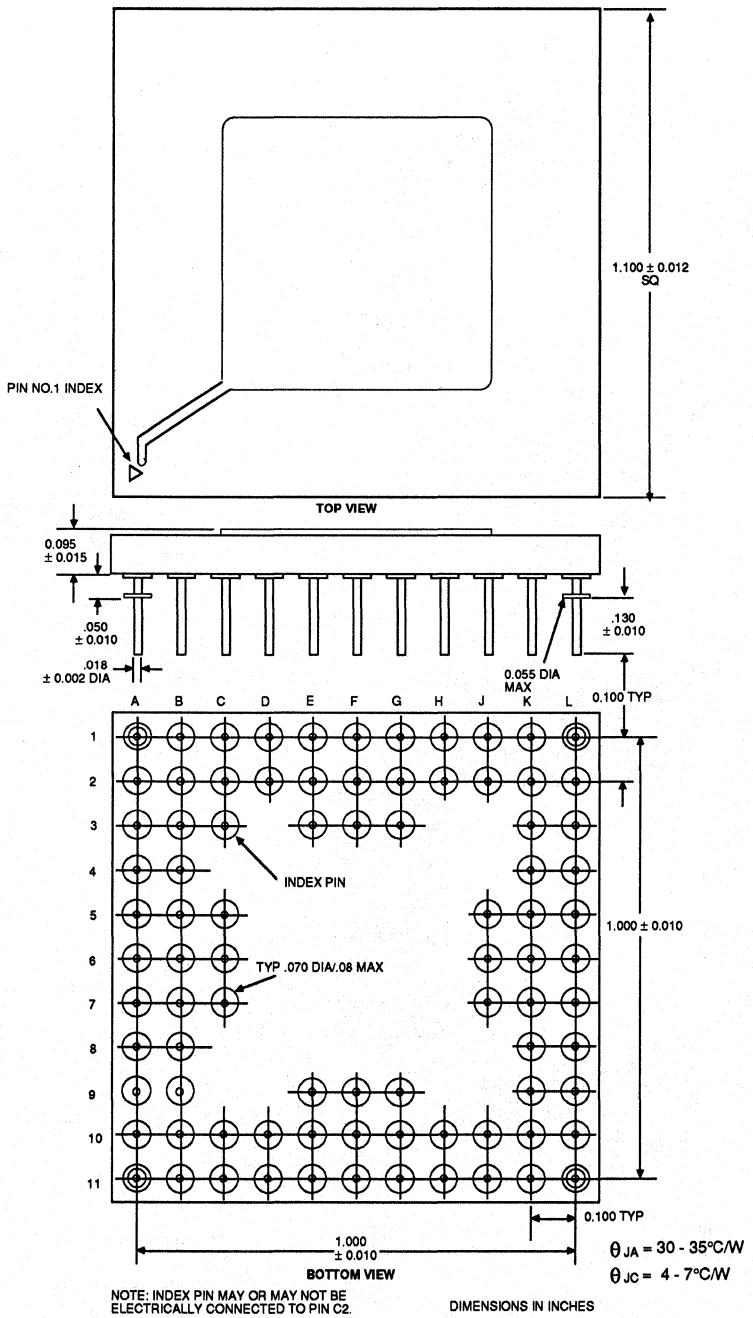
Pin No. CQFP	XC3020
1	GND
2	A13-I/O
3	A6-I/O
4	A12-I/O
5	A7-I/O
6	NC
7	NC
8	A11-I/O
9	A8-I/O
10	A10-I/O
11	A9-I/O
12	NC
13	NC
14	$\overline{\text{PWRDN}}$
15	I/O
16	NC
17	NC
18	NC
19	I/O
20	I/O
21	I/O
22	I/O
23	I/O
24	I/O
25	I/O
26	Vcc
27	I/O
28	I/O
29	I/O
30	I/O
31	I/O
32	I/O
33	I/O
34	I/O

Pin No. CQFP	XC3020
35	NC
36	NC
37	M1-$\overline{\text{RD}}$
38	NC
39	MO-RT
40	NC
41	M2-I/O
42	HDC-I/O
43	I/O
44	$\overline{\text{LDC}}$-I/O
45	NC
46	NC
47	I/O
48	I/O
49	I/O
50	$\overline{\text{INIT}}$-I/O
51	GND
52	I/O
53	I/O
54	I/O
55	I/O
56	I/O
57	I/O
58	I/O
59	NC
60	NC
61	XTAL2-I/O
62	NC
63	$\overline{\text{RESET}}$
64	NC
65	DONE-$\overline{\text{PG}}$
66	D7-I/O
67	XTAL1-I/O
68	D6-I/O

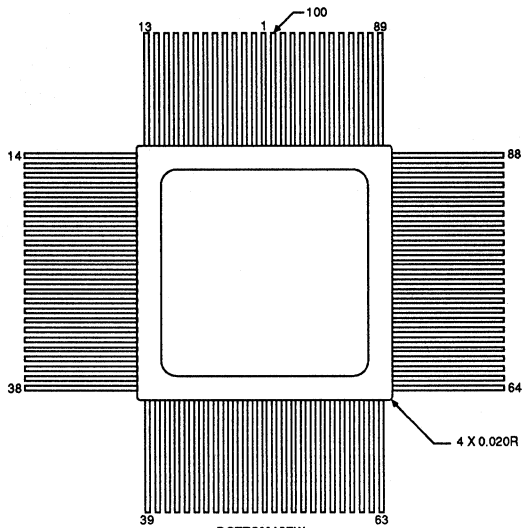
Pin No. CQFP	XC3020
69	NC
70	NC
71	I/O
72	D5-I/O
73	$\overline{\text{CS0}}$-I/O
74	D4-I/O
75	I/O
76	Vcc
77	D3-I/O
78	$\overline{\text{CS1}}$-I/O
79	D2-I/O
80	I/O
81	NC
82	NC
83	D1-I/O
84	$\overline{\text{RCLK}}$-$\overline{\text{BUSY}}$/RDY-I/O
85	DO-DIN-I/O
86	DOUT-I/O
87	CCLK
88	NC
89	NC
90	AO-$\overline{\text{WS}}$-I/O
91	A1-CS2-I/O
92	NC
93	A2-I/O
94	A3-I/O
95	NC
96	NC
97	A15-I/O
98	A4-I/O
99	A14-I/O
100	A5-I/O

Unprogrammed IOBs have a default pull-up. This prevents an undefined pad level for unbonded or unused IOBs. Programmed outputs are default slew-rate limited.

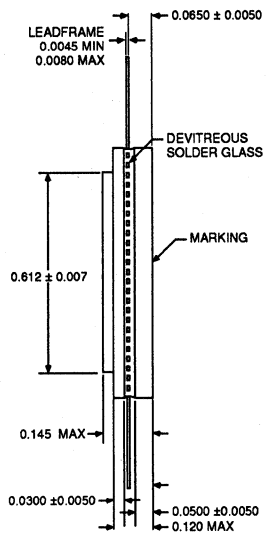
PHYSICAL DIMENSIONS – Conforms to MIL-M-38510 Appendix C, Case P-BC.



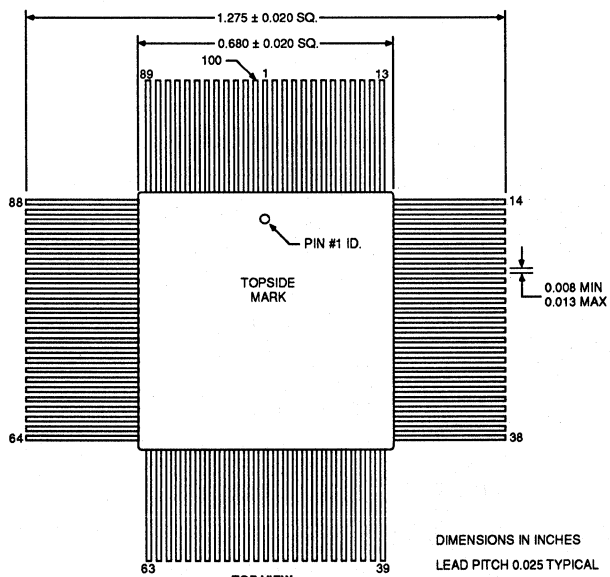
PHYSICAL DIMENSIONS (Continued)



BOTTOM VIEW
(LID SIDE UP)
(DIE FACING UP)



SIDE VIEW



TOP VIEW
(DIE FACING DOWN)

DIMENSIONS IN INCHES

LEAD PITCH 0.025 TYPICAL

PIN SPACING 0.025 TYPICAL

Θ_{JA} = 40-50° C/W
Θ_{JC} = 5-8° C/W

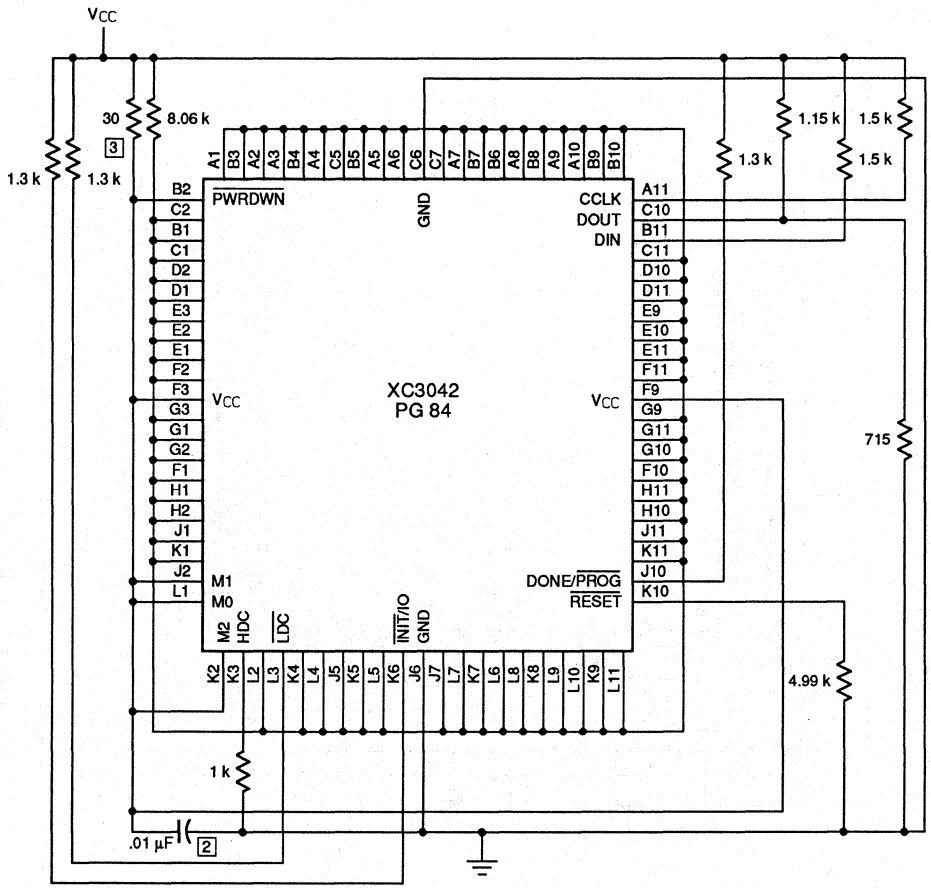
NOTES:
1. LEADS ARE SHIPPED UNFORMED IN CARRIERS IN TRAYS, TOPSIDE UP
2. FORMING TOOL INFORMATION:
– FANCORT INDUSTRIES – (201) 575-0610 WEST CALDWELL NJ.
– RISI INDUSTRIES (619) 425-3970 CHULA VISTA, CA.

1105 40-1

TSC0085

100-Pin CQFP Package

STATIC BURN-IN CIRCUITS



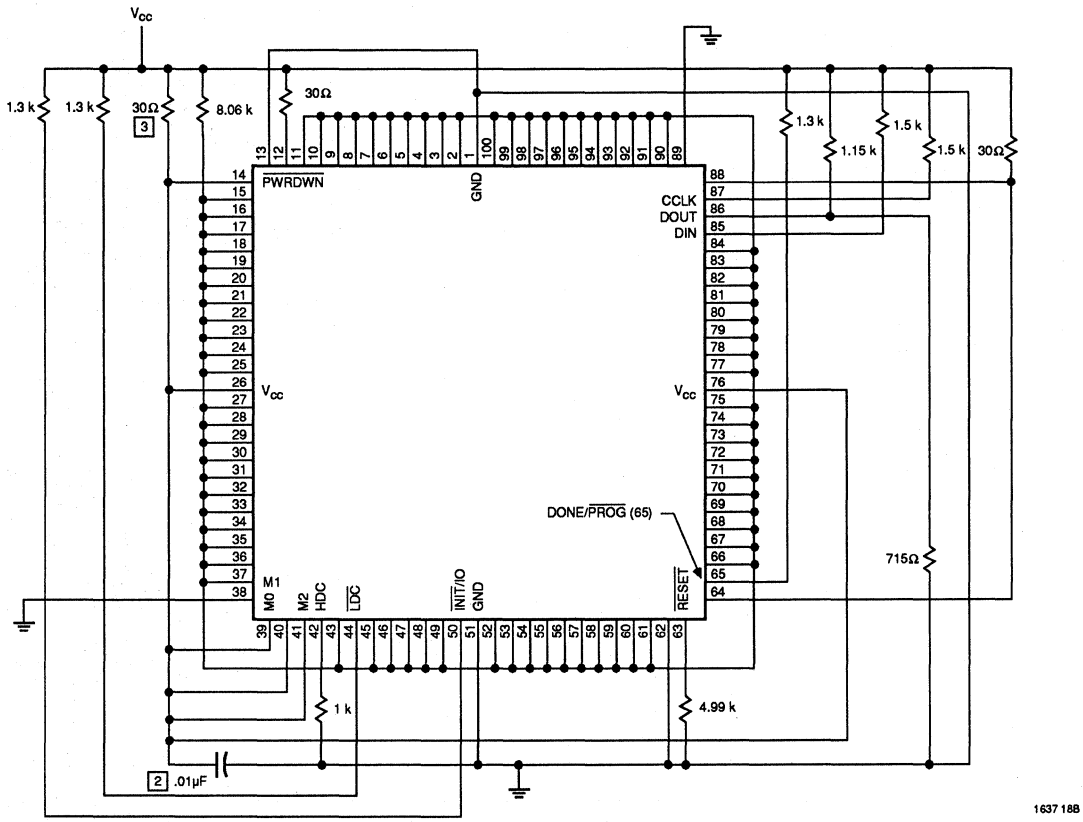
NOTES:

1. UNLESS OTHERWISE SPECIFIED, ALL RESISTORS ARE METAL FILM AND ARE RATED FOR 1/8 WATT AT 150°C WITH A BUILD TOLERANCE OF 1% AND A 5% TOLERANCE OVER LIFE.
2. CAPACITOR HAS 10% TOLERANCE, 50 V RATING WITH AN X7R TEMPERATURE CHARACTERISTIC.
3. 30 Ω RESISTOR IS METAL OXIDE AND IS RATED FOR 1 W AT 150°C WITH A TOLERANCE OF 5%.

1637 18

TSC0085

STATIC BURN-IN CIRCUITS (Continued)



NOTES:
1. UNLESS OTHERWISE SPECIFIED, ALL RESISTORS ARE METAL FILM AND ARE RATED FOR 1/8 W AT 150°C WITH A BUILD TOLERANCE OF 1% AND 5% TOLERANCE OVER LIFE.
2. CAPACITOR HAS 10% TOLERANCE, 50 V RATING WITH AN X7R TEMPERATURE CHARACTERISTICS.
3. 30-Ω RESISTOR IS METAL OXIDE AND IS RATED FOR 1 W AT 150°C WITH A TOLERANCE OF 5%.
4. USE ON: XC3020-XXCQ100X
5. UNLESS OTHERWISE SPECIFIED, SOCKET SHALL BE:
   ENPLAS
   PART NUMBER FPQ-132-0.635-01
   OR
   WELLS
   PART NUMBER CP-10582

XC3020-CQ100

TSC0085

TEST SPECIFICATION

Xilinx maintains test specifications for each product as controlled documents. To insure the use of the most recently released device performance parameters, please request a copy of the current test-specification revision.

Absolute Maximum Ratings		Limits	Units
V_{CC}	Supply voltage relative to GND	–0.5 to +7.0	V
V_{IN}	Input voltage with respect to GND	–0.5 to VCC +0.5	V
V_{TS}	Voltage applied to three-state output	–0.5 to VCC +0.5	V
T_{STG}	Storage temperature (ambient)	–65 to +150	°C
T_{SOL}	Maximum soldering temperature (10 sec @ 1/16 in.)	+260	°C
T_J	Maximum junction temperature	+150	°C

Note: Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those listed under Recommended Operating Conditions is not implied. Exposure to Absolute Maximum Ratings conditions for extended periods of time may affect device reliability.

Test	Symbol	Conditions –55°C ≤ Tc ≤ +125°C V_{CC} = 5.0 V ±10%	Group A Subgroups	Limits Min	Limits Max	Units
High Level Output Voltage	V_{OH}	Vcc = 4.5 V, I_{OH} = –4.0 mA	1,2,3	3.7		V
Low Level Output Voltage	V_{OL}	Vcc = 5.5 V, I_{OL} = 4.0 mA	1,2,3		0.4	V
Quiescent Operating[2] Power Supply Current	I_{CCO}	CMOS Mode, V_{IN} = V_{CC} = 5.5 V	1,2,3		1	mA
		TTL Mode, V_{IN} = V_{CC} = 5.5 V	1,2,3		15	mA
Power-Down Supply Current	I_{CCPD}	V_{IN} = V_{CC} = 5.5 V, $\overline{PWR}$ $\overline{DWN}$ = 0 V	1,2,3		0.5	mA
Leakage Current	I_{IL}	V_{CC} = 5.5 V, V_{IN} = V_{CC} and 0 V	1,2,3	–20	20	µA
Horizontal Long Line Pull-up Current	I_{RLL}	Measured as an average	1,2,3		2.4	mA
Input High Level TTL	V_{IHT}	Guaranteed Input High	1,2,3	2.0		V
Input Low Level TTL	V_{ILT}	Guaranteed Input Low	1,2,3		0.8	V
Input High Level CMOS	V_{IHC}	Guaranteed Input High	1,2,3	.7 V_{CC}		V
Input Low Level CMOS	V_{ILC}	Guaranteed Input Low	1,2,3		.2 V_{CC}	V

Table 1. Electrical Performance Characteristics

TSC0085

Test	Sym	Conditions −55°C ≤ Tc ≤ +125°C Vcc = 5.0 V ±10%	Group A Subgroup	-50 Min	-50 Max	-70 Min	-70 Max	Units
Switching Characteristics, General LCA								
DONE/$\overline{PROG}$								
Program Width (Low)	T_{PGW}	5 See Fig. 1	9,10,11	6		6		µs
Initialization	T_{PGI}	6	9,10,11		7		7	µs
$\overline{PWR\ DWN}$[3]								
Power Down Supply	V_{CCPD}		1,2,3	3.5		3.5		V
RESET[4]								
M2,M1,M0 Setup	T_{MR}	2	9,10,11	1		1		µs
M2,M1,M0 Hold	T_{RM}	3	9,10,11	1		1		µs
Width (low) abort	T_{MRW}	4	9,10,11	6		6		µs
Switching Characteristics, Peripheral Mode Programming[5]								
$\overline{WS}$ LOW	T_{CA}	1 See Fig. 4	9,10,11	0.5		0.5		µs
DIN Setup	T_{DC}	2	9,10,11	60		60		ns
DIN Hold	T_{CD}	3	9,10,11	0		0		ns
Ready/$\overline{Busy}$	T_{WTRB}	4	9,10,11		60		60	ns
Switching Characteristics, Slave Mode Programming[5]								
CCLK,								
To DOUT	T_{CCO}	3 See Fig. 5	9,10,11		100		100	ns
DIN Setup	T_{DCC}	1	9,10,11	60		60		ns
DIN Hold	T_{CCD}	2	9,10,11	0		0		ns
High Time	T_{CCH}	4	9,10,11	0.5		0.5		µs
Low Time	T_{CCL}	5	9,10,11	0.5	1.0	0.5	1.0	µs
Frequency	F_{CC}		9,10,11		1		1	MHz

Table 1. Electrical Performance Characteristics (Continued)

Test	Sym	Conditions −55°C ≤ Tc ≤ +125°C Vcc = 5.0 V ±10%	Group A Subgroups	-50 Min	-50 Max	-70 Min	-70 Max	Units
Switching Characteristics, Program Readback[6,7]								
RTRIG Setup	T_{RTH}	1 See Fig. 7	9,10,11	250		250		ns
CCLK,								
RTRIG Setup	T_{RTCC}	2	9,10,11	200		200		ns
RDATA Delay	T_{CCRD}	3	9,10,11		100		100	ns
Clock Low	T_{CCLR}	4	9,10,11	0.5	1.0	0.5	1.0	ns
Clock High	T_{CCHR}	5	9,10,11	0.5		0.5		ns
Benchmark Patterns[8]								
T_{PID} + interconnect + 8 (T_{ILO}) + T_{OP}. Measured on 8 cols.	T_{B1}		9,10,11		135		86	ns
T_{CKO} + T_{ICK} + T_{CKI} + interconnect	T_{B2}	Tested on all CLBs	9,10,11		32		21	ns
T_{CKO} + T_{QLO} + T_{ILO} + T_{DICK} + interconnect	T_{B3}	Tested on all CLBs	9,10,11		53		34	ns
T_{ILO} + T_{ECCK} + interconnect	T_{B4}	Tested on all CLBs	9,10,11		35		23	ns
T_{OKPO} + T_{OPS} − T_{OPF} + T_{PICK}	T_{B5}	Tested on all CLBs	9,10,11		73		53	ns
T_{CKO} + T_{QLO} + T_{PUS} + T_{ICK} + interconnect	T_{B6}	One long line pull-up	9,10,11		73		48	ns
T_{CKO} + T_{QLO} + T_{PUS} + T_{ICK} + interconnect	T_{B7}	The other long line pull-up	9,10,11		83		55	ns
T_{CKO} + T_{QLO} + T_{IO} + T_{ICK} + interconnect	T_{B8}	No pull-up, lower long lines	9,10,11		47		31	ns
T_{CKO} + T_{QLO} + T_{IO} + T_{ICK} + interconnect	T_{B9}	No pull-up, upper long lines	9,10,11		57		38	ns

Table 1. Electrical Performance Characteristics (Continued)

TSC0085

Test	Sym	Conditions $-55°C \le Tc \le +125°C$ $V_{CC} = 5.0\ V \pm 10\%$	Group A Subgroups	-50 Min	-50 Max	-70 Min	-70 Max	Units
Application Guidelines, Switching, CLB[8]								
Combinatorial	T_{ILO}	1 See Fig. 2	N/A		14		9	ns
Reset to CLB output	T_{RIO}	9	N/A		12		8	ns
Reset Direct width	T_{RPW}	13	N/A	12		8		ns
Master Reset pin to CLB out	T_{MRQ}		N/A		30		20	ns
K Clock[9]								
To CLB output	T_{CKO}	8	N/A		12		8	ns
Additional for Q returning through F or G to CLB out	T_{QLO}		N/A		11		7	ns
Logic-input setup	T_{ICK}	2	N/A	12		8		ns
Logic-input hold	T_{CKI}	3	N/A	1		1		ns
Data In setup	T_{DICK}	4	N/A	8		5		ns
Data In hold (1)	T_{CKDI}	5	N/A	6		4		ns
Enable Clock setup	T_{ECCK}	6	N/A	10		7		ns
Enable Clock hold	T_{CKEC}	7	N/A	0		0		ns
*Clock (High)	T_{CH}	11	N/A	9		7		ns
*Clock (Low)	T_{CL}	12	N/A	9		7		ns

*These parameters are for clock pulses within an LCA device. For externally applied clock, increase values by 20%.

Application Guidelines, Switching, Internal Buffers[8]								
Clock Buffer	T_{GCK}		N/A		9		6	ns
TBUF								
Data to Output	T_{IO}		N/A		8		5	ns
Three-state to Output								
Single Pull-up	T_{PUS}		N/A		34		22	ns
Pair of Pull-ups	T_{PUF}		N/A		17		11	ns
Bidirectional	T_{BIDI}		N/A		6		4	ns

Table 1. Electrical Performance Characteristics (Continued)

Test	Sym	Conditions −55°C ≤ Tc ≤ +125°C V_{CC} = 5.0 V ±10%	Group A Subgroups	-50		-70		Units
				Min	Max	Min	Max	
Application Guidelines, Switching, IOB[8,10]								
Pad (package pin)		See Fig. 3						
To inputs TCLKIN, BCLKIN	T_{PIDC}		N/A		5		5	ns
To inputs DIRECT IN	T_{PID}	3	N/A		9		6	ns
I/O Clock								
To I/O RI input (FF)	T_{IKRI}	4	N/A		11		7	ns
I/O pad-input setup	T_{PICK}	1	N/A	30		20		ns
I/O pad-input hold	T_{IKPI}	2	N/A	0		0		ns
To I/O pad (fast)	T_{OKPO}	7	N/A		18		13	ns
I/O pad output setup	T_{OOK}	5	N/A	15		10		ns
I/O pad output hold	T_{OKO}	6	N/A	0		0		ns
*Clock (High)	T_{IOH}	11	N/A	9		8		ns
*Clock (Low)	T_{IOL}	12	N/A	9		8		ns
Output								
To pad (enabled fast)	T_{OPF}	10	N/A		15		9	ns
To pad (enabled slow)	T_{OPS}	10	N/A		40		29	ns
Three-State								
To pad begin hi-Z (fast)	T_{TSHZ}	9	N/A		18		12	ns
To pad valid (fast)	T_{TSON}	8	N/A		20		14	ns
Master Reset								
To input RI	T_{RRI}	13	N/A		35		23	ns
To output (FF)	T_{RPO}	14	N/A		50		33	ns

*These parameters are for clock pulses within an LCA device. For externally applied clock, increase values by 20%.

Table 1. Electrical Performance Characteristics (Continued)

Test	Sym	Conditions –55°C ≤ Tc ≤ +125°C V_{CC} = 5.0 V ±10%	Group A Subgroups	-50		-70		Units
				Min	Max	Min	Max	
Application Guidelines, Switching, Master Parallel Mode Programming[8,11]								
$\overline{RCLK}$,								
To Address Valid	T_{RAC}	1 See Fig.6	N/A	0	200	0	200	ns
To Data Setup	T_{DRC}	2	N/A	60		60		ns
To Data Hold	T_{RCD}	3	N/A	0		0		ns
$\overline{RCLK}$ High	T_{RCH}	4	N/A	600		600		ns
$\overline{RCLK}$ Low	T_{RCL}	5	N/A	4.0		4.0		µs

Table 1. Electrical Performance Characteristics (Continued)

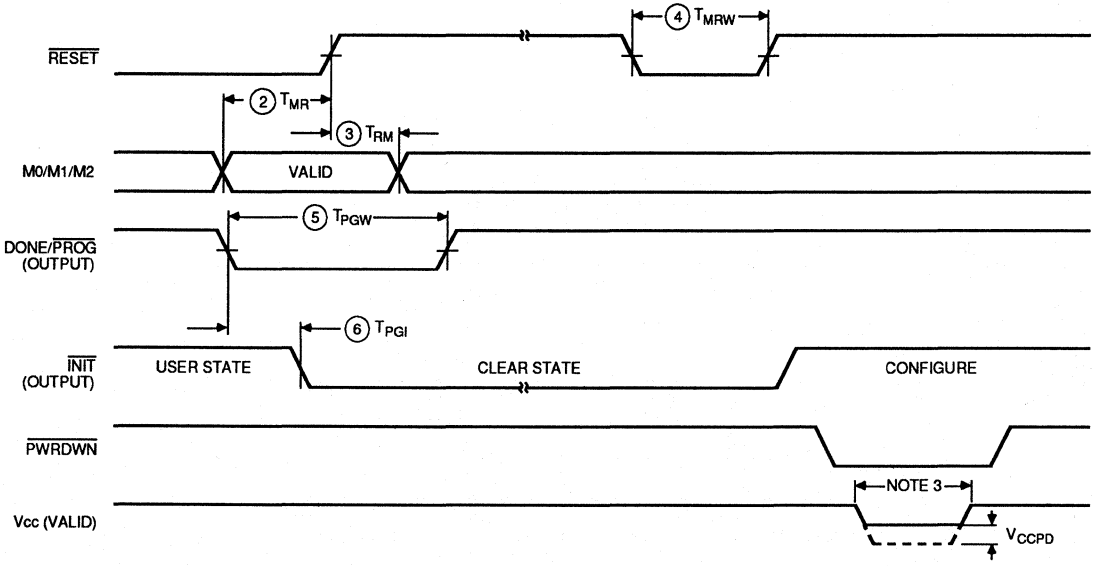


Figure 1. General LCA Waveforms

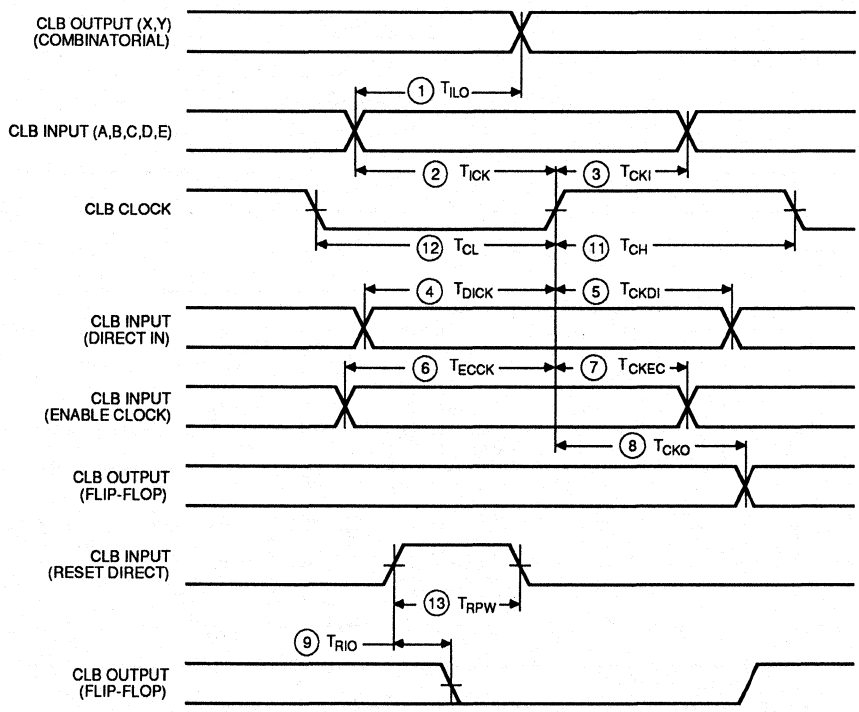


Figure 2. CLB Waveforms

TSC0085

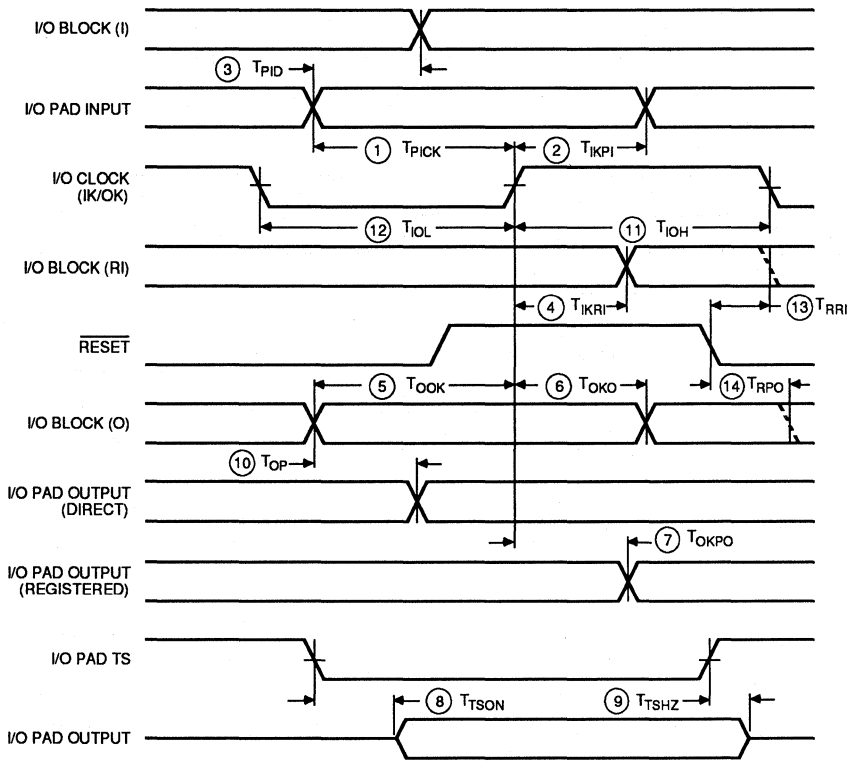


Figure 3. IOB Waveforms

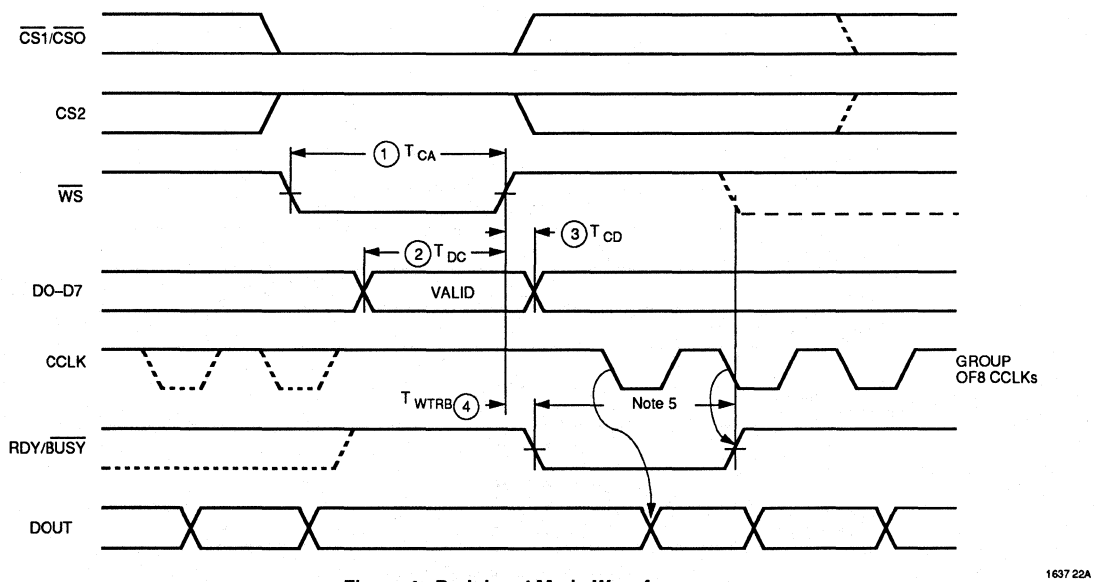


Figure 4. Peripheral Mode Waveforms

TSC0085

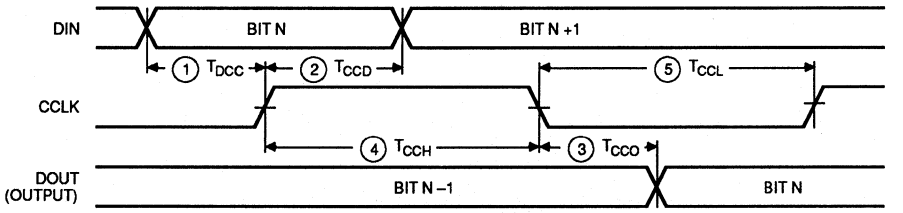


Figure 5. Slave Mode Waveforms

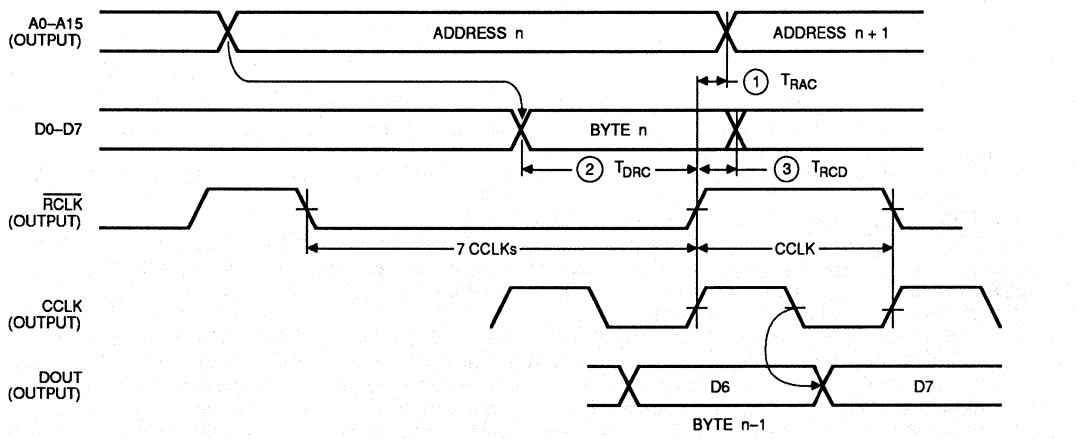
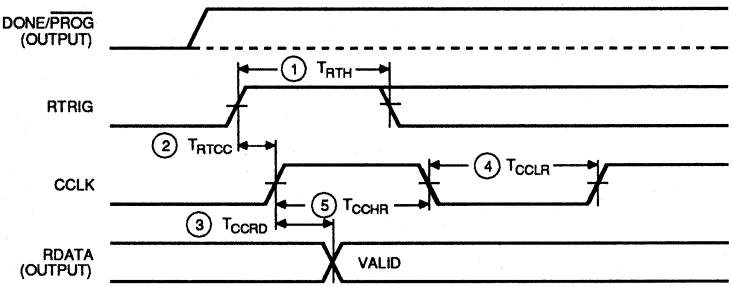


Figure 6. Master Parallel Mode Waveforms

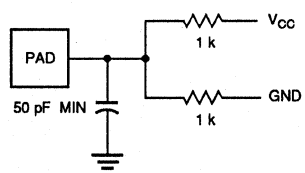


1637 25

Figure 7. Program Readback Waveforms

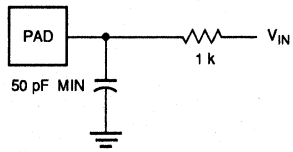
XC3020B Data Sheet Notes

1. Xilinx maintains this specification as a controlled document. To comply with the intent of MIL-STD-883, and to insure the use of the most recently released device performance parameters, please request a copy of the current revision of this Test Specification (TSC 0085) from Xilinx.
2. No output current loads, no active input or long line pull-up resistors, and with the device configured with the MAKEBITS "tie" option.
3. $\overline{\text{PWRDWN}}$ transitions must occur during operational Vcc levels.
4. $\overline{\text{RESET}}$ timing relative to valid mode lines (M0, M1, M2) is relevant only when $\overline{\text{RESET}}$ is used to delay configuration.
5. Configuration must be delayed until the INIT of all LCA's is HIGH. $\overline{\text{WS}}$ cannot go active until RDY/$\overline{\text{BUSY}}$ goes HIGH.
6. Readback should not be initiated until configuration is complete.
7. DOUT timing is the same as for slave mode.
8. Testing of the Applications Guidelines is modeled after testing specified by MIL-M-38510/605. Devices are first 100% functionally tested. Benchmark patterns are then used to measure the Application Guidelines. Characterization data are taken at initial device qualification, prior to introduction of significant changes, and at least twice yearly to monitor correlation between benchmark patterns, device performance, XACT software timings, and the data sheet.
9. The CLB K to Q output delay (T_{CKO}) plus the shortest possible interconnect delay is always longer than the Data In hold time requirement (T_{CKDI}) on the same die.
10. Voltage levels of unused pads must be valid logic levels. Each can be configured with the internal pull-up resistor, configured as a driven output, or driven from an external source.
11. At power-up, V_{CC} must rise from 2.0 V to Vcc minimum in less than 10 ms. Otherwise, delay configuration using $\overline{\text{RESET}}$.
12. All timings except T_{TSHZ} and T_{TSON} are measured at 1.5 V level with 50 pF minimum load output. For input signals, rise and fall times are less than 6 ns, with low amplitude = 0 V, and high = 3 V.
    T_{TSHZ} is determined when the output shifts 10% (of the output voltage swing) from V_{OL} level or V_{OH} level. The following circuit is used:



1637 13

12. (continued)
    T_{TSON} is measured at 0.5 V_{CC} level with V_{IN} = 0 for 3-State to active High, and V_{IN} = Vcc for 3-State to active Low. The following load circuit is used:



1637 14

XC3042B
Military Logic Cell™ Array

Product Specification. See Note 1.

FEATURES

- MIL-STD-883 Class B Processing. Complies with paragraph 1.2.1
- Field-programmable gate array
- Low power CMOS static memory technology
- Standard product. Completely tested at factory
- Design changes made in minutes
- Complete user control for design cycle. Secure design process
- Complete PC or workstation based development system
  - Schematic entry
  - Auto Place/ Route
  - Design Editor
  - Logic & Timing Simulator
  - XACTOR In-circuit Verifier

Part Number	Logic Capacity (gates)	Configurable Logic Blocks	User I/O's	Configuration Program (bits)
XC3042	4200	144	96	30784

DESCRIPTION

The Logic Cell™ Array (LCA) is a high density CMOS programmable gate array. Its patented array architecture consists of three types of configurable elements: Input/ Output Blocks, Configurable Logic Blocks and Interconnect. The designer can define individual I/O blocks for interface to external circuitry, define logic blocks to implement logic functions and define interconnection networks to compose larger scale logic functions.

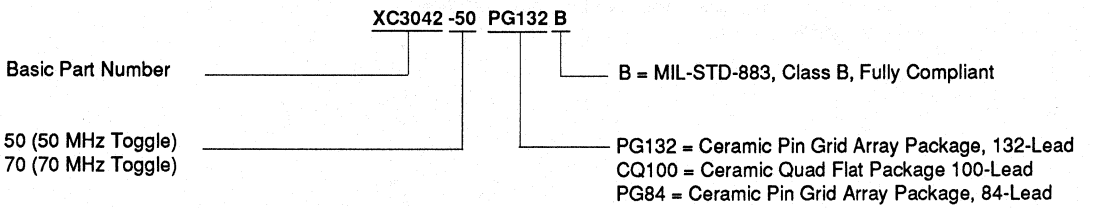
The Logic Cell Array's logic functions and interconnections are determined by the configuration program stored in internal static memory cells. On-chip logic provides for automatic loading of configuration data at power-up or on command. The program data can reside in an EEPROM, EPROM or ROM on the circuit board or on a floppy disk or hard disk.

Several methods of automatically loading the required data are designed into the Logic Cell Array and are determined by logic levels applied to mode selection pins at configuration time. The form of the data may be either serial or parallel, depending on the configuration mode. The programming data are independent of the configuration mode selected.

The XACT development system allows the user to define the logic functions of the device. Schematic capture is available for design entry, while logic and timing simulation, and in-circuit debugging are available for design verification. XACT is used to compile the data pattern which represents the configuration program. This data can then be converted to a PROM programmer format file to create the configuration program storage.

See the XC3000 Commercial data sheet for a full description.

ORDERING INFORMATION



TSC0117

1637 26

PIN ASSIGNMENTS

CONFIGURATION MODE: <M2:M1:M0>					84 PGA	100 CQFP	132 PGA	USER OPERATION
SLAVE <1:1:1>	MASTER-SER <0:0:0>	PERIPHERAL <1:0:1>	MASTER-HIGH <1:1:0>	MASTER-LOW <1:0:0>				
PWR DWN (I)	PWR DWN (I)	PWR DWN (I)	PWR DWN (I)	PWR DWN (I)	B2	14	A1	PWR DWN (I)
VCC	VCC	VCC	VCC	VCC	F3	26	C8	VCC
M1 (HIGH) (I)	M1 (LOW) (I)	M1 (LOW) (I)	M1 (HIGH) (I)	M1 (LOW) (I)	J2	37	B13	RDATA
M0 (HIGH) (I)	M0 (LOW) (I)	M0 (HIGH) (I)	M0 (LOW) (I)	M0 (LOW) (I)	L1	39	B14	RTRIG (I)
M2 (HIGH) (I)	M2 (LOW) (I)	M2 (HIGH) (I)	M2 (HIGH) (I)	M2 (HIGH) (I)	K2	41	C13	I/O
HDC (HIGH)	HDC (HIGH)	HDC (HIGH)	HDC (HIGH)	HDC (HIGH)	K3	42	B14	I/O
LDC (LOW)	LDC (LOW)	LDC (LOW)	LDC (LOW)	LDC (LOW)	L3	44	D14	I/O
INIT*	INIT*	INIT*	INIT*	INIT*	K6	50	G14	I/O
GND	GND	GND	GND	GND	J6	51	H12	GND
					L11	61	M13	XTL2 OR I/O
RESET (I)	RESET (I)	RESET (I)	RESET (I)	RESET (I)	K10	63	P14	RESET (I)
DONE	DONE	DONE	DONE	DONE	J10	65	N13	PROGRAM (I)
		DATA 7 (I)	DATA 7 (I)	DATA 7 (I)	K11	66	M12	I/O
					J11	67	P13	XTL1 OR I/O
		DATA 6 (I)	DATA 6 (I)	DATA 6 (I)	H10	68	N11	I/O
		DATA 5 (I)	DATA 5 (I)	DATA 5 (I)	F10	72	M9	I/O
		CS0 (I)			G10	73	N9	I/O
		DATA 4 (I)	DATA 4 (I)	DATA 4 (I)	G11	74	N8	I/O
VCC	VCC	VCC	VCC	VCC	F9	76	M8	VCC
		DATA 3 (I)	DATA 3 (I)	DATA 3 (I)	F11	77	N7	I/O
		CS1 (I)			E11	78	P6	I/O
		DATA 2 (I)	DATA 2 (I)	DATA 2 (I)	E10	79	M6	I/O
		DATA 1 (I)	DATA 1 (I)	DATA 1 (I)	D10	83	M5	I/O
		RDY/BUSY	RCLK	RCLK	C11	84	N4	I/O
DIN (I)	DIN (I)	DATA 0 (I)	DATA 0 (I)	DATA 0 (I)	B11	85	N2	I/O
DOUT	DOUT	DOUT	DOUT	DOUT	C10	86	M3	I/O
CCLK (I)	CCLK	CCLK	CCLK	CCLK	A11	87	P1	CCLK (I)
		WS (I)	A0	A0	B10	90	M2	I/O
		CS2 (I)	A1	A1	B9	91	N1	I/O
			A2	A2	A10	93	L2	I/O
			A3	A3	A9	94	L1	I/O
			A15	A15	B6	97	K1	I/O
			A4	A4	B7	98	J2	I/O
			A14	A14	A7	99	H1	I/O
			A5	A5	C7	100	H2	I/O
GND	GND	GND	GND	GND	C6	1	H3	GND
			A13	A13	A6	2	G2	I/O
			A6	A6	A5	3	G1	I/O
			A12	A12	B5	4	F2	I/O
			A7	A7	C5	5	E1	I/O
			A11	A11	A3	8	D1	I/O
			A8	A8	A2	9	D2	I/O
			A10	A10	B3	10	B1	I/O
			A9	A9	A1	11	C2	I/O

REPRESENTS A 50 kΩ TO 100 kΩ PULL-UP

* INIT IS AN OPEN-DRAIN OUTPUT DURING CONFIGURATION

(I) REPRESENTS AN INPUT

X1130

TSC0117

PIN ASSIGNMENTS (Continued)

84 PGA Pin Number	XC3042	84 PGA Pin Number	XC3042
B2	PWRDN	K10	RESET
C2	TCLKIN-I/O	J10	DONE-PG
B1	I/O	K11	D7-I/O
C1	I/O	J11	XTL1(OUT)-BLCKIN-I/O
D2	I/O	H10	D6-I/O
D1	I/O	H11	I/O
E3	I/O	F10	D5-I/O
E2	I/O	G10	CS0-I/O
E1	I/O	G11	D4-I/O
F2	I/O	G9	I/O
F3	VCC	F9	VCC
G3	I/O	F11	D3-I/O
G1	I/O	E11	CS1-I/O
G2	I/O	E10	D2-I/O
F1	I/O	E9	I/O
H1	I/O	D11	I/O
H2	I/O	D10	D1-I/O
J1	I/O	C11	RDY/BUSY-RCLK-I/O
K1	I/O	B11	D0-DIN-I/O
J2	M1-RDATA	C10	DOUT-I/O
L1	M0-RTRIG	A11	CCLK
K2	M2-I/O	B10	A0-WS-I/O
K3	HDC-I/O	B9	A1-CS2-I/O
L2	I/O	A10	A2-I/O
L3	LDC-I/O	A9	A3-I/O
K4	I/O	B8	I/O
L4	I/O	A8	I/O
J5	I/O	B6	A15-I/O
K5	I/O	B7	A4-I/O
L5	I/O	A7	A14-I/O
K6	INIT-I/O	C7	A5-I/O
J6	GND	C6	GND
J7	I/O	A6	A13-I/O
L7	I/O	A5	A6-I/O
K7	I/O	B5	A12-I/O
L6	I/O	C5	A7-I/O
L8	I/O	A4	I/O
K8	I/O	B4	I/O
L9	I/O	A3	A11-I/O
L10	I/O	A2	A8-I/O
K9	I/O	B3	A10-I/O
L11	XTL2(IN)-I/O	A1	A9-I/O

Unprogrammed IOBs have a default pull-up. This prevents an undefined pad level for unbonded or unused IOBs. Programmed outputs are default slew-rate limited.

TSC0117

PIN ASSIGNMENTS (Continued)

CQFP Pin No.	PGA Pin Number	Function
13	C4	GND
14	A1	$\overline{\text{PWRDN}}$
15	C3	I/O
16	B2	I/O
17	B3	I/O
	B4	I/O *
18	C5	I/O
19	A4	I/O
20	B5	I/O
21	C6	I/O
22	A5	I/O
23	B6	I/O
24	A6	I/O
25	B7	I/O
	C7	GND *
26	C8	VCC
27	A7	I/O
28	B8	I/O
29	A8	I/O
30	A9	I/O
31	B9	I/O
32	C9	I/O
33	A10	I/O
	B10	I/O *
34	C10	I/O
35	B11	I/O
36	B12	I/O
	C12	I/O *
37	B13	M1-$\overline{\text{RD}}$
38	C11	GND
39	A14	M0-RT
40	D12	VCC
41	C13	M2-I/O
42	B14	HDC-I/O
	C14	I/O *
	E12	I/O *
43	D13	I/O
44	D14	$\overline{\text{LDC}}$-I/O
45	F12	I/O
46	E14	I/O

CQFP Pin No.	PGA Pin Number	Function
47	F13	I/O
48	F14	I/O
49	G13	I/O
50	G14	INIT-I/O
	G12	VCC *
51	H12	GND
52	H14	I/O
53	H13	I/O
54	J14	I/O
55	J13	I/O
56	K14	I/O
57	J12	I/O
	K13	I/O *
	L13	I/O *
58	K12	I/O
59	M14	I/O
60	N14	I/O
61	M13	XTAL2-I/O
62	L12	GND
63	P14	$\overline{\text{RESET}}$
64	M11	VCC
65	N13	DONE-$\overline{\text{PG}}$
66	M12	D7-I/O
67	P13	XTAL1-I/O
	N12	I/O *
	P12	I/O *
68	N11	D6-I/O
69	M10	I/O
70	N10	I/O
71	P10	I/O
72	M9	D5-I/O
73	N9	$\overline{\text{CS0}}$-I/O
74	N8	D4-I/O
75	P7	I/O
76	M8	VCC
	M7	GND *
77	N7	D3-I/O
78	P6	$\overline{\text{CS1}}$-I/O
79	M6	D2-I/O
80	N5	I/O

CQFP Pin No.	PGA Pin Number	Function
81	P4	I/O
82	P3	I/O
83	M5	D1-I/O
84	N4	$\overline{\text{RCLK}}$-$\overline{\text{BUSY}}$/RDY-I/O
	P2	I/O *
	N3	I/O *
85	N2	D0-DIN-I/O
86	M3	DOUT-I/O
87	P1	CCLK
88	M4	VCC
89	L3	GND
90	M2	A0-$\overline{\text{WS}}$-I/O
91	N1	A1-CS2-I/O
92	M1	I/O
	K3	I/O *
93	L2	A2-I/O
94	L1	A3-I/O
95	K2	I/O
96	J3	I/O
97	K1	A15-I/O
98	J2	A4-I/O
99	H1	A14-I/O
100	H2	A5-I/O
1	H3	GND
	G3	VCC *
2	G2	A13-I/O
3	G1	A6-I/O
4	F2	A12-I/O
5	E1	A7-I/O
6	F3	I/O
7	E2	I/O
8	D1	A11-I/O
9	D2	A8-I/O
	E3	I/O *
	C1	I/O *
10	B1	A10-I/O
11	C2	A9-I/O
12	D3	VCC

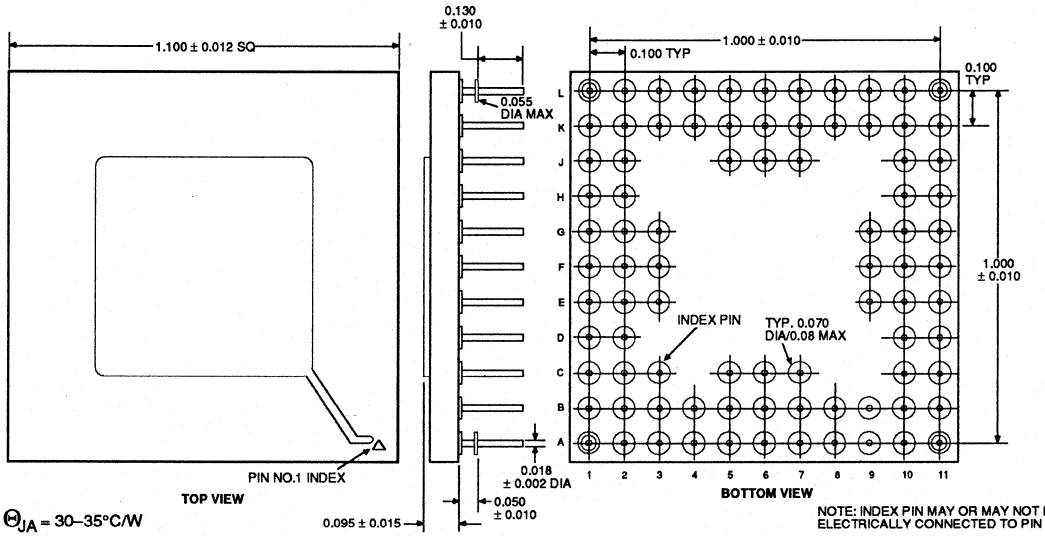
X1101A

Unprogrammed IOBs have a default pull-up. This prevents an undefined pad level for unbonded or unused IOBs. Programmed outputs are default slew-rate limited.

* Indicates unconnected bond pads for the CQFP-100 package

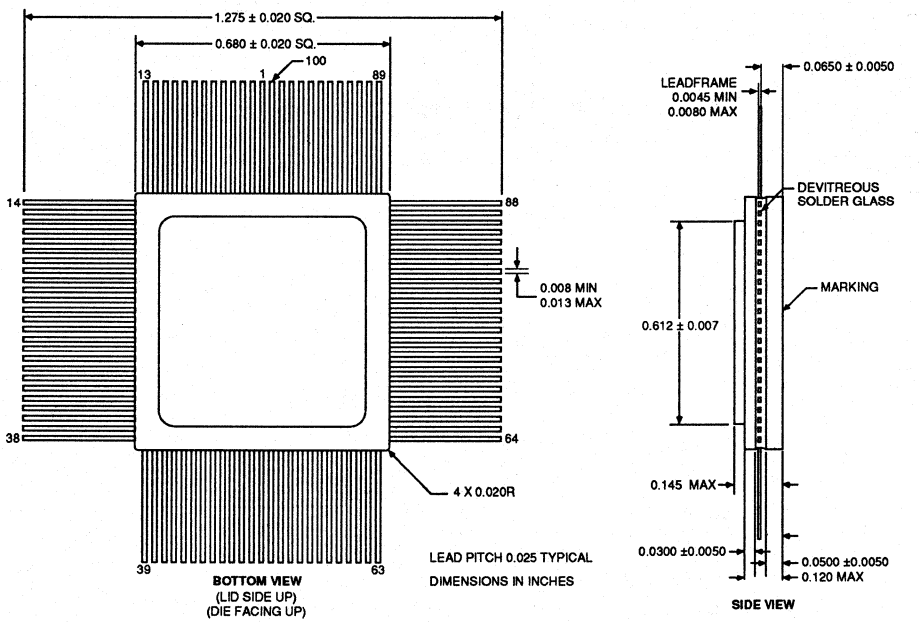
TSC0117

PHYSICAL DIMENSIONS - Conforms to MIL-M-38510 Appendix C, Case P-BC.



84-Pin PGA Package (Cavity Up)

1105 35C



ΘJA = 40-50° C/W
ΘJC = 5-8° C/W

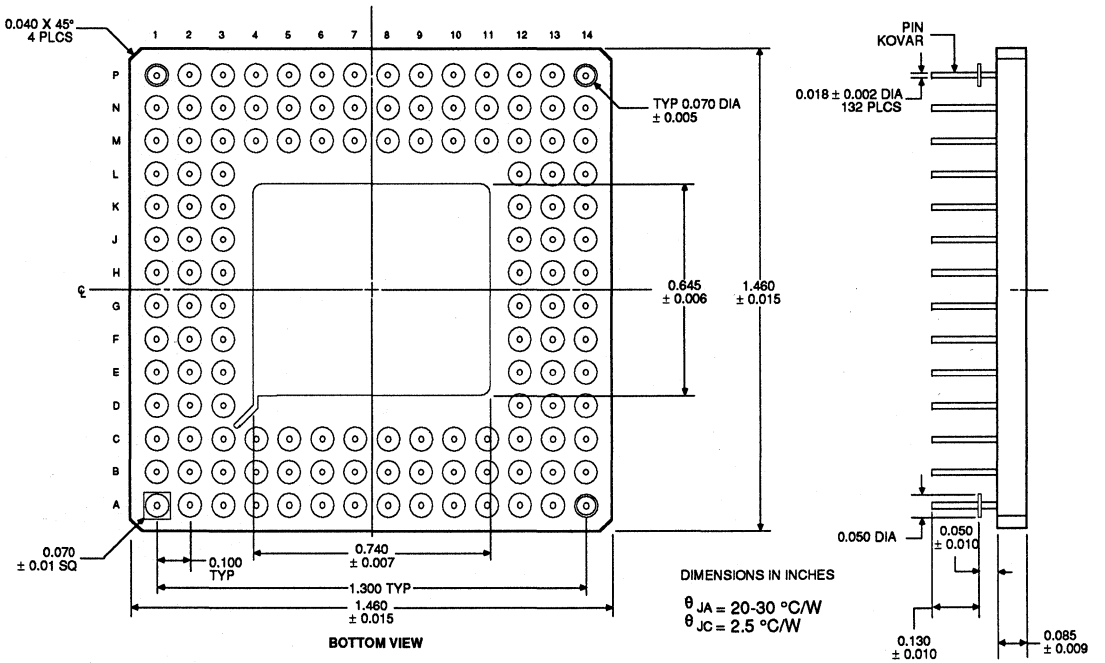
NOTES:
1. LEADS ARE SHIPPED UNFORMED IN CARRIERS IN TRAYS, TOPSIDE UP
2. FORMING TOOL INFORMATION:
– FANCORT INDUSTRIES – (201) 575-0610 WEST CALDWELL NJ.
– RISI INDUSTRIES (619) 425-3970 CHULA VISTA, CA.

TSC0117

100-Pin CQFP Package (Cavity Down)

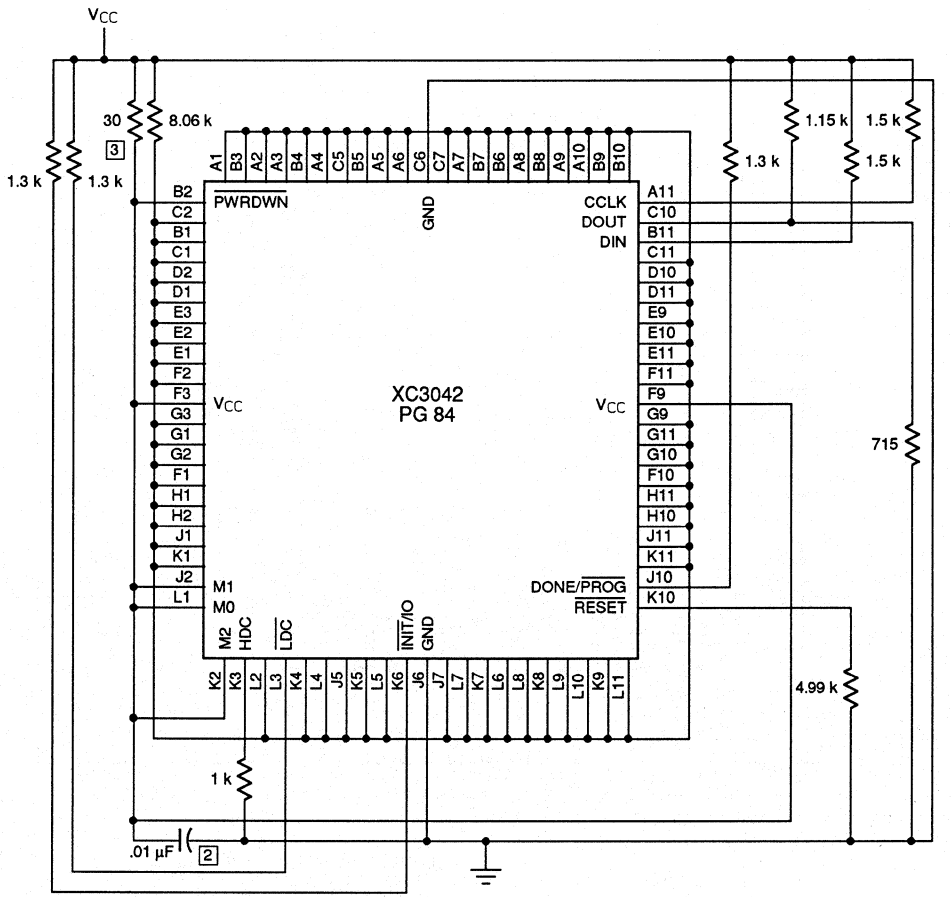
1105 40C

PHYSICAL DIMENSIONS (Continued)



132-Pin PGA Package (Cavity Down)

STATIC BURN-IN CIRCUITS



NOTES:

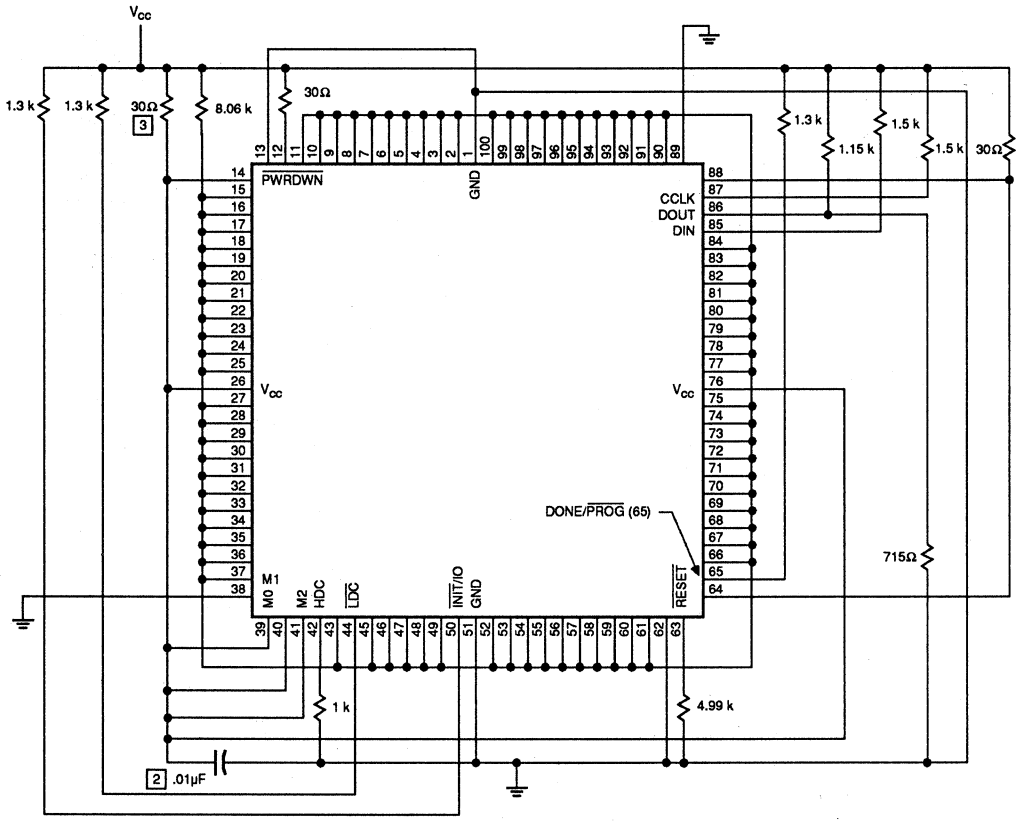
1. UNLESS OTHERWISE SPECIFIED, ALL RESISTORS ARE METAL FILM AND ARE RATED FOR 1/8 WATT AT 150°C WITH A BUILD TOLERANCE OF 1% AND A 5% TOLERANCE OVER LIFE.
2. CAPACITOR HAS 10% TOLERANCE, 50 V RATING WITH AN X7R TEMPERATURE CHARACTERISTIC.
3. 30 Ω RESISTOR IS METAL OXIDE AND IS RATED FOR 1 W AT 150°C WITH A TOLERANCE OF 5%.

1637 18

XC3042-PG84

TSC0117

STATIC BURN-IN CIRCUITS (Continued)

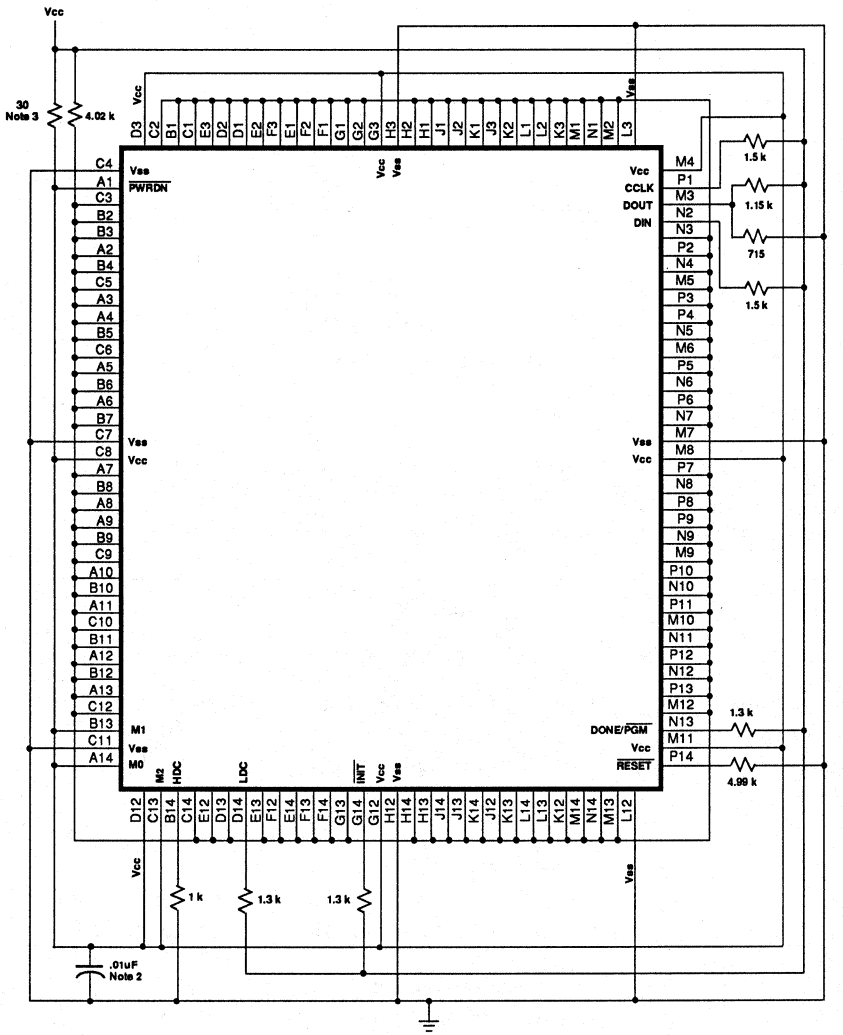


NOTES:
1. UNLESS OTHERWISE SPECIFIED, ALL RESISTORS ARE METAL FILM AND ARE RATED FOR 1/8 W AT 150°C WITH A BUILD TOLERANCE OF 1% AND 5% TOLERANCE OVER LIFE.
2. CAPACITOR HAS 10% TOLERANCE, 50 V RATING WITH AN X7R TEMPERATURE CHARACTERISTICS.
3. 30-Ω RESISTOR IS METAL OXIDE AND IS RATED FOR 1 W AT 150°C WITH A TOLERANCE OF 5%.
4. USE ON: XC3020-XXCQ100X
5. UNLESS OTHERWISE SPECIFIED, SOCKET SHALL BE:
   ENPLAS
   PART NUMBER FPQ-132-0.635-01
   OR
   WELLS
   PART NUMBER CP-10582

1637 18B

XC3042-CQ100

STATIC BURN-IN CIRCUITS (Continued)



X1129

NOTES:
1. UNLESS OTHERWISE SPECIFIED, ALL RESISTORS ARE METAL FILM AND ARE RATED FOR 1/8 W AT 150°C WITH A BUILD TOLERANCE OF 1% AND 5% TOLERANCE OVER LIFE.
2. CAPACITOR HAS 10% TOLERANCE, 50 V RATING WITH AN X7R TEMPERATURE CHARACTERISTICS.
3. 30-Ω RESISTOR IS METAL OXIDE AND IS RATED FOR 1 W AT 150°C WITH A TOLERANCE OF 5%.

XC3042-PG132

TSC0117

TEST SPECIFICATION

Xilinx maintains test specifications for each product as controlled documents. To insure the use of the most recently released device performance parameters, please request a copy of the current test-specification revision.

Absolute Maximum Ratings		Limits	Units
V_{CC}	Supply voltage relative to GND	–0.5 to +7.0	V
V_{IN}	Input voltage with respect to GND	–0.5 to V_{CC} +0.5	V
V_{TS}	Voltage applied to three-state output	–0.5 to V_{CC} +0.5	V
T_{STG}	Storage temperature (ambient)	–65 to +150	°C
T_{SOL}	Maximum soldering temperature (10 sec @ 1/16 in.)	+260	°C
T_J	Maximum junction temperature	+150	°C

Note: Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those listed under Recommended Operating Conditions is not implied. Exposure to Absolute Maximum Ratings conditions for extended periods of time may affect device reliability.

Test	Symbol	Conditions –55°C ≤ Tc ≤ +125°C V_{CC} = 5.0 V ±10%	Group A Subgroups	Limits Min	Limits Max	Units
High Level Output Voltage	V_{OH}	V_{CC} = 4.5 V, I_{OH} = –4.0 mA	1,2,3	3.7		V
Low Level Output Voltage	V_{OL}	V_{CC}= 5.5 V, I_{OL} = 4.0 mA	1,2,3		0.4	V
Quiescent Operating[2] Power Supply Current	I_{CCO}	CMOS Mode, Vin = V_{CC} = 5.5 V	1,2,3		1.650	mA
		TTL Mode, V_{in} = V_{CC} = 5.5 V	1,2,3		11.15	mA
Power-Down Supply Current	I_{CCPD}	V_{in} = V_{CC} = 5.5 V, $\overline{PWR}$ $\overline{DWN}$ = 0 V	1,2,3		1150	µA
Leakage Current	I_{IL}	V_{CC} = 5.5 V, V_{in} = V_{CC} and 0 V	1,2,3	–20	20	µA
Horizontal Long Line Pull-up Current	I_{RLL}	Measured as an average @ V_{CC} = 5.5 V	1,2,3		2.4	mA
Input High Level TTL	V_{IHT}	Guaranteed Input High	1,2,3	2.0		V
Input Low Level TTL	V_{ILT}	Guaranteed Input Low	1,2,3		0.8	V
Input High Level CMOS	V_{IHC}	Guaranteed Input High	1,2,3	.7 V_{CC}		V
Input Low Level CMOS	V_{ILC}	Guaranteed Input Low	1,2,3		.2 V_{CC}	V

Table 1. Electrical Performance Characteristics

TSC0117

Test	Sym	Conditions −55°C ≤ Tc ≤ +125°C V_{CC} = 5.0 V ±10%	Group A Subgroup	-50		-70		Units
				Min	Max	Min	Max	
Switching Characteristics, General LCA								
DONE/$\overline{PROG}$								
Program Width (Low)	T_{PGW}	5 See Fig. 1	9,10,11	6		6		µs
Initialization	T_{PGI}	6	9,10,11		7		7	µs
$\overline{PWR\ DWN}$[3]								
Power Down Supply	V_{CCPD}		1,2,3	3.5		3.5		V
RESET[4]								
M2,M1,M0 Setup	T_{MR}	2	9,10,11	1		1		µs
M2,M1,M0 Hold	T_{RM}	3	9,10,11	1		1		µs
Width (low) abort	T_{MRW}	4	9,10,11	6		6		µs
Switching Characteristics, Peripheral Mode Programming[5]								
$\overline{WRT}$ LOW	T_{CA}	1 See Fig. 4	9,10,11	0.5		0.5		µs
DIN Setup	T_{DC}	2	9,10,11	60		60		ns
DIN Hold	T_{CD}	3	9,10,11	0		0		ns
Ready/$\overline{Busy}$	T_{WTRB}	4	9,10,11		60		60	ns
Switching Characteristics, Slave Mode Programming[5]								
CCLK,								
To DOUT	T_{CCO}	3 See Fig. 5	9,10,11		100		100	ns
DIN Setup	T_{DCC}	1	9,10,11	60		60		ns
DIN Hold	T_{CCD}	2	9,10,11	0		0		ns
High Time	T_{CCH}	4	9,10,11	0.5		0.5		µs
Low Time	T_{CCL}	5	9,10,11	0.5	1.0	0.5	1.0	µs
Frequency	F_{CC}		9,10,11		1		1	MHz

Table 1. Electrical Performance Characteristics (Continued)

TSC0117

Test	Sym	Conditions −55°C ≤ Tc ≤ +125°C Vcc = 5.0 V ±10%	Group A Subgroups	-50 Min	-50 Max	-70 Min	-70 Max	Units
Switching Characteristics, Program Readback[6,7]								
RTRIG Setup	T_{RTH}	1 See Fig. 7	9,10,11	250		250		ns
CCLK,								
RTRIG Setup	T_{RTCC}	2	9,10,11	200		200		ns
RDATA Delay	T_{CCRD}	3	9,10,11		100		100	ns
Clock Low	T_{CCLR}	4	9,10,11	1.2	2.0	1.2	2.0	µs
Clock High	T_{CCHR}	5	9,10,11	0.5		0.5		µs
Benchmark Patterns[8]								
T_{PID} + interconnect + 8 (T_{ILO}) + T_{OP}. Measured on 8 cols.	T_{B1}		9,10,11		191		122	ns
$T_{CKO} + T_{ICK} + T_{CKI}$ + interconnect	T_{B2}	Tested on all CLBs	9,10,11		32		21	ns
$T_{CKO} + T_{QLO} + T_{ILO} + T_{DICK}$ + interconnect	T_{B3}	Tested on all CLBs	9,10,11		53		34	ns
$T_{ILO} + T_{ECCK}$ + interconnect	T_{B4}	Tested on all CLBs	9,10,11		35		23	ns
$T_{OKPO} + T_{OPS} - T_{OPF} + T_{PICK}$	T_{B5}	Tested on all CLBs	9,10,11		73		53	ns
$T_{CKO} + T_{QLO} + T_{PUS} + T_{ICK}$ + interconnect	T_{B6}	One long line pull-up	9,10,11		73		48	ns
$T_{CKO} + T_{QLO} + T_{PUS} + T_{ICK}$ + interconnect	T_{B7}	The other long line pull-up	9,10,11		83		55	ns
$T_{CKO} + T_{QLO} + T_{IO} + T_{ICK}$ + interconnect	T_{B8}	No pull-up, lower long lines	9,10,11		47		31	ns
$T_{CKO} + T_{QLO} + T_{IO} + T_{ICK}$ + interconnect	T_{B9}	No pull-up, upper long lines	9,10,11		57		38	ns

Table 1. Electrical Performance Characteristics (Continued)

TSC0117

Test	Sym	Conditions −55°C ≤ Tc ≤ +125°C V_{CC} = 5.0 V ±10%	Group A Subgroups	-50 Min	-50 Max	-70 Min	-70 Max	Units
Application Guidelines, Switching, CLB[8]								
Combinatorial	T_{ILO}	1 See Fig. 2	N/A		14		9	ns
Reset to CLB output	T_{RIO}	9	N/A		15		8	ns
Reset Direct width	T_{RPW}	13	N/A	12		8		ns
Master Reset pin to CLB out	T_{MRQ}		N/A		40		34	ns
K Clock[9]								
To CLB output	T_{CKO}	8	N/A		12		8	ns
Additional for Q returning through F or G to CLB out	T_{QLO}		N/A		11		7	ns
Logic-input setup	T_{ICK}	2	N/A	12		8		ns
Logic-input hold	T_{CKI}	3	N/A	1		1		ns
Data In setup	T_{DICK}	4	N/A	8		5		ns
Data In hold (1)	T_{CKDI}	5	N/A	6		4		ns
Enable Clock setup	T_{ECCK}	6	N/A	10		7		ns
Enable Clock hold	T_{CKEC}	7	N/A	0		0		ns
*Clock (High)	T_{CH}	11	N/A	9		7		ns
*Clock (Low)	T_{CL}	12	N/A	9		7		ns

*These parameters are for clock pulses within an LCA device. For externally applied clock, increase values by 20%.

Application Guidelines, Switching, Internal Buffers[8]								
Clock Buffer	T_{GCK}		N/A		9		6	ns
TBUF								
Data to Output	T_{IO}		N/A		8		5	ns
Three-state to Output								
Single Pull-up	T_{PUS}		N/A		42		36	ns
Pair of Pull-ups	T_{PUF}		N/A		22		16	ns
Bidirectional	T_{BIDI}		N/A		6		4	ns

Table 1. Electrical Performance Characteristics (Continued)

Test	Sym	Conditions $-55°C \le Tc \le +125°C$ $V_{CC} = 5.0\ V \pm 10\%$	Group A Subgroups	-50 Min	-50 Max	-70 Min	-70 Max	Units
Application Guidelines, Switching, IOB[8,10]								
Pad (package pin)		See Fig. 3						
To inputs TCLKIN, BCLKIN	T_{PIDC}		N/A		5		5	ns
To inputs DIRECT IN	T_{PID}	3	N/A		9		6	ns
I/O Clock								
To I/O RI input (FF)	T_{IKRI}	4	N/A		11		7	ns
I/O pad-input setup	T_{PICK}	1	N/A	30		20		ns
I/O pad-input hold	T_{IKPI}	2	N/A	0		0		ns
To I/O pad (fast)	T_{OKPO}	7	N/A		18		13	ns
I/O pad output setup	T_{OOK}	5	N/A	15		10		ns
I/O pad output hold	T_{OKO}	6	N/A	0		0		ns
*Clock (High)	T_{IOH}	11	N/A	9		8		ns
*Clock (Low)	T_{IOL}	12	N/A	9		8		ns
Output								
To pad (enabled fast)	T_{OPF}	10	N/A		15		9	ns
To pad (enabled slow)	T_{OPS}	10	N/A		40		29	ns
Three-State								
To pad begin hi-Z (fast)	T_{TSHZ}	9	N/A		14		12	ns
To pad valid (fast)	T_{TSON}	8	N/A		20		14	ns
Master Reset								
To input RI	T_{RRI}	13	N/A		37		33	ns
To output (FF)	T_{RPO}	14	N/A		55		43	ns

*These parameters are for clock pulses within an LCA device. For externally applied clock, increase values by 20%.

Table 1. Electrical Performance Characteristics (Continued)

Test	Sym	Conditions −55°C ≤ Tc ≤ +125°C V_{CC} = 5.0 V ±10%	Group A Subgroups	-50		-70		Units
				Min	Max	Min	Max	
Application Guidelines, Switching, Master Parallel Mode Programming[6,11]								
$\overline{RCLK}$,								
To Address Valid	T_{RAC}	1 See Fig.6	N/A	0	200	0	200	ns
To Data Setup	T_{DRC}	2	N/A	60		60		ns
To Data Hold	T_{RCD}	3	N/A	0		0		ns
$\overline{RCLK}$ High	T_{RCH}	4	N/A	600		600		ns
$\overline{RCLK}$ Low	T_{RCL}	5	N/A	4.0		4.0		µs

Table 1. Electrical Performance Characteristics (Continued)

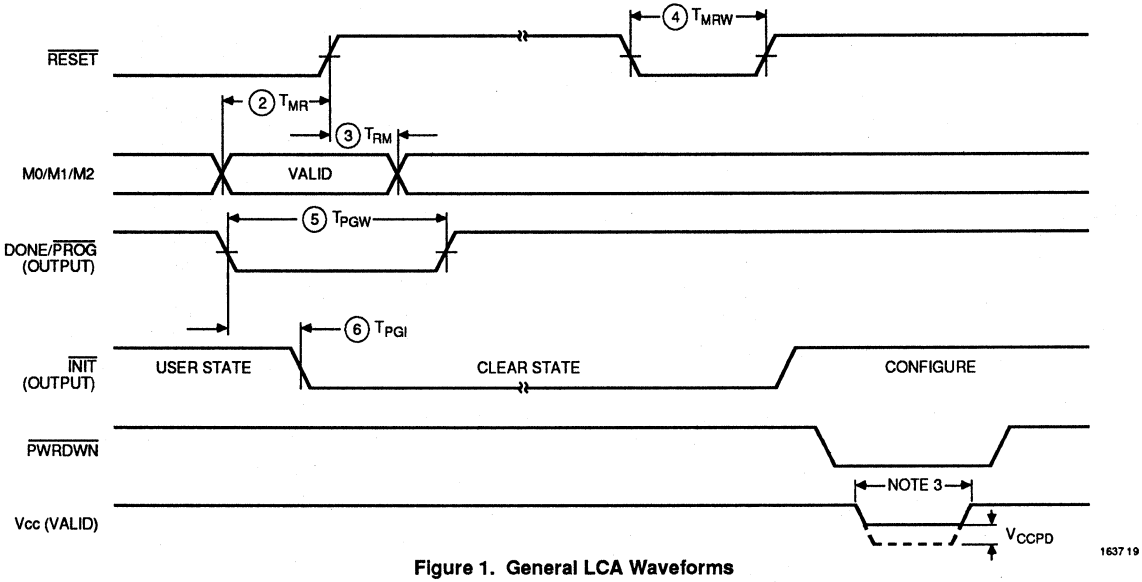


Figure 1. General LCA Waveforms

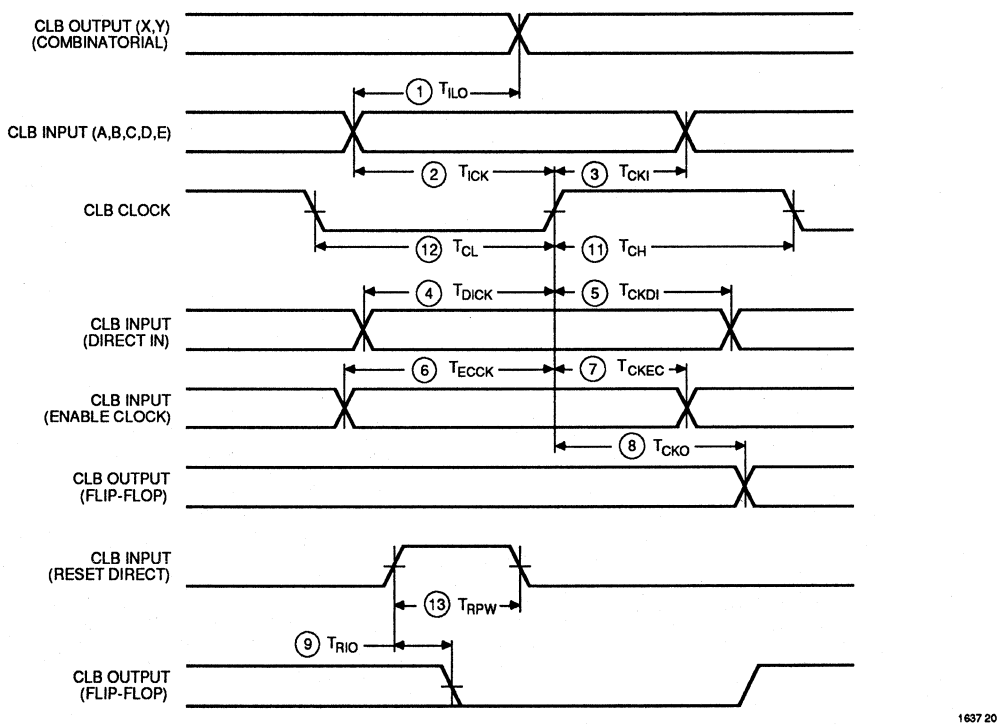


Figure 2. CLB Waveforms

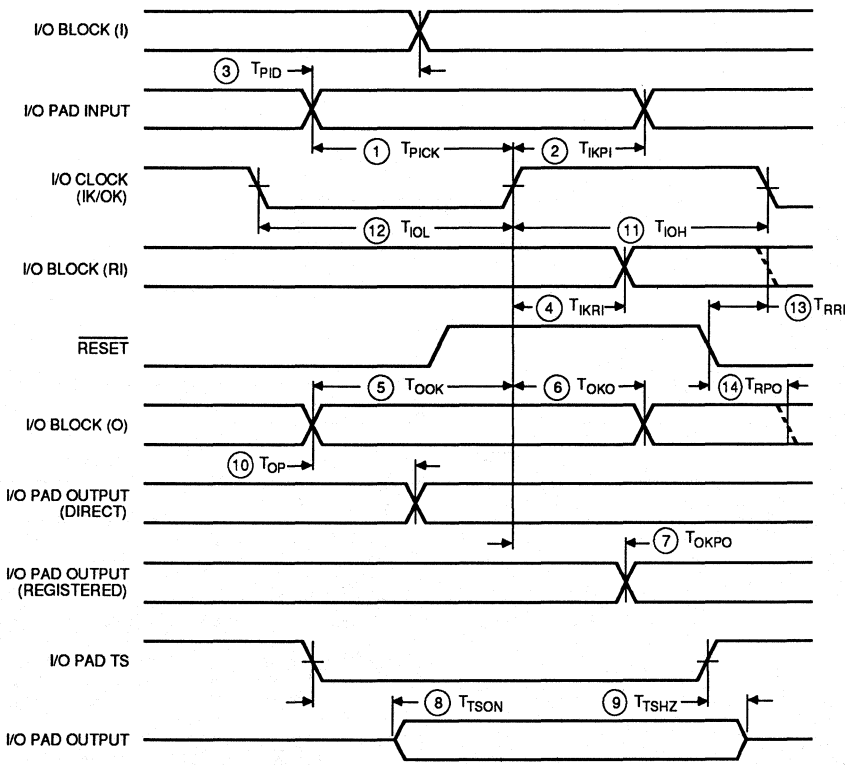
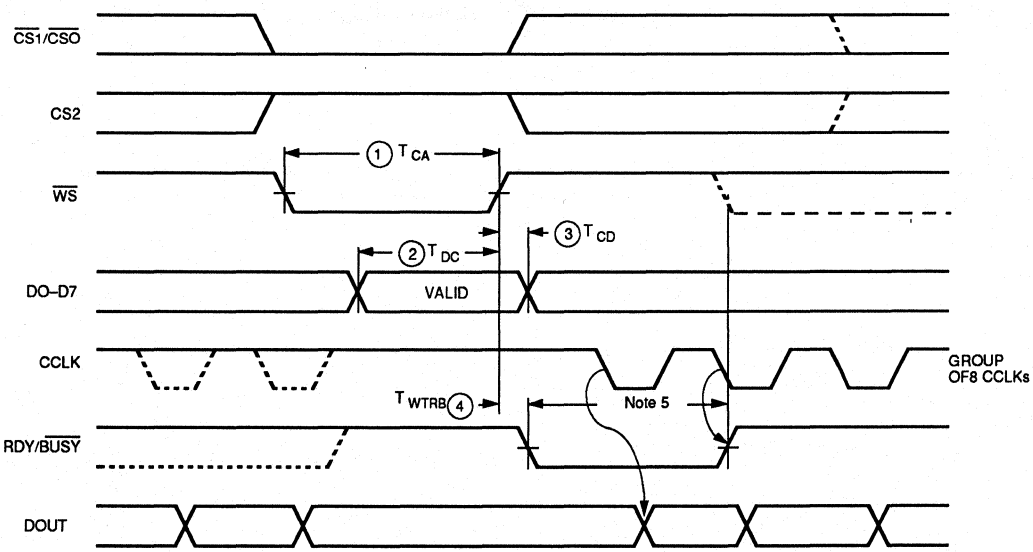


Figure 3. IOB Waveforms



TSC0117

Figure 4. Peripheral Mode Waveforms

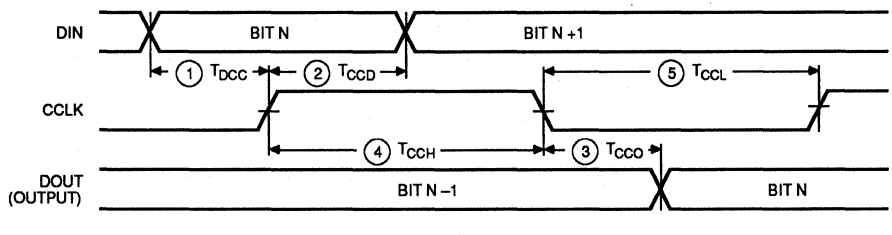


Figure 5. Slave Mode Waveforms

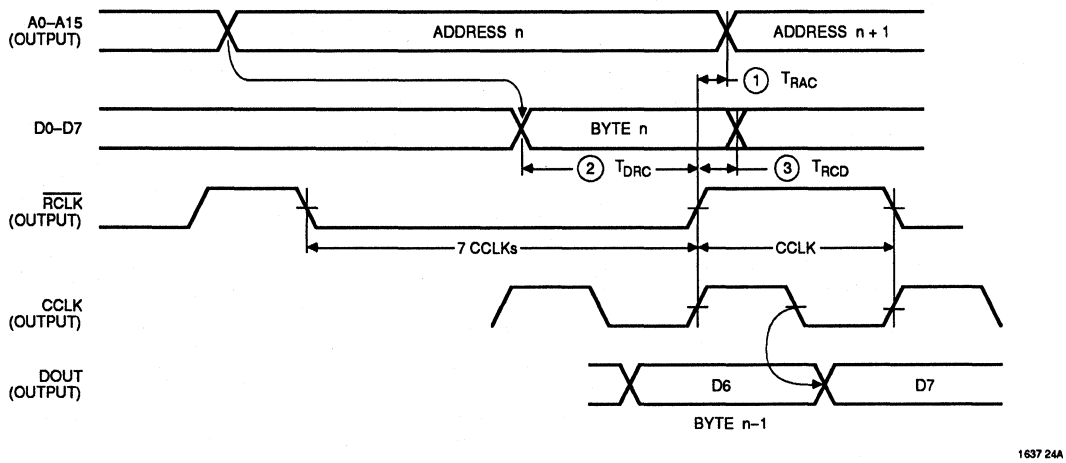
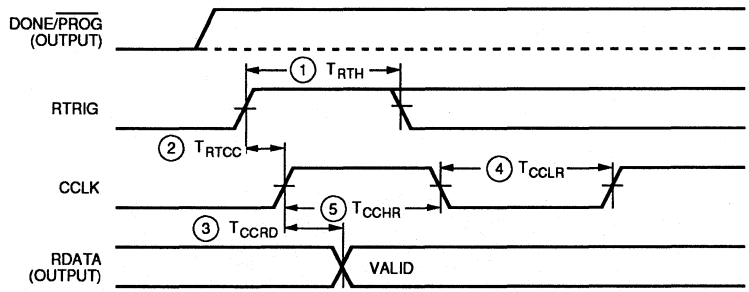


Figure 6. Master Parallel Mode Waveforms

TSC0117

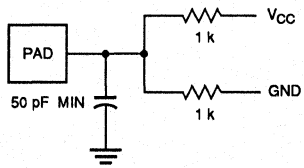


1637 25

Figure 7. Program Readback Waveforms

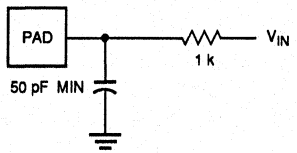
XC3042B Data Sheet Notes

1. Xilinx maintains this specification as a controlled document. To comply with the intent of MIL-STD-883, and to insure the use of the most recently released device performance parameters, please request a copy of the current revision of this Test Specification (TSC0117) from Xilinx.
2. No output current loads, no active input or long line pull-up resistors, and with the device configured with the MAKEBITS "tie" option.
3. $\overline{\text{PWRDWN}}$ transitions must occur during operational Vcc levels.
4. $\overline{\text{RESET}}$ timing relative to valid mode lines (M0, M1, M2) is relevant only when $\overline{\text{RESET}}$ is used to delay configuration.
5. Configuration must be delayed until the INIT of all LCAs is High. $\overline{\text{WS}}$ cannot go active until RDY/$\overline{\text{BUSY}}$ goes High.
6. Readback should not be initiated until configuration is complete.
7. DOUT timing is the same as for slave mode.
8. Testing of the Applications Guidelines is modeled after testing specified by MIL-M-38510/605. Devices are first 100% functionally tested. Benchmark patterns are then used to measure the Application Guidelines. Characterization data are taken at initial device qualification, prior to introduction of significant changes, and at least twice yearly to monitor correlation between benchmark patterns, device performance, XACT software timings, and the data sheet.
9. The CLB K to Q output delay (T_{CKO}) plus the shortest possible interconnect delay is always longer than the Data In hold time requirement (T_{CKDI}) on the same die.
10. Voltage levels of unused pads must be valid logic levels. Each can be configured with the internal pull-up resistor, configured as a driven output, or driven from an external source.
11. At power-up, V_{CC} must rise from 2.0 V to V_{CC} minimum in less than 10 ms. Otherwise, delay configuration using $\overline{\text{RESET}}$.
12. All timings except T_{TSHZ} and T_{TSON} are measured at 1.5 V level with 50 pF minimum load output. For input signals, rise and fall times are less than 6 ns, with low amplitude = 0 V, and high = 3 V.
    T_{TSHZ} is determined when the output shifts 10% (of the output voltage swing) from V_{OL} level or V_{OH} level. The following circuit is used:



1637 13

12. (continued)
    T_{TSON} is measured at 0.5 V_{CC} level with V_{IN} = 0 for 3-State to active High, and V_{IN} = V_{CC} for 3-State to active Low. The following load circuit is used:



1637 14

XC3090B
Military Logic Cell™ Array

Product Specification. See Note 1.

FEATURES

- MIL–STD–883 Class B Processing. Complies with paragraph 1.2.1
- Field-programmable gate array
- Low power CMOS static memory technology
- Standard product. Completely tested at factory
- Design changes made in minutes

  Complete user control for design cycle. Secure design process
- Complete PC or workstation based development system
  - Schematic entry
  - Auto Place/ Route (DS23)
  - Design Editor (DS21)
  - Logic & Timing Simulator (DS22)
  - XACTOR In-circuit Verifier (DS24)

Part Number	Logic Capacity (gates)	Configurable Logic Blocks	User I/Os	Configuration Program (bits)
XC3090	9000	320	144	64160

DESCRIPTION

The Logic Cell™ Array (LCA) is a high density CMOS programmable gate array. Its patented array architecture consists of three types of configurable elements: Input/ Output Blocks, Configurable Logic Blocks and Interconnect. The designer can define individual I/O blocks for interface to external circuitry, define logic blocks to implement logic functions and define interconnection networks to compose larger scale logic functions.

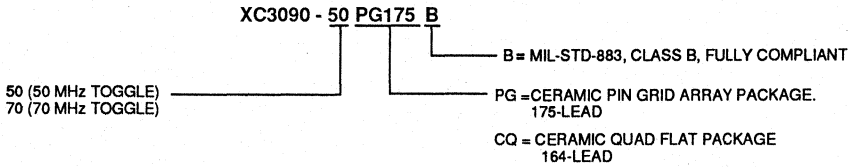
The Logic Cell Array's logic functions and interconnections are determined by the configuration program stored in internal static memory cells. On-chip logic provides for automatic loading of configuration data at power-up or on command. The program data can reside in an EEPROM, EPROM or ROM on the circuit board or on a floppy disk or hard disk.

Several methods of automatically loading the required data are designed into the Logic Cell Array and are determined by logic levels applied to mode selection pins at configuration time. The form of the data may be either serial or parallel, depending on the configuration mode. The programming data are independent of the configuration mode selected.

The XACT development system allows the user to define the logic functions of the device. Schematic capture is available for design entry, while logic and timing simulation, and in-circuit debugging are available for design verification. XACT is used to compile the data pattern which represents the configuration program. This data can then be converted to a PROM programmer format file to create the configuration program storage.

See the XC3000 Commercial data sheet for a full description.

ORDERING INFORMATION



1637 28A

TSC0097

PIN ASSIGNMENTS

CONFIGURATION MODE: <M2:M1:M0>					164 PGA	175 PGA	USER OPERATION
SLAVE <1:1:1>	MASTER-SER <0:0:0>	PERIPHERAL <1:0:1>	MASTER-HIGH <1:1:0>	MASTER-LOW <1:0:0>			
$\overline{\text{PWR DWN}}$ (I)	$\overline{\text{PWR DWN}}$ (I)	$\overline{\text{PWR DWN}}$ (I)	$\overline{\text{PWR DWN}}$ (I)	$\overline{\text{PWR DWN}}$ (I)	20	B2	$\overline{\text{PWR DWN}}$ (I)
VCC	VCC	VCC	VCC	VCC	42	D9	VCC
M1 (HIGH) (I)	M1 (LOW) (I)	M1 (LOW) (I)	M1 (HIGH) (I)	M1 (LOW) (I)	62	B14	$\overline{\text{RDATA}}$
M0 (HIGH) (I)	M0 (LOW) (I)	M0 (HIGH) (I)	M0 (LOW) (I)	M0 (LOW) (I)	64	B15	RTRIG (I)
M2 (HIGH) (I)	M2 (LOW) (I)	M2 (HIGH) (I)	M2 (HIGH) (I)	M2 (HIGH) (I)	66	C15	I/O
HDC (HIGH)	HDC (HIGH)	HDC (HIGH)	HDC (HIGH)	HDC (HIGH)	67	E14	I/O
$\overline{\text{LDC}}$ (LOW)	$\overline{\text{LDC}}$ (LOW)	$\overline{\text{LDC}}$ (LOW)	$\overline{\text{LDC}}$ (LOW)	$\overline{\text{LDC}}$ (LOW)	71	D16	I/O
* $\overline{\text{INIT}}$	$\overline{\text{INIT}}$	$\overline{\text{INIT}}$	$\overline{\text{INIT}}$	$\overline{\text{INIT}}$	81	H15	I/O
GND	GND	GND	GND	GND	83	J14	GND
					99	P15	XTL2 OR I/O
$\overline{\text{RESET}}$ (I)	$\overline{\text{RESET}}$ (I)	$\overline{\text{RESET}}$ (I)	$\overline{\text{RESET}}$ (I)	$\overline{\text{RESET}}$ (I)	101	R15	$\overline{\text{RESET}}$ (I)
DONE	DONE	DONE	DONE	DONE	103	R14	$\overline{\text{PROGRAM}}$ (I)
		DATA 7 (I)	DATA 7 (I)	DATA 7 (I)	104	N13	I/O
					105	T14	XTL1 OR I/O
		DATA 6 (I)	DATA 6 (I)	DATA 6 (I)	109	P12	I/O
		DATA 5 (I)	DATA 5 (I)	DATA 5 (I)	115	T11	I/O
		$\overline{\text{CS0}}$ (I)			116	R10	I/O
		DATA 4 (I)	DATA 4 (I)	DATA 4 (I)	121	R9	I/O
VCC	VCC	VCC	VCC	VCC	123	N9	VCC
		DATA 3 (I)	DATA 3 (I)	DATA 3 (I)	125	P8	I/O
		$\overline{\text{CS1}}$ (I)			126	R8	I/O
		DATA 2 (I)	DATA 2 (I)	DATA 2 (I)	131	R7	I/O
		DATA 1 (I)	DATA 1 (I)	DATA 1 (I)	137	R5	I/O
		RDY/$\overline{\text{BUSY}}$	$\overline{\text{RCLK}}$	$\overline{\text{RCLK}}$	138	P5	I/O
DIN (I)	DIN (I)	DATA 0 (I)	DATA 0 (I)	DATA 0 (I)	143	R3	I/O
DOUT	DOUT	DOUT	DOUT	DOUT	144	N4	I/O
CCLK (I)	CCLK	CCLK	CCLK	CCLK	145	R2	CCLK (I)
		$\overline{\text{WS}}$ (I)	A0	A0	148	P2	I/O
		CS2 (I)	A1	A1	149	M3	I/O
			A2	A2	152	P1	I/O
			A3	A3	153	N1	I/O
			A15	A15	156	M1	I/O
			A4	A4	157	L2	I/O
			A14	A14	160	K2	I/O
			A5	A5	161	K1	I/O
GND	GND	GND	GND	GND	164	J3	GND
			A13	A13	2	H2	I/O
			A6	A6	3	H1	I/O
			A12	A12	8	F2	I/O
			A7	A7	9	E1	I/O
			A11	A11	12	D1	I/O
			A8	A8	13	C1	I/O
			A10	A10	16	E3	I/O
			A9	A9	17	C2	I/O

REPRESENTS A 50 kΩ TO 100 kΩ PULL-UP

* INIT IS AN OPEN-DRAIN OUTPUT DURING CONFIGURATION

(I) REPRESENTS AN INPUT

X1161

TSC0097

PG175 PIN ASSIGNMENTS

PGA Pin Number	XC-3090
B2	PWRDN
D4	TCLKIN-I/O
B3	I/O
C4	I/O
B4	I/O
A4	I/O
D5	I/O
C5	I/O
B5	I/O
A5	I/O
C6	I/O
D6	I/O
B6	I/O
A6	I/O
B7	I/O
C7	I/O
D7	I/O
A7	I/O
A8	I/O
B8	I/O
C8	I/O
D8	GND
D9	VCC
C9	I/O
B9	I/O
A9	I/O
A10	I/O
D10	I/O
C10	I/O
B10	I/O
A11	I/O
B11	I/O
D11	I/O
C11	I/O
A12	I/O
B12	I/O
C12	I/O
D12	I/O
A13	I/O
B13	I/O
C13	I/O
A14	I/O

PGA Pin Number	XC-3090
D13	I/O
B14	M1-RDATA
C14	GND
B15	M0-RTRIG
D14	VCC
C15	M2-I/O
E14	HDC-I/O
B16	I/O
D15	I/O
C16	I/O
D16	LDC-I/O
F14	I/O
E15	I/O
E16	I/O
F15	I/O
F16	I/O
G14	I/O
G15	I/O
G16	I/O
H16	I/O
H15	INIT-I/O
H14	VCC
J14	GND
J15	I/O
J16	I/O
K16	I/O
K15	I/O
K14	I/O
L16	I/O
L15	I/O
M16	I/O
M15	I/O
L14	I/O
N16	I/O
P16	I/O
N15	I/O
R16	I/O
M14	I/O
P15	XTAL2(IN)-I/O
N14	GND
R15	RESET
P14	VCC

PGA Pin Number	XC-3090
R14	DONE-PG
N13	D7-I/O
T14	XTAL1(OUT)-BCLKIN-I/O
P13	I/O
R13	I/O
T13	I/O
N12	I/O
P12	D6-I/O
R12	I/O
T12	I/O
P11	I/O
N11	I/O
R11	I/O
T11	D5-I/O
R10	CS0-I/O
P10	I/O
N10	I/O
T10	I/O
T9	I/O
R9	D4-I/O
P9	I/O
N9	VCC
N8	GND
P8	D3-I/O
R8	CS1-I/O
T8	I/O
T7	I/O
N7	I/O
P7	I/O
R7	D2-I/O
T6	I/O
R6	I/O
N6	I/O
P6	I/O
T5	I/O
R5	D1-I/O
P5	RDY/BUSY-RCLK-I/O
N5	I/O
T4	I/O
R4	I/O
P4	I/O

PGA Pin Number	XC-3090
R3	D0-DIN-I/O
N4	DOUT-I/O
R2	CCLK
P3	VCC
N3	GND
P2	A0-WS-I/O
M3	A1-CS2-I/O
R1	I/O
N2	I/O
P1	A2-I/O
N1	A3-I/O
L3	I/O
M2	I/O
M1	A15-I/O
L2	A4-I/O
L1	I/O
K3	I/O
K2	A14-I/O
K1	A5-I/O
J1	I/O
J2	I/O
J3	GND
H3	VCC
H2	A13-I/O
H1	A6-I/O
G1	I/O
G2	I/O
G3	I/O
F1	I/O
F2	A12-I/O
E1	A7-I/O
E2	I/O
F3	I/O
D1	A11-I/O
C1	A8-I/O
D2	I/O
B1	I/O
E3	A10-I/O
C2	A9-I/O
D3	VCC
C3	GND

Unprogrammed IOBs have a default pull-up. This prevents an undefined pad level for unbonded or unused IOBs.
Programmed outputs are default slew-rate limited.

Pins A2, A3, A15, A16, T1, T2, T3, T15 and T16 are not connected.
Pin A1 does not exist.

TSC0097

164-PIN CQFP PINOUTS

CQFP Pin Number	XC-3090
20	PWRDN
21	TCLKIN-I/O
22	I/O
23	I/O
24	I/O
25	I/O
26	I/O
27	I/O
28	I/O
29	I/O
30	I/O
31	I/O
32	I/O
33	I/O
34	I/O
35	I/O
36	I/O
37	I/O
38	I/O
39	I/O
40	I/O
41	GND
42	VCC
43	I/O
44	I/O
45	I/O
46	I/O
47	I/O
48	I/O
49	I/O
50	I/O
51	I/O
52	I/O
53	I/O
54	I/O
55	I/O
56	I/O
57	I/O
58	I/O
59	I/O
60	I/O

CQFP Pin Number	XC-3090
61	I/O
62	M1-RDATA
63	GND
64	M0-RTRIG
65	VCC
66	M2-I/O
67	HDC-I/O
68	I/O
69	I/O
70	I/O
71	LDC-I/O
72	I/O
73	I/O
74	I/O
75	I/O
76	I/O
77	I/O
78	I/O
79	I/O
80	I/O
81	INIT-I/O
82	VCC
83	GND
84	I/O
85	I/O
86	I/O
87	I/O
88	I/O
89	I/O
90	I/O
91	I/O
92	I/O
93	I/O
94	I/O
95	I/O
96	I/O
97	I/O
98	I/O
99	XTAL2(IN)-I/O
100	GND
101	RESET
102	VCC

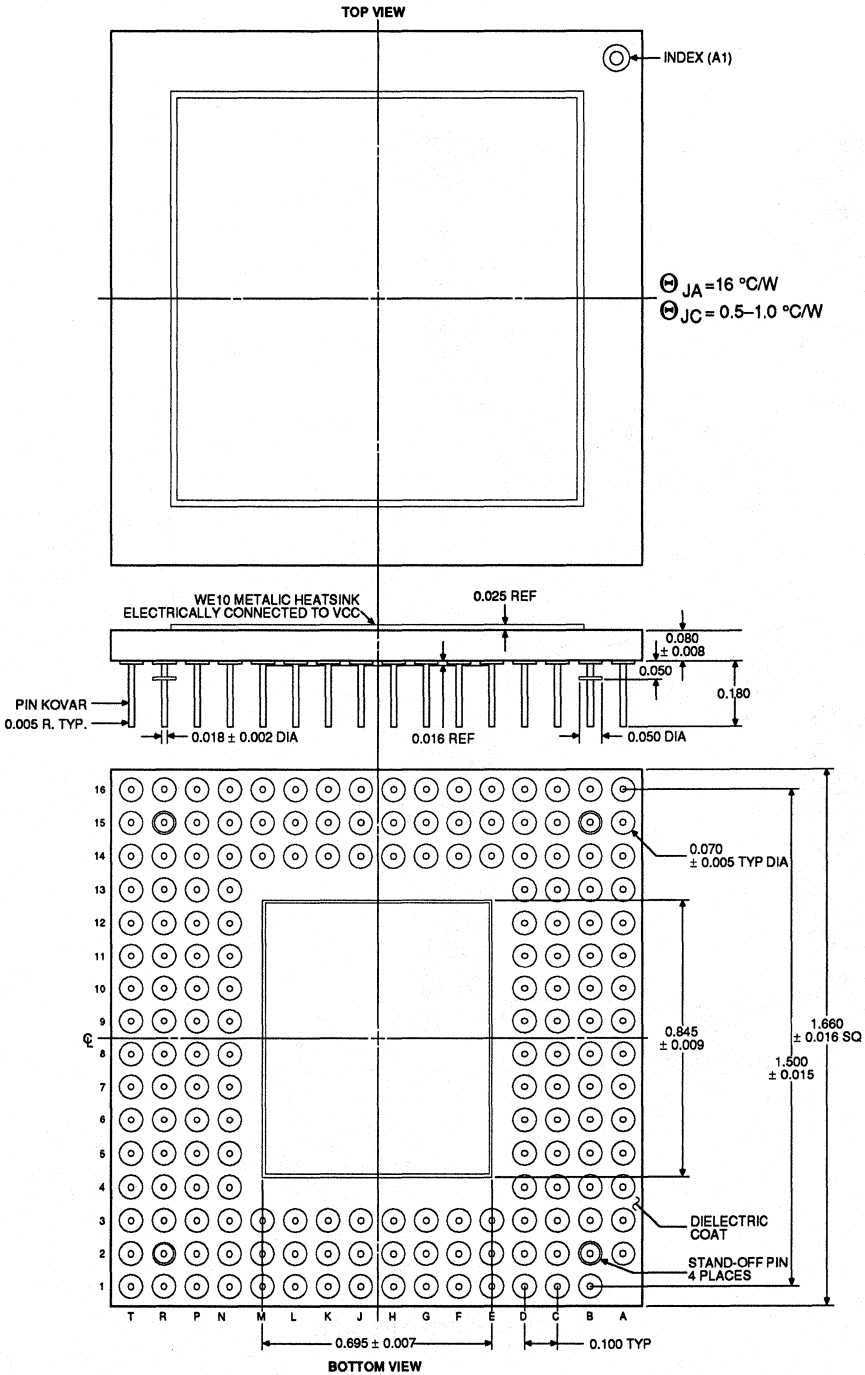
CQFP Pin Number	XC-3090
103	DONE-PG
104	D7-I/O
105	XTAL1(OUT)-BCLKIN-I/O
106	I/O
107	I/O
108	I/O
109	D6-I/O
110	I/O
111	I/O
112	I/O
113	I/O
114	I/O
115	D5-I/O
116	CS0-I/O
117	I/O
118	I/O
119	I/O
120	I/O
121	D4-I/O
122	I/O
123	VCC
124	GND
125	D3-I/O
126	CS1-I/O
127	I/O
128	I/O
129	I/O
130	I/O
131	D2-I/O
132	I/O
133	I/O
134	I/O
135	I/O
136	I/O
137	D1-I/O
138	RDY/BUSY-RCLK-I/O
139	I/O
140	I/O
141	I/O
142	I/O

CQFP Pin Number	XC-3090
143	D0-DIN-I/O
144	DOUT-I/O
145	CCLK
146	VCC
147	GND
148	A0-WS-I/O
149	A1-CS2-I/O
150	I/O
151	I/O
152	A2-I/O
153	A3-I/O
154	I/O
155	I/O
156	A15-I/O
157	A4-I/O
158	I/O
159	I/O
160	A14-I/O
161	A5-I/O
162	I/O
163	I/O
164	GND
1	VCC
2	A13-I/O
3	A6-I/O
4	I/O
5	I/O
6	I/O
7	I/O
8	A12-I/O
9	A7-I/O
10	I/O
11	I/O
12	A11-I/O
13	A8-I/O
14	I/O
15	I/O
16	A10-I/O
17	A9-I/O
18	VCC
19	GND

Unprogrammed IOBs have a default pull-up.
This Prevents an undefined pad level for unbonded or unused IOBs.
Programmed outputs are default slew-rate limited.

TSC0097

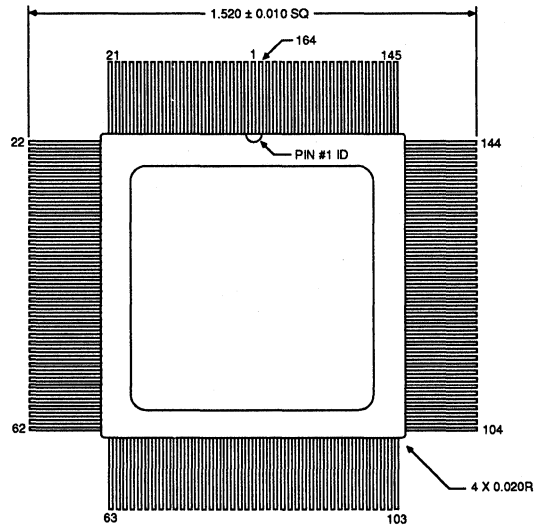
PHYSICAL DIMENSIONS – Conforms to MIL-M-38510 Appendix C, Case P-BC.

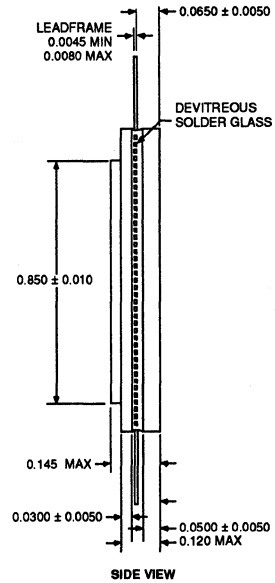


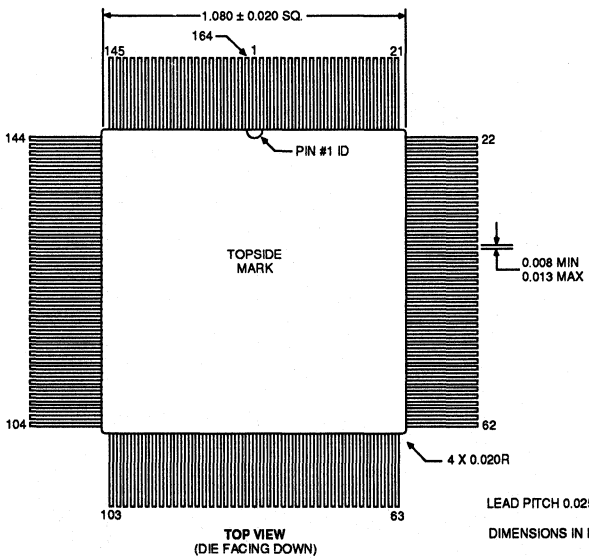
TSC0097

175-Pin PGA Package (Cavity Down)

PHYSICAL DIMENSIONS (Continued)







LEAD PITCH 0.025 TYPICAL

DIMENSIONS IN INCHES

NOTES:
1. LEADS ARE SHIPPED UNFORMED IN CARRIERS IN TRAYS, TOPSIDE UP
2. FORMING TOOL INFORMATION:
   – FANCORT INDUSTRIES – (201) 575-0610 WEST CALDWELL NJ.
   – RISI INDUSTRIES INC. (619) 425-3970 CHULA VISTA, CA.

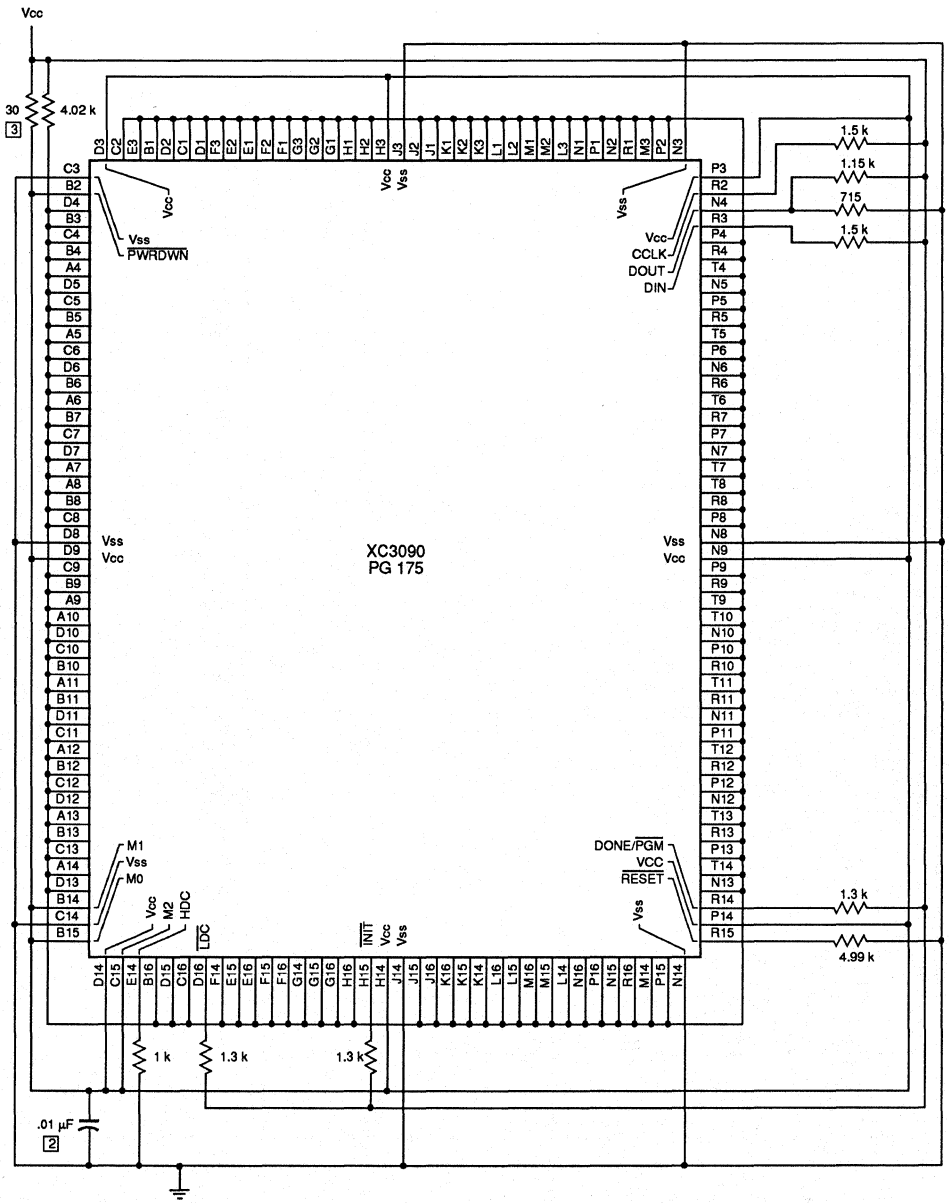
Θ_{JA} = 35-45° C/W
Θ_{JC} = 3-5° C/W

1105 41D

164-PIN CQFP Package

TSC0097

STATIC BURN-IN CIRCUITS



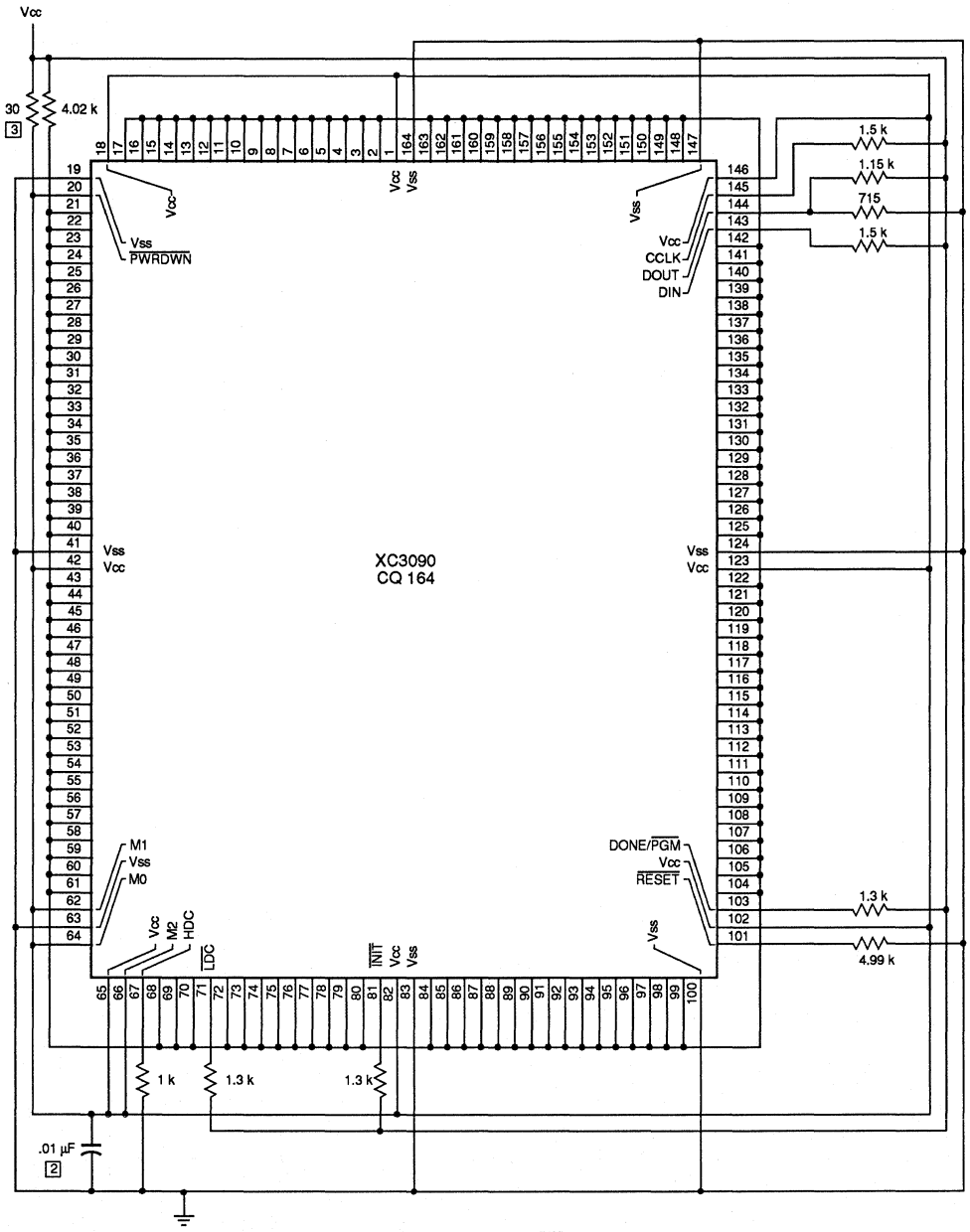
NOTES:

1. UNLESS OTHERWISE SPECIFIED, ALL RESISTORS ARE METAL FILM AND ARE RATED FOR 1/8 W AT 150°C WITH A BUILD TOLERANCE OF 1% AND A 5% TOLERANCE OVER LIFE.
2. CAPACITOR HAS 10% TOLERANCE, 50 V RATING WITH AN X7R TEMPERATURE CHARACTERISTIC.
3. 30 Ω RESISTOR IS METAL OXIDE AND IS RATED FOR 1 W AT 150°C WITH A TOLERANCE OF 5%.

1637 31A

TSC0097

STATIC BURN-IN CIRCUITS (Continued)



NOTES:

1. UNLESS OTHERWISE SPECIFIED, ALL RESISTORS ARE METAL FILM AND ARE RATED FOR 1/8 W AT 150°C WITH A BUILD TOLERANCE OF 1% AND A 5% TOLERANCE OVER LIFE.
2. CAPACITOR HAS 10% TOLERANCE, 50 V RATING WITH AN X7R TEMPERATURE CHARACTERISTIC.
3. 30 Ω RESISTOR IS METAL OXIDE AND IS RATED FOR 1 W AT 150°C WITH A TOLERANCE OF 5%.

TSC0097

TEST SPECIFICATION

Xilinx maintains test specifications for each product as controlled documents. To insure the use of the most recently released device performance parameters, please request a copy of the current test-specification revision.

Absolute Maximum Ratings		Limits	Units
V_{CC}	Supply voltage relative to GND	–0.5 to +7.0	V
V_{IN}	Input voltage with respect to GND	–0.5 to V_{CC} +0.5	V
V_{TS}	Voltage applied to three-state output	–0.5 to V_{CC} +0.5	V
T_{STG}	Storage temperature (ambient)	–65 to +150	°C
T_{SOL}	Maximum soldering temperature (10 sec @ 1/16 in.)	+260	°C
T_J	Maximum junction temperature	+150	°C

Note: Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those listed under Recommended Operating Conditions is not implied. Exposure to Absolute Maximum Ratings conditions for extended periods of time may affect device reliability.

Test	Symbol	Conditions –55°C ≤ Tc ≤ +125°C V_{CC} = 5.0 V ±10%	Group A Subgroups	Limits Min	Limits Max	Units
High Level Output Voltage	V_{OH}	V_{CC} = 4.5 V, I_{OH} = –4.0 mA	1,2,3	3.7		V
Low Level Output Voltage	V_{OL}	V_{CC} = 5.5 V, I_{OL} = 4.0 mA	1,2,3		0.4	V
Quiescent Operating[2] Power Supply Current	I_{CCO}	CMOS Mode, V_{in} = V_{CC} = 5.5 V	1,2,3		3	mA
		TTL Mode, V_{in} = V_{CC} = 5.5 V	1,2,3		15	mA
Power-Down Supply Current	I_{CCPD}	V_{in} = V_{CC} = 5.5 V, $\overline{PWR}$ $\overline{DWN}$ = 0 V	1,2,3		2.5	mA
Leakage Current	I_{IL}	V_{CC} = 5.5 V, V_{in} = V_{CC} and 0 V	1,2,3	–20	20	µA
Horizontal Long Line Pull-up Current	I_{RLL}	Measured as an average	1,2,3		2.4	mA
Input High Level TTL	V_{IHT}	Guaranteed Input High	1,2,3	2.0		V
Input Low Level TTL	V_{ILT}	Guaranteed Input Low	1,2,3		0.8	V
Input High Level CMOS	V_{IHC}	Guaranteed Input High	1,2,3	.7 V_{CC}		V
Input Low Level CMOS	V_{ILC}	Guaranteed Input Low	1,2,3		.2 V_{CC}	V

Table 1. Electrical Performance Characteristics

TSC0097

Test	Sym	Conditions −55°C ≤ Tc ≤ +125°C V_{CC} = 5.0 V ±10%	Group A Subgroup	-50		-70		Units
				Min	Max	Min	Max	
Switching Characteristics, General LCA								
DONE/$\overline{PROG}$								
Program Width (Low)	T_{PGW}	5 See Fig. 1	9,10,11	6		6		µs
Initialization	T_{PGI}	6	9,10,11		7		7	µs
$\overline{PWR}$ $\overline{DWN}$[3]								
Power Down Supply	V_{CCPD}		1,2,3	3.5		3.5		V
RESET[4]								
M2,M1,M0 Setup	T_{MR}	2	9,10,11	1		1		µs
M2,M1,M0 Hold	T_{RM}	3	9,10,11	1		1		µs
Width (low) abort	T_{MRW}	4	9,10,11	6		6		µs
Switching Characteristics, Peripheral Mode Programming[5]								
$\overline{WS}$ LOW	T_{CA}	1 See Fig. 4	9,10,11	0.5		0.5		µs
DIN Setup	T_{DC}	2	9,10,11	60		60		ns
DIN Hold	T_{CD}	3	9,10,11	0		0		ns
Ready/$\overline{Busy}$	T_{WTRB}	4	9,10,11		60		60	ns
Switching Characteristics, Slave Mode Programming[5]								
CCLK,								
To DOUT	T_{CCO}	3 See Fig. 5	9,10,11		100		100	ns
DIN Setup	T_{DCC}	1	9,10,11	60		60		ns
DIN Hold	T_{CCD}	2	9,10,11	0		0		ns
High Time	T_{CCH}	4	9,10,11	0.5		0.5		µs
Low Time	T_{CCL}	5	9,10,11	0.5	1.0	0.5	1.0	µs
Frequency	F_{CC}		9,10,11		1		1	MHz

Table 1. Electrical Performance Characteristics (Continued)

TSC0097

Test	Sym	Conditions −55°C ≤ Tc ≤ +125°C Vcc = 5.0 V ±10%	Group A Subgroups	-50 Min	-50 Max	-70 Min	-70 Max	Units
Switching Characteristics, Program Readback[6,7]								
RTRIG Setup	T_{RTH}	1 See Fig. 7	9,10,11	250		250		ns
CCLK,								
RTRIG Setup	T_{RTCC}	2	9,10,11	200		200		ns
RDATA Delay	T_{CCRD}	3	9,10,11		100		100	ns
Clock Low	T_{CCLR}	4	9,10,11	1.2		1.2	2.0	µs
Clock High	T_{CCHR}	5	9,10,11	0.5		0.5		µs
Benchmark Patterns[8]								
T_{PID} +interconnect +20 (T_{ILO}) + T_{OP}. Measured on 8 cols.	T_{B1}		9,10,11		303		194	ns
$T_{CKO} + T_{ICK} + T_{CKI}$ + interconnect	T_{B2}	Tested on all CLBs	9,10,11		32		21	ns
$T_{CKO} + T_{QLO} + T_{ILO} + T_{DICK}$ + interconnect	T_{B3}	Tested on all CLBs	9,10,11		53		34	ns
$T_{ILO} + T_{ECCK}$ + interconnect	T_{B4}	Tested on all CLBs	9,10,11		35		23	ns
$T_{OKPO} + T_{OPS} - T_{OPF} + T_{PICK}$	T_{B5}	Tested on all CLBs	9,10,11		73		53	ns
$T_{CKO} + T_{QLO} + T_{PUS} + T_{ICK}$ + interconnect	T_{B6}	One long line pull-up	9,10,11		73		48	ns
$T_{CKO} + T_{QLO} + T_{PUS} + T_{ICK}$ + interconnect	T_{B7}	The other long line pull-up	9,10,11		83		55	ns
$T_{CKO} + T_{QLO} + T_{IO} + T_{ICK}$ + interconnect	T_{B8}	No pull-up, lower long lines	9,10,11		47		31	ns
$T_{CKO} + T_{QLO} + T_{IO} + T_{ICK}$ + interconnect	T_{B9}	No pull-up, upper long lines	9,10,11		57		38	ns

Table 1. Electrical Performance Characteristics (Continued)

TSC0097

Test	Sym	Conditions $-55°C \le Tc \le +125°C$ $V_{CC} = 5.0\ V \pm 10\%$	Group A Subgroups	-50 Min	-50 Max	-70 Min	-70 Max	Units
Application Guidelines, Switching, CLB[8]								
Combinatorial	T_{ILO}	1 See Fig. 2	N/A		14		9	ns
Reset to CLB output	T_{RIO}	9	N/A		12		8	ns
Reset Direct width	T_{RPW}	13	N/A	12		8		ns
Master Reset pin to CLB out	T_{MRQ}		N/A		40		34	ns
K Clock[9]								
To CLB output	T_{CKO}	8	N/A		12		8	ns
Additional for Q returning through F or G to CLB out	T_{QLO}		N/A		11		7	ns
Logic-input setup	T_{ICK}	2	N/A	12		8		ns
Logic-input hold	T_{CKI}	3	N/A	1		1		ns
Data In setup	T_{DICK}	4	N/A	8		5		ns
Data In hold (1)	T_{CKDI}	5	N/A	6		4		ns
Enable Clock setup	T_{ECCK}	6	N/A	10		7		ns
Enable Clock hold	T_{CKEC}	7	N/A	0		0		ns
*Clock (High)	T_{CH}	11	N/A	9		7		ns
*Clock (Low)	T_{CL}	12	N/A	9		7		ns

*These parameters are for clock pulses within an LCA device. For externally applied clock, increase values by 20%.

Test	Sym	Conditions $-55°C \le Tc \le +125°C$ $V_{CC} = 5.0\ V \pm 10\%$	Group A Subgroups	-50 Min	-50 Max	-70 Min	-70 Max	Units
Application Guidelines, Switching, Internal Buffers[8]								
Clock Buffer	T_{GCK}		N/A		9		6	ns
TBUF								
Data to Output	T_{IO}		N/A		8		5	ns
Three-state to Output								
Single Pull-up	T_{PUS}		N/A		46		38	ns
Pair of Pull-ups	T_{PUF}		N/A		22		16	ns
Bidirectional	T_{BIDI}		N/A		6		4	ns

Table 1. Electrical Performance Characteristics (Continued)

TSC0097

Test	Sym	Conditions –55°C ≤ Tc ≤ +125°C V_{CC} = 5.0 V ±10%	Group A Subgroups	-50		-70		Units
				Min	Max	Min	Max	
Application Guidelines, Switching, IOB[8,10]								
Pad (package pin)		See Fig. 3						
To inputs TCLKIN, BCLKIN	T_{PIDC}		N/A		5		5	ns
To inputs DIRECT IN	T_{PID}	3	N/A		9		6	ns
I/O Clock								
To I/O RI input (FF)	T_{IKRI}	4	N/A		11		7	ns
I/O pad-input setup	T_{PICK}	1	N/A	30		20		ns
I/O pad-input hold	T_{IKPI}	2	N/A	0		0		ns
To I/O pad (fast)	T_{OKPO}	7	N/A		18		13	ns
I/O pad output setup	T_{OOK}	5	N/A	15		10		ns
I/O pad output hold	T_{OKO}	6	N/A	0		0		ns
*Clock (High)	T_{IOH}	11	N/A	9		8		ns
*Clock (Low)	T_{IOL}	12	N/A	9		8		ns
Output								
To pad (enabled fast)	T_{OPF}	10	N/A		15		9	ns
To pad (enabled slow)	T_{OPS}	10	N/A		40		29	ns
Three-State								
To pad begin hi-Z (fast)	T_{TSHZ}	9	N/A		14		12	ns
To pad valid (fast)	T_{TSON}	8	N/A		20		14	ns
Master Reset								
To input RI	T_{RRI}	13	N/A		37		33	ns
To output (FF)	T_{RPO}	14	N/A		53		47	ns

*These parameters are for clock pulses within an LCA device. For externally applied clock, increase values by 20%.

Application Guidelines, Switching, Master Parallel Mode Programming[8,11]

$\overline{RCLK}$,								
To Address Valid	T_{RAC}	1 See Fig.6	N/A	0	200	0	200	ns
To Data Setup	T_{DRC}	2	N/A	60		60		ns
To Data Hold	T_{RCD}	3	N/A	0		0		ns
$\overline{RCLK}$ High	T_{RCH}	4	N/A	600		600		ns
$\overline{RCLK}$ Low	T_{RCL}	5	N/A	4.0		4.0		µs

Table 1. Electrical Performance Characteristics (Continued)

TSC0097

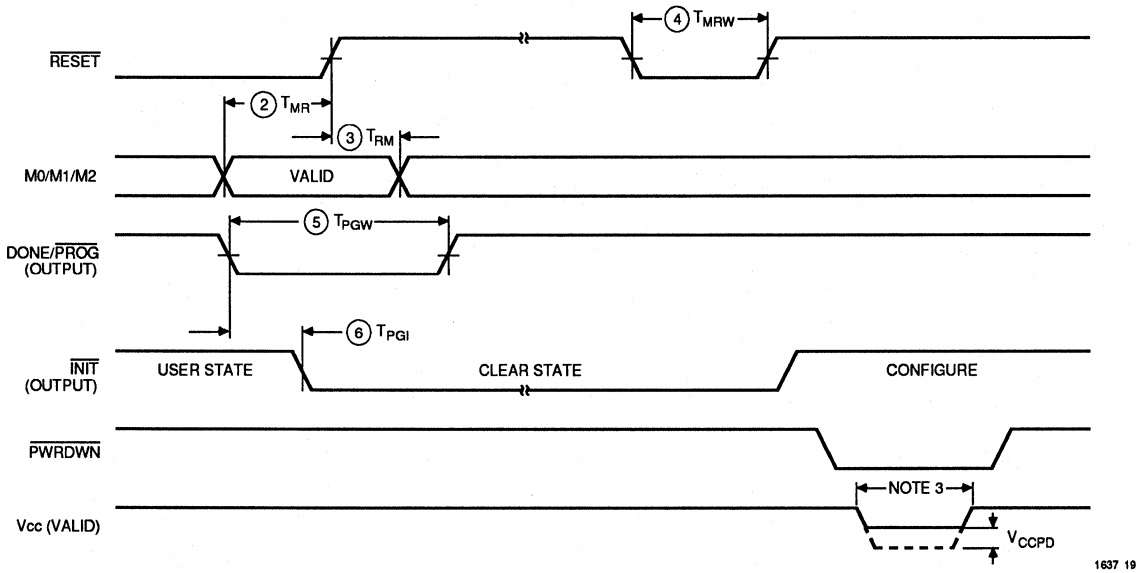


Figure 1. General LCA Waveforms

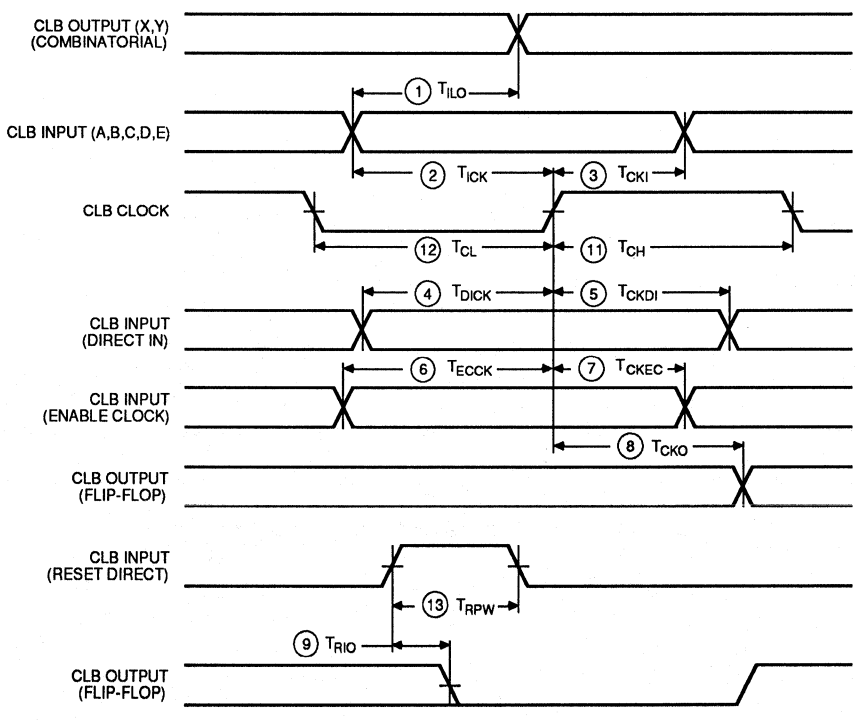


Figure 2. CLB Waveforms

TSC0097

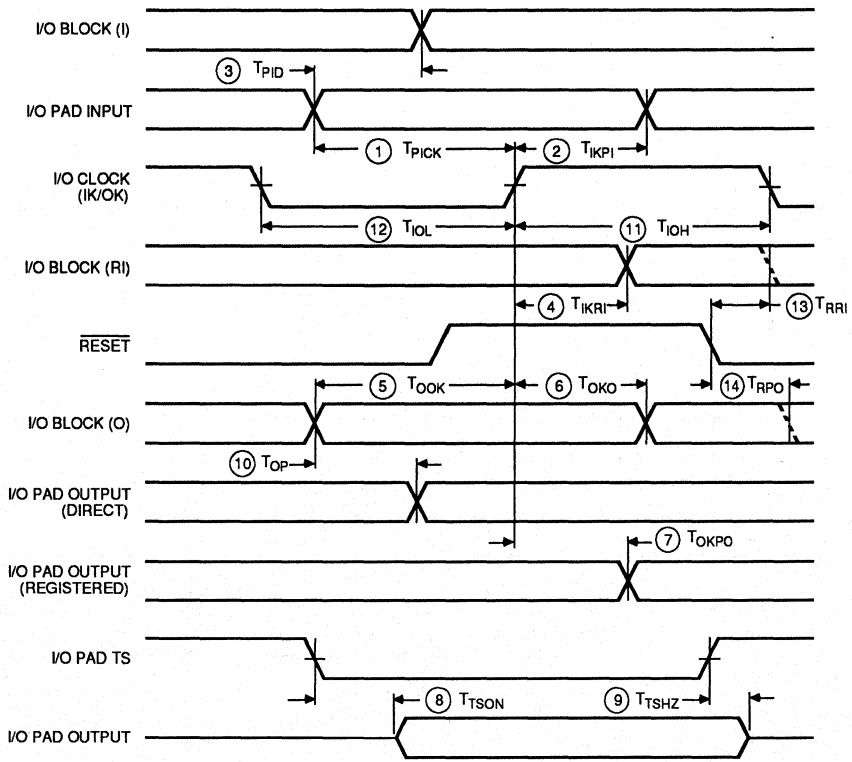
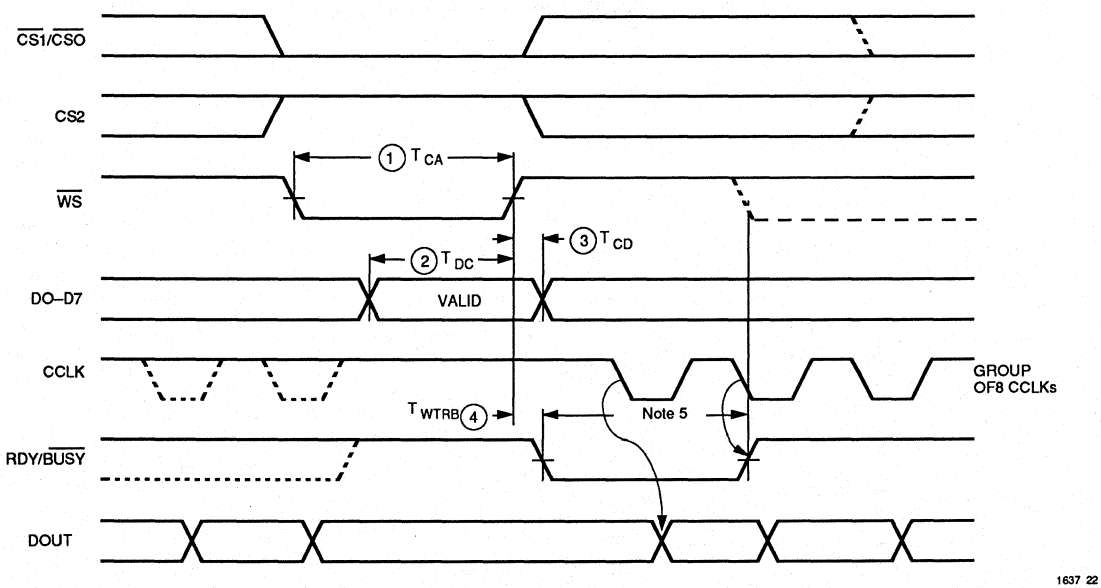


Figure 3. IOB Waveforms



TSC0097

Figure 4. Peripheral Mode Waveforms

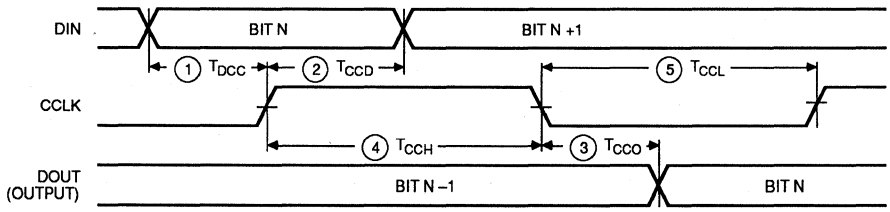


Figure 5. Slave Mode Waveforms

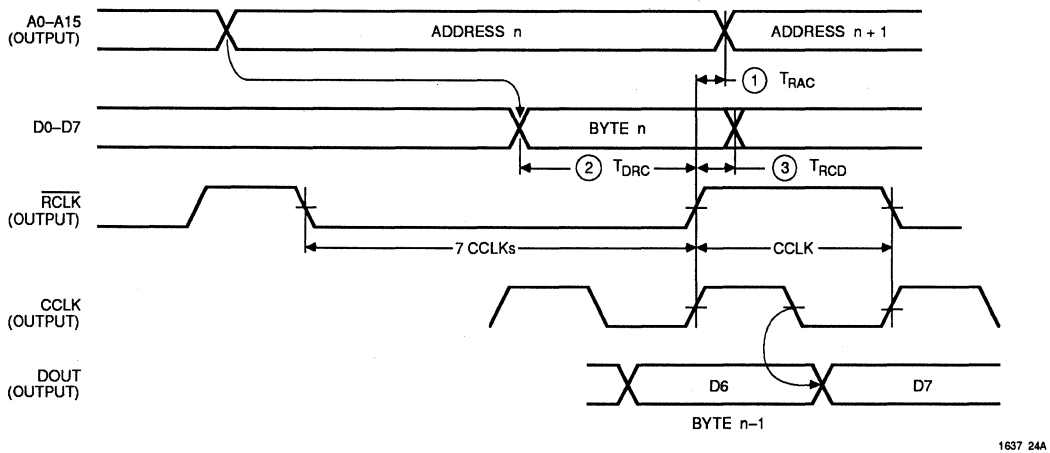


Figure 6. Master Parallel Mode Waveforms

TSC0097

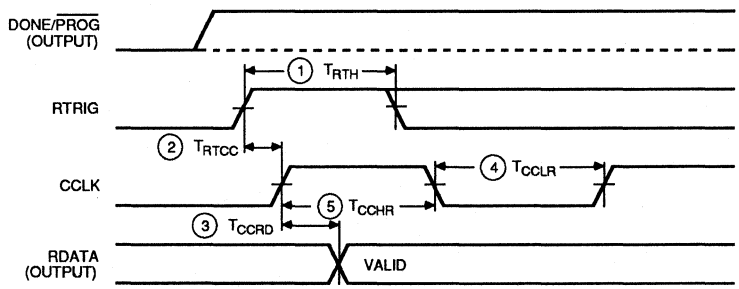
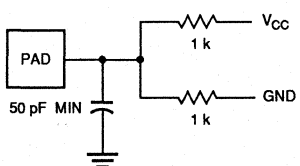


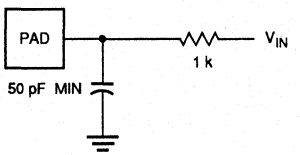
Figure 7. Program Readback Waveforms

XC3090B Data Sheet Notes

1. Xilinx maintains this specification as a controlled document. To comply with the intent of MIL-STD-883, and to insure the use of the most recently released device performance parameters, please request a copy of the current revision of this Test Specification (TSC 0097) from Xilinx.
2. No output current loads, no active input or long line pull-up resistors, and with the device configured with the MAKEBITS "tie" option.
3. $\overline{\text{PWRDWN}}$ transitions must occur during operational Vcc levels.
4. $\overline{\text{RESET}}$ timing relative to valid mode lines (M0, M1, M2) is relevant only when $\overline{\text{RESET}}$ is used to delay configuration.
5. Configuration must be delayed until the $\overline{\text{INIT}}$ of all LCAs is High. $\overline{\text{WS}}$ cannot go active until RDY/$\overline{\text{BUSY}}$ goes High.
6. Readback should not be initiated until configuration is complete.
7. DOUT timing is the same as for slave mode.
8. Testing of the Applications Guidelines is modeled after testing specified by MIL-M-38510/605. Devices are first 100% functionally tested. Benchmark patterns are then used to measure the Application Guidelines. Characterization data are taken at initial device qualification, prior to introduction of significant changes, and at least twice yearly to monitor correlation between benchmark patterns, device performance, XACT software timings, and the data sheet.
9. The CLB K to Q output delay (T_{CKO}) plus the shortest possible interconnect delay is always longer than the Data In hold time requirement (T_{CKDI}) on the same die.
10. Voltage levels of unused pads must be valid logic levels. Each can be configured with the internal pull-up resistor, configured as a driven output, or driven from an external source.
11. At power-up, V_{CC} must rise from 2.0V to V_{CC} minimum in less than 10 ms. Otherwise, delay configuration using $\overline{\text{RESET}}$.
12. All timings except T_{TSHZ} and T_{TSON} are measured at 1.5 V level with 50 pF minimum load output. For input signals, rise and fall times are less than 6 ns, with low amplitude = 0V, and high = 3V.
    T_{TSHZ} is determined when the output shifts 10% (of the output voltage swing) from V_{OL} level or V_{OH} level. The following circuit is used:



12. (continued)
    T_{TSON} is measured at 0.5 V_{cc} level with V_{IN} = 0 for 3-State to active High, and V_{IN} = V_{CC} for 3-State to active Low. The following load circuit is used:



TSC0097

TSC0097

XC1736A/XC1765 Serial Configuration PROM

Product Specification

FEATURES

- One-Time Programmable (OTP) 36,288 x 1 bit and 65,536 x 1 bit serial memories designed to store configuration programs for Programmable Gate Arrays
- Simple interface to Logic Cell™ Arrays (LCA) requires only one user I/O pin
- Daisy chain configuration support for multiple XC2000 or XC3000 LCAs
- Cascadable to support additional configurations or future higher-density arrays
- Military XC1765R screening and quality conformance inspection is patterned after the requirements of MIL-STD-883, methods 5004 and 5005.
- Low-power CMOS EPROM process
- Programmable reset polarity for the XC1765
- Available in the space-efficient 8-pin plastic or ceramic DIP, or in 20-pin surface-mount PLCC package
- PC-based programming supported by the XILINX DS112 and other leading programmer manufacturers

DESCRIPTION

The XC1736A/XC1765 Serial Configuration PROMs (SCP) provide an easy-to-use, cost-effective configuration memory for Xilinx Field Programmable Gate Arrays. Both the XC1736A and the XC1765 are packaged in the economical 8-pin plastic DIP and are also available in the popular 20-pin Plastic Leaded Chip Carrier. The XC1765 is also available in an 8-pin ceramic DIP that supports the military temperature range. The XC17XX family uses a simple serial-access procedure to configure one or more LCA devices. The XC1765 organization (65,536 x 1) supplies enough memory to configure one XC3090 or multiple smaller LCAs. Multiple configurations for a single LCA can also be loaded from the XC17XX family. Using a special feature of the XC1765, the user can select the polarity of the reset function by programming a special EPROM bit.

The XC1736A/XC1765 can be programmed with the PC-based Xilinx XC-DS112 Configuration PROM Programmer or with programmers from other manufacturers. The LCA design file is first compiled into a standard HEX format with the XC-DS501 Development System. It can then be transferred to the programmer through a serial port on the PC.

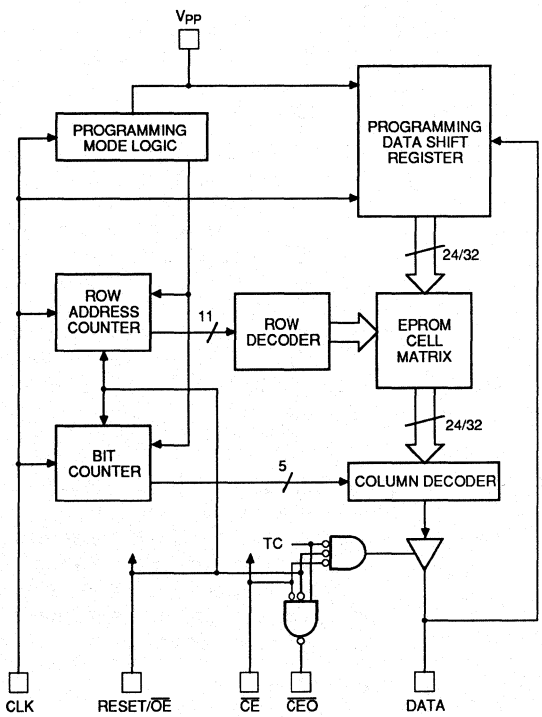
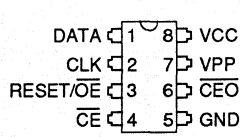
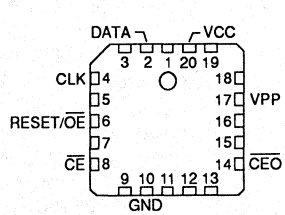


Figure 1. XC1736A/XC1765 Block Diagram



XC1736A/XC1765 8-Pin DIP Pin Assignments



XC1736A/XC1765 20-Pin PLCC Pin Assignments

Table 1. XC1736A/XC1765 Pin Assignments for 8-Pin DIP

PLCC Pin	DIP Pin	Name	I/O	Description
2	1	DATA	O	Three-state DATA output for reading. Input/Output pin for programming.
4	2	CLK	I	Clock input. Used to increment the internal address and bit counters for reading and programming.
6	3	RESET/ $\overline{OE}$	I	Output Enable input. A Low level on both the $\overline{CE}$ and RESET/$\overline{OE}$ inputs enables the data output driver. A High level on RESET/$\overline{OE}$ resets both the address and bit counters. In the XC1765, the logic polarity of this input is programmable as either RESET/$\overline{OE}$ or OE/$\overline{RESET}$. This document describes the pin as RESET/$\overline{OE}$ although the opposite polarity is also possible on the XC1765.
8	4	$\overline{CE}$	I	Chip Enable input. Used for device selection. A Low level on both $\overline{CE}$ and $\overline{OE}$ enables the data output driver. A High level on $\overline{CE}$ disables both the address and bit counters and forces the device into a low power mode.
10	5	GND		Ground pin.
14	6	$\overline{CEO}$	O	Chip Enable Out output. This signal is asserted Low on the clock cycle following the last bit read from the memory. It will stay Low as long as $\overline{CE}$ and $\overline{OE}$ are both Low. It will then follow $\overline{CE}$ until $\overline{OE}$ goes High. Thereafter $\overline{CEO}$ will stay High until the entire PROM is read again and senses the status of RESET polarity.
17	7	V_{PP}		Programming Voltage Supply. Used to enter programming mode (+6 V) and to program the memory (+15 V) Must be connected directly to Vcc for normal Read operation. No overshoot above +15.5 V permitted.
20	8	V_{CC}		+5 V power supply input.

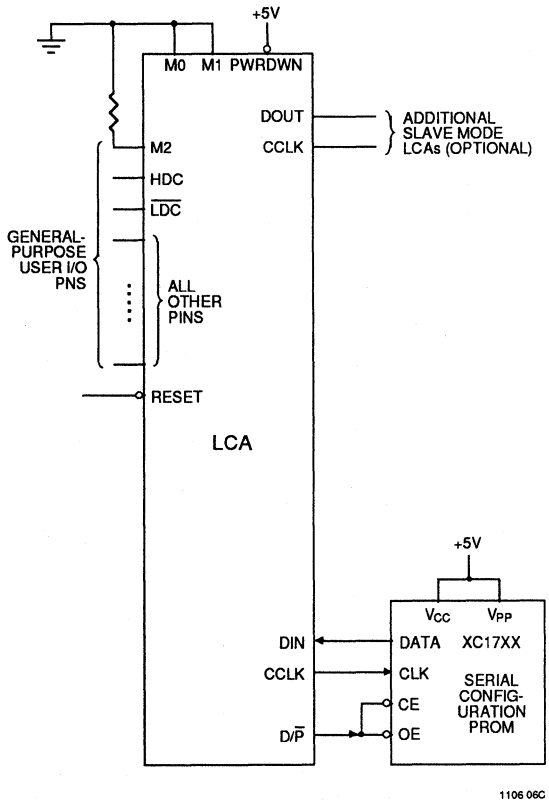


Figure 2. Master Serial Mode Configuration

CONTROLLING THE XC1736A/XC1765 SERIAL PROMS

Most connections between the LCA and the Serial PROM are simple and self-explanatory:

- The DATA output of the XC1736A (or XC1765) drives DIN of the LCA devices.
- The master LCA CCLK output drives the CLK input of the XC1736A/XC1765(s)
- The $\overline{CEO}$ output of any XC1736A/XC1765 can be used to drive the $\overline{CE}$ input of the next XC1736A/XC1765 in a cascade chain of PROMs.
- V_{PP} must be connected to V_{CC}. Leaving V_{PP} open can lead to unreliable, temperature-dependent operation.

There are, however, two different ways to use the inputs $\overline{CE}$ and $\overline{OE}$.

1. The LCA D/$\overline{P}$ output drives both $\overline{CE}$ and $\overline{OE}$ in parallel. This is the simplest connection, but it fails when a user applies $\overline{RESET}$ during the LCA configuration process. The LCA will abort the configuration and then restart a new configuration, as intended, but the

XC1736A/XC1765 does not reset its address counter, since it never saw a High level on its $\overline{OE}$ input. The new configuration will, therefore, read whatever data is stored at the higher address locations in the PROM and re-configuration will fail.

2. The $\overline{LDC}$ output from the LCA drives the $\overline{CE}$ input of the XC1736A/XC1765, while its $\overline{OE}$ input is driven by the inversion of the LCA $\overline{RESET}$ input. This connection works under all normal circumstances, even when the user aborts a configuration before D/$\overline{P}$ has gone High. The High level on the $\overline{OE}$ input during RESET clears the PROM internal address pointer, so that the reconfiguration starts at the beginning. Most designs have a "spare" inverter or inverting gate that can be used for this purpose.

LCA MASTER SERIAL MODE SUMMARY

The I/O and logic functions of the Logic Cell Array and their associated interconnections, are established by a configuration program. The program is loaded either automatically upon power up, or on com-mand, depending on the state of the three LCA mode pins. In Master Mode, the Logic Cell Array automatically loads the configuration program from an external memory. The Serial Configuration PROM has been designed for compatibility with the Master Serial Mode.

Upon power-up or upon reconfiguration, an LCA device will enter Master Serial Mode whenever all three of the LCA mode-select pins are Low (M0=0, M1=0, M2=0). Data are read from the Serial Configuration PROM sequentially on a single data line. Synchronization is provided by the rising edge of the temporary signal CCLK, which is generated during configuration.

Master Serial Mode provides a simple configuration interface. Only a serial data line and two control lines are required to configure an LCA. Data from the Serial Configuration PROM is read sequentially, accessed via the internal address and bit counters which are incremented on every valid rising edge of CCLK.

Programming the LCA With Counters Reset Upon Completion

Figure 2 shows the connections between an LCA and its SCP. The DATA line from the SCP is connected to the DIN input of the LCA. CCLK is connected to the CLK input of the SCP. At power-up or upon reconfiguration, the D/$\overline{P}$ signal goes Low (pulled Low by the LCA at reset, or by external circuitry for reconfiguration), enabling the SCP and its DATA output. During the configuration process, CCLK clocks data out of the SCP on every rising clock edge. At the completion of configuration, the DONE/$\overline{PROG}$ signal goes High and resets the internal address counters of the SCP.

If the user-programmable, dual-function DIN pin is used only for configuration, it must still be held at a defined level during normal operation. The XC3000 family takes care of this automatically with an on-chip default pull-up resistor. On XC2000-family devices, the user must either configure DIN as an active output, or somehow provide a defined level, e.g., by using an external pull-up resistor, if DIN is configured as an inputs.

If the LCA is to be reprogrammed after initial power-up, note that the LCA requires several microseconds to respond after the D/$\overline{P}$ pin is pulled Low. In this case, the $\overline{LDC}$ pin can be used instead of the D/$\overline{P}$ pin to control the SCP.

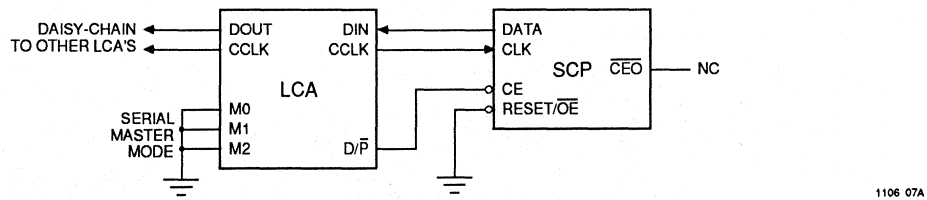


Figure 3. Address Counters Not Reset

Notes: 1. If M2 is tied directly to ground, it should be programmed as an input during operation.
2. If the LCA is reset during configuration, it will abort back to initialization state. D/$\overline{P}$ will not go High, so an external signal is required to reset the 17XX counters.

Programming the LCA With Counters Unchanged Upon Completion

When multiple LCA configurations for a single LCA are stored in a Serial Configuration PROM, the $\overline{OE}$ pin should be tied Low as shown in Figure 3. Upon power-up, the internal address counters are reset and configuration begins with the first program stored in memory. Since the $\overline{OE}$ in is held Low, the address counters are left unchanged after configuration is complete. Therefore, to reprogram the LCA with another program, the DONE/$\overline{PROG}$ line is pulled Low and configuration begins at the last value of the address counters.

Cascading Serial Configuration PROMs

For multiple LCAs configured as a daisy-chain, or for future LCAs requiring larger configuration memories, cascaded SCPs provide additional memory.

After the last bit from the first SCP is read, the SCP asserts its $\overline{CEO}$ output Low and disables its DATA line. The next SCP recognizes the Low level on its $\overline{CE}$ input and enables its DATA output. See Figure 4.

After configuration is complete, the address counters of all cascaded SCPs are reset when DONE/$\overline{PROG}$ goes High, forcing the RESET/$\overline{OE}$ on each SCP to go High.

If the address counters are not to be reset upon completion, then the $\overline{OE}$ inputs can be tied to ground, as shown in Figure 3. To reprogram the LCA with another program, the DONE/$\overline{PROG}$ line goes Low and configuration begins where the address counters had stopped. In this case, avoid contention between DATA and the configured I/O use of DIN.

Extremely large, cascaded memories in some systems may require additional logic if the rippled chip enable is too slow to activate successive SCPs.

STANDBY MODE

The XC17XX enters a low-power standby mode whenever $\overline{CE}$ is asserted High. In this mode, the SCP consumes less than 0.5 mA of current. The output remains in a high-impedance state regardless of the state of the $\overline{OE}$ input.

PROGRAMMING MODE

Figures 5 and 6 show the programming algorithm for the XC1736A/XC1765. Note that programming mode is entered by holding V_{PP} High for at least two clock edges and is exited by removing power from the device or by a Low on both $\overline{CE}$ and $\overline{OE}$.

XC1765 RESET POLARITY

The XC1765 lets the user choose the reset polarity as either RESET/$\overline{OE}$ or OE/$\overline{RESET}$. The Xilinx DS112 programmer and its XPP software prompt the user for the desired reset polarity. Any third-party commercial programmer should do the same.

(The polarity is programmed into the first four overflow byte locations, 2000H through 2003H. 00000000 in these locations makes the reset active Low, FFFFFFFF in these locations makes the reset active High. The programming of these overflow bytes should be handled transparently by the PROM programmer; it is mentioned here only as additional information.)

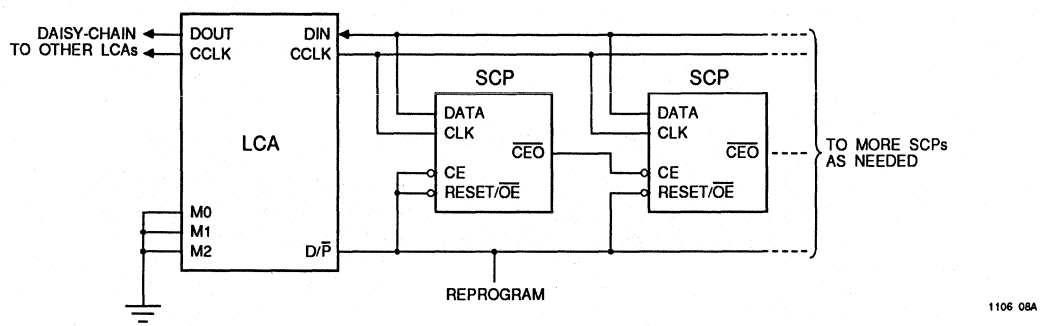


Figure 4. Cascading SCPs

ABSOLUTE MAXIMUM RATINGS

Symbol	Description		Units
V_{CC}	Supply voltage relative to GND	–0.5 to +7.0	V
V_{PP}	Supply voltage relative to GND	–0.5 to +15.5	V
V_{IN}	Input voltage with respect to GND	–0.5 to V_{CC} +0.5	V
V_{TS}	Voltage applied to 3-state output	–0.5 to V_{CC} +0.5	V
T_{STG}	Storage temperature (ambient)	–65 to +125	°C
T_{SOL}	Maximum soldering temperature (10 s @ 1/16 in.)	+260	°C

Note: Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those listed under Operating Conditions is not implied. Exposure to Absolute Maximum Ratings conditions for extended periods of time may affect device reliability.

OPERATING CONDITIONS

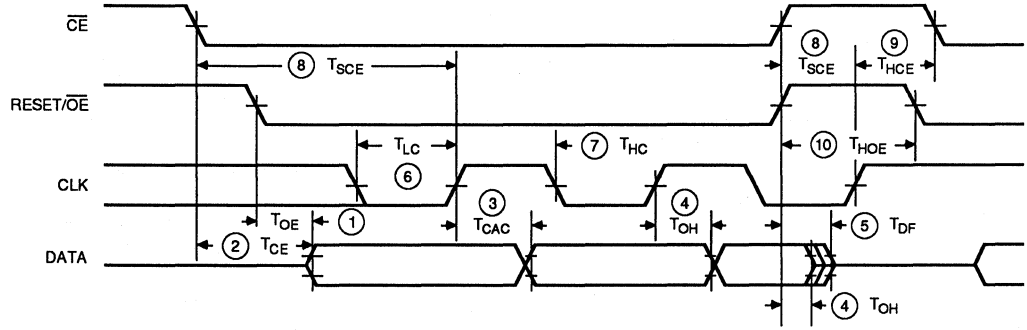
Symbol	Description			Min	Max	Units
V_{CC}	Commercial	Supply voltage relative to GND	–0°C to +70°C	4.75	5.25	V
	Industrial	Supply voltage relative to GND	–40°C to +85°C	4.5	5.5	V
	Military	Supply voltage relative to GND	–55 °C +125°C	4.5	5.5	V

DC CHARACTERISTICS OVER OPERATING CONDITIONS

Symbol	Description		Min	Max	Units
V_{IH}	High-level input voltage		2.0	Vcc	V
V_{IL}	Low-level input voltage		0	0.8	V
V_{OH}	High-level output voltage (I_{OH} = –4 mA)	Commercial	3.86		V
V_{OL}	Low-level output voltage (I_{OL} = +4 mA)	Commercial		0.32	V
V_{OH}	High-level output voltage (I_{OH} = –4 mA)	Industrial	3.76		V
V_{OL}	Low-level output voltage (I_{OL} = +4 mA)	Industrial		0.37	V
V_{OH}	High-level output voltage (I_{OH} = –4 mA)	Military	3.7		V
V_{OL}	Low-level output voltage (I_{OL} = +4 mA)	Military		0.4	V
I_{CCA}	Supply current, active mode			10	mA
I_{CCS}	Supply current, standby mode			0.5	mA
I_L	Input or output leakage current		–10	10	µA

Note: During normal read operation, V_{PP} ***must*** be connected to V_{CC}.

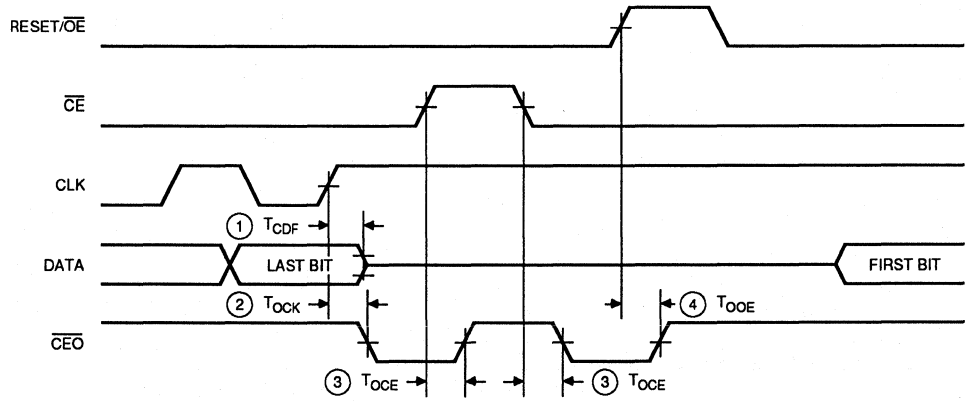
AC CHARACTERISTICS OVER OPERATING CONDITIONS



1106 10

Symbol		Description	Limits Min	Limits Max	Units
1	T_{OE}	$\overline{OE}$ to Data Delay		100	ns
2	T_{CE}	$\overline{CE}$ to Data Delay		250	ns
3	T_{CAC}	CLK to Data Delay		400	ns
4	T_{OH}	Data Hold From $\overline{CE}$, $\overline{OE}$, or CLK	0		ns
5	T_{DF}	$\overline{CE}$ or $\overline{OE}$ to Data Float Delay		50	ns
6	T_{LC}	CLK Low Time[4]	200		ns
7	T_{HC}	CLK High Time	200		ns
8	T_{SCE}	$\overline{CE}$ Setup Time to CLK (to guarantee proper counting)	100		ns
9	T_{HCE}	$\overline{CE}$ Hold Time to CLK (to guarantee proper counting)[4]	0		ns
10	T_{HOE}	$\overline{OE}$ High Time (Guarantees Counters Are Reset)	100		ns

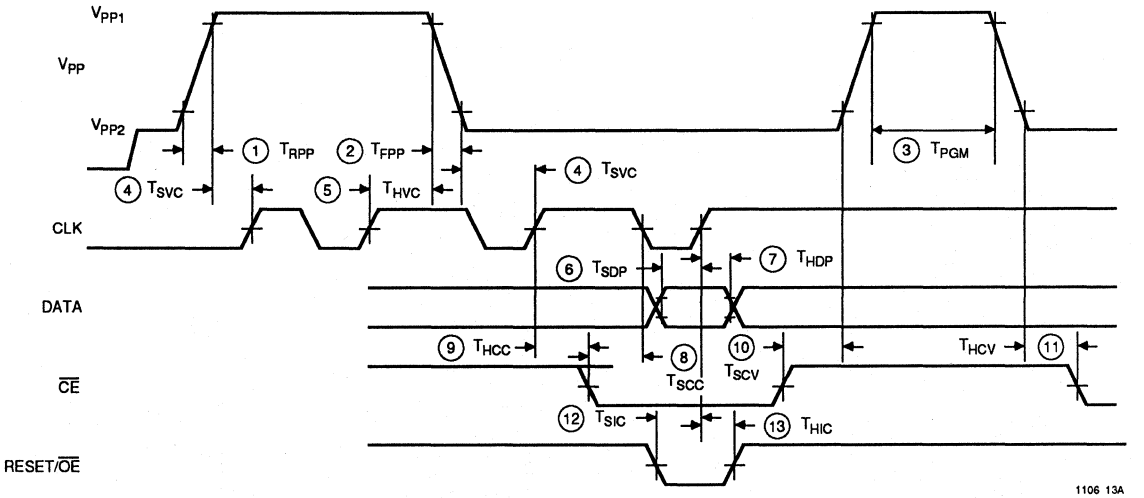
Notes: 1. Preliminary specifications for military operating range only.
2. AC test load = 50 pF.
3. Float delays are measured with minimum tester ac load and maximum dc load.
4. Guaranteed by design, not tested.



1106 11

Symbol			Limits Min	Limits Max	Units
1	T_{CDF}	CLK to Data Float Delay		50	ns
2	T_{OCK}	CLK to $\overline{CEO}$ Delay		100	ns
3	T_{OCE}	$\overline{CE}$ to $\overline{CEO}$ Delay		100	ns
4	T_{OOE}	RESET/$\overline{OE}$ to $\overline{CEO}$ Delay		100	ns

PROGRAMMING SPECIFICATIONS (Guaranteed by design, but not fully tested)



DC PROGRAMMING SPECIFICATIONS

Symbol		Description	Limits Min	Limits Max	Units
1	V_{CCP}	Supply voltage during programming	4.5	5.5	V
2	V_{IL}	Low-level input voltage	0.0	0.5	V
3	V_{IH}	High-level input voltage	2.4	V_{CC}	V
4	V_{OL}	Low-level ouptput voltage		0.4	V
5	V_{OH}	High-level output voltage	3.7		V
6	V_{PP1}	Programming voltage*	14.5	15.5	v
7	V_{PP2}	Programming-mode access voltage	5.75	6.25	V
8	I_{PPP}	Supply current in programming mode		60	mA
9	I_L	Input or output leakage current	-10	10	µA

*No overshoot is permitted on this signal. V_{PP} must not be allowed to exceed V_{PP1} max

AC PROGRAMMING SPECIFICATIONS

Symbol		Description	Limits Min	Limits Max	Units
1	T_{RPP}	10% to 90% Rise Time of VPP	50	70	µs
2	T_{FPP}	90% to 10% Fall Time of VPP	50	70	µs
3	T_{PGM}	VPP Programming Pulse Width	0.95	1.05	ms
4	T_{SVC}	Setup of VPP to CLK to Enter Programming Mode	100		ns
5	T_{HVC}	Hold of VPP to CLK to Enter Programming Mode	300		ns
6	T_{SDP}	Data Setup to CLK for Programming	50		ns
7	T_{HDP}	Data Hold to CLK for Programming	0		ns
8	T_{SCC}*	$\overline{CE}$ Setup to CLK for Programming/Verifying	100		ns
9	T_{HCC}*	$\overline{CE}$ Hold From CLK for Programming/Verifying	200		ns
10	T_{SCV}	$\overline{CE}$ Setup to VPP for Programming	100		ns
11	T_{HCV}	$\overline{CE}$ Hold From VPP for Programming	50		ns
12	T_{SIC}	$\overline{OE}$ Setup to CLK to Increment Address Counter	100		ns
13	T_{HIC}	$\overline{OE}$ Hold From CLK to Increment Address Counter	0		ns

*During programming, $\overline{CE}$ should only be changed while CLK is High and has been High for 200 ns.

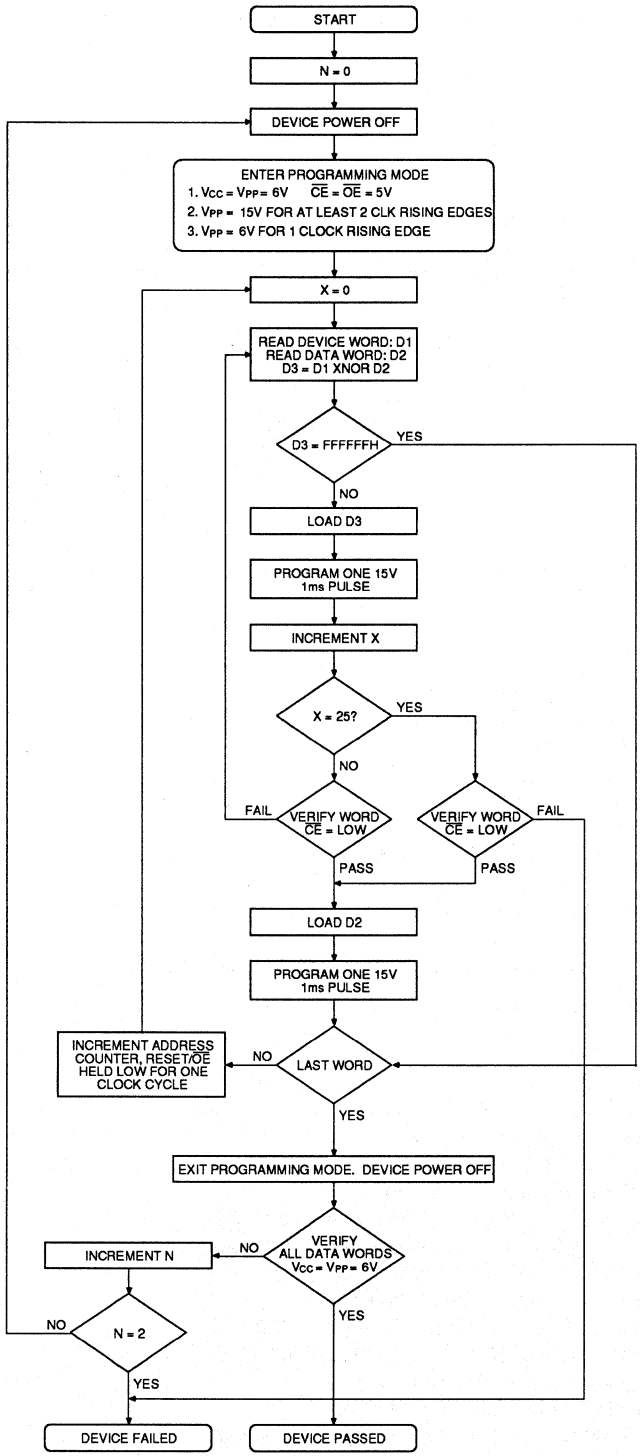


Figure 5. Programming Sequence (XC1736A)

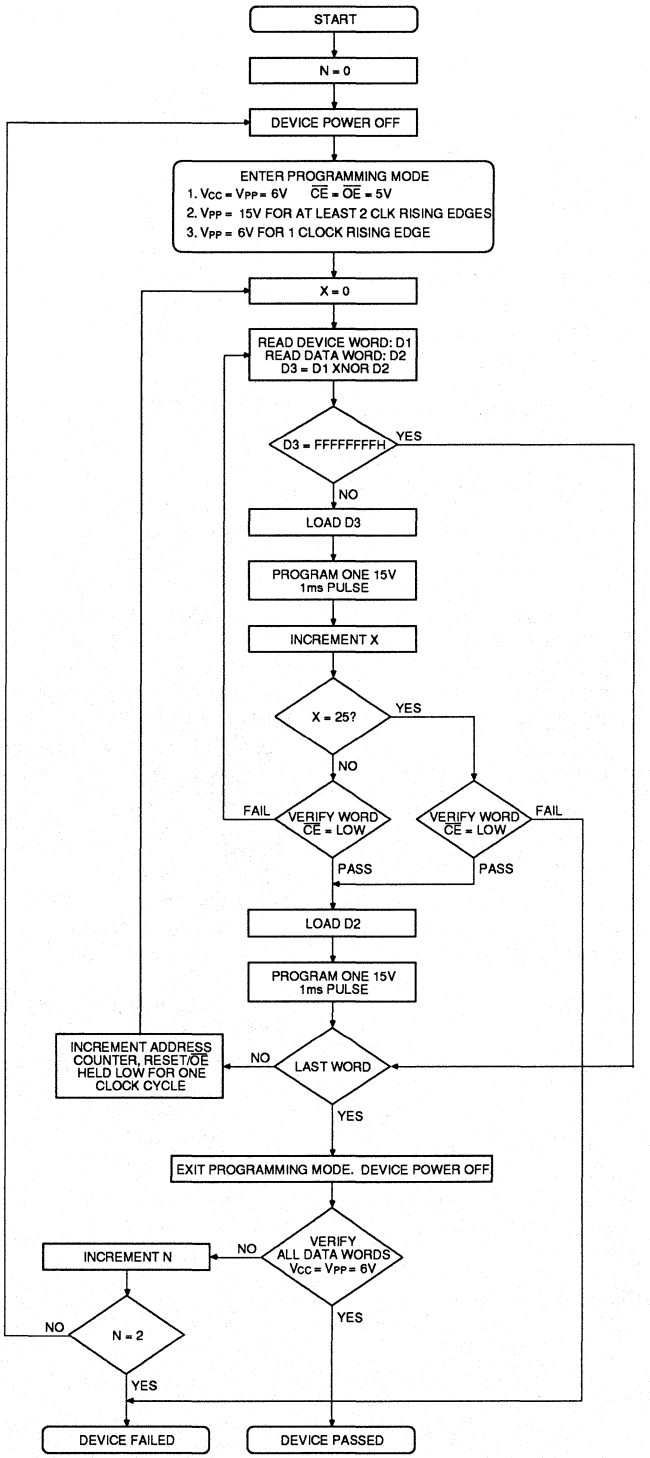


Figure 6. Programming Sequence (XC1765)

SERIAL PROM-PROGRAMMER SUPPORT

Xilinx offers PROM-programmer support for the XC1736A and XC1765 through the DS112 programmer. The latest release of XPP is revision 3.10 which supports the XC1736A and XC1765. The Am1736 is supported by an update to that release: P/N 1060265 for 5-1/4" disk or P/N 1060266 for 3-1/2" disk. Contact Customer Service for details on availability.

Do not program the XC1736A with the DS81 programmer or the XC1736 non "A" algorithm, as this stresses the device causing potential reliability problems. Use the DS112 with the XC1736A algorithm only!

There have been numerous inquiries regarding other vendor support for the serial PROM family (XC1736A and XC1765). Below is the latest list of PROM-programmer manufacturers that offer support for the XC1736A, XC1765, and the Am1736.

Data I/O
(206) 881-6444
Model 29B Unipak 2B V21 XC1736A
V22 XC1765
Model Unisite V3.0 XC1736A
V3.1 XC1765
Model 2900 V1.1 XC1736A
V1.1 XC1765
V1.1 Am1736

Advin Systems
(408) 984-8600
Model SAILOR-PAL/SA
SAILOR-PAL/SB
PILOT 142 143 144 145
Supports XC1736A Am1736 XC1765

Logical Devices
(305) 974-0967
Model ALLPRO
XC1736A
XC1765 V1.51
XC1736 V1.5

Oliver Advanced Engineering
(818) 240-0080
Model OMNI 40 and ONMI 64
XC1736A Rel 2.51Q
XC1765 Rel 2.51Q
Am1736 Rel 2.51Q

Stag Microsystems LTD
UK 707 332-148
US (408) 998-1118
Model System 3000
XC1736A
XC1765

Bytek Corporation
(408) 437-2414
Model 135H-U
XC1736A
XC1765
Am1736

System General
(408) 263-6667
Model SGUP-85A V1.7 XC1736A
V1.7 XC1765

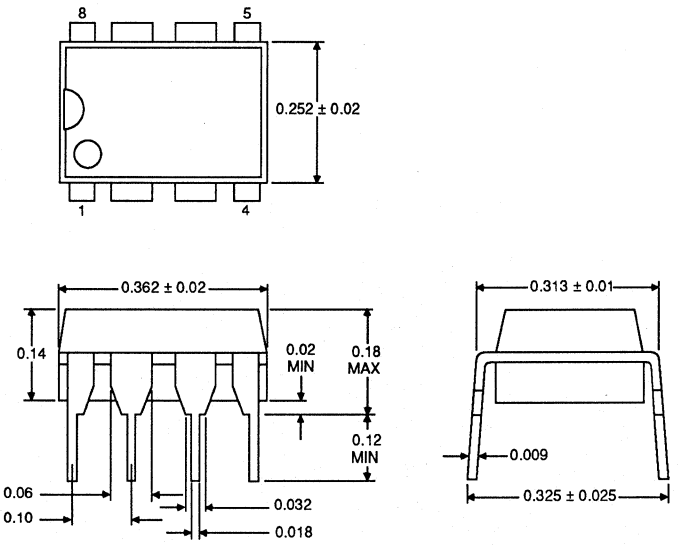
BP Microsystems
(713) 461-9606
Model EP1140 Head 40A
V1.40 XC1736A
V1.40 XC1765
V1.40 Am1736

Link Computer Graphics
(201) 994-6669
Model CLK 3100 V3.1 or greater
XC1736A
XC1765

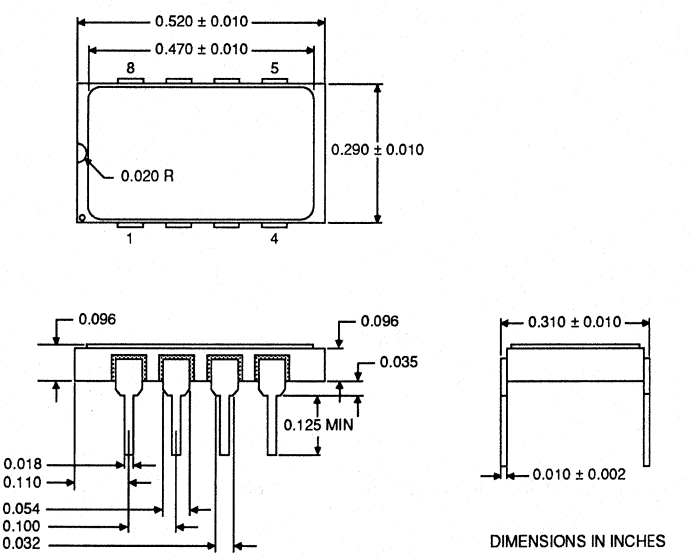
Xeltek
(408) 727-6995
Model Unipro
V2.13 XC1736A
V2.13 XC1765

Pistohl Electronic Tool Company
(408) 255-2422
Model PET 110 PET 120 PET 130
XC1736A
XC1765
Am1736

PHYSICAL DIMENSIONS

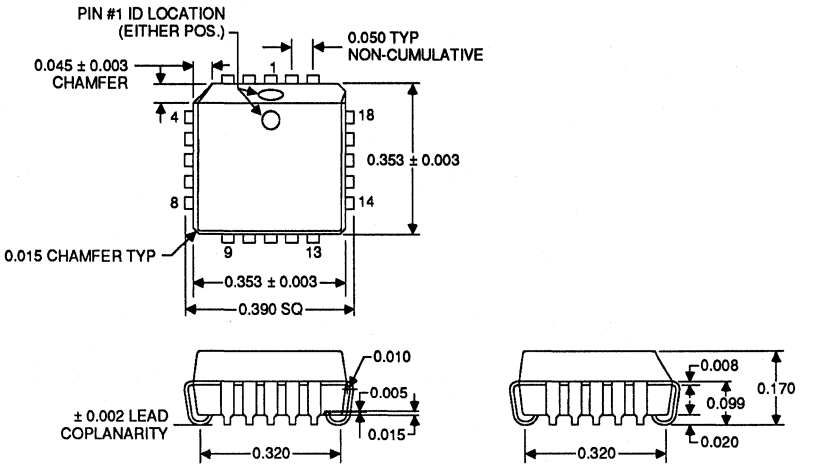


8-Pin Plastic DIP (PD8)

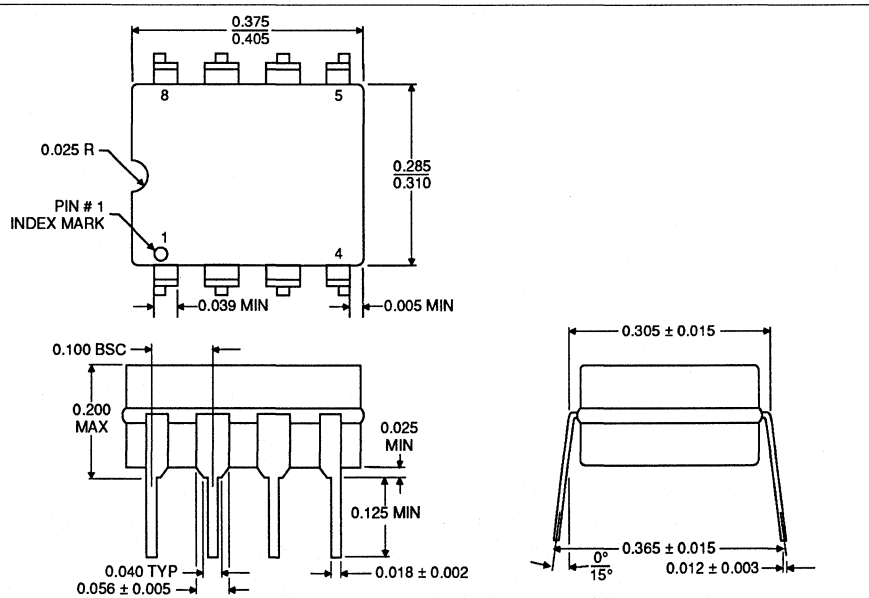


8-Pin Ceramic Sidebrazed DIP (CD8)

PHYSICAL DIMENSIONS (Continued)



20-Pin PLCC (PC20)



8-Pin CerDIP (DD8)

ORDERING INFORMATION AND VALID ORDERING COMBINATIONS

XC17XX - PC20C

DEVICE NUMBER
- XC1736A
- XC1765

PACKAGE TYPE
- PC8 = 8-PIN PLASTIC DIP
- DD8 = 8-PIN CERDIP
- CD8 = 8-PIN CERAMIC SIDE-BRAZED DIP
- PC20 = 20-PIN PLASTIC LEADED CHIP CARRIER

OPERATING RANGE/PROCESSING
- C = COMMERCIAL/INDUSTRIAL (-40° TO +85°C)
- M = MILITARY (-55° TO +125°)
- R = MILITARY (-55° TO +125°C) WITH MIL-STD-883 LEVEL B EQUIVALENT PROCESSING

X1188

XC1736A-PD8C	XC1765-PD8C
XC1736A-PC20C	XC1765-PC20C
XC1736A-CD8M	XC1765-CD8M
	XC1765-DD8M
	XC1765-DD8R

Sockets

Below are two lists of manufactures known to offer sockets for Xilinx package types. This list does not imply an endorsement by Xilinx. Each user must evaluate the particular socket type.

There are no wire-wrap sockets for PLCCs. One solution is to piggy-back a through-hole PLCC socket mounted in a compatible PGA socket with wire-wrap pins. Note that the board-layout then differs from a PGA board layout.

Zero Insertion Force (ZIF) sockets, recommended for prototyping with 132 and 175 pin PGA devices, also lack the wire-wrap option. Piggy-back the ZIF socket in a normal PGA wire-wrap socket.

PLCC Sockets

AMP Inc.
Harrisburg, PA 17105
(717) 564-0100

Burndy Corp.
Richards Ave.
Norwalk, CT 06856
(203) 852-8437

Garry Electronics
9 Queen Anne Court
Langhorne, PA 19047-1803
(215) 949-2300

Honda - MHOtronics
Deerfield. IL 60015
444 Lake Cook Road, Suite 8
(312) 948-5600

ITT Cannon
10550 Talbert Ave.
P.O.Box 8040
Fountain Valley, CA 92728
(714) 964-7400

Kycon Cable & Connector
1772 Little Orchard Street
San Jose, CA 95125
(408) 295-1110

Maxconn Inc.
1855 O'Toole Ave., D102
San Jose, CA 95131
(408) 435-8666

Methode Electronics Inc.
1700 Hicks Road
Rolling Meadows, IL 47150
(312) 392-3500

Mill-Max Mfg. Corp.
190 Pine Hollow Road
Oyster Bay, N.Y. 11771-0300
(516) 922-6000

Precicontact Inc.
835 Wheeler Way
Langhorne, PA 19047
(215) 757-1202

Robinson Nugent
800 East Eighth Street
New Albany, IN 47150
(812) 945-0211

Samtec Inc.
P.O.Box 1147
New Albany, IN 47150
(812) 944-6733

3M Textool
Austin, TX
(800) 328-7732

Thomas & Betts Corp.
920 Route 202
Raritan, NJ 08869
(201) 469-4000

Wells Electronics, Inc.
1701 South Main Street
South Bend, IN 46613
(219) 287-5941

Yamaichi Electronics, Inc.
1420 Koll Circle
Suite B
San Jose, CA 95112
(408) 452-0799

PGA Sockets

Advanced Interconnections
5 Energy Way
West Warwick, RI 02893
(401) 823-5200

AMP Inc.
Harrisburg, PA 17105
(717) 564-0100

Aries Electronics, Inc.
P.O.Box 130
Frenchtown, NJ 08825
(201) 996-6841

Augat
33 Perry Ave.
P.O.Box 779
Attleboro, MA 02703
(617) 222-2202

Bevmar Industries, Inc.
20601 Annalee Ave.
Carson, CA 90746
(213) 631-5152

Bevmar Industries, Inc.
1 John Clarke Rd.
Middletown, RI 02840
(401) 849-4803

Electronic Molding Corp.
96 Mill Street
Woonsocket, RI 02895
(401) 769-3800

Garry Electronics
9 Queen Anne Court
Langhorne, PA 19047-1803
(215) 949-2300

Mark Eyelet Inc.
63 Wakelee Road
Wolcott, CT 06716
(203) 756-8847

McKenzie Technology
44370 Old Warm Springs Blvd.
Fremont CA 94538
(415) 651-2700

Methode Electronics Inc.
1700 Hicks Road
Rolling Meadows, IL 47150
(312) 392-3500

Mill-Max Mfg. Corp.
190 Pine Hollow Road
Oyster Bay, N.Y. 11771-0300
(516) 922-6000

Precicontact Inc.
835 Wheeler Way
Langhorne, PA 19047
(215) 757-1202

Robinson Nugent
800 East Eighth Street
New Albany, IN 47150
(812) 945-0211

Samtec Inc.
P.O.Box 1147
New Albany, IN 47150
(812) 944-6733

Texas Instruments
CSD Marketing, MS 14-1
Attleboro, MA 02703
(617) 699-5206

Thomas & Betts Corp.
920 Route 202
Raritan, NJ 08869
(201) 469-4000

Yamaichi Electronics, Inc.
1420 Koll Circle
Suite B San Jose, CA 95112
(408) 452-0799

Quality, Testing, and Packaging

Quality Assurance and Reliability

QUALITY ASSURANCE PROGRAM

All aspects of the Quality Assurance Program at Xilinx have been designed in compliance with the requirements of Appendix A of MIL-M-38510. This program emphasizes heavily the aspects of operator training and certification, the use of "accept only on zero defects" lot sampling plans, and extensive audits of both internal departments and outside suppliers.

Xilinx utilizes the world-class wafer fabrication facilities of Seiko-Epson's plant in Fujimi, Suwa, Japan and the high-volume assembly resources of ANAM in Seoul, the Republic of Korea. Periodic quality assurance audits of these facilities to the full requirements of MIL-STD-883 are routinely performed.

Xilinx calculates its outgoing component quality level, expressed in PPM (defective parts per million devices shipped), using the industry-standard methods now adopted by JEDEC and published in JEDEC Standard 16. These figures of merit are revised and published quarterly by Xilinx Quality Assurance and are available from local manufacturer's representatives or from Xilinx. These summary data are available for downloading from the Xilinx Electronic Bulletin Board at (408) 559-9327 [1200/2400 baud; 8 data bits; no parity; 1 stop bit] supporting all of the following communications protocols: ASCII, Kermit, XModem, -CRC, and Telink.

RELIABILITY INTRODUCTION

From its inception, Xilinx has been committed to delivering the highest quality, most reliable programmable gate arrays available. A strong Quality Assurance and Reliability program begins at the initial design stages and is carried through to final shipment. The final proof of our success is in the performance of the Logic Cell™ Array (LCA) in our customers' systems applications. An extensive, on-going reliability-testing program is used to predict the field performance of our devices.

These tests provide an accelerated means of emulating long-term system operation in severe field environments. From the performance of the devices during these tests, predictions of actual field performance under a variety of conditions can be calculated.

This report describes the nature and purpose of the various reliability tests performed on finished devices. Updated summaries are available upon request from the Quality Assurance and Reliability Department at Xilinx.

OUTLINE OF TESTING

Qualification testing of devices is performed to demonstrate the reliability of the die used in the device, and the materials and methods used in the assembly of the device. Testing methods are derived from and patterned after the methods specified in MIL-STD-883.

Referral to the test methods of MIL-STD-883 is not intended to imply that nonhermetic products comply with the requirements of MIL-STD-883. These test methods are recognized industry-wide as stringent tests of reliability and are commonly used for nonmilitary-grade semiconductor devices, as well as for fully compliant military-grade products.

Hermetic packages are qualified using the test methods specified in MIL-STD-883. The Group D package qualification tests are performed on one lot of each package type from each assembly facility every twelve months.

A summary of the reliability demonstration tests used at Xilinx is contained in Table 1.

DIE QUALIFICATION

Name of Test	Test Conditions	Lot Tolerance Percent Defective Minimum Sample Size/ Maximum Acceptable Failures
1. High Temperature Life	1000 hr min equivalent at temperature = 125°C Actual test temperature = 145°C Max. rated operating voltage. Life test circuit equivalent to MIL-STD-883	LTPD = 5, s = 105, c = 2
2. Biased Moisture Life	1000 hr min exposure T = 85°C, RH = 85% Max. rated operating voltage. Biased moisture life circuit equivalent to MIL-STD-883	LTPD = 5, s = 105, c = 2

NON-HERMETIC PACKAGE INTEGRITY and ASSEMBLY QUALIFICATION

Name of Test	Test Conditions	Lot Tolerance Percent Defective Minimum Sample Size/ Maximum Acceptable Failures
3. Unbiased Pressure Pot	96 hr min. exposure T = 121°C, P = 2 atm H_2O sat.	LTPD = 5, s = 105, c = 2
4. Thermal Shock	MIL-STD-883, Method 1011, Cond. C –65°C to +150°C 100 cycles	LTPD = 5, s = 75, c = 2
5. Temperature Cycling	MIL-STD-883, Method 1010, Cond. C –65°C to +150°C 200 cycles	LTPD = 5, s = 105, c = 2
6. Salt Atmosphere	MIL-STD-883, Method 1009, Cond. A 24 hrs	s = 25, c = 0
7. Resistance to Solvents	MIL-STD-883, Method 2015	s = 4, c = 0
8. Solderability	MIL-STD-883, Method 2003	s = 3, c = 0
9. Lead Fatigue	MIL-STD-883, Method 2004	s = 2, c = 0

Table 1A. Reliability Testing Sequence for Non-Hermetic Logic Cell Arrays

HERMETIC PACKAGE INTEGRITY and ASSEMBLY QUALIFICATION

	Name of Test	Test Conditions	Lot Tolerance Percent Defective Minimum Sample Size/ Maximum Acceptable Failures
1.	Subgroup D1: Physical Dimensions	MIL-STD-883, Method 2016	LTPD = 15, s = 34, c = 2
2.	Subgroup D2		LTPD = 15, s = 34, c = 2
	a. Lead Integrity	MIL-STD-883, Method 2028	
	b. Seal (fine and gross leak)	MIL-STD-883, Method 1014 (not required for PGA's)	
3.	Subgroup D3		LTPD = 15, s = 34, c = 2
	a. Thermal Shock-15 cycles	MIL-STD-883, Method 1011, Cond. B	
	b. Temp. cycling-100 cycles	MIL-STD-883, Method 1010, Cond. C	
	c. Moisture Resistance	MIL-STD-883, Method 1004	
	d. Seal (fine & gross leak)	MIL-STD-883, Method 1014	
	e. Visual Examination	MIL-STD-883, Method 1004 *and* Method 1010.	
	f. End-point electricals	Group A, subgroup 1	
4.	Subgroup D4		LTPD = 15, s = 34, c = 2
	a. Mechanical Shock	MIL-STD-883, Method 2002, Cond. B	
	b. Vibration, Variable Freq.	MIL-STD-883, Method 2007, Cond. A	
	c. Constant Acceleration min, Y_1 only (Cond. D for large PGAs)	MIL-STD-883, Method 2001, Cond. E	
	d. Seal (fine & gross leak)	MIL-STD-883, Method 1014	
	e. Visual Examination	MIL-STD-883, Method 1010	
	f. End-point electricals	Group A, subgroup 1	
5.	Subgroup D5		LTPD = 15, s = 34, c = 2
	a. Salt Atmosphere	MIL-STD-883, Method 1009, Cond. A	
	b. Seal (fine & gross leak)	MIL-STD-883, Method 1014	
	c. Visual Examination	MIL-STD-883, Method 1009	
6.	Subgroup D6: Internal Water Vapor Content	MIL-STD-883, Method 1018, 5000 ppm water at 100°C	s = 3; c = 0 or s = 5; c = 1
7.	Subgroup D7: Lead Finish Adhesion	MIL-STD-883, Method 2025	LTPD = 15, s = 34 leads, (3 device min) c = 0
8.	Subgroup D8: Lid Torque	MIL-STD-883, Method 2024 (for ceramic quad flat pack, CQFP only)	LTPD = 5, s = 5, c = 0

Table 1B. Reliability Testing Sequence for Hermetic Logic Cell Arrays

DESCRIPTION OF TESTS

Die Qualification

1. **High Temperature Life** This test is performed to evaluate the long-term reliability and life characteristics of the die. It is defined by the Military Standard from which it is derived as a "Die-Related Test" and is contained in the Group C Quality Conformance Tests. Because of the acceleration factor induced by higher temperatures, data representing a large number of equivalent hours at a normal temperature of 70°C can be accumulated in a reasonable period of time. Xilinx performs its High Temperature Life test at a higher temperature, 145°C, than the more common industry practice of 125°C. For comparison, the Reliability Testing Data Summary in Table 2 gives the equivalent testing hours at 125°C.

2. **Biased Moisture Life** This test is performed to evaluate the reliability of the die under conditions of long-term exposure to severe, high-moisture environments which could cause corrosion. Although it clearly stresses the package as well, this test is typically grouped under the die-related tests. The device is operated at maximum-rated voltage, 5.5 V_{dc}, and is exposed to a temperature of 85°C and a relative humidity of 85% throughout the test.

Package Integrity and Assembly Qualification

3. **Unbiased Pressure Pot** This test is performed at a temperature of 121°C and a pressure of 2 atm of saturated steam to evaluate the ability of the plastic encapsulating material to resist water vapor. Moisture penetrating the package could induce corrosion of the bonding wires and nonglassivated metal areas of the die [bonding pads only for LCA devices]. Under extreme conditions, moisture could cause drive-in and corrosion under the glassivation. Although it is difficult to correlate this test to actual field conditions, it provides a well-established method for relative comparison of plastic packaging materials and assembly and molding techniques.

4. **Thermal Shock** This test is performed to evaluate the resistance of the package to cracking and resistance of the bonding wires and lead frame to separation or damage. It involves nearly instantaneous change in temperature from –65°C to +150°C.

5. **Temperature Cycling** This test is performed to evaluate the long-term resistance of the package to damage from alternate exposure to extremes of temperature or to intermittent operation at very low temperatures. The range of temperatures is –65°C to +150°C. The transition time is longer than that in the Thermal Shock test but the test is conducted for many more cycles.

6. **Salt Atmosphere** This test was originally designed by the US Navy to evaluate resistance of military-grade ship-board electronics to corrosion from sea water. It is used more generally for non-hermetic industrial and commercial products as a test of corrosion resistance of the package marking and finish.

7. **Resistance to Solvents** This test is performed to evaluate the integrity of the package marking during exposure to a variety of solvents. This is an especially important test, as an increasing number of board-level assemblies are subjected to severe conditions of automated cleaning before system assembly operations occur. This test is performed according to the methods specified by MIL-STD-883.

8. **Solderability** This test is performed to evaluate the solderability of the leads under conditions of low soldering temperature following exposure to the aging effects of water vapor.

9. **Lead Fatigue** This test is performed to evaluate the resistance of the completed assembly to vibrations during storage, shipping, and operation.

TESTING FACILITIES

Xilinx has the complete capability to perform High Temperature Life Tests, Thermal Shock, Biased Moisture Life Tests, and Unbiased Pressure Pot Tests in its own Reliability Testing Laboratory. Other tests are being performed by outside testing laboratories with DESC laboratory suitability for each of the test methods they perform.

SUMMARY

Tables 2 and 3 testing data show the actual performance of the Logic Cell Arrays during the initial qualification tests to which they have been subjected. These test results demonstrate the reliability and expected long life inherent in the non-hermetic product line. This series of tests is ongoing as a part of the Quality Conformance Program on non-hermetic devices.

Table 2. Xilinx Reliability Testing Summary

Device Types: XC2018, XC2064, XC3020, XC3030, XC3042 Process/Technology: 1.2 Micron Double-Layer Metal CMOS
Die Attach Method: Silver Epoxy Package Type: 68- & 84-Pin PLCC
Molding Compound: Sumitomo 6300H Date: 1Q 1990

Test	Combined Sample	Failures	Equivalent Mean Hrs/Device at T_A = 125°C	Total Device Hrs at T_A = 125°C	Equivalent Failure Rate in FIT at T_J = 70°C
High Temperature Life Test 145°C	6,418	15	964	Equivalent Device Hrs 6,190,193	36*
Biased Moisture Life Test T = 85°C; RH = 85%	820	10	at T_A = 85°C 696	at T_A = 85°C 570,745	
Unbiased Pressure Pot Test +121°C, 2 atm sat. steam	420	4	372	156,496	
Thermal Shock Test −65°C/+150°C 100 cycles (min)	1,566	3	Mean Cycles per Device 328	Total Device Cycles 514,025	
Temperature Cycling Test −65°C/+150°C 100 cycles (min)	1,026	2	Mean Cycles per Device 192	Total Device Cycles 197,300	
Salt Atmosphere Test MIL-STD-883, Method 1009, Cond. A	55	0	24	1,320	
Resistance to Solvents Test MIL-STD-883, Method 2105	12	0			
Solderability Test MIL-STD-883, Method 2003	12	0			
Lead Fatigue Test MIL-STD-883, Method 2004	2	0			

* Assumed activated energy 0.90 eV

Table 3. Xilinx Reliability Testing Summary

Device Types: XC3090
Die Attach Method:
Molding Compound:

Process/Technology: 1.2 Micron CMOS
Package Type: 175-Pin PPGA
Date: 2Q 1990

Test	Combined No. Lots	Failures	Devices on Test	Equivalent Mean Hrs/Cycle at T_A = 125°C	Total Device Hrs at T_A = 125°C	Equivalent Failure Rate in FIT at T_J = 70°C
High Temperature Life Test 145°C	1	0	45	260	11,700	0*
Biased Moisture Life Test T = 85°C; RH = 85%	1	0**	45	1,000	44,500	
Unbiased Pressure Pot Test +121°C, 2 atm sat. steam	1	0	45	96	4,320	
Thermal Shock Test –65°C/+150°C 100 cycles (min)	1	0	45	200	9,000	
Temperature Cycling Test –65°C/+150°C 100 cycles (min)	1	0	45	200	9,000	
Salt Atmosphere Test MIL-STD-883, Method 1009, Cond. A	1	0	8	24	192	0 (no rejects)
Resistance to Solvents Test MIL-STD-883, Method 2105	1	0	3			0 (no rejects)
Solderability Test MIL-STD-883, Method 2003	1	0	3			0 (no rejects)

*Assumed activation energy 0.90 eV
* *Two non-85/85 anomalies were discounted

Table 4. Reliability Summary Package Qualification
Ceramic Pin Grid Array (PGA) and Ceramic Quad Flat Pack (CQFP)

Code	Test	PG84		PG175		CQ100		PG132	
		Combined Sample	Failures	Combined Sample	Failures	Combined Sample	Failures	Combined Sample	Failures
D1	Physical Dimension	45	0	15	0	40	0	15	0
D2	Lead Integrity Seal	55	0	15	0	40	0	15	0
D3	Thermal Shock	75	0	25	0	59	0	25	0
		Mean Hrs/ Cycle/Device	Total Device Hours	Mean Hrs/ Cycle/Device	Total Device Hours	Mean Hrs/ Cycle/Device	Total Device Hours	Mean Hrs/ Cycle/Device	Total Device Hours
	Thermal Shock	15	1,125	15	375	15	885	15	375
	Temperature Cycle	100	7,500	100	2,500	100	5,900	100	2,500
	Seal								
	Visual								
	End-Point Electrical Parameters								
D4	Mechanical Shock	75	0	25	0	69	1	25	0
	Vibration, Var. Freqency								
	Constant Acceleration								
	Seal								
	Visual Examination								
	End-Point Electrical Parameters								
D5	Salt Atmosphere	55	0	25	0	40	0	15	0
	Seal								
	Visual								
D6	Internal Water-Vapor Content	9	0	5	0	10	0	5	0
D7	Adhesion of Lead Finish	13	0	3	0	6	0	3	0
D8	Lead Torque					10	0		

DATA INTEGRITY

Memory Cell Design

An important aspect of the LCA reliability is the robustness of the static memory cells used to store the configuration program.

The basic cell is a single-ended five-transistor memory element (Figure 1). By eliminating a sixth transistor, which would have been used as a pass transistor for the complementary bit line, a higher circuit density is achieved. During normal operation, the outputs of these cells are fixed, since they determine the user configuration. Write and readback times, which have no relation to the device performance during normal operation, will be slower without the extra transistor. In return, the user receives more functionality per unit area.

This explains the basic cell, but how is the LCA user assured of high data integrity in a noisy environment? Consider three different situations: normal operation, a Write operation and a Read operation. In the normal operation, the data in the basic memory element is not changed. Since the two circularly linked inverters that hold the data are physically adjacent, supply transients result in only small relative differences in voltages. Each inverter is truly a complementary pair of transistors. Therefore, whether the output is High or Low, a low-impedance path exists to the supply rail, resulting in extremely high noise immunity. Power supply or ground transients of several volts have no effect on stored data.

The transistor driving the bit line has been carefully designed so that whenever the data to be written is opposite the data stored, it can easily override the output of the feedback inverter. The reliability of the Write operation is

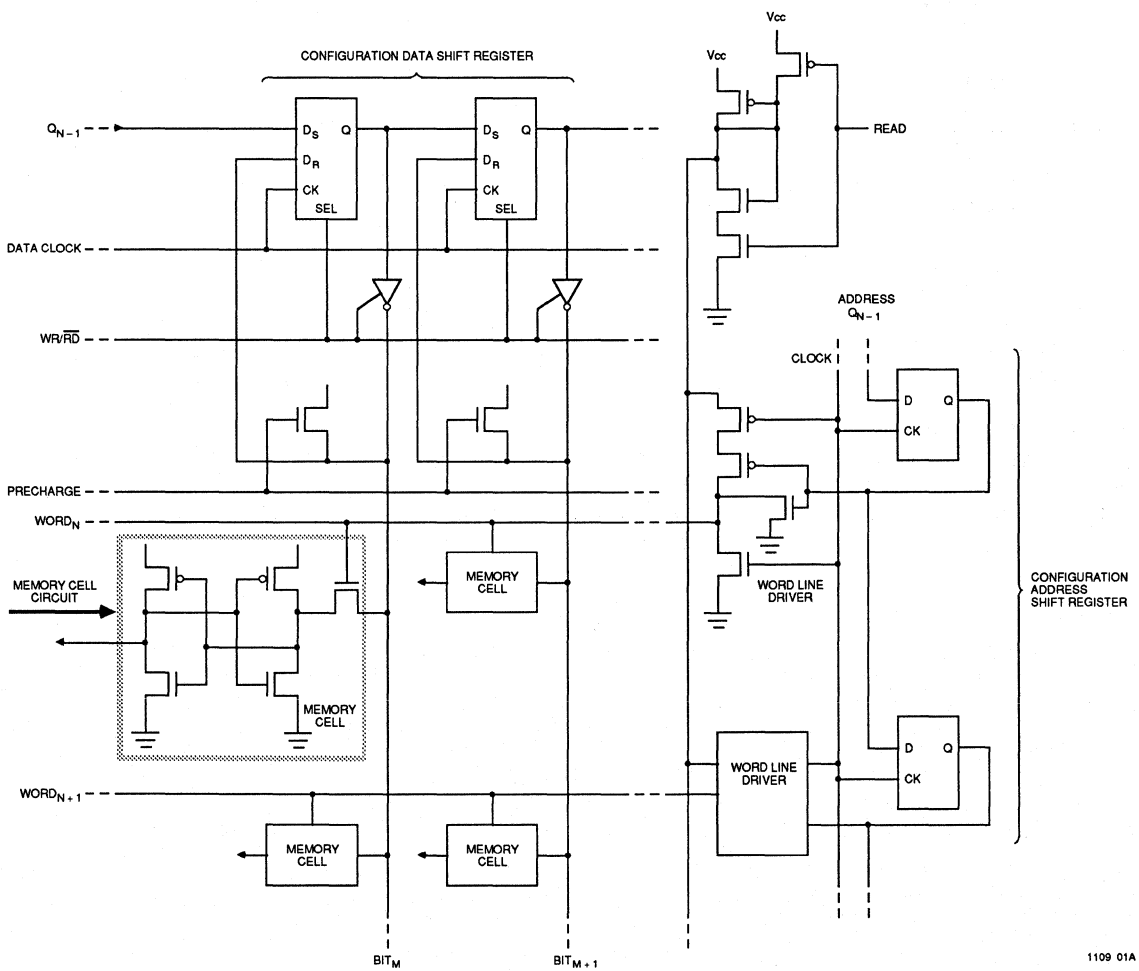


Figure 1. Configuration Memory Cell

guaranteed within the tolerances of the manufacturing process.

In the Read mode, the bit line, which has a significant amount of parasitic capacitance, is precharged to a logic one. The pass transistor is then enabled by driving the word line High. If the stored value is a zero, the line is then discharged to ground. Reliable reading of the memory cell is achieved by reducing the word line High level during reading to a level that insures that the cell will not be disturbed.

Alpha Particle (Soft Error) Sensitivity

The CMOS static memory cell was designed to be insensitive to alpha particle emissions. To verify that this design goal was achieved, the following tests were performed.

A one-microcurie alpha-particle source (Americum 241) was placed in direct contact with the top surface of an XC2064 die. This allows the die to capture at least 40% of the emissions from the radiation source. The following sequence of tests was performed:

1. A complex pattern containing roughly 50% logic ones was loaded into the XC2064. The operating conditions were 25°C and 5.0 V.
2. A pause of variable duration was permitted.
3. The entire contents of the XC2064 were read back and compared with the original data.

Validation tests to ensure that the test setup would detect errors were performed before and after the alpha-particle tests. The results are as follows:

Test	Time Duration	Readback Time	Total Time Exposed	Number of Errors
1	10 s	70 s	80 s	0
2	120 s	70 s	190 s	0
3	300 s	70 s	370 s	0
4	1500 s	70 s	1570 s	0
Total			2210 s (0.61 hours)	

Analysis

A one-microcurie source emits 3.7×10^4 alpha particles per second. Assuming that 40% of these are captured by the XC2064 during this experiment, this corresponds to 5.3×10^7 alpha particles per hour.

The alpha-particle emission rate of the molding compound used by Xilinx is specified to emit fewer than 0.003 alpha particles per square centimeter per hour (alpha particles/cm²/hr). The surface area of the XC-2064 die is less than 0.5 cm², so less than 0.0015 alpha particles per hour will be captured by the XC2064 in normal operation. The error rate acceleration in this test is therefore equal to:

$$\frac{5.3 \times 10^7 \text{ particles/hour}}{0.0015 \text{ particles/hour}} = 3.6 \times 10^{10}$$

The 0.61 hours of error-free test time thus is equivalent to 2.2×10^{10} hours or 2.5 million years of error-free operation.

Most ceramic packages are specified to emit less than 0.01 alpha particles/cm²/hr which is about three times more than the plastic compound. For an XC2064 in a ceramic package, this still results in error-free operation for almost a million years.
The highest rate of alpha-particle emission comes from the sealing glass used in cerdip packages and some ceramic packages (frit lids). For instance, KCIM glass emits about 24 alpha particles/cm²/hr. Low-alpha glasses are specified at 0.8 alpha particles/cm²/hr.

Because these glasses are used only for the package seal, they present a relatively small emitting cross section to the die (less than 0.1 cm²). A low-alpha glass would therefore cause fewer than 0.8 alpha particle hits per hour. The acceleration factor is then 6.6×10^8, which translates to about 46,000 years without an error.

The LCA memory cell has been designed so that soft errors caused by alpha particles can safely be ignored.

ELECTROSTATIC DISCHARGE

Electrostatic-discharge (ESD) protection for each pad is provided by a circuit that uses forward and reverse-biased distributed resistor-diodes (Figure 2). In addition, inherent capacitance integrates any current spikes. This give sufficient time for the diode and breakdown protections to provide a low-impedance path to the power-supply rail. Geometries and doping levels are optimized to provide sufficient ESD protection for both positive and negative discharge pulses.

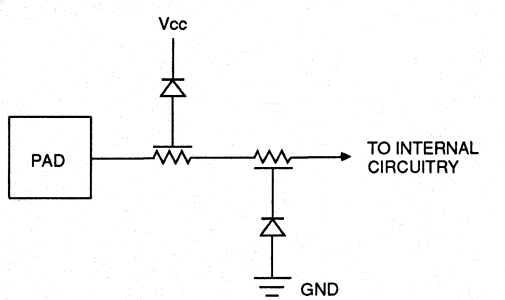


Figure 2. Input Protection Circuitry

LATCHUP

Latchup is a condition in which parasitic bipolar transistors form a positive feedback loop (Figure 3), which quickly reaches current levels that permanently damage the device. Xilinx uses techniques based on doping levels and circuit placement to avoid this phenomenon. The cross section of a typical transistor (Figure 4) shows several features. The beta of each parasitic transistor is minimized by increasing the base width. This is achieved with large physical spacings. The butting contacts effectively short the n+ and p+ regions for both wells, which makes the V_{BE} of each parasitic very close to zero. This also makes the parasitic transistors very hard to forward bias. Finally, each well is surrounded by a dummy collector, which forces the V_{CE} of each parasitic almost to zero and creates a structure in which the base width of each parasitic is large, thus making latchup extremely difficult to induce.

At elevated temperatures, 100 mA will ***not*** cause latchup. At room temperature, the device can withstand more than 300 mA without latchup. However, ***continuous*** currents in excess of 10 mA are not recommended.

HIGH TEMPERATURE PERFORMANCE

Although Xilinx guarantees parts to perform only within the specifications of the data sheet, extensive high temperature life testing has been been done at 145°C with excellent results. In plastic packages, the maximum junction temperature is 125°C.

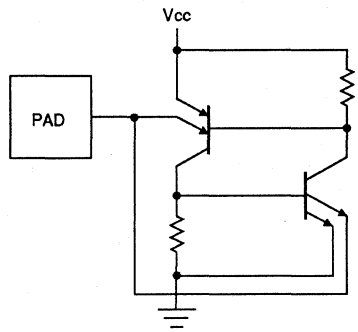


Figure 3. SCR Model

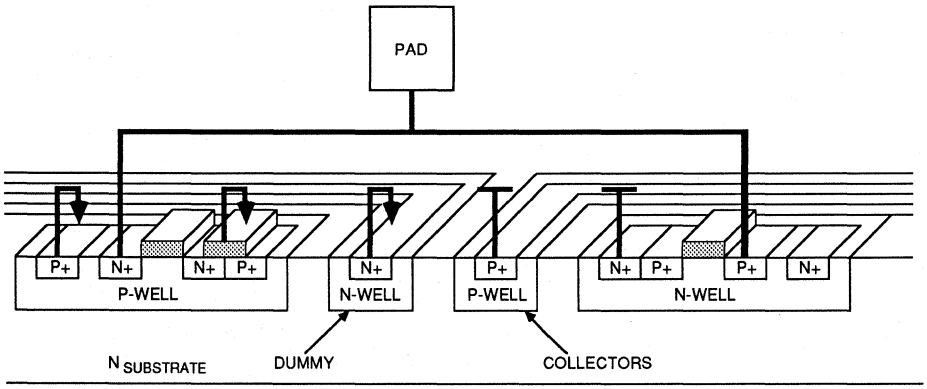


Figure 4. CMOS Input Circuit Layout

RADIATION HARDNESS—GAMMA TOTAL DOSE TEST

Outline of Testing

Xilinx has conducted a series of radiation hardness tests to demonstrate the capability of the 1.2-micron CMOS process being used to produce both the XC2000 and XC3000 family of devices. The test vehicle used was the XC3020-70PC84C. Other devices in other packages are expected to show similar results.

Four random devices taken from normal production lots were individually subjected to gamma radiation levels under 5V Vcc bias in power-down mode. The four radiation levels were 20, 30, 50 and 75 krads (Si). Each device was exposed to a different radiation level. The radiation was applied at the rate of 10 rads (Si) per second. After exposure each device was functionally tested in a system at the irradiation site. A device failure is defined as inability to load configuration data and begin functional operation. The devices were also tested at Xilinx after irradiation. Standard dc functionality and ac parametric tests were performed on the Xilinx production tester. Both the system and the device tests showed device failure at the 75 krad (Si) level. Test results are summarized in Table 1.

Table 1

Device Number	Gamma Exposure Level	System Functional Test	Tester Results
1	20 krad (Si)	Passed	Passed
2	30 krad (Si)	Passed	Passed
3	50 krad (Si)	Passed	Passed
4	75 krad (Si)	Failed*	Failed*

* Would not reconfigure

Total Dose Level: 50 krad (Si)

Testing closely followed the procedures of MIL-STD-883C, Method 1019, but was not strictly in full compliance with the requirements since the irradiation and final test sites were in different locations. The devices were transported by air (with bias continually applied) from the location in southern California, where they had been exposed to the radiation, to the Xilinx test facility in San Jose. Up to two hours elapsed, from the time of exposure to final testing, due to air transportation.

In summary, four devices were tested following exposure to a Cobalt-60 gamma-radiation source. The data obtained from these tests indicate that devices are capable of normal operation up to the 50 krad (Si) total dose level. The test results suggest that Xilinx XC2000 and XC3000 devices manufactured on the 1.2-micron CMOS process are acceptable in any application requiring a radiation design level of 50 krad (Si) or less.

Xilinx personnel did not verify the total radiation dosage applied the calibration of the Cobalt-60 source, or the application of bias during irradiation. Because of these factors, coupled with the two-hour test-time lag and the small sample size, these tests are not fully conclusive.

Test Methodology

Xilinx is committed to providing the highest level of quality and reliability for the Logic Cell (LCA) Array. Quality is best assured by taking the necessary steps to achieve zero defects. Comprehensive testing confirms that every LCA device is free from defects and conforms to the data sheet specifications. The memory-cell design assures integrity of the configuration program.

TESTING

As quality consciousness has grown among semiconductor users, awareness of the importance of testability has also increased. Testing for standard components, including memories and microprocessors, is accomplished with carefully developed programs which exhaustively test the function and performance of each part. For reasons explained below, most application specific ICs cannot be comprehensively tested. Without complete testing, defective devices might escape detection and be installed into a system. In the best case, the failure will be detected during system testing at a higher cost. In the worst case, the failure will be detected only after shipment of the system to a customer.

Testing advantages of the Logic Cell Array can be illustrated through comparison with two other application specific ICs: Erasable Programmable Logic Devices (EPLDs) and gate arrays.

EPLDs: In order to test all memory cells and logic paths of programmable logic devices controlled by EPROM memory cells, the part must be programmed with many different patterns. This in turn requires expensive quartz lid packages and many lengthy program/test/erase cycles. To save time and reduce costs, this process is typically abbreviated.

Gate Arrays: Since each part is programmed with metal masks, the part can only be tested with a program tailored to the specific design. This in turn requires that the designer provide sufficient controllability and observability for comprehensive testability. The design schedule must also include time for the development of test vectors and a test program specification. If the gate array user requires a comprehensive test program, then he must perform exhaustive and extensive fault simulation and test grading. This requires substantial amounts of expensive computer time. Additionally, it typically requires a series of time-consuming and expensive iterations in order to reach even 80% fault coverage. The cost of greater coverage is often prohibitive. In production, many gate array vendors either limit the number of vectors allowed or charge for using additional vectors.

The replacement of all storage elements with testable storage elements, known as scan cells, improves testability. Although this technique can reduce the production testing costs, it can add about 30% more circuitry, decrease performance by up to 20%, and increase design time.

Logic Cell Arrays: The testability of the LCA device is similar to other standard products, including micro-processors and memories. This is the result of the design and the test strategies:

Design strategy:

- Incorporates testability features because each functional node can be configured and routed to outside pads
- Permits repeated exercise of the part without removing it from the tester because of the short time to load a new configuration program
- Produces a standard product which guarantees that every valid configuration will work.

Test strategy:

- Performs Reads and Writes of all bits in the configuration memory, as in memory testing
- Uses an efficient parallel testing scheme in which multiple configurable logic blocks are fully tested simultaneously
- Is exhaustive since the circuits in every block are identical

The Logic Cell Array user can better appreciate the LCA test procedure by examining each of the testing requirements:

- All configuration-memory bits must be exercised and then verified. This is performed using readback mode.
- All possible process-related faults, such as short circuits, must be detected. The Logic Cell Array is configured such that every metal line can be driven and observed directly from the input/output pads.

- All testing configurations must provide good controllability and observability. This is possible since all configurable logic blocks can be connected to input/output pads. This makes them easy to control by testing different combinations of inputs and easy to observe by comparing the actual outputs with expected values.

These points bring out an important issue: the Logic Cell Array was carefully designed to achieve 100% fault coverage. With the Xilinx testing strategy, the number of design configurations needed to fully test the Logic Cell Array is minimized and the test fault coverage of the test patterns is maximized. In addition, the user's design time is reduced because the designer does not have to be concerned about testability requirements during the design cycle. The LCA concept not only removes the burden of the test-program and test-vector generation from the user, but also removes the question of fault coverage and eliminates the need for fault grading. The Logic Cell Array is a standard part that guarantees any valid design will work. These issues are critically important in quality-sensitive applications. The designer who uses the Logic Cell Array can build significant added value into his design by providing higher quality levels.

TESTING OF THE LOGIC CELL ARRAY

The LCA device is tested as a standard product. Every device is tested for: 1) 100% functionality; 2) dc parametrics; and 3) speed. This allows the end-user to design and use the logic cell array without worrying about testing for a particular application.

The strategy for testing the LCA device is to test the functionality of every internal element. These elements consist of memory cells, metal interconnects, transistor switches, bidirectional buffers, inverters, decoders, and multiplexers. If each element is functional, then the user's design will also be functional if the proper design procedures are used.

The static memory cells and the symmetry of the Logic Cell Array make it 100% testable. The Logic Cell Array can be programmed and reprogrammed with as many patterns as required to fully test it. This is done with as many as 50 configuration/test patterns. Each configuration/te pattern consist of: 1), A set of test vectors that configure the LCA device with a hardware design that utilizes specific elements; and 2), A set of test vectors that exercise those specific elements. The symmetry of the LCA device allows the test engineer to develop the test for one CLB or IOB and then apply it to all others. All configuration/test patterns are exercised at both Vcc minimum and maximum.

Memory Cell Testing

The static memory cells have been designed specifically for high reliability and noise immunity. The basic memory cell consists of two CMOS inverters and a pass transistor used for both writing and reading the memory cell data (See Figure 1). The cell is only written during configuration. Writing is accomplished by raising the gate of the pass transistor to Vcc and forcing the two CMOS inverters to conform to the data on the word line. During normal operation, the memory-cell provides continuous control of the logic, and the pass transistor is "off" and does not affect memory-cell stability. The output capacitive load and the CMOS levels of the inverters provide high stability. The memory cells are not affected by extreme power-supply excursions.

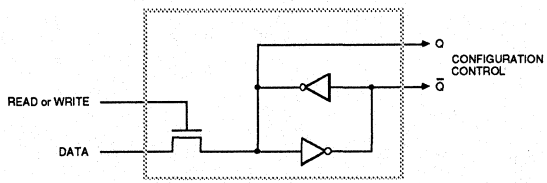


Figure 1. Configuration Memory Cell

The memory cells are directly tested in the Logic Cell Array with three test patterns that are equivalent to those used on a RAM device. The first test pattern writes 95% of all the RAM cells to a logic zero and then reads each RAM cell back to verify its contents. The second test pattern writes 95% of all the RAM cells to a logic one and also verifies the contents. The third pattern is used to verify that all I/O and configurable logic blocks can have their logic value read back correctly. All RAM cells are thus written and verified for both logic levels.

Interconnect Testing

The programmable interconnect is implemented using transistor switches to route signals through a fixed two layer grid of metal conductors. The transistor switches "on" or "off" depending on the logic value of the static memory cell that controls the switch. The interconnect is tested with configuration/test patterns that: 1) Test for continuity of each metal segment; 2) Test for shorts between metal segments; and 3) Check the ability for each switch to connect two metal lines. This can be accomplished with a pattern similar to Figure 2. Each interconnect line will be set to a logic one while the others are set to logic zero. This checks for shorts between adjacent interconnects while at the same time checking for continuity of the line.

Print World: PATN11.LCA (2018PC84-70), XACT 2.05b Eng, Wed Mar 02 16:28:18 198{Print World: PATN11.LCA (2018PC84-7

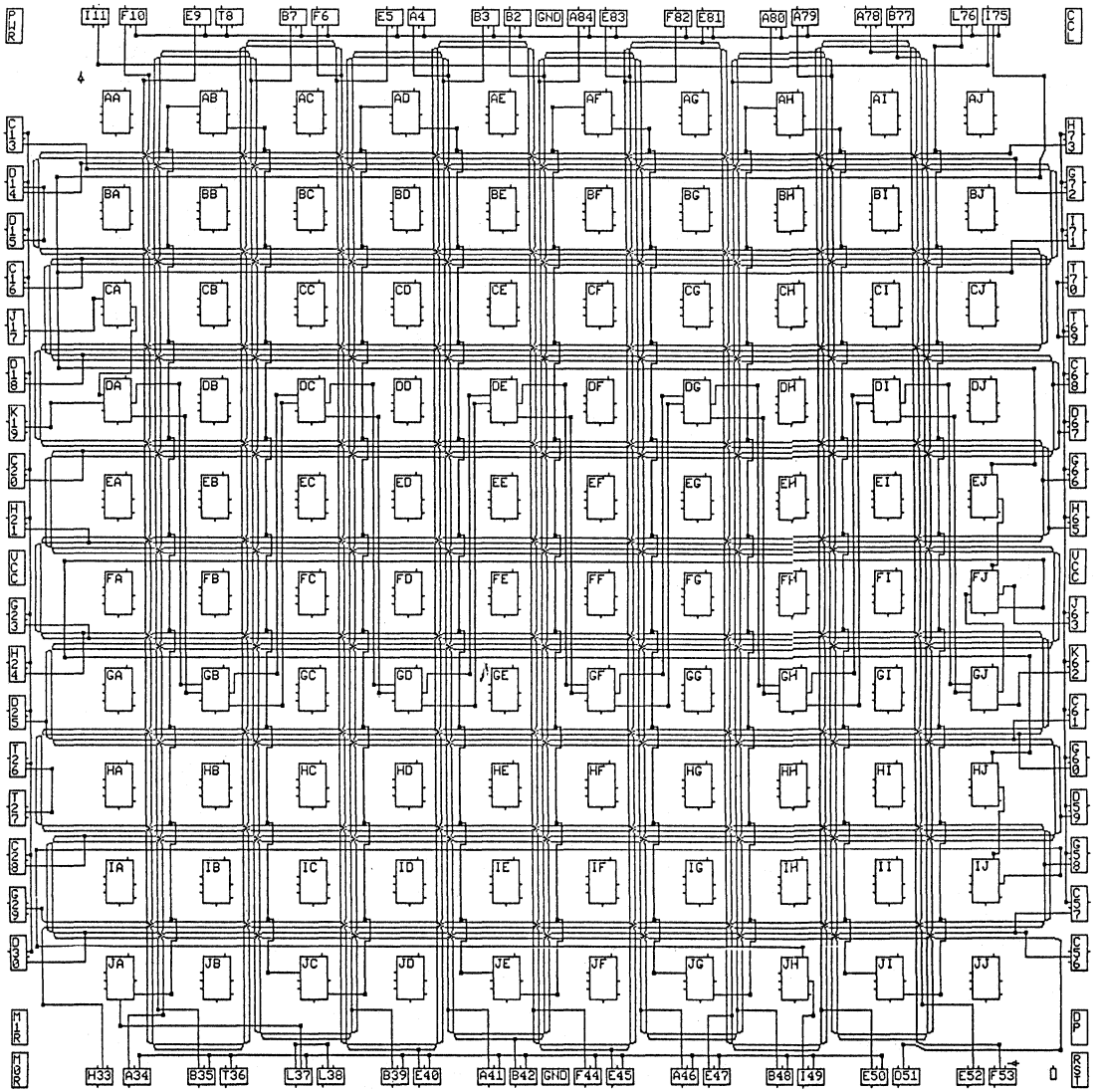


Figure 2. Interconnect Test Pattern

I/O Block Testing

Each I/O block includes registered and direct input paths and a programmable 3-state output buffer. The testing of these functions is accomplished by several configuration/ test patterns that implement and test each option that is available to the user. One method used to test the I/O blocks is to configure them as a shift register that has a 3-state control (See Figure 3). This allows a test pattern to check the ability of each I/O block to latch and to output data that is derived from either the previous I/O block or from the tester. Several of these patterns are used to exercise different input and output combinations allowed for each I/O block. Configuration/test patterns are also used to precondition the device to test dc parameters such as V_{IH}, V_{IL}, V_{OH}, V_{OL}, TTL standby current, CMOS standby current and input/output leakage. The V_{OH}/V_{OL} Test is done while all outputs are either all Low or all High.

Configurable Logic Block Testing

Each configurable logic block has a combinatorial-logic section, a flip-flop section, and an internal-control section. The combinatorial-logic section of the logic block uses an array of RAM cells (16x1 in or 32X1 in) as a look-up table to implement the Boolean functions. This section is tested as an array of memory cells. Configuration/test patterns are used to verify that each RAM cell can be logically decoded as the output of the array. The flip-flop section of the logic block is tested with configuration/test patterns that configure the LCA device as shift registers. Each shift register pattern will have different data in the look-up tables and will have a different pin used as the input to each shift register. Other configuration/test patterns are used to implement and test the internal-control section.

TESTING THE SPEED OF THE LOGIC CELL ARRAY

LCA speed is checked with configuration/test patterns that have been correlated to data sheet ac values.

Most of these patterns are shift registers with interconnect, IOBs and CLBs in the data path (See Figure 4). They are designed with the idea that all elements in the path must be fast enough for the proper data to get to the next input of the shift register before the next clock occurs. If any element doesn't meet the specified ac value, then the shift register will clock in the wrong data and fail the test. The complexity of the logic between two shift register cells determines the maximum frequency required for the clock pulse input of the shift register. This can be used to reduce the performance requirement of the tester in use. The patterns used consist of a TCKO + TILO + INTERCONNECT + TICK for each shift register. This increases the shift register clock pulse separation time to 30 to 40 ns. The configuration of each pattern is varied so that all of the interconnect, IOBs, and CLBs are tested at speed.

HARDWARE TESTING CONSIDERATIONS FOR THE LCA DEVICE

Currently the Logic Cell Array is being tested on Sentry testers. The 68 and 84 pin versions can be tested on a 60-pin tester with 256K of extended local memory. The 3000 series products are being tested on a 120-pin Sentry Series 21 tester with 1 million vectors required for 3042 - 3090, 512K vectors required for 3020 - 3030 and multiple PMU measurement systems.

Print World: patn01.lca (2018PC84-70), XACT 2.05b Eng, Wed Mar 02 16:02:02 1988 Print World: patn01.lca (2018PC84

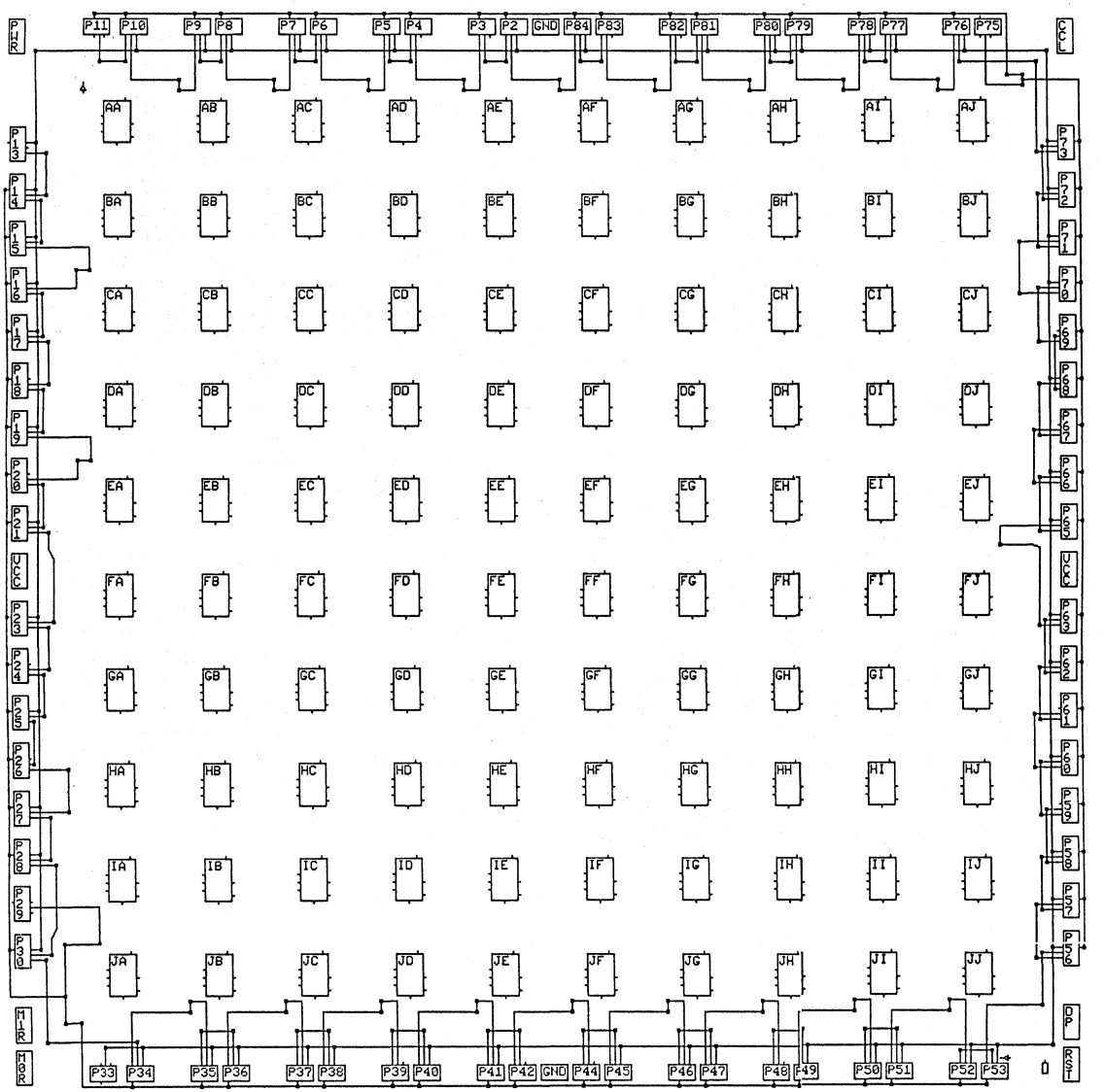


Figure 3. IOB Test Pattern

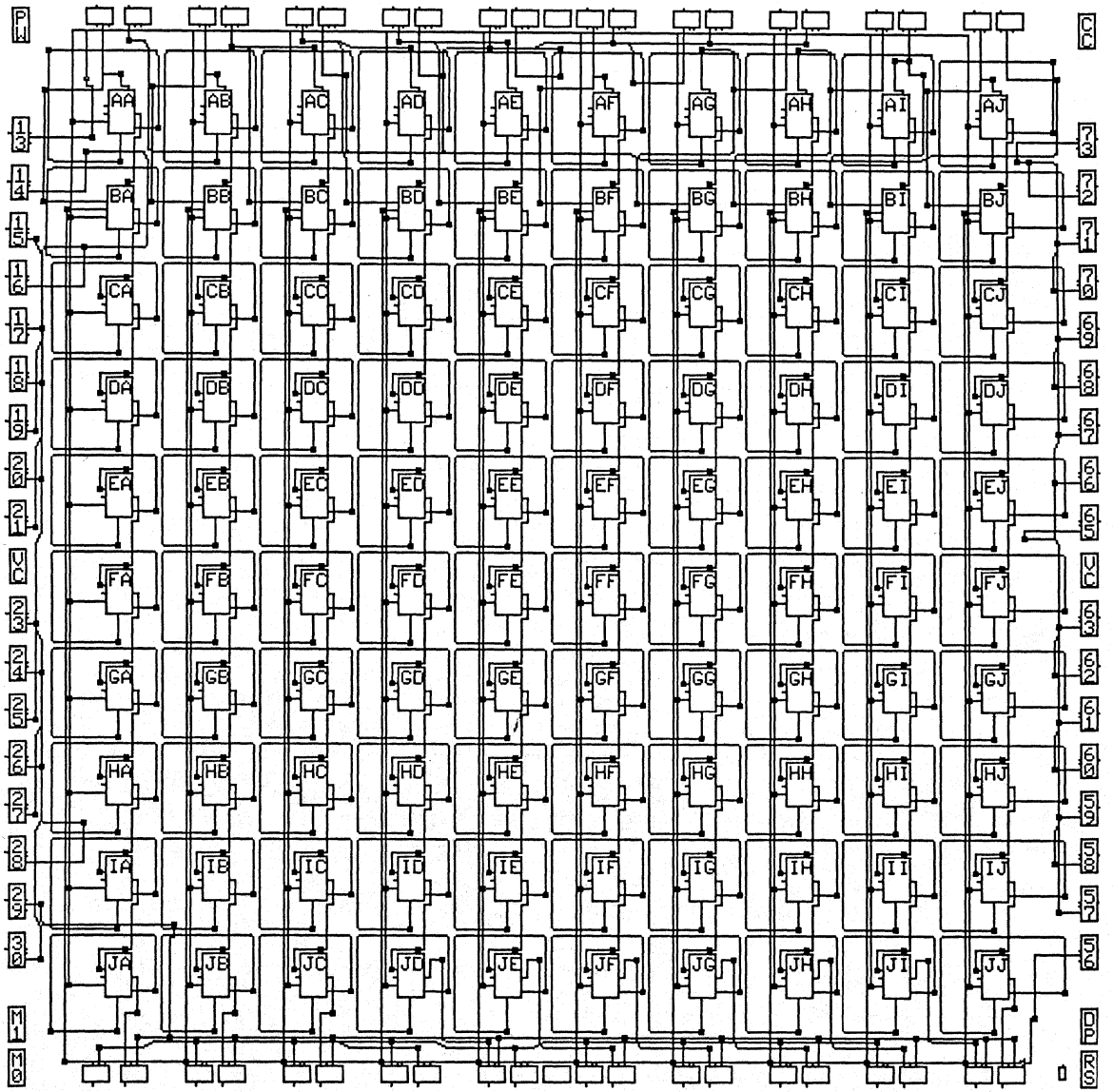


Figure 4. Speed Test Pattern

PACKAGE AND USER I/O AVAILABILITY

Number of User I/O Available

	48 PIN	68 PIN	84 PIN	100 PIN	132 PIN	160 PIN	164 PIN	175 PIN
XC2064	40	58						
XC2018		58	74					
XC3020		58	64	64				
XC3030			74	80				
XC3042			74	82	96			
XC3064					110			
XC3090						135	142	144

1112 02B

PACKAGE/SPEED/TEMPERATURE SELECTIONS

		44 PIN	48 PIN		68 PIN		84 PIN		100 PIN		132 PIN		160 PIN	164 PIN	175 PIN	
		PLASTIC PLCC	PLASTIC DIP	CERAMIC DIP	PLASTIC PLCC	CERAMIC PGA	PLASTIC PCC	CERAMIC PGA	PLASTIC PQFP	CERAMIC CQFP	PLASTIC PGA	CERAMIC PGA	PLASTIC PQFP	CERAMIC CQFP	PLASTIC PGA	CERAMIC PGA
		PC44	PD48	CD48	PC68	PG68	PC84	PG84	PQ100	CQ100	PP132	PG132	PQ160	CQ164	PP175	PG175
XC2064	-50		C	I	CI	CIM										
	-70				CI	CIM										
	-100				C	C										
XC2018	-33							MB								
	-50				CI		CI	CIMB								
	-70				CI		CI	CIMB								
	-100				C		C	C								
XC3020	-50				CI		CI	CIMB	CI	CIMB						
	-70				CI		CI	CIMB	CI	CIMB						
	-100				C		C	C	C	C						
XC3030	-50	CI			CI		CI	CIM	C							
	-70	CI			CI		CI	CIM	CI							
	-100	C			C		C	C	C							
XC3042	-50						CI	CIMB	C	CIMB	CI	CIMB				
	-70						CI	CIMB	CI	CIMB	CI	CI				
	-100						C	C	C	C	C	C				
XC3064	-50						C				CI	CIM				
	-70						C				CI	CIM				
	-100						C				C	C				
XC3090	-50						C						CI*	CIMB	CI	CIMB
	-70						C						CI*	CIMB	CI	CIMB
	-100						C						C*	C	C	C

X1104

PACKAGE THERMAL CHARACTERIZATION

Method and Calibration

Xilinx uses the indirect electrical method for thermal-resistance characterization of packages. The forward-voltage drop of an isolated diode residing on a special test die is calibrated at a constant forcing current of 0.520 mA with respect to temperature over a correlation temperature range of 22°C to 125°C. The calibrated device is then mounted in an appropriate environment, e.g. still air, forced convection, FC-40, etc. Power (Pd) is applied to the device through diffused resistors on the same thermal die; usually between 0.5 to 4 W is applied, depending on the package. The resulting rise in junction temperature (T_J) is monitored with the forward-voltage drop of the pre-calibrated diode. Typically, three identical samples are tested at each data point. The reproducibility error is close to 6%.

Junction-to-Case Measurement – θ_{JC}

The junction-to-case characterization is measured in a 3M Flourinert (FC-40) isothermal circulating fluid stabilized at 25°C. During the measurement, the Device Under Test (DUT) is completely immersed in the fluid; initial stable conditions are recorded, then Pd is applied. Case temperature (T_C) is measured at the primary heat-flow path of the particular package. Junction temperature (T_J) is calculated from the diode forward-voltage drop from the initial condition before power is applied, i.e.

$$\theta_{JC} = \frac{T_J - T_C}{Pd}$$

The junction-to-isothermal-fluid measurement θ_{JL} can also be calculated from the above data as follows:

$$\theta_{JC} = \frac{T_J - T_L}{Pd}$$

where T_L = isothermal fluid temperature.

The latter data is considered as the ideal θ_{JA} data for the package that can be obtained with the most efficient heat removal scheme–airflow, copper-clad board, heat sink or some combination of these. Since this is not a widely used parameter in the industry, and it is not very realistic for normal application of Xilinx packages, the data are not published. The thermal lab keeps such data for package comparisons.

Junction-to-Ambient Measurement – θ_{JA}

θ_{JA} is measured on a 4.5" x 6.0" x .0625" (11.4 cm x 15.2 cm x 0.16 cm) FR-4 board. The data may be taken with the package in a socket or, for packages used primarily for surface mount, with the package mounted directly on traces on the FR-4 board. The copper-trace density is limited to the pads needed for the leads and the 10 or so traces required for signal conditioning and measurement. The board is mounted in a cylindrical enclosure and data is taken at the prevailing temperature and pressure–between 22°C and 25°C ambient (T_A). The power application and signal monitoring proceed in the same way as the θ_{JC} measurement with enclosure (ambient) thermocouple substituted for the fluid thermocouple and two extra thermocouples brought in to monitor room and board temperatures. The junction-to-ambient thermal resistance is calculated as follows:

$$\theta_{JA} = \frac{T_J - T_A}{Pd}$$

The setup lends itself to the application of various airflow velocities from 0 - 800 Linear Feet per Minute (LFM), i.e., 0 - 4.06 m/s. Since the board selection (copper trace density, mounting distance, board thermal conductivity etc) affects the results of the thermal resistance, the data from these tests shall always be qualified with the board-mounting information.

Data Acquisition and Package Thermal Database

Data for a package type is gathered for various die sizes, power levels, cooling modes (airflow and sometimes heat-sink effects) with an IBM-PC based Data Acquisition and Control System (DAS). The system controls and conditions the the power supplies and other ancillary equipment for a hands-free data taking. Different custom-tailored setups within the DAS software are used to run calibration, θ_{JA}, θ_{JC}, fan test as well as power-effects characteristics of a package. A package is completely characterized with respect to the major variables that influence the thermal resistance. A database is generated for the package. From the database, thermal resistance data is interpolated as typical values for the individual Xilinx devices that are assembled in the characterized package. (See data in following table.)

Thermal Resistance Data

Product and Package (Socketed unless noted)	θ_{JA}°C/W	θ_{JC}°C/W
XC1736A CD8	112.0	7.1
XC1765 CD8	108.4	6.4
XC1765 PC20	79.0	17.4
XC3030 PC44	43.6	10.7
XC2018 PC68	40.7	8.1
XC2318 PC68	42.4	9.9
XC2364 PC68	42.4	9.9
XC2064 PC68	42.1	9.5
XC3020 PC68	40.9	8.3
XC3030 PC68	39.4	7.0
XC2064 PC84	36.7	7.9
XC2018 PC84	35.1	6.7
XC3020 PC84	35.3	6.9
XC3030 PC84	33.7	5.7
XC3042 PC84	32.3	4.8
XC3064 PC84	30.5	3.6
XC3090 PC84	29.1	2.8
XC2064 PG84	36.7	6.7
XC2018 PG84	35.1	6.0
XC2020 PG84	35.4	6.1
XC3030 PG84	33.7	5.4
XC3042 PG84	32.3	4.9
XC3064 PG84	30.5	4.4
XC3020 PQ100*	67.6	9.3
XC3030 PQ100*	62.7	6.8
XC3042 PQ100*	58.2	5.0
XC3020 PQ100	75.3	–
XC3030 PQ100	71.1	–
XC3042 PQ100	68.1	–
XC3042 PG132	26.5	2.5
XC3064 PG132	24.1	2.0
XC3090 CQ164*	32.9	1.5
XC3090 PG175	16.4	0.9
XC3090 PP175	21.7	1.6

*Surface mounted

PACKAGE CHARACTERISTICS

Component Average Mass by Package Type and Lead Count

Package Type	Lead Count	Mass (Grams)	Comment
PLCC	20	0.75	
PLCC	44	1.20	
PLCC	68	4.80	
PLCC	84	6.80	
PDIP	8	0.52	300 mil
PDIP	48	7.90	600 mil
Side Braze	8	0.95	
Side Braze	48	8.00	
PQFP	100	1.6	14 mm x 10 mm
PQFP	160	5.80	28 mm x 28 mm
CQFP	100	3.60	Unformed
CQFP	164	8.35	Unformed
CPGA	68	6.95	11 x 11 Array
CPGA	84	7.25	11 x 11 Array
CPGA	132	11.75	14 x 14 Array
CPGA	175	28.40	16 x 16 KCW10 H/S
PPGA	132	8.10	14 x 14 Cu Slug
PPGA	175	10.60	16 x 16 Cu Slug

*Data represents average values for typical packages with typical devices. For accuracy between 7% to 10%, these numbers will be adequate.

*More precise numbers (below 5% accuracy) for specific devices may be obtained from Xilinx through a factory representative.

Ceramic Quad Flat Pack (CQFP)

The Ceramic Quad Flat Pack (also called Quad Cerpack) is a cavity down, pressed ceramic package. The leads are gull-wing, on four sides, with 25-mil pitch. It is for surface mount Commercial, Industrial, and Military (including MIL-STD-883 Class B) applications. JEDEC has developed a standard that Xilinx will follow.

Plastic Quad Flat Pack (PQFP)

The Plastic Quad Flat Pack is an EIAJ standard package. The leads are gull-wing on four sides. It is for surface mount Commercial applications.

For more information on SMT

The following organizations provide SMT consulting and training, component part lists, and related services:

D Brown Associates
(Surface Mounting Directory
and SMT: How to Get Started)
Box 43
Warrington, PA 18976
(215) 343-0123

Electronics Manufacturing Productivity Facility (EMPF)
1417 North Norma Street
Ridgecrest, CA 93555-2510
(619) 446-7706

International Quality Technologies, Inc.
4300 Stevens Creek Blvd
Suite 203
San Jose, CA 95129
(408) 246-6071

National Training Center
Northhampton Area College
3835 Greenpond Rd
Bethlehem, PA 18017
(215) 861-5486

Surface Mount Council
C/O IPC
7380 N Lincoln Ave
Lincolnwood, Il 60646

SOCKETS

Below are two lists of manufactures known to offer sockets for Xilinx package types. This list does not imply an endorsement by Xilinx. Each user must evaluate the particular socket type.

There are no wire-wrap sockets for PLCCs. One solution is to piggy-back a through-hole PLCC socket mounted in a compatible PGA socket with wire-wrap pins. Note that the board-layout then differs from a PGA board layout.

Zero Insertion Force (ZIF) sockets, recommended for prototyping with 132 and 175 pin PGA devices, also lack the wire-wrap option. Piggy-back the ZIF socket in a normal PGA wire-wrap socket.

PLCC Sockets

AMP Inc.
Harrisburg, PA 17105
(717) 564-0100

Burndy Corp.
Richards Ave.
Norwalk, CT 06856
(203) 852-8437

Garry Electronics
9 Queen Anne Court
Langhorne, PA 19047-1803
(215) 949-2300

Honda - MHOtronics
Deerfield. IL 60015
444 Lake Cook Road, Suite 8
(312) 948-5600

ITT Cannon
10550 Talbert Ave.
P.O.Box 8040
Fountain Valley, CA 92728
(714) 964-7400

Kycon Cable & Connector
1772 Little Orchard Street
San Jose, CA 95125
(408) 295-1110

Maxconn Inc.
1855 O'Toole Ave., D102
San Jose, CA 95131
(408) 435-8666

Methode Electronics Inc.
1700 Hicks Road
Rolling Meadows, IL 47150
(312) 392-3500

Mill-Max Mfg. Corp.
190 Pine Hollow Road
Oyster Bay, N.Y. 11771-0300
(516) 922-6000

Precicontact Inc.
835 Wheeler Way
Langhorne, PA 19047
(215) 757-1202

Robinson Nugent
800 East Eighth Street
New Albany, IN 47150
(812) 945-0211

Samtec Inc.
P.O.Box 1147
New Albany, IN 47150
(812) 944-6733

3M Textool
Austin, TX
(800) 328-7732

Thomas & Betts Corp.
920 Route 202
Raritan, NJ 08869
(201) 469-4000

Wells Electronics, Inc.
1701 South Main Street
South Bend, IN 46613
(219) 287-5941

Yamaichi - Electronics, Inc.
1420 Koll Circle
Suite B
San Jose, CA 95112
(408) 452-0799

PGA Sockets

Advanced Interconnections
5 Energy Way
West Warwick, RI 02893
(401) 823-5200

AMP Inc.
Harrisburg, PA 17105
(717) 564-0100

Aries Electronics, Inc.
P.O.Box 130
Frenchtown, NJ 08825
(201) 996-6841

Augat
33 Perry Ave.
P.O.Box 779
Attleboro, MA 02703
(617) 222-2202

Bevmar Industries, Inc.
20601 Annalee Ave.
Carson, CA 90746
(213) 631-5152

Bevmar Industries, Inc.
1 John Clarke Rd.
Middletown, RI 02840
(401) 849-4803

Electronic Molding Corp.
96 Mill Street
Woonsocket, RI 02895
(401) 769-3800

Garry Electronics
9 Queen Anne Court
Langhorne, PA 19047-1803
(215) 949-2300

Mark Eyelet Inc.
63 Wakelee Road
Wolcott, CT 06716
(203) 756-8847

McKenzie Technology
44370 Old Warm Springs Blvd.
Fremont CA 94538
(415) 651-2700

Methode Electronics Inc.
1700 Hicks Road
Rolling Meadows, IL 47150
(312) 392-3500

Mill-Max Mfg. Corp.
190 Pine Hollow Road
Oyster Bay, N.Y. 11771-0300
(516) 922-6000

Precicontact Inc.
835 Wheeler Way
Langhorne, PA 19047
(215) 757-1202

Robinson Nugent
800 East Eighth Street
New Albany, IN 47150
(812) 945-0211

Samtec Inc.
P.O.Box 1147
New Albany, IN 47150
(812) 944-6733

Texas Instruments
CSD Marketing, MS 14-1
Attleboro, MA 02703
(617) 699-5206

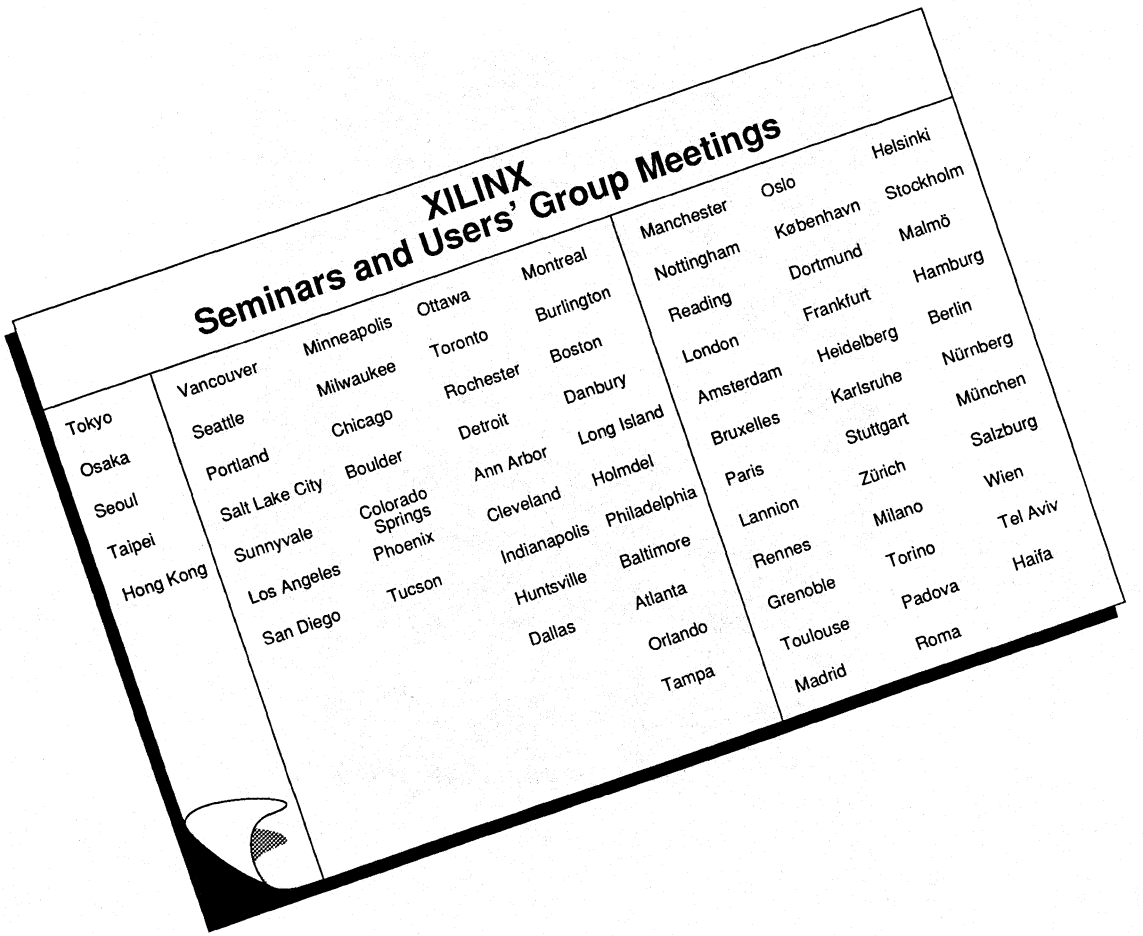
Thomas & Betts Corp.
920 Route 202
Raritan, NJ 08869
(201) 469-4000

Yamaichi - Electronics, Inc.
1420 Koll Circle
Suite B
San Jose, CA 95112
(408) 452-0799

Technical Support

Beyond the technical data in this book, Xilinx provides a wealth of additional technical information to LCA users. The following pages give an overview of the existing material, beginning with Technical Seminars and ending with detailed Technical Manuals.

Technical Seminars and Users' Group Meetings



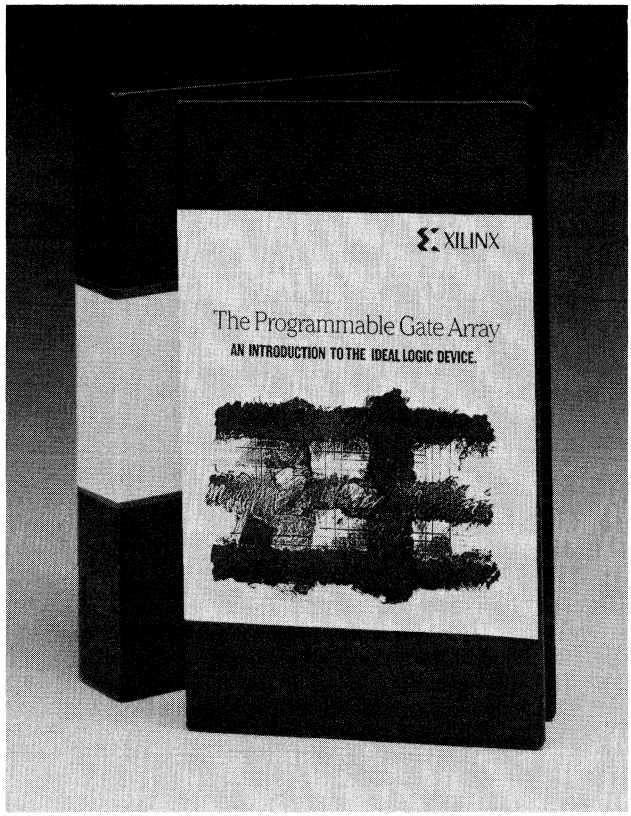
Xilinx sponsors technical seminars at locations throughout North America, Europe, and Asia.

Product-oriented seminars are directed toward new and potential users of Field Programmable Gate Arrays. These seminars include a basic description of the Logic Cell Array architecture and the benefits of this technology. Experienced users will also find these seminars useful for learning about newly released products from Xilinx.

Users' Group meetings are intended for experienced users of Xilinx Field Programmable Gate Arrays, and emphasize the use of the various development system tools to generate LCA-based designs.

Contact your local Xilinx sales office, sales representative, or distributor for information about seminars in your area.

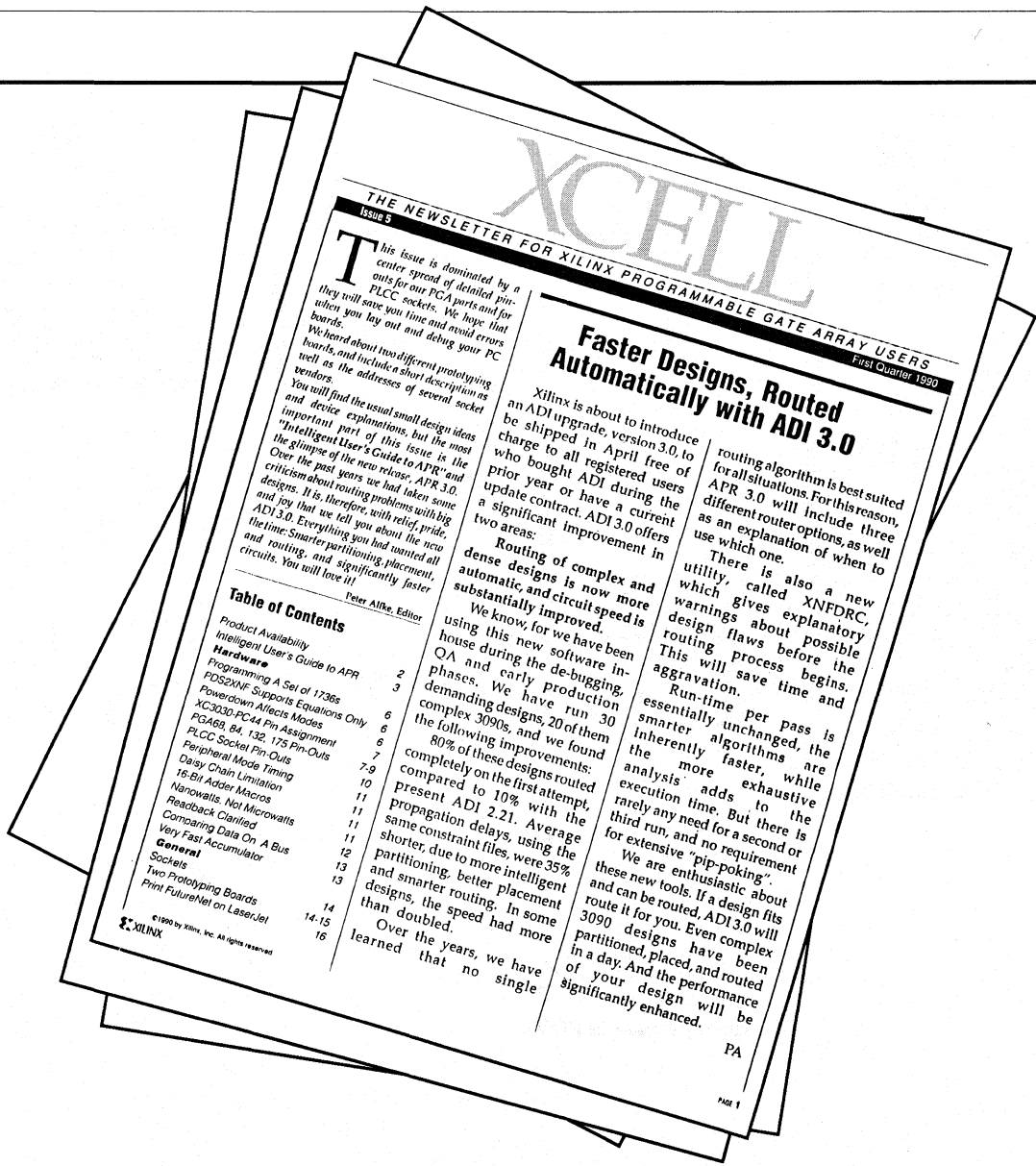
Video Tapes



A one-hour video tape, entitled "Programmable Gate Arrays: The Ideal Logic Device," is available from Xilinx. The presentation is divided into three main sections. The first portion of the video tape is an overview of the Logic Cell Array architecture and the development system, including some example applications. The second section contains a description of the Xilinx product families, a more detailed description of the XC3000 series architecture, a description of the LCA configuration modes, and a brief discussion of programmable gate array performance in terms of speed, density, and cost. Development systems and the design methodology are discussed in the last third of the presentation, including on-screen demonstrations of some of the software tools. Additional video tapes covering specific details are in preparation.

VHS copies are available in NTSC, PAL, and SECAM formats; contact your local Xilinx sales office, sales representative, or distributor.

Newsletter



XCELL

THE NEWSLETTER FOR XILINX PROGRAMMABLE GATE ARRAY USERS

Issue 5 — First Quarter 1990

Faster Designs, Routed Automatically with ADI 3.0

This issue is dominated by a center spread of detailed pin-outs for our PGA parts and for PLCC sockets. We hope that they will save you time and avoid errors when you lay out and debug your PC boards.

We heard about two different prototyping boards, and include a short description as well as the addresses of several socket vendors.

You will find the usual small design ideas and device explanations, but the most important part of this issue is the "Intelligent User's Guide to APR" and the glimpse of the new release, APR 3.0. Over the past years we had taken some criticism about routing problems with big designs. It is, therefore, with relief, pride, and joy that we tell you about the new ADI 3.0. Everything you had wanted all the time: Smarter partitioning, placement, and routing, and significantly faster circuits. You will love it!

Peter Alfke, Editor

Table of Contents

XILINX

Xilinx is about to introduce an ADI upgrade, version 3.0, to be shipped in April free of charge to all registered users who bought ADI during the prior year or have a current update contract. ADI 3.0 offers a significant improvement in two areas:

Routing of complex and dense designs is now more automatic, and circuit speed is substantially improved.

We know, for we have been using this new software in-house during the de-bugging, QA and early production phases. We have run 30 demanding designs, 20 of them complex 3090s, and we found the following improvements:

80% of these designs routed completely on the first attempt, compared to 10% with the present ADI 2.21. Average propagation delays, using the same constraint files, were 35% shorter, due to more intelligent partitioning, better placement and smarter routing. In some designs, the speed had more than doubled.

Over the years, we have learned that no single routing algorithm is best suited for all situations. For this reason, APR 3.0 will include three different router options, as well as an explanation of when to use which one.

There is also a new utility, called XNFDRC, which gives explanatory warnings about possible design flaws before the routing process begins. This will save time and aggravation.

Run-time per pass is essentially unchanged, the smarter algorithms are inherently faster, while the more exhaustive analysis adds to the execution time. But there is rarely any need for a second or third run, and no requirement for extensive "pip-poking".

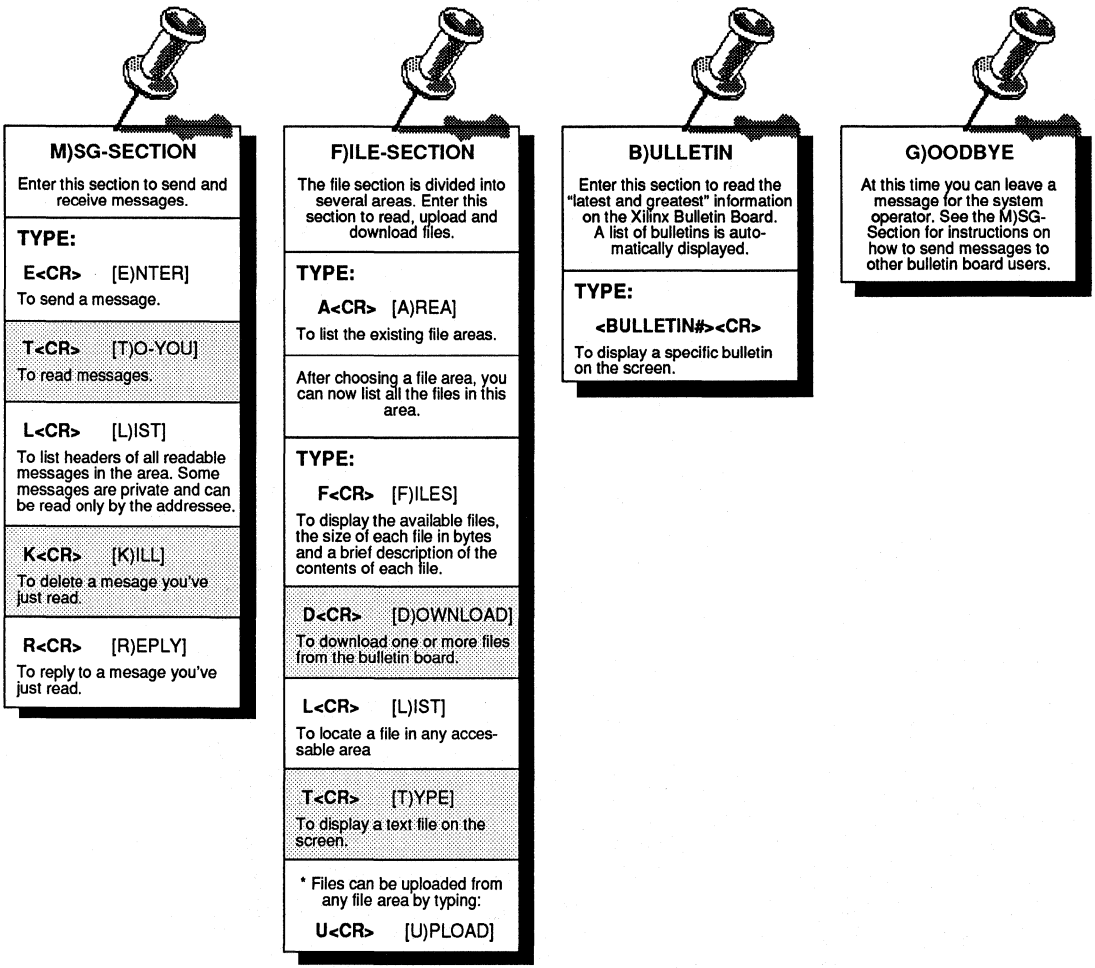
We are enthusiastic about these new tools. If a design fits and can be routed, ADI 3.0 will route it for you. Even complex 3090 designs have been partitioned, placed, and routed in a day. And the performance of your design will be significantly enhanced.

PA

PAGE 1

In September '88, Xilinx started a quarterly technical newsletter to supply up-to-date information to registered Xilinx customers. This newsletter gives updates on hardware and software availability and revision levels. It also carries information on PC-clone compatibility, software bugs and work-arounds. Applications ideas and user tips and a list of relevant magazine articles make this a valuable source of information for the systems designer using LCA devices.

Xilinx Technical Bulletin Board



To provide customers with up-to-date information and an immediate response to questions, Xilinx provides a 24-hour electronic bulletin board. The Xilinx Technical Bulletin Board (XTBB) is available to all registered XACT customers. Users with full privilege can read files on the bulletin board, download those of interest to their own systems or upload files to the XTBB. They can also leave messages for other XTBB users.

New bulletin board users must answer a questionnaire when they first access the XTBB. After answering the questionnaire callers can browse through the bulletin and general information file areas. Before exiting, they should leave a message for the system operator requesting full access. A caller with a valid XACT protection key will be given full user privileges within 24 hours.

The software and hardware requirements for accessing the Xilinx Technical Bulletin Board are:

Baud Rate	1200 or 2400
Character Format	8 data bits, no parity, 1 stop bit
Phone Number	(408) 559-9327
Transfer Protocols	ASCII, Kermit, Xmodem, - CRC, Telink

Information contained on the XTBB is divided into three general categories: 1. Bulletins, 2. Files and 3. Messages.

1. Bulletins contain tidbits of up-to-date information; they can be displayed on screen but cannot be downloaded.

2. Files can contain just about anything (text, user programs, etc.). XTBB users can download files to their own systems or upload files to the bulletin board.

3. Messages are used to communicate with other XTBB users; they can be general—available to everyone—or private.

The XTBB is based on a bulletin board system called FIDO. FIDO is a menu-driven system—you choose commands from menus to decide what happens next. To choose a menu command, simply type the first letter of the command and press return <CR>. Listed below are some helpful hints for using the XTBB.

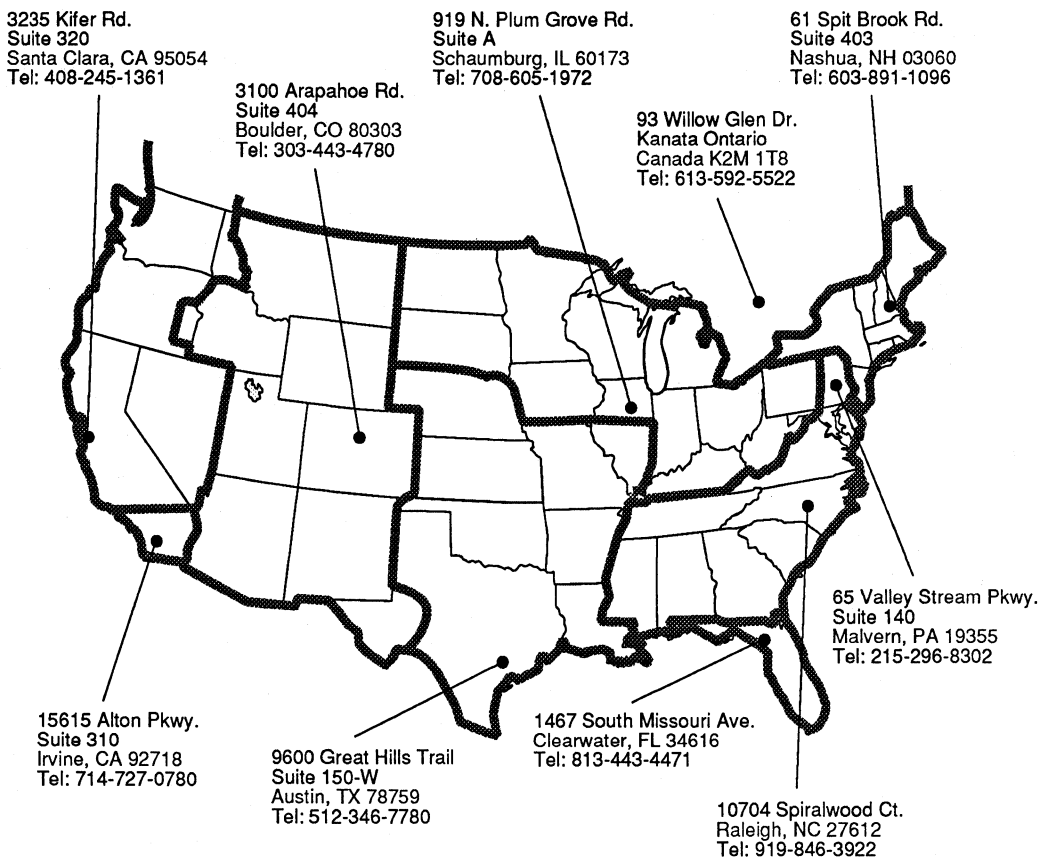
- To perform a sequence of commands, type the first letter of each command, followed by a space, and press return. For example, typing **F A 1 F <CR>** [F)ile A)rea 1 F)iles] from the main menu will list all of the files contained in file area 1.

- Often the user is asked a question and promped to choose between two options (e.g. [yes NO]). The option displayed in all capital letters is the default choice. To select this option, simply press return. Otherwise, type your choice and hit return.

- The XTBB has an extensive help section. To get help, type **?<CR>**. If you have questions about a specific command, type the first letter of the command followed by a question mark and a carriage return (e.g.**F?<CR>**). A short explanation of the command will be displayed.

- For more information, read the XTBBHLP.TXT file located in the GENINFO file area (file area 1).

Field Applications Engineers



North America

There are 13 Xilinx Field Applications Engineers in the locations shown above. Additional technical support is provided by Headquarters Applications.
Dial (408) 879-5199 or (800) 255-7778.

The world-wide network of Xilinx Representatives and Distributors also gives technical support.

Europe

Each of the Xilinx European sales offices in England and Germany has a resident Field Applications Engineer: England (tel 44-932-349401); Germany (tel 49-89-6110851.

Japan

Xilinx Japan is located in Tokyo and has a resident Field Applications Engineer (tel 81-3-297-9191).

Programmable Gate Array Training Courses

The Xilinx Programmable Gate Array Training Courses are comprehensive classes covering the Logic Cell Array component architecture and Xilinx development systems. These courses are intended for design engineers using Programmable Gate Arrays in their applications who want to get "up-to-speed" as quickly as possible. Two courses are offered: a four-day course on the XC3000 series, and a two-day course on the XC4000 series.

A substantial amount of the class time will be spent performing lab exercises on the Xilinx Development System (two students per development system). These development systems will be available to the students on the day following the class for optional individual work and consultations with the instructor(s).

TUITION: The tuition fee is $1,000 per student for the XC3000 course, and $750 for the XC4000 course.

ENROLLMENT: To enroll, call the Training Administrator at Xilinx headquarters at (408) 879-5090 or contact your local Xilinx sales office.

Class size is limited, so early enrollment is recommended. Students are not considered to be enrolled until a check, money order or P.O. for the course tuition is received. Please mail your payment to:

Xilinx, Training Administrator
2100 Logic Drive, San Jose, CA 95124

Enrollments will be confirmed by letter.

There will be a 25% cancellation fee if you cancel within two weeks of your scheduled course date. Failure to cancel at least one week in advance of the scheduled course date will result in forfeiture of the full course tuition.

PREREQUISITES: Students are assumed to have a background in digital logic design.

LOCATION: All courses are held at Xilinx corporate headquarters in San Jose, CA. A detailed map, including a list of nearby hotels, will be included with the enrollment confirmation letter. For regional and international classes, contact your local sales office.

XC3000 COURSE OUTLINE

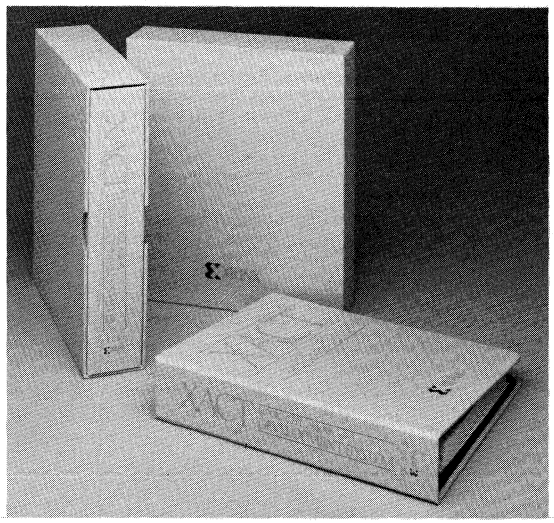
DAY 1:	I.	**BASIC ARCHITECTURE**
	II.	**DESIGN METHODOLOGY AND THE XACT DESIGN MANAGER**
	III.	**DESIGN ENTRY TOOLS**
DAY 2:	IV.	**DESIGN IMPLEMENTATION** LOGIC PARTITIONING AUTOMATIC PLACE AND ROUTE
	V.	**LCA CONFIGURATION**
DAY 3:	VI.	**DESIGN VERIFICATION** SIMULATION XACTOR2 IN-CIRCUIT DEBUGGER
	VII.	**XACT DESIGN EDITOR (XDE)**
DAY 4:	VIII.	**LCA ARCHITECTURE DETAILS**
	IX.	**SUMMARY** ESTIMATING SIZE AND PERFORMANCE BENEFITS OF PGA TECHNOLOGY

XC4000 COURSE OUTLINE

DAY 1:	I.	**INTRODUCTION**
	II.	**THE XACT DESIGN MANAGER (XDM)** AUTOMATIC DESIGN TRANSLATOR (XMAKE)
	III.	**XC4000 ARCHITECTURE**
	IV.	**ESTIMATING SIZE AND PERFORMANCE**
	V.	**DESIGN ENTRY TOOLS**
	VI.	**ROM/RAM MEMORY COMPILER (MEMGEN)**
	VII.	**DESIGN IMPLEMENTATION (PPR)**
DAY 2:	VIII.	**CONFIGURATION**
	IX.	**XACT DESIGN EDITOR AND MAKEBITS**
	X.	**MAKEPROM**
	XI.	**DESIGN VERIFICATION**
	XII.	**DOWNLOAD CABLE**
	XIII.	**READBACK**
	XIV.	**XACT DESIGN EDITOR (XDE)**

Technical Literature

XACT Development System Manuals

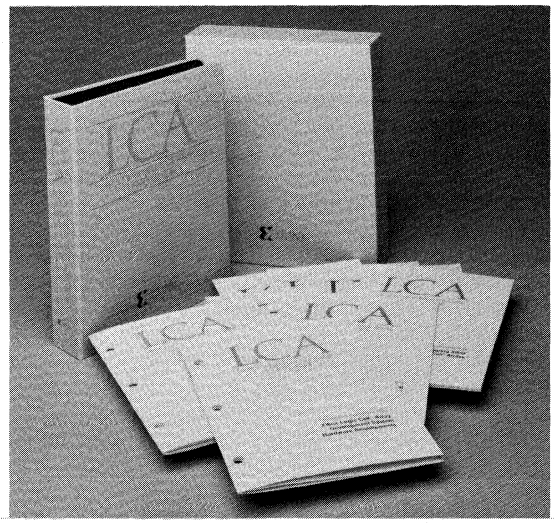


The first two binders of this 3-volume set are the LCA Development System Manuals, providing exhaustive reference information on:

- Executive Program
- LCA Editor
- Macros
- Simulator (SILOS)
- PROM Formatter
- Bit-stream Generator
- Demo Board
- Place and Route
- XNF to LCA
  and information on schematic capture

The third volume provides detailed information on each of the 2000 and 3000 series XACT macros, including schematics, block count, and examples of typical placements.

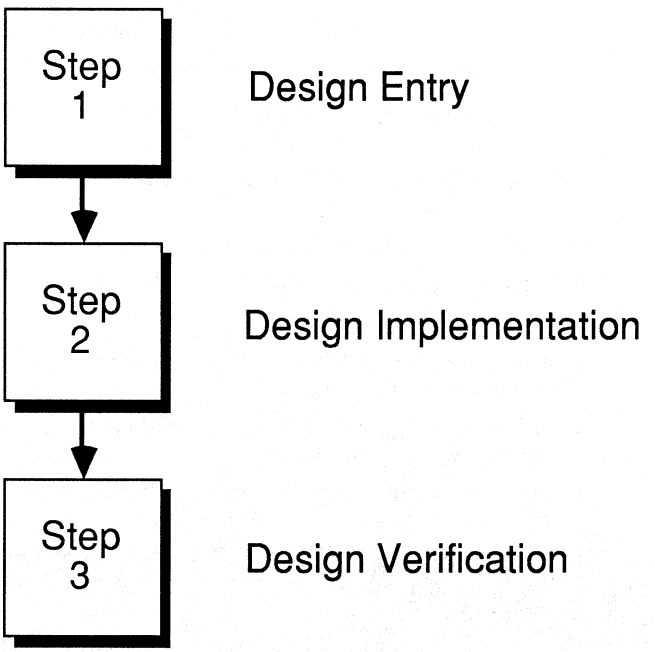
User's Guide



The Xilinx User's Guide, included with every system, is a binder with several self-contained application notes giving practical and tutorial information. The following notes are currently available:

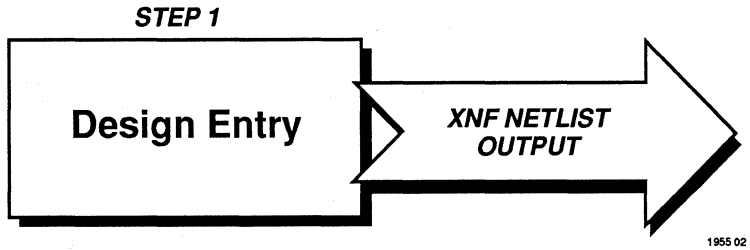
- Introduction
  Basic Design Flow
  Hierarchical Design and Merging
- Getting Started
  Programmable Gate Array Design Flow 1990
  Xilinx Tutorial Using Schema II+
  Xilinx Tutorial Using FutureNet DASH-LCA 1990
  Xilinx Tutorial Using OrCAD/SDT
  XACT Design Editor Tutorial
- Design Entry
  Designing LCAs with Boolean Equations
- Design Implementation
  Advanced Design Methodology
  Fundamentals of Placement and Routing
  LCA Configuration and Debugging Hints
  Automatic Design Translation with Xilinx Design Manager
- Design Verification
  Verification by Readback and Signature Analysis
  Simulating Bidirectional I/O Using SILOS

Xilinx Design Flow Overview (PC or Workstation)

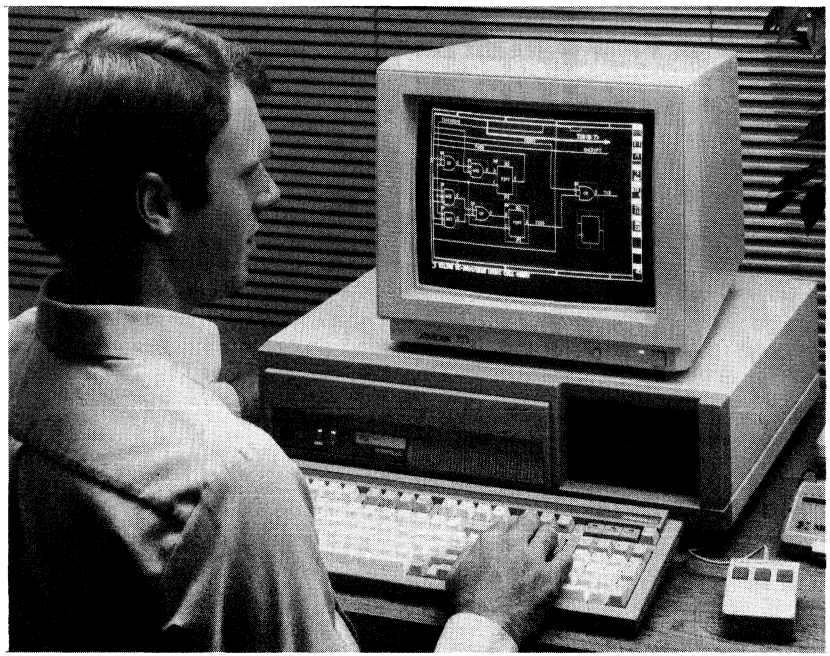


1955 01A

Logic Cell Array Design Flow

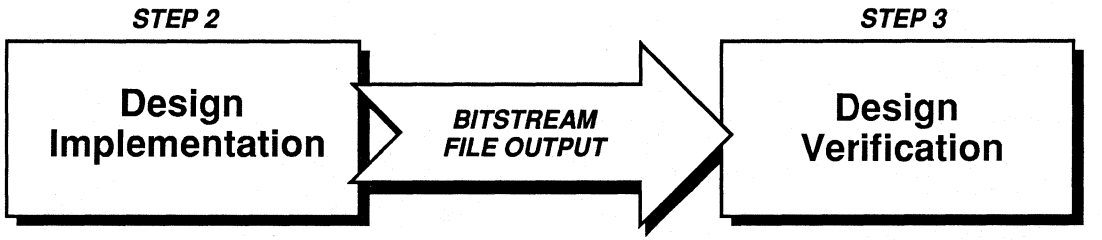


- Open development system supports design entry and simulation on popular CAE systems
- Interfaces available from Xilinx for PC and workstation-based environments:
  - FutureNet, VIEWdraw, OrCAD, Mentor
- Standard macro library includes over 100 elements
- TTL library includes over 100 elements
- ABEL, CUPL, Log/IC, PGADesigner designs may be merged with schematic input
- Several other PC and workstation environments are supported by third-party vendors
- Xilinx State Machine Compiler provides efficient state machine implementation for LCA architecture



Schematic entry on your PC/workstation

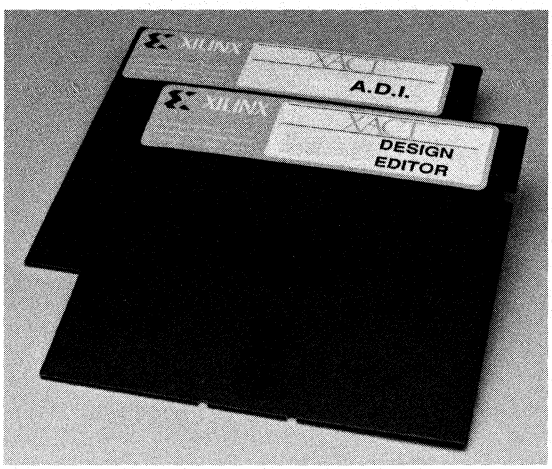
Logic Cell Array Design Flow



1955 03

- Complete system translates design into programmable gate arrays
- Partitions gate-level design logic into LCA architecture (CLB/IOB)
- Automatic logic reduction and partitioning removes unused logic, e.g. unused counter outputs
- Logic synthesis software optimizes design for LCA architecture
- All programs run on PC/AT*-compatible personal computers and Apollo, Sun-4, Sun-3, and DECstation engineering workstations

- Interfaces available from Xilinx to popular simulators for logic and full timing simulations
  - Mentor Graphics
  - VIEWlogic
  - SILOS
  - OrCAD VST
- LCA user-programmability permits real time, in-circuit debugging
- Download cable allows the LCA to be programmed in-circuit during debugging
- XC-DS28 XACTOR or MESA** In-Circuit Design Verifier provides additional hardware debugging capabilities



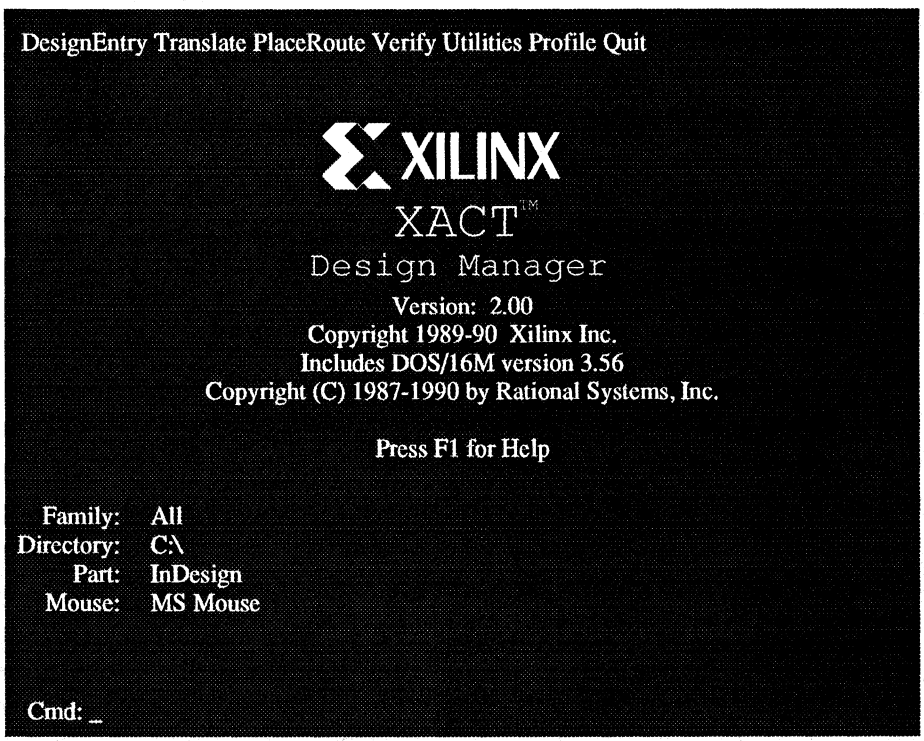
Automated Design Implementation software

Immediate production

*PC/AT is a trademark of IBM Corporation
**MESA is a trademark of Data I/O Inc.

The Xilinx Design Manager—Simplifies the Design Flow

- Lets you run all Xilinx software from menus
- XMAKE facility automates the translation of a design into an LCA file, including optimization, mapping and merging.
- Provides on-line help for all menus, programs and options.



X1248

XMAKE Command

- Automatically invokes all other translation programs as required to compile a design into an LCA file ready for placement and routing
- Supports hierarchically structured designs

Extensive On-line Help

The Design Manager contains on-line Help for

- Every menu
- Every program
- Every program option
- Design-flow suggestions

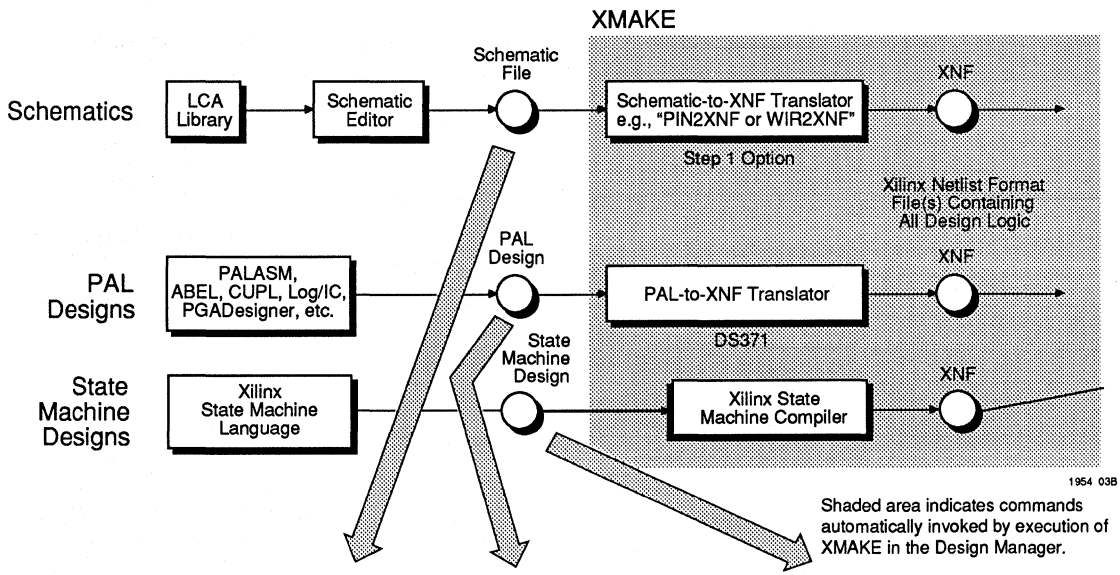
Design Processing Sequence	Design Manager Menu	Description
STEP 1	Design Entry	Create your design by running your schematic editor while inside the Design Manager
	Translate	With one mouse click, *automatically* • Translate all schematic pages and PAL designs to Xilinx format • Merge PAL designs and state machines into schematics
STEP 2	Optimization & Mapping	• Optimize PAL designs and state machines using Logic Synthesis • Map all logic into CLBs and IOBs
	Place Route	Run Automatic Place and Route (APR) program and/or XACT Design Editor to place/route design
	Bitstream Generation	Run MAKEBITS program to compile design into a configuration bitstream for an LCA
STEP 3	Verify	Run simulator, e.g., SILOS or VIEWsim, or in-circuit design verifier, e.g., XACTOR or MESA, to debug your design

1955 04

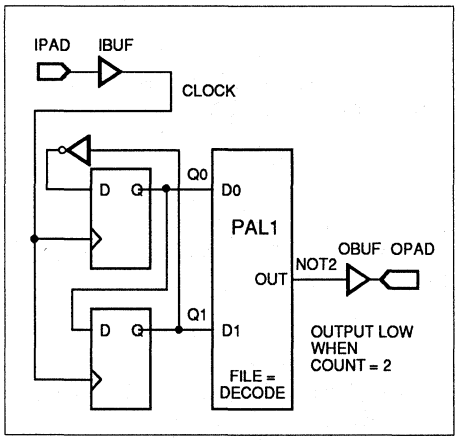
The Xilinx Design Manager provides a highly automated environment for converting your designs into working field-programmable gate-array designs.

This sequence is illustrated — for a very simple design — on the following pages...

Step 1: Design Entry and XNF Translation



Shaded area indicates commands automatically invoked by execution of XMAKE in the Design Manager.



Schematics can include any number of PAL devices created with Boolean equations and/or state machines

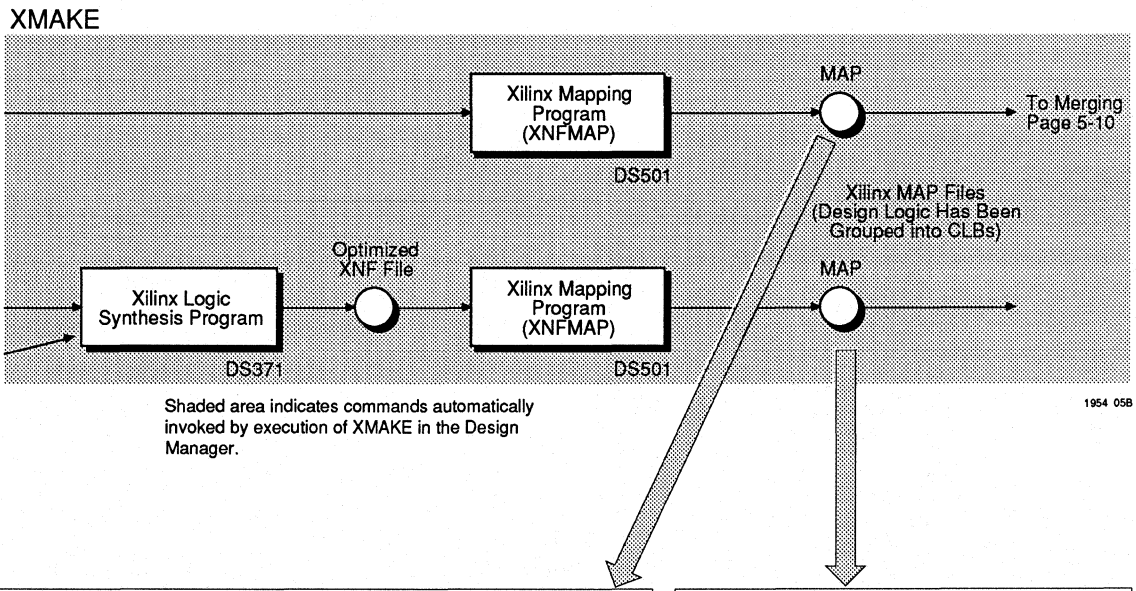
```
TITLE        DECODE.PDS
AUTHOR
COMPANY      XILINX
DATE
CHIP         DECODE  PAL10H8
;Input Pins
              d0 d1
;Output Pins
              out
;
;Define counter states
;
STRING ZERO       '/d1 * /d0'
STRING ONE        '/d1 *  d0'
STRING TWO        ' d1 * /d0'
STRING THREE      ' d1 *  d0'

EQUATIONS
out = ZERO + ONE + THREE
```

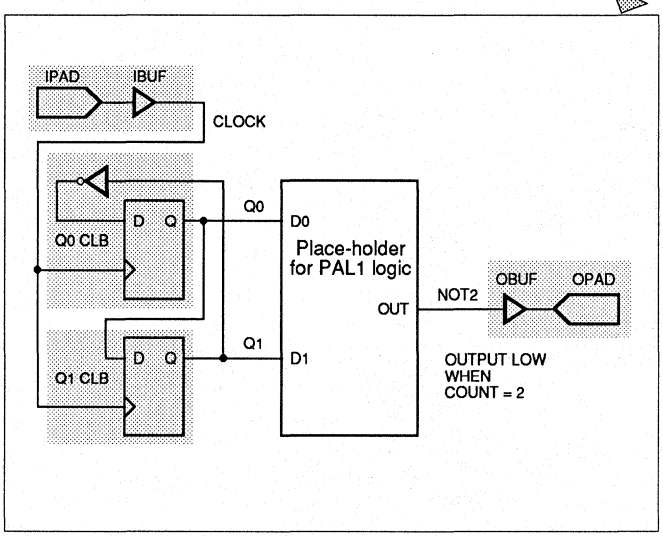
Very simple PAL design included in schematic at left with PAL1 symbol

```
-an example of a VHDL-State
description of a traffic light
controller
-(cases omitted for readability)

CHIP traffic <device_type;

PINS
clock, sensor1, sensor2:IN;
red1, yellow1, green1, red2,
yellow2, green2:OUT;

STATE_MACHINE(state, clock):
  CASE state IS
  WHEN s0 →
    green1 ← '1';
    red2 ← '1';
    IF sensor2=sensor1 THEN
      state ← s1;
    ELSIF (sensor1 ='0'AND
    sensor2= '1')THEN
      state ← s2;
    ELSE
      state ← s0;
    END IF;
  WHEN s1 →
            .
            .
            .
    END CASE;
END STATE MACHINE.
```

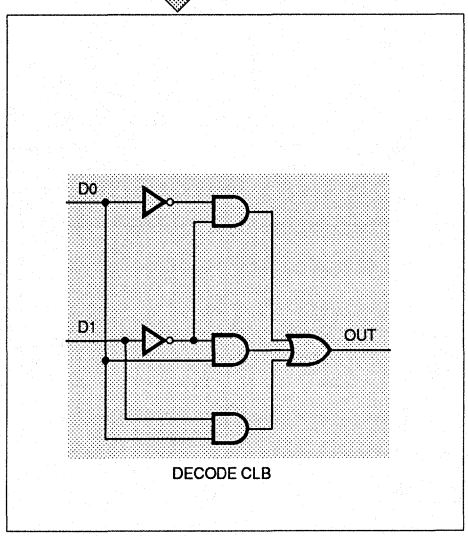
Example of using VHDL-like State Machine Language.

Step 2: Optimization and Mapping



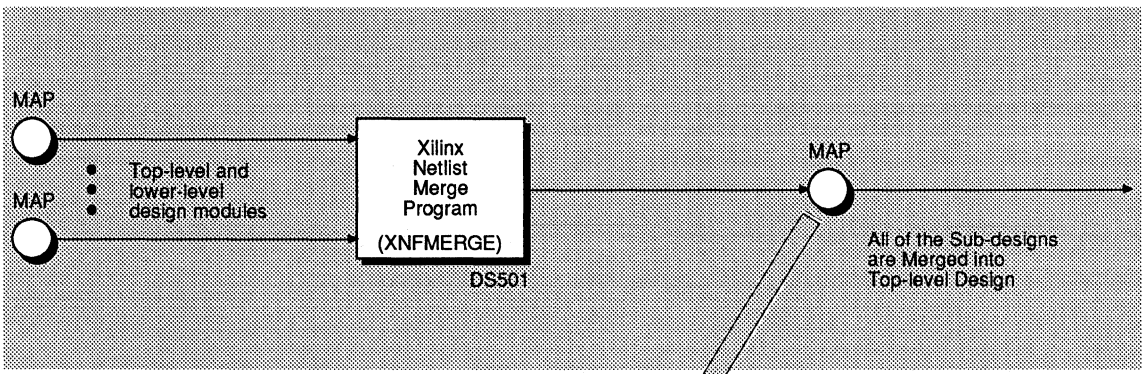


A graphic representation of the top-level MAP file. Unused logic (if any) has been deleted and the remaining logic has been grouped (mapped) into Configurable Logic Blocks and I/O Blocks.

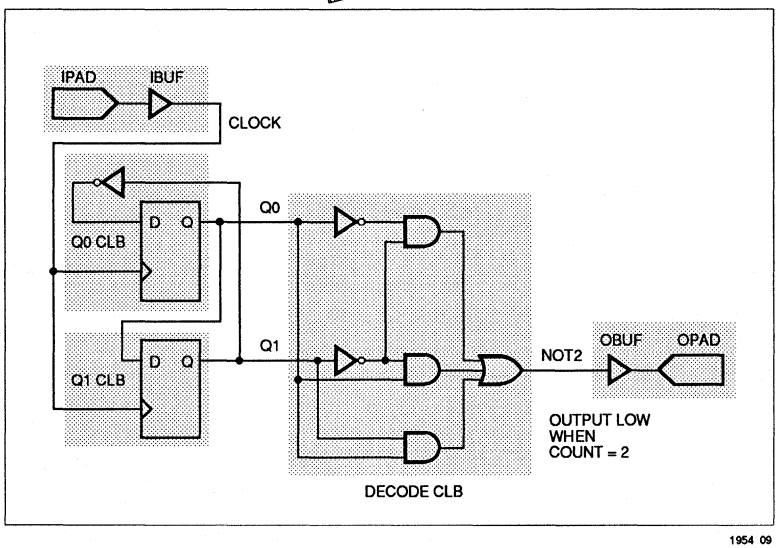


A graphic representation of the CLB containing the PAL logic. In a typical PAL design, of course, several CLBs would be used.

Step 2: Merging

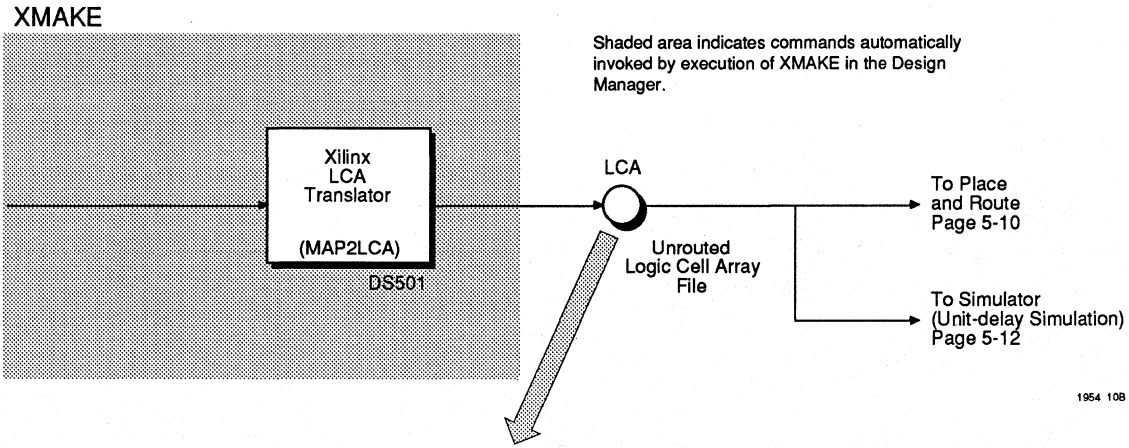


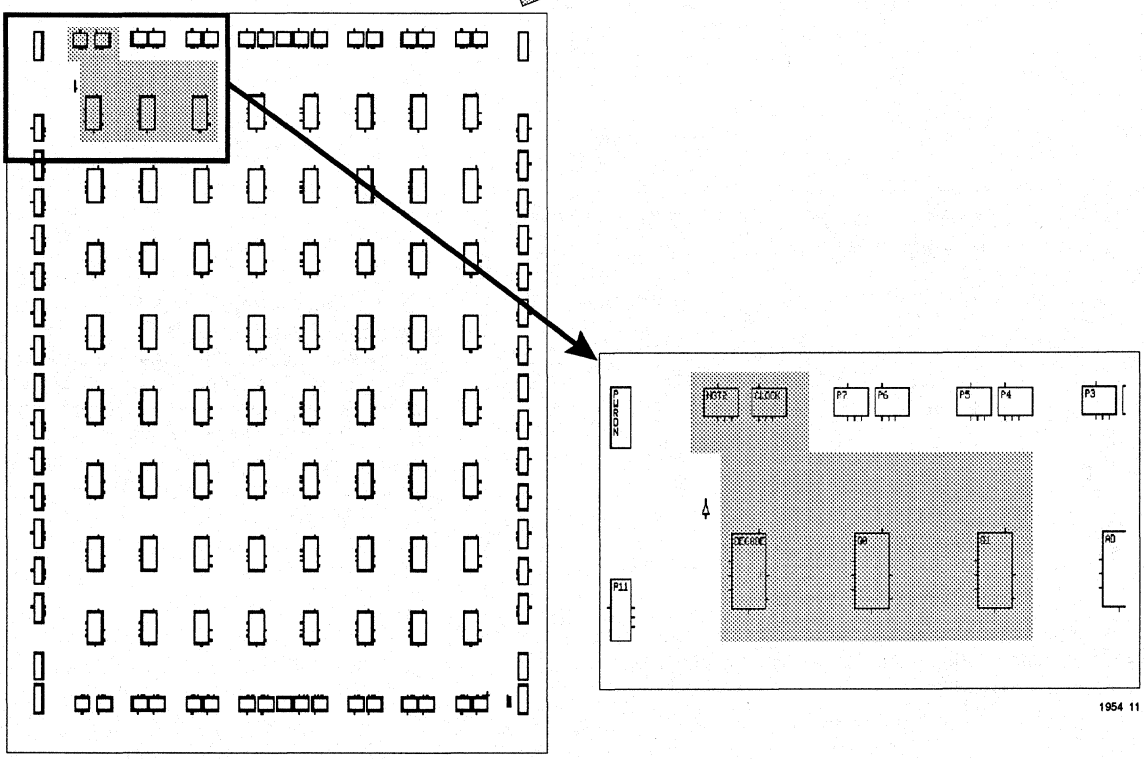
Shaded area indicates commands automatically invoked by execution of XMAKE in the Design Manager.



The merged design contains the CLBs and IOBs for the entire design.

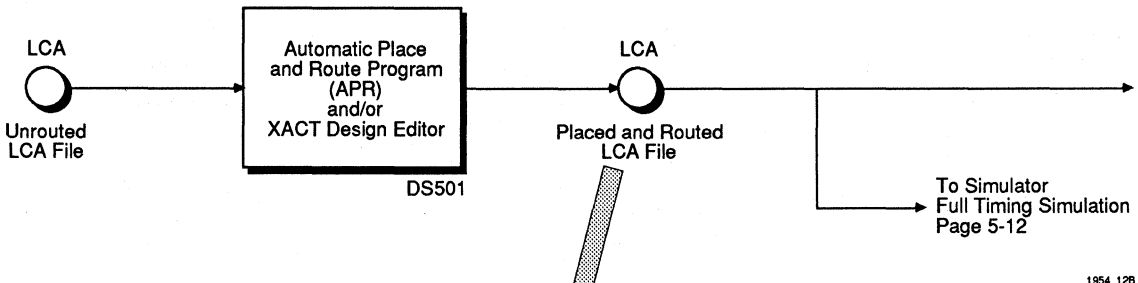
Step 2: Translating to an LCA File

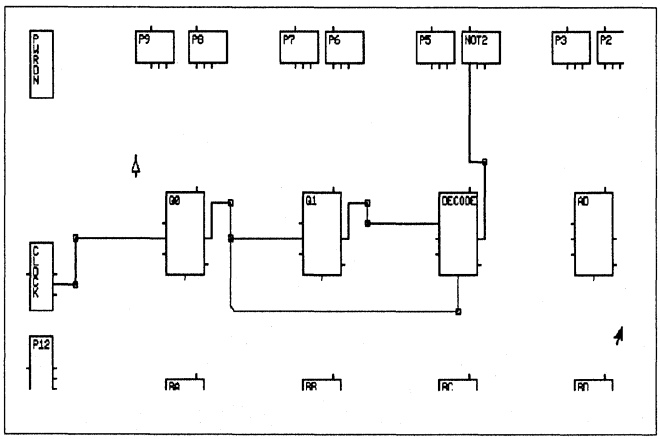




Initially (before Place and Route) the LCA design is unrouted, and the Configurable Logic and I/O blocks are put in random locations.

Step 2: Placing and Routing the LCA File

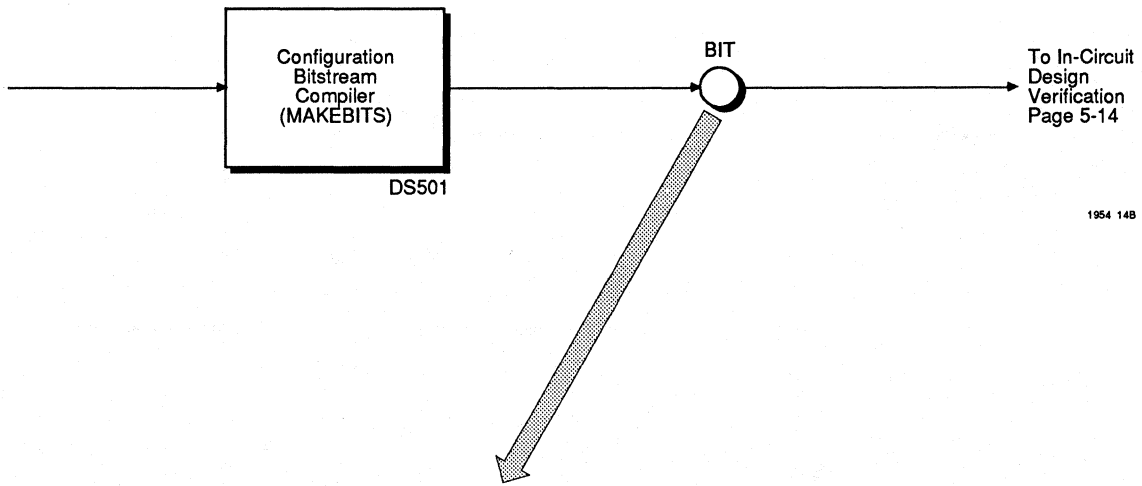




A simple placed and routed design (closeup of upper-left corner of 2064PC68).

The Automatic Place and Route (APR) program uses sophisticated algorithms to determine the optimal placement and routing for a design. The XACT Design Editor, an interactive graphics-based placement and routing tool, is available for the experienced designer who wants to pre-place critical portions of the design or "tweak" the output of the APR program.

Step 2: Bitstream Generation

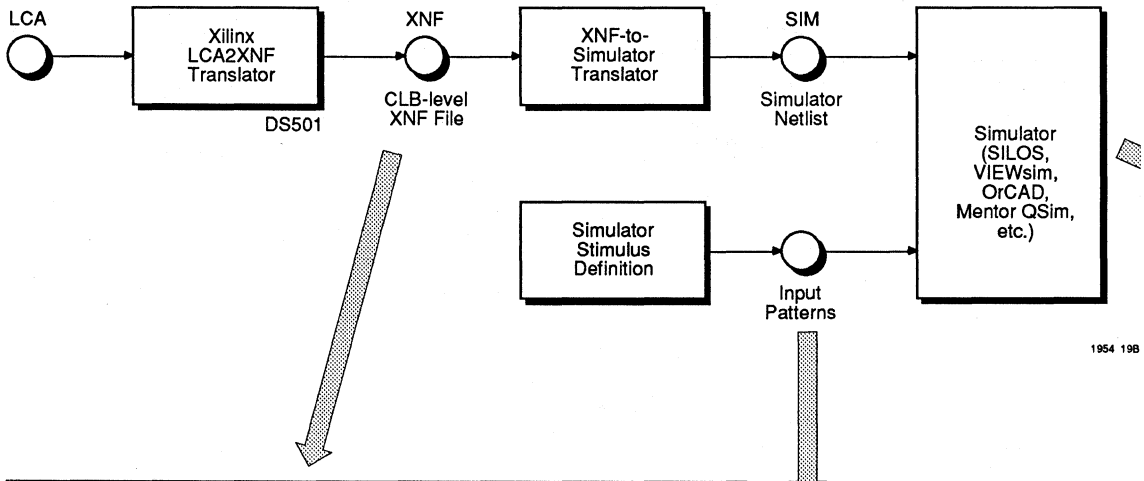


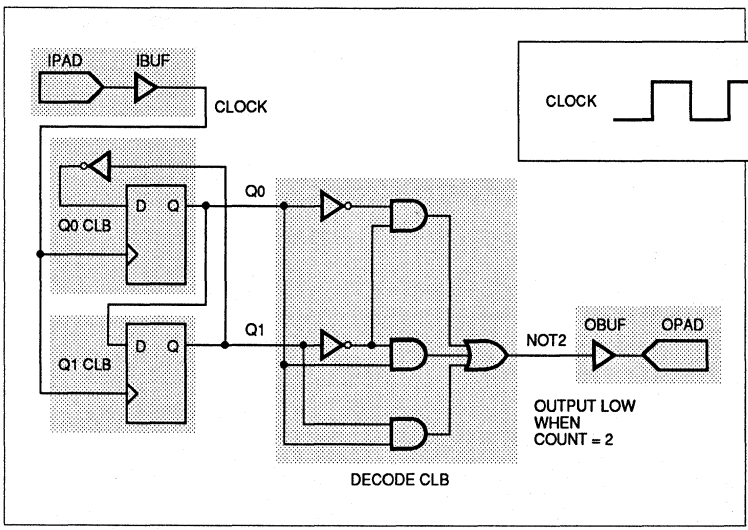

```
11111111001000000000001110011110010011111
00111111110111111111110001011110111111101
01111110101101110111101110111111110111111
01110111011011110111111101111111011111111
00111111111111111111111111111011111111111
01110011011101110111011101111111011111111
11111111001000000000001110011100010011111
00111000111101111101111111111111111111111
00111111101111111111011111101111111111111
00111111111111101111111111111101111101111
01111110111011101111111011111110111111110
01111110101101110111...
```

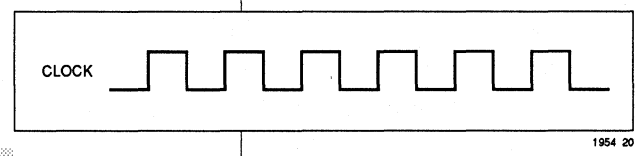
LCA Configuration Bitstream

The BIT file contains the binary configuration data that programs an LCA to perform the design function.

Step 3: Functional and Full Timing Simulation





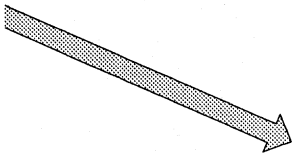


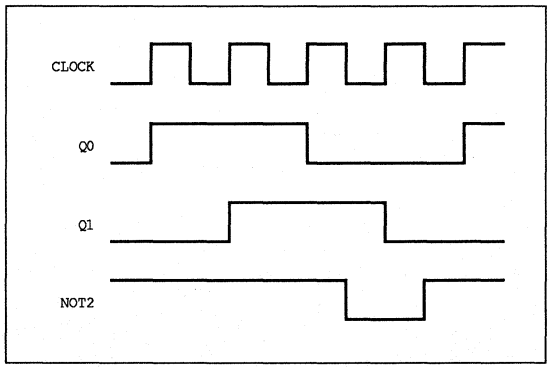
Designer defines design inputs.

LCA designs are simulated at the physical CLB and IOB level with worst-case timing.

Step 3: Functional and Full Timing Simulation

Simulation ***provides for design analysis*** under worst-case temperature, voltage, and process conditions.

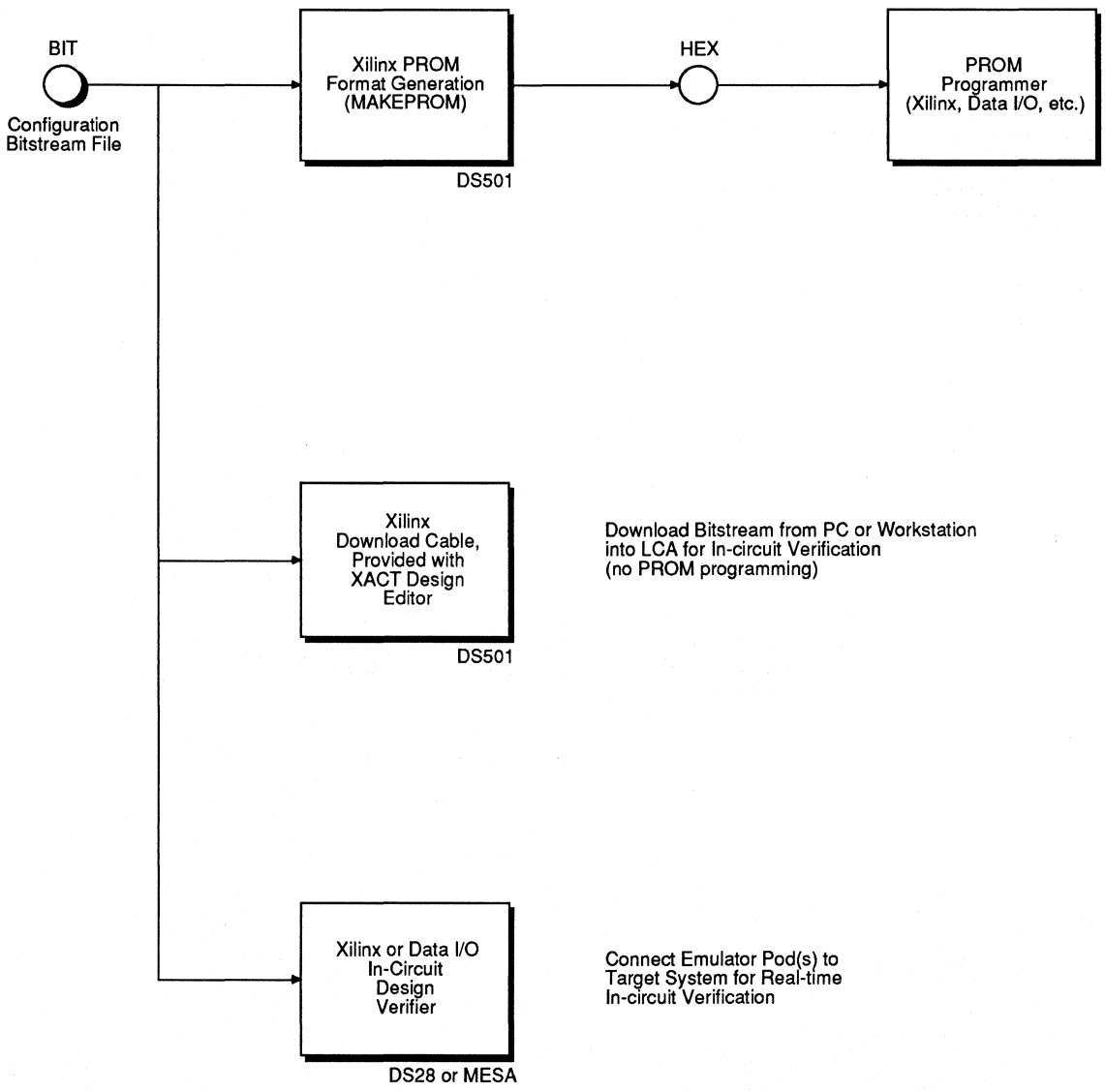




1954 21

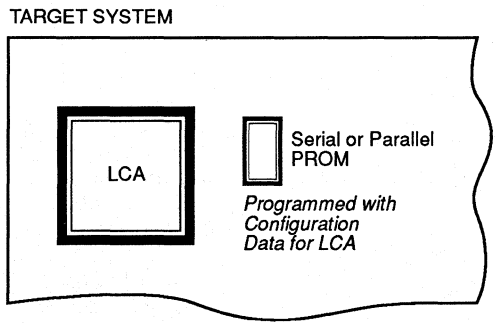
Each I/O pin and CLB output can be observed with a simulator driven by input stimuli. The simulator displays the logic behavior and ac performance of the design in graphics or text form.

Step 3: Real-time, In-circuit Verification



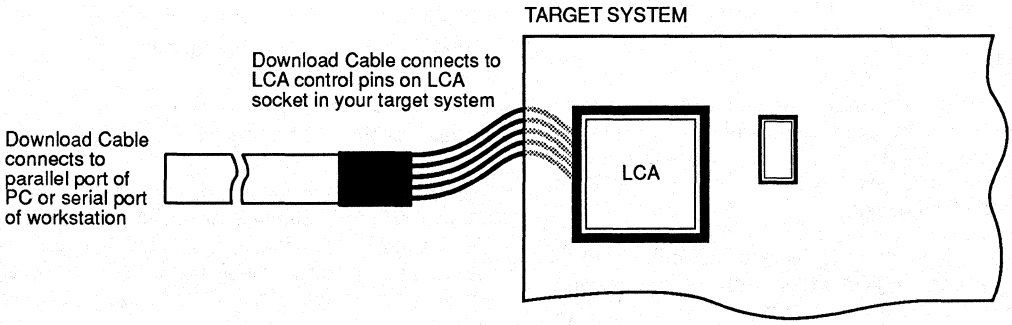
In-circuit verification lets you immediately see how your LCA designs function...

Program a PROM...



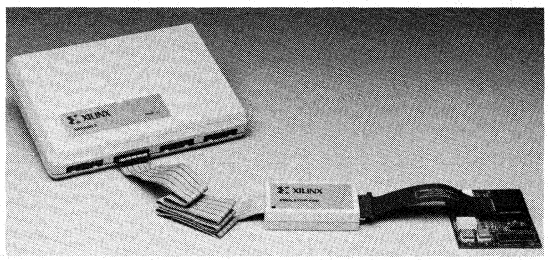
1954 16A

Use the Download Cable...



1954 17A

... Or use the Xilinx XACTOR or Data I/O MESA Design Verifier



The XACTOR controller (left) can control up to four emulation pods (center). An emulation header (right) connects each pod to an LCA socket in the target system.

Xilinx Automatic CAE Tools Product Overview

DESIGN FLOW

The Xilinx Automatic CAE Tools (XACT Development System) use a 3-step design process:

- Step 1: DESIGN ENTRY
- Step 2: DESIGN IMPLEMENTATION
- Step 3: DESIGN VERIFICATION

The Xilinx Logic Libraries and XNF Interface Products support design entry with popular schematic logic drawing systems supplied by multiple vendors, providing easy entry to the XACT Development System. Logic entry from Boolean equations or a variety of state machine language systems is also supported in the Design Implementation phase.

Logic synthesis, partitioning, and optimization programs translate the design specifications into CLBs and IOBs unique to the LCA architecture. Subsequent programs perform automatic placement and routing (APR) to complete the LCA design.

While completely automatic implementation is desirable for both low and high-complexity designs, the designer may prefer an interactive process, especially in high-performance designs. This interactive editing can range from rerouting a few previously automatically routed nets, to prerouting critical nets or preplacing CLBs prior to design completion using APR, to more extensive control over logic partitioning and placement into CLBs. The Automatic Place and Route software gives the designer an option for direct control over specific logic mapped into CLBs (partitioning) to provide better distribution of logic signal routing through the LCA device. The XACT Design Editor, XDE, is extremely versatile, ranging from design entry to CLB and signal routing manipulations. This combination of automatic and interactive design editing capability is a unique feature provided by Xilinx.

Logic simulation or actual in-circuit emulation provides for functional verification, while timing analysis permits verification of critical timing paths under worst-case conditions. The system contains a compiler to generate bitstream patterns to configure the LCA device according to the designer's specification. The overall design flow is illustrated on page 5-17.

An important feature of the XACT Development System is the capability to incorporate design changes, frequently encountered during verification. Small changes can be made to the schematics and then automatically incorporated into the existing design with minimal impact on existing routing and performance. Using this "incremental design" capability, the designer can develop "production quality" programmable gate arrays on a PC or engineering workstation.

PLATFORM AND ENVIRONMENT SUPPORT

The Xilinx Automatic CAE Tools, XACT, are currently available for the following platforms:

- IBM PC/AT, PS/2, and compatibles
- Apollo DN4000 Series
- Sun-4 and SparcStation Series
- Sun-3 Series 960 and above
- DECstation 3100 Series

Xilinx and third-party vendors have developed library and interface products compatible with a variety of design entry and simulation environments. Xilinx has provided a standard interface file specification, XNF, to simplify file transfers into and out of the XACT Development System.

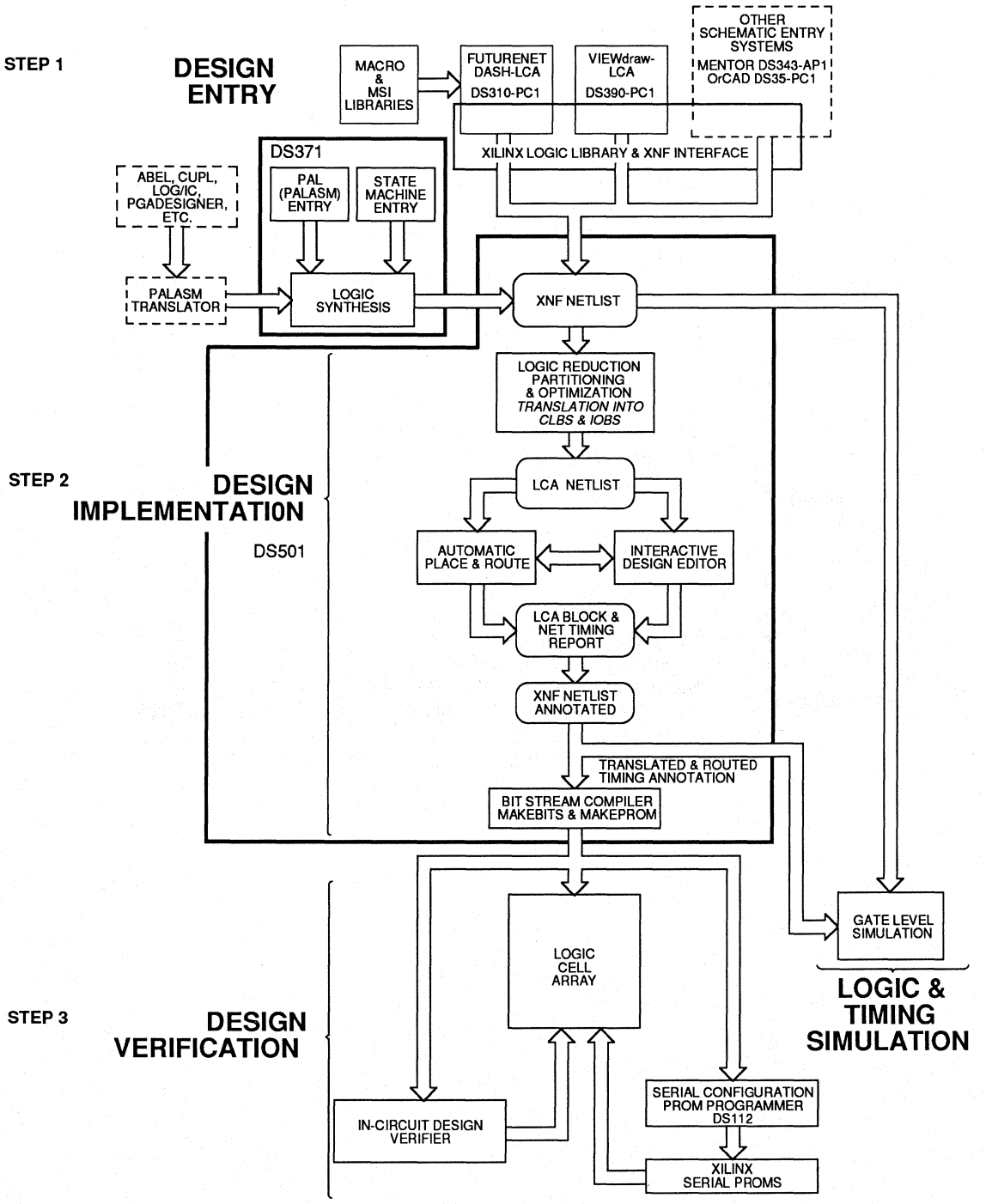
Xilinx directly supports the following design environments:

- FutureNet DASH
- VIEWlogic VIEWdraw and VIEWsim
- Mentor Graphics NetED and Qsim
- OrCAD SDT and VST
- SILOS

Several other environments are supported by third-party vendors.

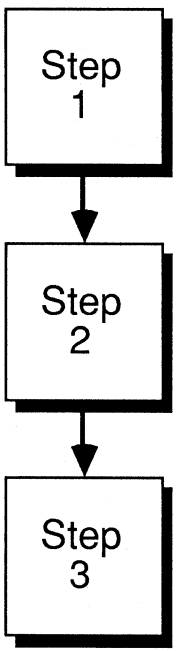
A collection of over 100 TTL logic macrofunctions is available for the schematic editors, and is included in the appropriate packages at no charge.

The XACT Design Manager, XDM, simplifies the selection of command-line options with pull-down menus and on-line help text. Application programs ranging from schematic capture to APR can be accessed from the XDM, while the sequence of program commands is generated and stored for documentation prior to execution. The XMAKE command in the XDM automates the entire translation, optimization, merging, and mapping process.


STEP 1
DESIGN ENTRY
MACRO & MSI LIBRARIES
FUTURENET DASH-LCA DS310-PC1
VIEWdraw-LCA DS390-PC1
OTHER SCHEMATIC ENTRY SYSTEMS
MENTOR DS343-AP1
OrCAD DS35-PC1
XILINX LOGIC LIBRARY & XNF INTERFACE
DS371
ABEL, CUPL, LOG/IC, PGADESIGNER, ETC.
PAL (PALASM) ENTRY
STATE MACHINE ENTRY
PALASM TRANSLATOR
LOGIC SYNTHESIS
XNF NETLIST
LOGIC REDUCTION PARTITIONING & OPTIMIZATION
TRANSLATION INTO CLBS & IOBS
STEP 2
DESIGN IMPLEMENTATION
DS501
LCA NETLIST
AUTOMATIC PLACE & ROUTE
INTERACTIVE DESIGN EDITOR
LCA BLOCK & NET TIMING REPORT
XNF NETLIST ANNOTATED
TRANSLATED & ROUTED TIMING ANNOTATION
BIT STREAM COMPILER MAKEBITS & MAKEPROM
GATE LEVEL SIMULATION
LOGIC & TIMING SIMULATION
STEP 3
DESIGN VERIFICATION
LOGIC CELL ARRAY
IN-CIRCUIT DESIGN VERIFIER
SERIAL CONFIGURATION PROM PROGRAMMER DS112
XILINX SERIAL PROMS

1956 01C

Xilinx Automatic CAE Tools Product Options



• LCA Logic Synthesis Tools	DS371
• DASH-LCA Editor and interface	DS310
• FutureNet DASH Interface	DS31
• Mentor Graphics NetEd and Qsim Interface	DS343
• OrCAD SDT Interface	DS35
• VIEW-LCA Schematic Editor and Interface	DS390
• VIEWdraw and VIEWsim Interface	DS391
• EDIF 2.0.0 Netlist Interface	DS361
• XC2000 & XC3000 Series XACT Design Implementation System	DS501
• SILOS Simulator and Interface	DS22
• VIEWsim Simulator and Interface	DS290
• OrCAD VST Interface	DS351
• Serial Configuration PROM Programmer	DS112
• XACTOR In-Circuit Design Verifier	DS28

XC-DS310 DASH-LCA Schematic Editor, Interface and Library

Step 1 Option

Product Brief

FEATURES

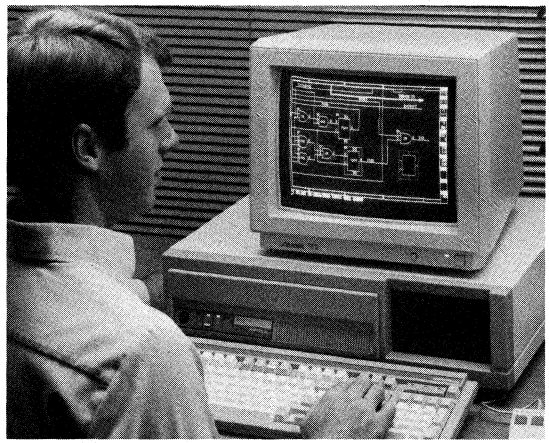
- Xilinx only FutureNet DASH-LCA schematic editor provides easy-to-use hierarchical LCA design capability
- Macro library of over 100 standard logic family equivalents derived from the XACT Macro Library
- Library of logic symbols including all two-input, three-input and four-input AND, OR and XOR gates plus storage, input/output and clock elements
- Additional 100 7400 TTL library elements. See page 5-34 for a listing of TTL macros
- User control for flagging critical paths for the Automated Placement and Routing program
- Converts schematic drawings to a Xilinx Netlist Format (XNF) output file
- Output compatibility with XC-DS501 XACT Design Implementation System
- Runs on PC/AT or compatible personal computers

GENERAL

Schematic entry and automatic partitioning of LCA designs shortens logic reduction and product development times. Complex designs can be specified schematically and quickly implemented for in-circuit design verification.

DASH-LCA supports unlimited levels of hierarchy. The Xilinx DASH-LCA Schematic Library provides the symbol library and conversion utility to permit designers to enter LCA designs with the DASH-LCA Schematic Editor. The Xilinx library provides the logic, I/O, and macro symbols to be used in the schematic. A Xilinx conversion utility converts the schematic into an XNF output file.

Once partitioned, the design may be placed and routed with the XC-DS501 XACT Design Implementation System. The Xilinx symbol library includes symbols to flag critical data and clock signals which the Automatic Placement and Routing Program uses to prioritize those signals for minimum delay.



3002

XC-DS31 DASH Schematic Interface and Library

Step 1 Option

Product Brief

FEATURES

- Library and translator for users of the DASH Schematic Designer
- Macro library of over 100 standard logic family equivalents derived from the XACT Macro Library
- Library of logic symbols including all two-input, three-input and four-input AND, OR and XOR gates plus storage, input/output and clock elements
- Additional one hundred 7400 MSI library elements. See page 5-34 for a listing of all macros.
- User control for flagging critical paths for the Automated Placement and Routing program
- Converts schematic drawings to a Xilinx Netlist Format (XNF) output file
- Output compatibility with XC-DS501 XACT Design Implementation System
- Runs on PC/AT or compatible personal computers, Sun 3 and Sun 4

GENERAL

Schematic entry and automatic partitioning of LCA designs shortens logic reduction and product development times. Complex designs can be specified schematically and quickly implemented for in-circuit design verification.

The Xilinx DASH Schematic Designer Library provides the symbol library and conversion utility to permit designers to enter LCA designs with the DASH Schematic Designer. The Xilinx library provides the logic, I/O, and macro symbols to be used in the schematic. A Xilinx conversion utility converts the schematic into an XNF output file.

Once partitioned, the design may be placed and routed with the XC-DS501 XACT Design Implementation System. The Xilinx symbol library includes symbols to flag critical data and clock signals which the Automatic Placement and Routing Program uses to prioritize those signals for minimum delay.

Xilinx provides ongoing support for users of the DASH Schematic Designer Library. For the first year, software updates are included. After that, the user may purchase the XC-SC31 Annual Support Agreement to continue to receive the latest software releases.

1964

XC-DS343 Mentor Graphics Schematic and Simulation Interfaces and Library

Step 1 Option

Product Brief

FEATURES

- Mentor Graphics certified interfaces
- The IDEA* Interface Station can be used for schematic entry and simulation of programmable-gate-array designs
- Full timing simulation with post placement/routing information
- Primitive library includes flip-flops, latches, AND, OR, XOR, NAND, NOR gates
- Macro library includes over 100 standard logic elements (counters, multiplexers, registers, etc.)
- Additional one hundred 7400 MSI library elements included at no charge. See page 5-34 for a listing of all macros (available 1H91).
- Xilinx Netlist Format (XNF) output is compatible with XC-DS501 Design Implementation System
- Available on Apollo SR10.1 and Mentor IDEA V7.0

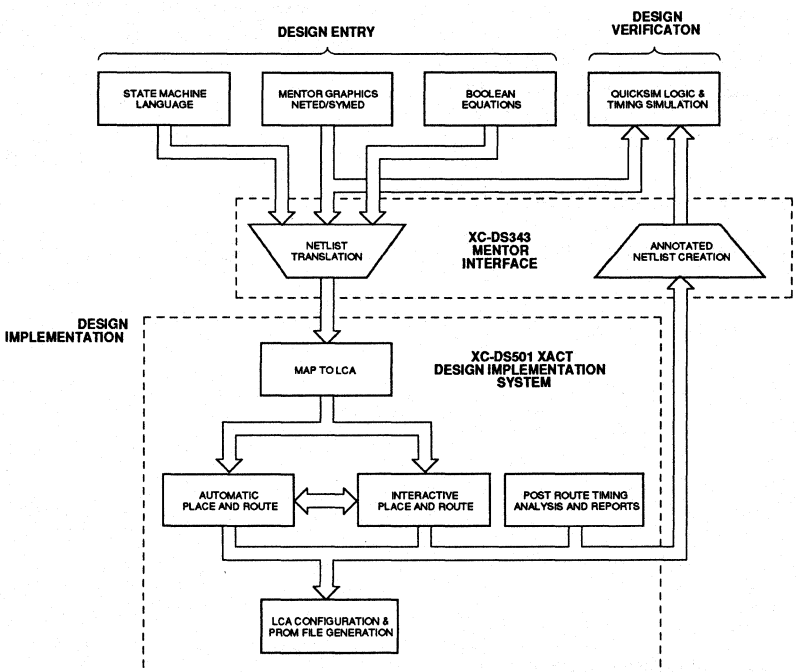
*IDEA is a registered trademark of Mentor Graphics

GENERAL

Schematic entry and automatic partitioning of LCA designs shorten logic reduction and product-development times. Complex designs can be specified schematically and quickly implemented for full timing simulation and in-circuit design verification.

The Xilinx DS343 package provides the symbol library and conversion utility to permit designers to enter LCA designs with the Mentor Graphics NetED Schematic Editor. The Xilinx library provides the logic, I/O, macro, and TTL symbols to be used in the schematic. A Xilinx conversion utility converts the schematic into an XNF output file.

Once partitioned, the design may be placed and routed with the Apollo-based XC-DS501 XACT Design Implementation System. The Xilinx symbol library includes symbols to flag critical data and clock signals which the Automatic Placement and Routing Program uses to prioritize those signals for minimum delay.



1958 02B

XC-DS35 OrCAD* SDT Schematic Entry Interface and Design Library

Step 1 Option

Product Brief

FEATURES

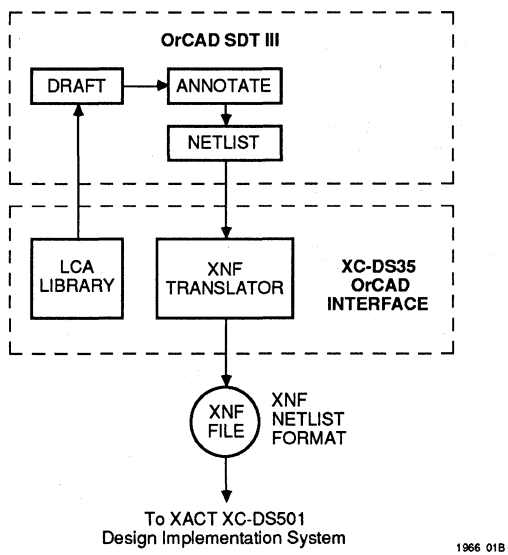
- Library and translator for users of the OrCAD* SDT Schematic Editor
- Library of over 100 standard logic macros
- Library of logic symbol primitives includes AND, OR, NAND, NOR, and XOR gates plus storage, input/output and clock elements
- Additional one hundred 7400 MSI library elements included at no charge. See page 5-34 for a listing of all macros (available 1H91).
- User control for flagging critical paths for the Automated Placement and Routing Program
- Converts schematic drawings to a Xilinx Netlist Format (XNF) output file
- Output compatibility with XACT Design Implementation System
- Runs on a PC/AT or compatible personal computer

GENERAL

Schematic entry and automatic partitioning of LCA designs shorten logic reduction and product development times. Complex designs can be specified schematically and quickly implemented for in-circuit design verification.

The Xilinx OrCAD Schematic Entry Interface provides the symbol library and conversion utility to permit designers to enter LCA designs with the SDT Schematic Editor. The Xilinx library provides the logic, I/O, and macro symbols to be used in the schematic. A Xilinx conversion utility converts the schematic into an XNF output file.

Once partitioned, the design may be placed and routed with the PC-based XACT Automated Design Implementation Program. The Xilinx symbol library includes symbols to flag critical data and clock signals which the Automatic Placement and Routing Program uses to prioritize those signals for minimum delay.



*OrCAD is a registered trademark of OrCAD Systems Corp.

XC-DS361 EDIF Netlist Interface

Step 1 & 3 Options

Product Brief

FEATURES

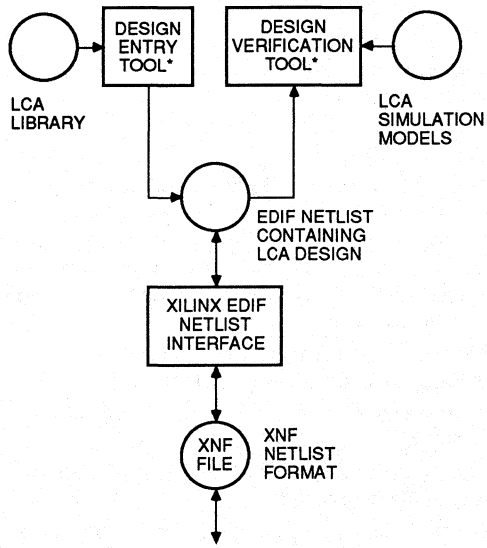
- Provides Xilinx netlist interfaces to many design-entry and verification tools that support EDIF
- Reads and writes LCA designs in EDIF v 2.0 format netlist
- Supports all logic symbols and timing parameters supported by XACT Design Implementation System
- Runs on all Xilinx-supported PC and workstation platforms

GENERAL

The Xilinx EDIF Netlist Interface, used in conjunciton with an LCA library, permits designers to enter and verify LCA designs using popular CAE tools that support EDIF.

The design is created using an LCA library for the desired CAE tool (available from either Xilinx or the CAE vendor). The design is converted to an EDIF netlist using the CAE tool, then translated into a Xilinx netlist using the Xilinx EDIF Netlist Interface.

The design is then partitioned, placed and routed using the XC-DS501 XACT Design Implementation System to simulate the design. It is then translated back to EDIF format – with full timing – using the Xilinx EDIF Netlist Interface. Once in EDIF format, the design can be read into an EDIF-compatible design environment for simulation.



* Contact Xilinx for current list of compatible third-party CAE tools

1992 01B

XC-DS390 VIEWdraw-LCA Schematic Editor, Interface and Library

Step 1 Option

Product Brief

FEATURES

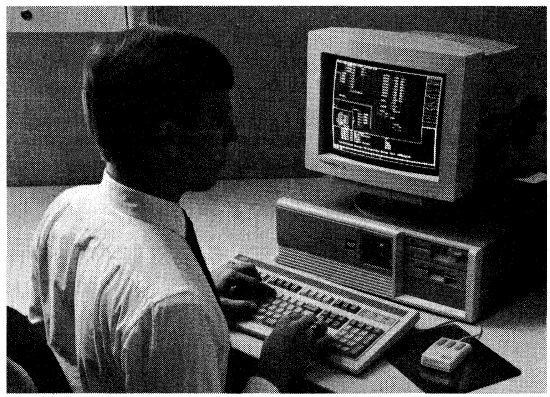
- VIEWlogic VIEWdraw-LCA Schematic Editor provides easy-to-use hierarchical LCA design capability
- Macro library of over 100 standard logic family equivalents derived from the XACT Macro Library
- Library of logic symbols including all two-input, three-input and four-input AND, OR and XOR gates plus storage, input/output and clock elements
- Additional one hundred 7400 MSI library elements included at no charge. See page 5-34 for a listing of all macros (available 1H91).
- User control for flagging critical paths for the Automated Placement and Routing
- Converts schematic drawings to a Xilinx Netlist Format (XNF) output file
- Output compatibility with XC-DS501 XACT Design Implementation System
- Runs on PC/AT or compatible personal computers

GENERAL

Schematic entry and automatic partitioning of LCA designs shorten logic-reduction and product-development times. Complex designs can be specified schematically and quickly implemented for in-circuit design verification.

VIEWdraw-LCA is the Xilinx-only VIEWdraw Schematic editor and supports unlimited levels of hierarchy. The Xilinx VIEWdraw-LCA Library provides the symbol library and conversion utility to permit designers to enter LCA designs with the Xilinx-only VIEWdraw Schematic Editor. The Xilinx library provides the logic, I/O, and macro symbols to be used in the schematic. A Xilinx conversion utility converts the schematic into an XNF output file.

Once partitioned, the design may be placed and routed with the PC-or workstation-based XC-DS501 XACT Design Implementation System. The Xilinx symbol library includes symbols to flag critical data and clock signals which the Automatic Placement and Routing Program uses to prioritize those signals for minimum delay.



XC-DS391 VIEWlogic VIEWdraw and VIEWsim Interfaces and Library

Step 1 and Step 3 Options

Product Brief

FEATURES

- Library and translator for users of the VIEWlogic VIEWdraw Schematic Editor and VIEWsim Simulator
- Macro library of over 100 standard logic family equivalents derived from the XACT Macro Library
- Library of logic symbols including all 2-input, 3-input and 4-input AND, OR and XOR gates plus storage, input/output and clock elements
- Additional 100 7400 TTL library elements. See page 5-34 for listing of the TTL macros.
- User control for flagging critical paths for the Automated Placement and Routing
- Converts schematic drawings to a Xilinx Netlist Format (XNF) output file
- Converts XNF files to format accepted by VIEWsim Simulator for logic and timing simulation
- Output compatibility with XC-DS501 XACT Design Implementation System
- Runs on PC/AT-compatible personal computers, Sun-3, Sun-4 and DECstation 3100

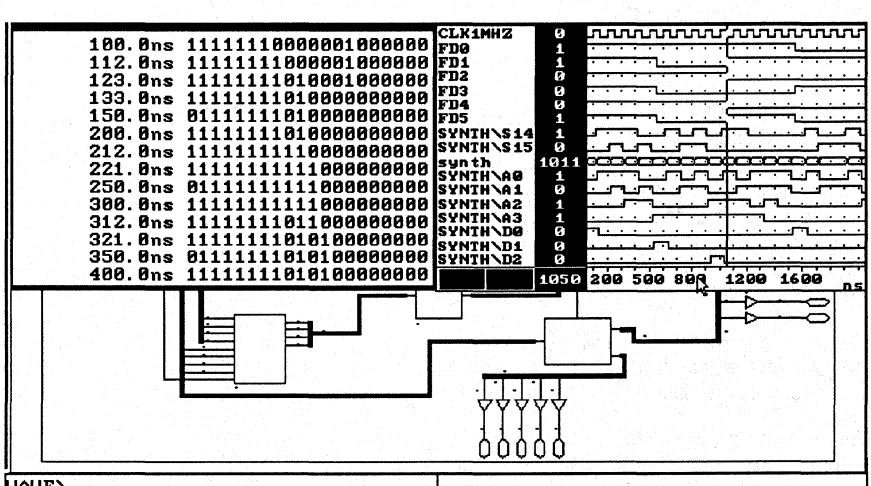
GENERAL

Schematic entry and automatic partitioning of LCA designs shorten logic-reduction and product-development times. Complex designs can be specified schematically and quickly implemented for in-circuit design verification.

The Xilinx VIEWdraw Library provides the symbol library and conversion utility to permit designers to enter LCA designs with the VIEWdraw Schematic Designer. The Xilinx library provides the logic, I/O, and macro symbols to be used in the schematic. A Xilinx conversion utility converts the schematic into an XNF output file.

Once partitioned, the design may be placed and routed with the PC- or workstation-based XC-DS501 XACT Design Implementation System. The Xilinx symbol library includes symbols to flag critical data and clock signals which the Automatic Placement and Routing Program uses to prioritize those signals for minimum delay.

With the Xilinx VIEWsim Simulation Interface, designers can use the VIEWlogic simulation environment to perform post-layout simulation. All post-layout timing information, including pin-to-pin delays, is back annotated into the VIEWlogic environment for full timing simulation.



X1251

XC-DS371 LCA Logic Synthesis Tools

Step 1 Option

Product Brief

FEATURES

Xilinx State Machine Language

- Combines the simplicity of popular PLD languages with the power and expressiveness of VHDL.
- Allows complex state machine implementation using a simplified VHDL-like language.
- Supports multiple state machines.
- External-file-reference capability and syntax-alias capability provide versatile environment for state-machine entry.

Xilinx State Machine Compiler

- Automatically synthesizes the Xilinx State Machine Language, and generates an optimized Xilinx Netlist Format (XNF) file.
- Supports both automatic state-encoding algorithms optimized for the LCA architectures and user-defined state-bit assignments.
- User can optimize for speed or area

Xilinx PLD Synthesis

- Automatically synthesizes PALASM2 format, and generates an optimized Xilinx netlist

GENERAL

Xilinx Logic Synthesis Tools allow designers to describe designs in easily understood and modified forms such as Boolean equations or state-machine description language, in addition to schematic diagrams. Xilinx Logic Synthesis Tools efficiently map technology-independent logic descriptions such as state-machine descriptions into the LCA architecture. The implementation can be optimized for either speed or the amount of logic used.

The figure on page 5-17 illustates the design flow from a design schematic with some glue logic, a TTL macro, PLD symbol, or state machine description.

The PLD design is entered using ABEL, CUPL, Log/IC, PGADesigner, or directly through PALASM2 Boolean Equations. The state-machine design is entered using the Xilinx State Machine Language, which is an intuitive and powerful language for state-machine entry. An example is given below.

The multiple-mode design is combined into the LCA architecture in the following steps. The schematic is converted by the netlist translator to the Xilinx Netlist Format (XNF) file. Then, the PALASM2 and Xilinx State Machine Language are translated and optimized, resulting in a second XNF file. The design files are then merged and partitioned into CLBs and IOBs.

PLD synthesis and XNF optimization will continue to be included in the DS501 package until the DS371 becomes available.

```
-an example of a VHDL-State description of
 a traffic light controller
-(cases omitted for readability)

CHIP traffic <device_type;

PINS
clock, sensor1, sensor2:IN;
red1, yellow1, green1, red2, yellow2,
green2:OUT;

STATE_MACHINE(state, clock):
  CASE state IS
  WHEN s0 →
    green1 ← '1';
    red2 ← '1';
    IF sensor2=sensor1 THEN
      state ←s1;
    ELSIF (sensor1 ='0'AND
    sensor2= '1')THEN
      state ← s2;
    ELSE
      state ← s0;
    END IF;
  WHEN s1 →
    green1 ← '1';
    red2 ← '1';
    state ← s2;
  WHEN s4 →
    red1 ← '1';
    green2 ← '1';
    IF (sensor1 = '1' AND
    sensor2 = '0') THEN
      state ← s5;
    ELSIF (sensor1 = '1' AND
    sensor2 = '0') THEN
      state ← s6;
    ELSE
      state ← s4;
    END IF;
  WHEN s7 →
    red1 ← '1';
    yellow2 ← '1';
    state ←
    END CASE;
END STATE MACHINE.
```

Example of using VHDL-like State Machine Language.

XC-DS501 XACT Design Implementation System

Step 2

Product Brief

FEATURES

- Complete system for implementing programmable gate array designs into LCA architecture
- Accepts Xilinx netlists created from schematic editors, Boolean equations, or state-machine descriptions
- Automated logic reduction and partitioning removes unused, disabled logic
- Automated placement and routing of logic minimizes design cycle time
- Interactive editor for design optimization
- Point-to-point timing calculations for critical-path analysis
- Demo board for training and trying out concept designs
- Download cable to transfer configuration programs from PC or workstation to LCA in target system
- Available on PC/AT, Apollo, Sun-3, Sun-4 and DECstation computers

DESIGN IMPLEMENTATION PROCESS

Designers often describe portions of their design, such as counters and glue logic, with schematics, and other portions of the design, such as decoders, with Boolean equations. Automatic Placement and Routing software, APR, permits designers to merge multiple modes of design entry into a single design.

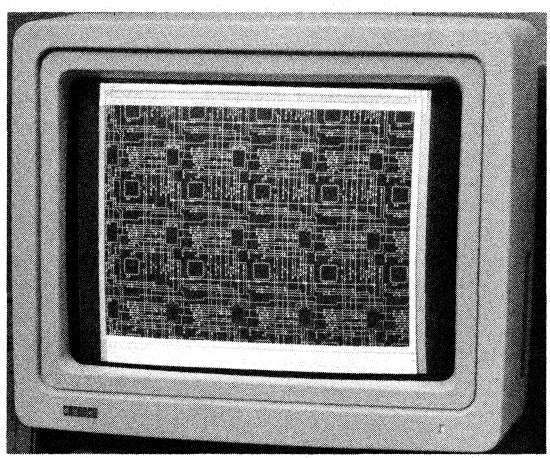
The figure on page 5-17 illustrates the design flow from a design schematic with some glue logic, a 7400-MSI macro, and a PLD symbol. The design files are merged and partitioned into CLBs and IOBs.

The Automatic Placement and Routing software, APR, is very flexible. Routing resources can be specified to eliminate clock skews and minimize routing delays for critical paths.

The XACT Design Editor (XDE), can then be used to modify design placement and routing, when required to meet critical timing requirements. Checks for logic connectivity and design rule violation are easily performed using the XDE. All unused internal nodes are automatically configured to minimize power dissipation.

Interactive point-to-point timing delay calculation is provided for timing analysis and critical-path determination. This ability enables the user to quickly identify and correct timing problems while the design is in progress.

A download cable is included with the DS501. It is useful for transferring configuration programs serially from the PC or workstation to an LCA device installed in a system or on the demo board. During product development and verification this capability can be used to save the time required to write a modified configuration program into an EPROM.



XC-DS22 PC-SILOS* Simulators

Step 3 Option

Product Brief

FEATURES

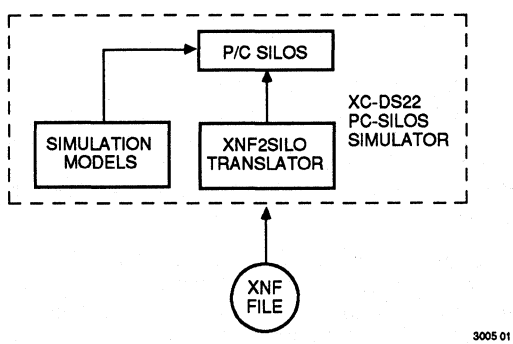
- PC-based simulator for LCA-design verification
- Simulates any LCA design, regardless of design input format (combined logic schematics, Boolean equations, and state machine descriptions)
- General-purpose event-driven logic and timing simulator
- Input automatically generated from XNF file
- Control and observation of any physical circuit node
- Multiple-file input for vectors and commands
- Interactive or batch-mode operation
- Output available in printed or tabular formats
- Simulates logic complexities up to 16,000 gates
- Runs on a PC/AT or compatible personal computer

GENERAL

P/C-SILOS is a powerful PC-based simulator that provides event-driven logic and timing simulation of LCA designs. Simulation is particularly useful for testing designs or design segments as well as for verifying critical timing over worst-case power supply, temperature and process conditions.

Simulation is useful in several stages of the design cycle. After design entry, simulation may be used to debug logic in an unplaced and unrouted design. This saves design time because logic errors can be detected and corrected prior to final placement and routing. After a circuit has been placed, routed and then fully debugged using in-circuit emulation, worst-case timing may be verified. This enables the user to select the correct Logic Cell Array speed grade for a particular application.

Network inputs for LCA designs are automatically created by the XNF2SILO utility. The network includes logic and routing-delay parameters and set-up and hold times based upon the selected speed grade operating under worst-case conditions. Simulation stimuli are created with a set of clock statements or with an input pattern for either pad inputs or internal nodes. Simulation results are available in tabular, plotted and graphic formats. This flexibility makes the "logic debug" easy for both the circuit function and timing.



3005 01

*P/C SILOS is a trademark of SimuCad Corp.

XC-DS290 VIEWlogic VIEWsim Simulator

Step 3 Option

Product Brief

FEATURES

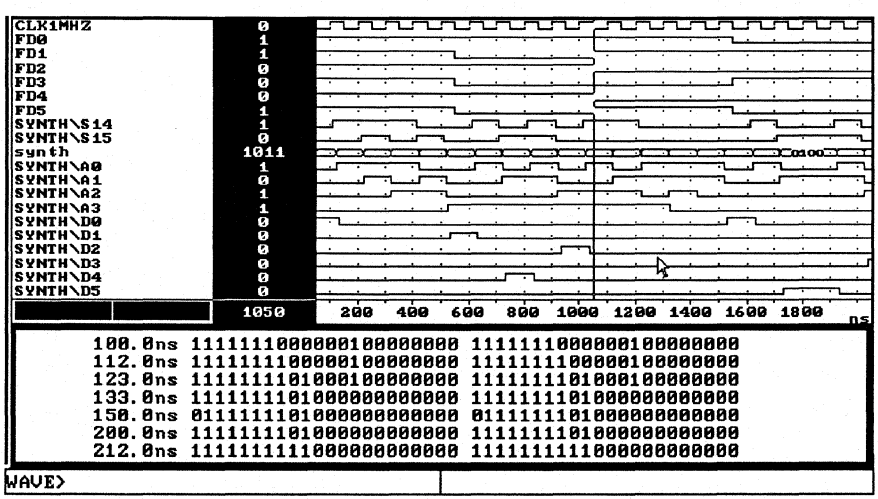
- PC-based simulator for LCA-design verification
- Simulates any LCA design, regardless of design input format (combined logic schematics, Boolean equations, and state machine descriptions)
- General-purpose event-driven logic and timing simulator
- Input automatically generated from XNF file
- Control and observation of any physical circuit node
- Sophisticated waveform display
- Schematic capture and simulation integrated in one environment (when using with VIEWdraw)
- Simulates logic complexities of the largest LCAs
- Runs on an PC/AT or compatible personal computer

GENERAL

VIEWsim is a powerful PC-based simulator that provides event-driven logic and timing simulation of LCA designs. Simulation is particularly useful for testing designs or design segments as well as for verifying critical timing over worst-case power supply, temperature and process conditions.

Simulation is useful in several stages of the design cycle. After design entry, simulation may be used to debug logic in an unplaced and unrouted design. This saves design time because logic errors can be detected and corrected prior to final placement and routing. After a circuit has been placed, routed and then fully debugged using in-circuit emulation, timing simulation may be performed. This enables the user to select the correct LCA speed grade for a particular application.

Network inputs for LCA designs are automatically created by the XNF2WIR utility. The network includes logic and routing-delay parameters and set-up and hold times based upon the selected speed grade operating under worst-case conditions. Simulation stimuli are created with a set of clock statements or with an input pattern for either pad inputs or internal nodes. Simulation results are available in tabular, plotted and graphic formats. This flexibility makes the "logic debug" easy for both the circuit function and timing.



X1252

XC-DS351 OrCAD VST Simulator Interface

Step 3 Option

Product Brief

FEATURES

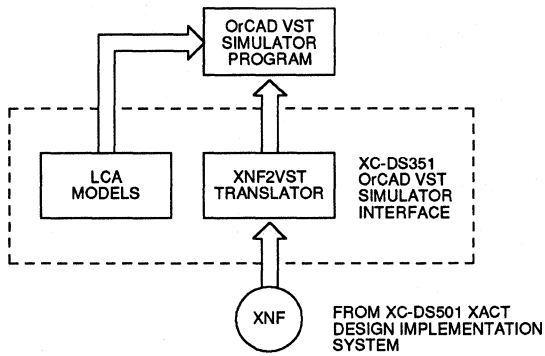
- Model library and netlist translator for users of the OrCAD VST Simulator
- Supports full timing simulation of routed LCA designs, and unit-delay simulation of unrouted designs
- Permits simulation of schematics which include PAL logic defined with PALASM, ABEL, CUPL, Log/IC, or PLDesigner.
- Input compatible with XACT Design Implementation System
- Runs on a PC/AT or compatible personal computer

GENERAL

Simulation is particularly useful for testing designs or design segments as well as for verifying critical timing over worst-case power supply, temperature and process conditions.

Simulation is useful in several stages of the design cycle. After design entry, simulation may be used to debug logic in an unplaced and unrouted design. This saves design time because logic errors can be detected and corrected prior to final placement and routing. After a circuit has been placed, routed and then fully debugged using in-circuit emulation, worst case timing may be verified. This enables the user to select the correct LCA speed grade for a particular application.

Network inputs for LCA designs are automatically created by the XNF2VST utility from the XNF output of the XACT Design Implementation System. The network includes logic and routing delay parameters and setup and hold times based upon the selected speed grade operating under worst case conditions.



1970 01

1970

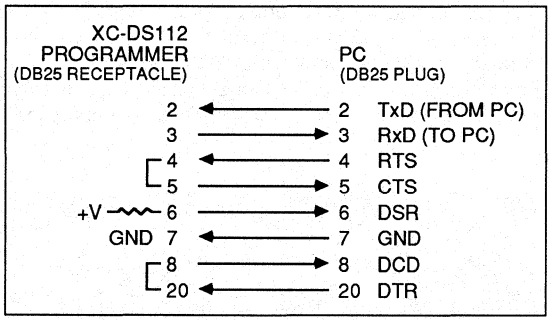
XC-DS112 and XC-DS113 Serial Configuration PROM Programmer and Adapter

Step 3 Option

Product Brief

FEATURES

- Programs XC1736, XC1736A, XC1765 Serial Configuration PROMs
- Connects to serial port of PC/AT or compatibles
- Operates from PC via software provided with program-ming unit
- Accepts HEX-format data files created by the DS501 XACT Design Implementation System
- Supports 8-pin mini-DIPs directly
- Supports 20-pin PLCC packages with optional DS113 adapters
- Runs on PC/AT-compatible personal computers Apollo, Sun-4, Sun-3, DECstation engineering workstations



XC-DS112 Interface to PC

1960 01

GENERAL

When using Xilinx Serial Configuration PROMs to configure programmable gate arrays, the designer can program them with XC-DS112 Configuration PROM Programmer.

The programming unit connects to a serial port of a PC/AT or workstation and is controlled using the software included with the XC-DS112.

Designers compile their LCA designs into a standard HEX format file using the XACT development system. The programming software provided with the XC-DS112 is then used to download the HEX file into the programming unit and to program a serial PROM.

1960

XC-DS28, XC-DS27, XC-DS26 XACTOR™ In-Circuit Design Verifier

Step 3 Option

Product Brief

FEATURES

- Real time in-circuit verification in user's target system
- Concurrent emulation of up to four devices
- Readback and display of LCA internal storage-element states
- Device status display with automatic update of asynchronous events
- Control and I/O pin isolation from target system
- Support for daisy-chain programming of up to seven devices in a daisy chain
- Support for multiple device and package types
- Runs on a PC/AT or compatible personal computer

GENERAL

The XACTOR real-time in-circuit design verifier provides interactive target system emulation of up to four Logic Cell Arrays from the host PC system. In-circuit "debug" provides a powerful productivity enhancement to simulation, providing capabilities to verify functionality in the target system at full speed with all other circuits and system software.

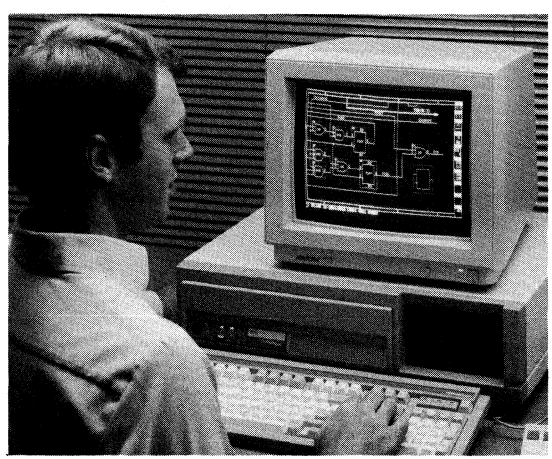
The design verifier is composed of a microcomputer-based controller, and from one to four universal emulation pods, each with an emulation header. The controller is connected to the host PC through a serial port and provides local storage of configuration programs, control of individual device configurations, and control of the isolation of the pod device(s) from the target system. The user can set the state and isolation for each of the control signals to provide debugging of target hardware. Four general I/O pins are available to provide test points which may also be isolated from the target system.

Target Logic Cell Arrays can be programmed individually or in a daisy chain. Daisy chains of up to seven devices may be supported from any of the four pods. Individual device isolation and configuration is controlled with mouse or keyboard commands and may be supplemented with user-defined set-up files for easy system debugging.

Readback of device configuration may be performed on command for verification of the configuration process and interrogation of the internal states. The state of all internal storage elements is displayed after readback has been performed. Status displays showing the state of all isolation switches and control signal states are provided. The status display includes automatic reporting of asynchronous status changes in the target system.

UNIVERSAL IN-CIRCUIT EMULATOR PODS

Additional pods may be connected to the XACTOR controller, up to a maximum of four pods per controller. Pod headers are interchangeable for different device and package types. Each pod provides a direct in-socket connection without disruption of the target system. Test points are provided to allow connection of a logic analyzer or other test equipment to aid in the system debugging.



Xilinx Development System Support Agreements

All Xilinx development systems come with free software support for one year. To receive software updates, the customer must become a registered customer by returning the registration card. Registering your software also places you on the mailing list for XCELL, the Xilinx customer newsletter.

Benefits include:

- Free software updates. Customer will receive new releases of the software programs covered by the agreement. Customer can enjoy enhanced functionalities, performance and bug fixes free of charge. Xilinx typically provides about two new update releases per year that include the necessary diskettes and documentation required to implement each update.
- Toll free applications hot line and local field application engineering support. Customers can receive expert help instantly by calling our application hot-line: (408) 897-5199 or (800) 225-7778 and ask for Applications Engineering.
- Free use of the Xilinx Technical Bulletin Board. To provide customers with up-to-date information and an immediate response to questions, Xilinx provides a 24-hour electronic bulletin board. The customers who are current with their support agreements and have full privileges, can read files on the bulletin board, download those of interest to their own systems or upload files to the bulletin board.
- Free quarterly technical newsletter. This newsletter gives updates on hardware and software availability and revision levels, as well as known software bugs and work-arounds. In addition, application ideas and user tips and a list of relevant magazine articles make this a valuable source of information.

After the first year, the above benefits are available through Xilinx annual software support agreements. The cost of the agreements are typically 12% to 15% of the original price per year.

PADS		XC 3000 # CLBs	XC2000 #CLBs
BPAD	Bidirectional Package Pin Symbol	0	0
IPAD	Input Package Pin Symbol	0	0
OPAD	Output Package Pin Symbol	0	0
UPAD	Unbonded Die Pad Symbol	0	0
IOB SCHEMATIC ELEMENTS			
TBUF	Internal 3-State Buffer	0	-
ACLK	Auxiliary Buffer	0	0
GCLK	Global Buffer	0	0
IBUF	Input Buffer	0	0
INFF	Input Flip-Flop	0	0
INLAT	Input Latch	0	-
OBUF	Output Buffer	0	0
OBUFZ	Output Buffer with Output Enable	0	0
OUTFF	Output Flip-Flop	0	-
OUTFFZ	Output Flip-Flop with OBUFZ	0	-
BUF	Internal non-inverting Buffer	0	0
INV	Inverter	1	1
PULLUP	Input pull-up Resistor	0	-
OINV	Inverting Output Buffer	0	-
GENERAL			
BRLSHFT4	4-Input Barrel Shifter	4	-
BRM	Binary Rate Multiplier	5	-
C3BIT8, 7 ..4	3-Bit Divide-by-8, 7 .. 4 Shift Register Counter	2	-
C5BIT32, 31..9	5-Bit Divide-by-32, 31 ..9 Shift Register Counter	3	-
C3 Square	Divide-by-3 with 50% Duty Cycle	2	-
C5 Square	Divide-by-5 with 50% Duty Cycle	2	-
GADD	1-Bit Full Adder	1	1
GCOMP	2-Bit Comparator	1	1
GLTGT	2-Bit Less Than/Greater Than Comparator	1	-
GEQGT	2-Bit Equal/Greater Than Comparator	1	1
GMUX	2-to-1 Mux	1	1
OSC	Crystal Osc	0	0
GXTL	Crystal Osc (XACT: 3020 GXTL20, 2018 GXTL2)	0	0
GOSC	Low Frequency Resistor-Capacitor Oscillator	1	1
GMAJ	Majority Gate	1	1
GXOR	Exclusive-OR	-	1
GPAR	Parity Test (Even = Low)	-	1

GENERAL (Continued)		XC 3000 # CLBs	XC2000 #CLBs
HX83	4-Bit Binary Adder With Fast Carry	6	-
HX85	4-Bit Magnitude Comparator	7	-
HX280	9-Bit Parity Checker / Generator	3	-
HX283	4-Bit Binary Full Adder6	6	-
HX518	8-Bit Identity Comparator	5	-
HX521	8-Bit Identity Comparator	5	-
HX125	3-State Bus Buffer	0	-
HX240	Octal Inverting Buffer, 3-State Outputs	4	-
HX241	Octal Non-inverting Buffer, 3-State Outputs	1	-
HX244	Octal Non-inverting Buffer, 3-State Outputs	0	-
HX245	Octal Bidirectional Transceiver	1	-
HX540	Octal Inverting, 3-State Outputs	0	-
HX541	Octal Non-inverting, 3-State Outputs	0	-
MCOMP	Magnitude Comparator	4	-
PHFRCOMP	Phase/Frequency Comparator	2	-
SAR	Successive Approximation Register	9	-
LATCHES			
LD	Data Latch	1	1
LDRD	Data Latch with Reset Direct	1	1
LDSD	Data Latch with Set Direct	1	1
LRS	Set-Reset Data Latch with Reset Dominant	1	-
LDM	Data Latch with 2-Input Data Mux	-	1
LDMRD	Data Latch with 2-Input Data Mux with Reset Direct	-	1
LDMSD	Data Latch with 2-Input Data Mux with Set Direct	-	1
LDSRD	Data Latch with Set Direct, Reset Direct	-	1
HX77	2-Bit Latch	1	-
HX259	8-Bit Addressable Latch	8	-
HX373	Octal Latch with 3-State Outputs	4	-
FLIP-FLOPS			
FD	D Flip-Flop	1	1
FDRD	D Flip-Flop with Reset Direct	1	1
FDSD	D Flip-Flop with Set Direct	-	1
FDSRD	D Flip-Flop with Set Direct, Reset Direct	-	
FDC	D Flip-Flop with Clock Enable	1	1
FDCRD	D Flip-Flop with Clock Enable, Reset Direct	1	1
FDCR	D Flip-Flop with Clock Enable, Reset	1	1
FDCS	D Flip-Flop with Clock Enable, Set	1	1
FDR	D Flip-Flop with Reset	1	1
FDS	D Flip-Flop with Set	1	1
FRS	Set-Reset Flip-Flop with Reset Dominant	1	1
FSR	Set-Reset Flip-Flop with Set Dominant	1	1
FDM	D Flip-Flop with 2-Input Data Mux	1	1
FDMRD	D Flip-Flop with 2-Input Data Mux with Reset Direct	1	1
FDMSD	D Flip-Flop with 2-Input Data Mux with Set Direct	-	1

FLIP-FLOPS (Continued)		XC 3000 # CLBs	XC2000 #CLBs
FDMR	D Flip-Flop with 2-Input Data Mux with Reset	1	1
FD\MS	D Flip-Flop with 2-Input Data Mux with Set	1	1
FJK	J-K Flip-Flop	1	1
FJKRD	J-K Flip-Flop with Reset Direct	1	1
FJKSD	J-K Flip-Flop with Set Direct	-	1
FJKSRD	J-K Flip-Flop with Set Direct, Reset Direct	-	1
FJKS	J-K Flip-Flop with Set	1	1
FT0	Self Toggle Flip-Flop	1	1
FT0RD	Self Toggle Flip-Flop with Reset Direct	1	-
FT0R	Self Toggle Flip-Flop with Reset	1	1
FT	Toggle Flip-Flop	1	1
FTRD	Toggle Flip-Flop with Reset Direct	1	1
FTP	Toggle Flip-Flop with Parallel Enable	1	1
FTPRD	Toggle Flip-Flop with Parallel Enable, Reset Direct	1	1
FTR	Toggle Flip-Flop with Reset	1	1
FTS	Toggle Flip-Flop with Set	1	1
FT2	2-Input Toggle Flip-Flop	-	1
FT2R	2-Input Toggle Flip-Flop with Reset	-	1
NDFF	Negative Edge Flip-Flop Primitive	1	1
PDFF	Positive Edge Flip-Flop Primitive	1	1
DECODERS/ENCODERS			
D2-4	1-of-4 Decoder	2	2
D2-4E	1-of-4 Decoder with Enable	2	2
74-139	1-of-4 Single Decoder with Enable, Low Output	2	2
D3-8	1-of-8 Decoder	4	4
D3-8E	1-of-8 Decoder with Enable	4	5
74-138	1-of-8 Decoder with Enables, Low Output	5	6
74-42	1-of-10 Decoder with Low Output	5	7
HX42	4-to-10 Line Decoder	5	-
HX48	BCD to Seven Segment Decoder	5	-
HX138	1-of-8 Decoder/Demultiplexer	5	-
HX139	1-of-4 Decoder	2	-
HX147	10-to-4 Line Priority Encoder	5	-
HX148	3-to-8 Line Priority Encoder	9	-
HX154	1-of-16 Decoder/Demultiplexer	9	-
HX278	4-Bit Cascadable Priority Encoder	6	-
MULTIPLEXERS			
M3-1	3-to-1 Mux	1	2
M3-1E	3-to-1 Mux with Enable	2	2
M4-1	4-to-1 Mux	2	3
M4-1C	4-to-1 Mux	1	-
M4-1E	4-to-1 Mux with Enable	2	3
74-352	4-to-1 Mux with Enable, Low Output	2	3

MULTIPLEXERS (Continued)		XC 3000 # CLBs	XC2000 #CLBs
M4-2	4-to-2 Mux	1	-
M8-1	8-to-1 Mux	4	7
M8-1E	8-to-1 Mux with Enable	4	7
74-151	8-to-1 Mux with Enable, Complementary Outputs	4	7
74-152	8-to-1 Mux with Low Output	4	7
HX151	8 Input Multiplexer	5	-
HX152	8 Input Multiplexer	4	-
HX153	Dual 4 Input Multiplexer	6	-
HX157	Quad 2 Input Multiplexer	4	-
HX158	Quad 2 Input Multiplexer	3	-
HX257	Quad 2-to-1 Multiplexer with Enable	2	-
HX258	Quad 2-to-1 Inverting Multiplexer	2	-
HX352	4-to-1 Data Selector / Multiplexer	5	-
REGISTERS			
Data Registers			
RD4	4-Bit Data Register	2	4
RD4RD	4-Bit Data Register	2	-
RD8	8-Bit Data Register	4	8
RD8RD	8-Bit Data Register with Reset Direct	4	-
RD8CR	8-Bit Data Register with Clock Enable, Reset	4	8
HX174	Hex D Register with Master Reset	4	-
HX273	Octal D Flip-flop	4	-
HX298	Quad 2 Input Flip-flop	4	-
HX374	Octal D Flip-flops with 3-State Outputs	4	-
HX377	Octal D Flip-flops with Clock Enable	4	-
HX577	Octal D Flip-flops with Reset and 3-State Outputs	4	-
Serial to Parallel			
RS4	4-Bit Shift Register	2	4
RS4RD	4-Bit Shift Register with Reset Direct	2	-
RS4C	4-Bit Shift Register with Clock Enable	2	-
RS4CRD	4-Bit Shift Register with Clock Enable, Reset Direct	2	-
RS4CR	4-Bit Shift Register with Clock Enable, Reset	2	-
74-195	4-Bit Serial to Parallel SR with ParEna, MRLow	3	5
74-194	4-Bit Bi-Directional SR with ClkEna, ParEna, MRLow	5	12
RS8	8-Bit Shift Register	4	8
RS8RD	8-Bit Shift Register with Reset Direct	4	-
RS8R	8-Bit Shift Register with Reset	4	8
RS8C	8-Bit Shift Register with Clock Enable	4	-
RS8CRD	8-Bit Shift Register with Clock Enable, Reset Direct	4	-
RS8CR	8-Bit Shift Register with Clock Enable, Reset	4	8
RS8PR	8-Bit Shift Register with Parallel Enable, Reset	4	8
74-164	8-Bit Serial to Parallel SR with Master Reset Low	5	8
HX164	8-Bit Serial In-Parallel Out Shift Register	5	-
HX166	Parallel Load 8-Bit Shift Register	6	-

REGISTERS (Continued)		XC 3000 # CLBs	XC2000 #CLBs
HX179	4-Bit Parallel Access Shift Register	5	-
HX194	4-Bit Bidirectional Universal Shift Register	7	-
HX195	4-Bit Parallel Access Shift Register	3	-
HX198	8-Bit Bidirectional Shift Register	14	-
HX199	8-Bit Shift Register with Clock Inhibit	7	-
HX595	8-Bit Shift Register with 3-State Register Output	9	-
COUNTERS			
Modulo 2			
C2BCP	1-Bit Binary Counter w/ Clock Enable, Parallel Enable	1	-
C2BCPRD	1-Bit Binary Counter w/ ClkEna, ParEna, Reset Direct	1	-
C2BCR	1-Bit Binary Counter with Clock Enable, Reset	1	1
C2BCRD	1-Bit Binary Counter with Clock Enable, Reset Direct	1	1
C2BP	1-Bit Binary Counter with Parallel Enable	1	1
C2BR	1-Bit Binary Counter with Reset	1	1
C2BRD	1-Bit Binary Counter with Reset Direct	1	1
Modulo 4			
C4BCP	2-Bit Binary Counter with Clock Enable, Parallel Enable	2	3
C4BCPRD	2-Bit Binary Counter w/ ClkEna, ParEna, Reset Direct	2	-
C4BCR	2-Bit Binary Counter with Clock Enable, Reset	2	2
C4BCRD	2-Bit Binary Counter with Clock Enable, Reset Direct	2	2
C4JX	2-Bit Expandable Johnson Counter	1	-
C4JXRD	2-Bit Expandable Johnson Counter with Reset Direct	1	-
C4JXC	2-Bit Expandable Johnson Counter with Clock Enable	1	-
C4JXCRD	2-Bit Expandable Johnson Cntr w/ClkEna, Reset Dir	1	-
C4JXCR	2-Bit Expandable Johnson Counter with ClkEna, Reset	1	-
C4JCR	2-Bit Johnson Counter with Clock Enable, Reset	-	2
Modulo 6			
C6JCR	3-Bit Johnson Counter with Clock Enable, Reset	2	3
Modulo 8			
C8BCP	3-Bit Binary Counter w/ ClkEna, Parallel Enable	3	5
C8BCPRD	3-Bit Binary Counter w/ ClkEna, ParEna, Reset Dir	3	-
C8BCR	3-Bit Binary Counter with Clock Enable, Reset	3	4
C8BCRD	3-Bit Binary Counter with Clock Enable, Reset Direct	2	4
C8JCR	4-Bit Johnson Counter with Clock Enable, Reset	2	4
Modulo 10			
C10BCRD	4-Bit BCD Counter with Clock Enable, ResetDir	3	4
C10BCPRD	4-Bit BCD Counter with Parallel Enable, ResetDir	4	7
74-160	4-Bit BCD Counter w/ ClkEna, ParEnaL, MRLow	6	8
74-162	4-Bit BCD Counter w/ ClkEna, ParEnaL, Reset Low	7	-
C10BPRD	4-Bit BCD Counter w/ Parallel Enable, Reset Direct	4	6

COUNTERS (Continued)		XC 3000 # CLBs	XC2000 #CLBs
C10JCR	5-Bit Johnson Counter with Clock Enable, Reset	3	5
HX160	Presettable Decade Counter	8	-
HX162	Presettable Decade Counter with Sync Clear	10	-
HX168	4-Bit BCD Synchronous Up/Down Counter	11	-
HX390	4-Bit Decade Counters with Clear	3	-
Modulo 12			
C12JCR	6-Bit Johnson Counter with Clock Enable, Reset	3	6
Modulo 16			
C16BARD	4-Bit Binary Ripple Counter with Reset Direct	2	4
C16BCRD	4-Bit Binary Counter w/ ClkEna, Reset Direct	3	4
C16BCP	4-Bit Binary Counter w/ ClkEna, Parallel Enable	5	-
C16BCPRD	4-Bit Binary Counter w/ ClkEna, ParEna, Reset Direct	5	6
74-161	4-Bit Binary Counter w/ ClkEna, ParEna, MRLow	6	8
C16BCPR	4-Bit Binary Counter w/ Clock Enable, ParEna, Reset	6	10
74-163	4-Bit Binary Counter w/ ClkEna, ParEnaL, Reset Low	7	-
C16BPRD	4-Bit Binary Counter w/ Parallel Enable, Reset Direct	4	5
C16BUDRD	4-Bit Binary Up-Down Cntr w/ ParEna, ResetDir	5	8
C16JCR	8-Bit Johnson Counter with Clock Enable, Reset	4	8
HX161	Presettable Binary Counter	6	-
HX163	Synchronous Binary Counter with Sync Clear	8	-
HX169	4-Bit Binary Synchronous Up/Down Counter	7	-
HX393	4-Bit Binary Counters with Clear	5	-
HX590	8-Bit Counter with Register and 3-State Output	13	-
Modulo 256			
C256BCRD	8-Bit Binary Counter with Clock Enable, Reset Direct	7	-
C256BCR	8-Bit Binary Counter with Clock Enable, Reset	7	-
C256BCP	8-Bit Binary Counter w/ ClkEna, Parallel Enable	8	-
C256BCPRD	8-Bit Counter w/ ClkEna, ParEna, Reset Direct	8	-
C256FCRD	8-Bit Mod 256 Feedback SR w/ ClkEna, ResetDir	6	9
C8UDLD	8-Bit Loadable Up/Down Counter	9	-
C16UDLD	16-Bit Loadable Up/Down Counter	18	-

Development System Hardware Requirements

Xilinx provides an integrated Development System for design and implementation of LCA devices. The XACT Development System operates on an PC/AT or PS/2 model 60 or 80, Apollo, Sun-3, Sun-4, and DECstation 3100 and provides a range of support features. This provides the user with an effective, convenient, low-risk method of logic-design entry, simulation, LCA generation and verification for single-chip logic designs of up to 9000 gates. In addition, several popular PC and workstation CAE vendors have developed and offer design-entry and simulation programs compatible with the XACT Development System

PC REQUIREMENTS

The recommended PC-system configuration needed to run the XC2000 and XC3000 Series XACT Development System consists of:

- A 20-MHz "386" PC/AT or PS/2 model 60 or better
- 40 M byte hard disk drive plus a 1.2 Mbyte high-density floppy disk drive
- Two RS-232-C serial ports
- One parallel port
- EGA or VGA Graphics Display
- Mouse
- MS-DOS version 3.0 or higher
- IBM-compatible BIOS and keyboard
- A math co-processor can enhance performance of APR by 10 or 20%
- Extended memor as follows.

LCA Gates	Total Memory Required for XACT 3.0
2000 or less	2.50 Mbytes
3000	3.25 Mbytes
4200	4.00 Mbytes
6400	5.25 Mbytes
9000	6.50 Mbytes

To assure integrity, all Xilinx software is tested on IBM systems and several compatible systems. LCA development software includes some of the first DOS-based programs to make extensive use of the "protected" mode of the processor. This has exposed protected mode IBM-incompatibilities of some clones, usually in the BIOS or Keyboard Controller. Xilinx software includes system exercises called PMTEST and PMINFO to help test IBM compatibility and measure relative performance.
Note that this amount of memory must be available to XACT, i.e., it does not include the memory used for other resident programs.

Note also that the Compaq 386 has only 640 Kbytes of its first megabyte available to any user.

PC I/O Ports

The LCA Development System requires several I/O ports. A parallel port is needed for the software execution protection key. The key must be in place to allow Xilinx software to execute but is virtually transparent, and the port can be used simultaneously for a parallel printer or the Xilinx download cable. Several printer types are supported for text or graphic hard copy. Serial COM ports are used for a mouse, the XACTOR In-Circuit Design Verifier and the Configuration PROM Programmer.

PC Mouse

The Xilinx Development System programs are compatible with several varieties of mice offered for the PC. These include Mouse Systems PC Mouse (no device driver required), Microsoft (serial or parallel), LogiTech C7 and the FutureNet mouse. The Xilinx software supports any mouse directly that emulates the PC mouse or has a device driver that provides Microsoft compatibility and defines the PC COM port.

Please note however, only the Mouse System M4 and the LogiTech C7 will work with the VIEWlogic software, VIEWdraw-LCA and VIEWsim.

1164

PC Setup

When the system is powered up it uses commands from the DOS CONFIG.SYS file to install selected device driver-programs (such as Mouse driver) in memory and define buffer and file sizes. Exampls of these statements are:

```
device=C:\lib\msmouse.sys /1
files=10
buffers=20
```

After CONFIG.SYS functions are implemented the system executes the commands found in the AUTOEXEC.BAT file. This file contains DOS commands such s:

```
path=c:\; … c:\xact;c:\dash-lca; …
set xact=c:\xact
set grmode=ega
set swmode=9
set minbytes=65000
```

The first line shows the portion of the path established by the XACT and DASH-LCA installation procedures. These are the default directories created and used in the Xilinx installation procedures. The `SET SWMODE=` sets a parameter defining one of several alternative ways of switching the processor from protected to real mode. Several alternatives are made available in order to accommodate various "clone" idiosyncrasies. Possible values are 9 (default), 10, 7, 4, and for 80386 based systems, 3.

See the Xilinx installation instructions and PC manuals for additional information.

WORKSTATION REQUIREMENTS

The workstation system requirements needed to run Xilinx software are:

Apollo Requirements

DN4000 Series

- Apollo Operating System SR10.1
- Mentor Graphics Version 7.0
- 60 Mbytes allocated for Xilinx designs
- 16 Mbyes of RAM
- Color Monitor
- DOMAIN XII VI

Sun-3 Requirements

Series 960 and above

- Sun Operating System OS4.0
- 60 Mbytes allocated for Xilinx designs
- 16 Mbytes of RAM
- Color Monitor
- X-Windows

Sun-4 Requirements

Sun-4 & SparcStation Series

- Sun Operating System OS4.0
- 60 Mbytes allocated for Xilinx designs
- 16 Mbytes of RAM
- Color Monitor
- X-Windows

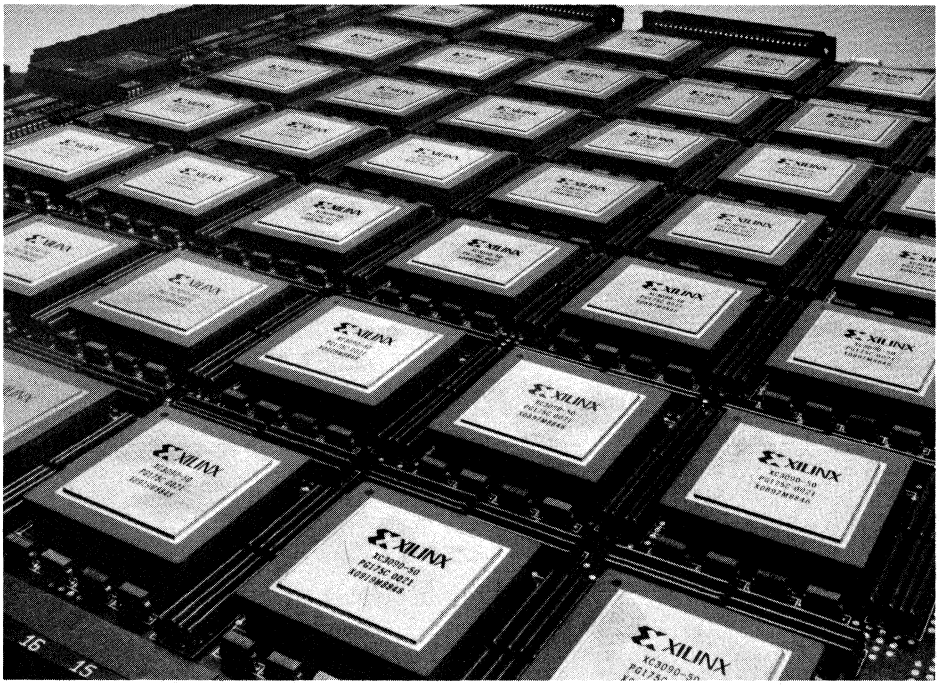
DECStation Requirements

DECstation 3100 Series

- Worksystem V2.2
- 60 Mbytes allocated for Xilinx designs
- 16 Mbytes of RAM
- Color Monitor

XILINX

The Programmable Gate Array Company



XILINX

Applications

Applications

INTRODUCTION

The following pages show examples of systems and subsystems solutions using Xilinx Field Programmable Gate Arrays. Some of these designs have been implemented, a few are in production, but most are conceptual designs. These are intended to demonstrate the device capabilities, to highlight special advantage, to emphasize the best design methods, and in general to stimulate the designer's imagination.

A Logic Cell Array (LCA) can implement virtually any digital design. Xilinx offers a software package that covers the gamut from schematic capture through logic optimization to automatic place and route and to the generation of a programming bit stream. The designer can use these tools and achieve a working LCA design while paying very little attention to the architectural details of the Logic Cell Array.

Such an approach, however, will not always achieve the highest possible performance and the lowest possible cost. For specific, well-structured designs it may pay to work out a good match with the LCA architecture. This chapter gives several examples of such solutions. The XC2000 and XC3000 families of LCA devices have inherent features different from those of LSTTL MSI circuits, or PAL devices, or conventional gate arrays. These four technologies all have different structures, which lead to different strengths and weaknesses when they are being used to implement any specific type of logic.

TTL-MSI was originally defined to fit into a 16-pin package and to provide maximum flexibility, so that each standard part could be used in a myriad of applications. Some functions are therefore overdesigned (counters and shift registers have parallel inputs and outputs, when few applications need both) and some are crippled by the 16-pin limitation (notably the up-down counters).

PAL devices suffer from the rigidity of the AND-OR architecture and from the fixed assignment of flip-flops to output pins. While the number of inputs is generous, ideal for wide decoding, the limited number of product terms that can be ORed together makes many designs inefficient and slow. The number of flip-flops available in PALs is very limited.

Gate Arrays offer maximum flexibility and a high level of integration, but burden the user with high risk, high cost, and a long delay from finished design to working prototype. Generating test vectors and worrying about testability is another price the gate array user has to pay.

Field Programmable Gate Arrays offer a very large number of flip-flops (128 in the CLBs and another 128 in the IOBs of the XC3020, a total of 928 in the XC3090). Unlike the situation with gate arrays, these LCA flip-flops cannot, or need not be, traded off against logic. Combinatorial logic coexists with the flip-flops in the form of function generators. The function generators are surprisingly versatile pieces of logic, unlimited in their flexibility, limited only by their fan-in of four or five signals.

When the logic has five inputs or less and is interspersed with flip-flops driven by a common clock, the LCA devices are extremely efficient. Certain high fan-in functions like ALUs tend to be less efficient, and bus-oriented designs must be routed carefully to utilize the long lines of the 3-state drivers of the XC3000-series.

Fortunately, the user normally has some freedom in structuring the system design. Whenever possible, this freedom should be used to improve either the performance or the efficiency of the implementation.

GENERAL TOPICS

Most designers want to estimate density and performance before they begin an LCA design, and some want to know the definition of equivalent gates. While the data sheets provide worst-case guaranteed parameters, many designers need additional information about input and output characteristics, power consumption, crystal oscillator design, and the exact interpretation or certain ac parameters. CLB flip-flops show excellent recovery from metastable problems, an important concern with asynchronous interfaces.

COMBINATORIAL FUNCTIONS

The 5-input function generator of the XC3000 family CLBs offers unlimited flexibility to implement any one of the more than 4 billion (2^{32}) possible functions of up to five variables in one CLB, all with the same combinatorial delay. The 4-input function generator in the XC2000 family can implement any one of the 64K (2^{16}) possible functions of four variables. The logic designer should take advantage of this flexibility while avoiding the possible speed penalty imposed by the limitation to only five or four inputs. This may lead to logic partitioning that is different from traditional design or from MSI or PAL implementation.

Majority logic is just one example in which the CLB excels: A 5-input majority function would use 29 gates when implemented with 2-input NANDs and inverters, but it fits into the combinatorial portion of one XC3000-series CLB.

Address decoding is the classical strength of PAL devices. It is done efficiently in LCA devices if the complete function includes the combination of several addresses or groups of addresses.

ALUs consume many LCA-device resources, but adders or subtractors can be implemented quite efficiently, even using carry-look-ahead for functions that exceed a width of eight bits.

SEQUENTIAL FUNCTIONS

LCA devices offer an abundance of flip-flops, from 119 in the XC2064 to 928 in the XC3090. Each CLB flip-flop (64 in the XC2064, 128 in the XC3020, 640 in the XC3090) has a "free" combinatorial function generator available as its input. This simplifies the design of shift registers and counters.

The "Corner Bender" serial-parallel or parallel-serial converter design, is a two-dimensional shift register array that fits very efficiently into an XC2064 or half of an XC3020, with 100% utilization of the CLB flip-flops.

Using the fast flip-flops and distributed logic in the LCA to their best advantages, a synchronous presettable counter of arbitrary length has been demonstrated to run at 40 MHz. This is much faster than any available popular microprocessor peripheral counter/timer.

State-machine design is another example in which the creative use of CLB resources can result in a straightforward and easily understood solution.

As explained in the beginning of this chapter, the CLB flip-flops are "metastable-resistant;" they resolve metastable situations typically within a few nanoseconds. Designers are nevertheless encouraged to avoid asynchronous designs whenever possible. The combination of very fast CLB flip-flops with relatively slow and layout-dependent interconnects can lead to internal decoding spikes and glitches that cannot be observed with an oscilloscope. However, they can play havoc with internal asynchronous logic. The high-speed, low-skew global clock lines and the individual Clock Enable inputs on each CLB favor synchronous design approaches that are inherently safer and more predictable.

SYSTEM DESCRIPTIONS

LCA devices are universal programming building blocks that are used in a wide variety of systems. An 8-digit frequency counter implemented in a XC2064 is a simple illustration. A PS/2 Micro Channel Controller and a DRAM Controller/Error Corrector demonstrate the versatility of the LCA in speed-critical applications. Some of the designs are available, as indicated, from Xilinx and may be obtained by calling the applications hot line.

The purpose of this applications chapter is not to provide cookbook solutions, but rather to stimulate the imagination, convey ideas and demonstrate that LCA devices offer a better solution for a large variety of digital designs.

XILINX

Estimating Size and Performance

BY DAVE LAUTZENHEISER

INTRODUCTION

Fileld Programmable Gate Arrays are available in a range of densities and speed grades. Before committing resources to design implementation, the user should make an estimate to determine which FPGA best fits the specific application. Size and performance estimates cannot be expected to provide exact details, but they provide useful guidelines for device selection and cost estimates. A complete design is always the final test for both density and performance.

Design-fit estimates can be done in two steps. The first is a quick I/O and storage element count, with no regard for performance. The second step counts logic blocks based on details of the intended circuit, and includes gross performance estimates, still without regard for routing delays. Performance estimates should always be considered "best-case," recognizing that actual system performance can only be verified on a completed design.

STEP 1: I/O and Storage Element Fit

A quick initial estimate of how a system fits a specific LCA device can be made by counting the required input and output pins and internal storage elements. Table 1 lists the Xilinx XC2000- and XC3000-series Logic Cell Array devices and their respective I/O and storage element counts. To estimate a fit, first count the required inputs and outputs and compare the total with the I/O pin count of the desired device. If the desired functions require more I/O than listed for a device, the designer must either select a larger device or package, or reduce the I/O requirements.

Device	Maximum I/O	Logic Block Storage	I/O Block Storage
XC2064	58	58	58
XC2018	74	100	100
XC3020	64	128	128
XC3030	80	200	160
XC3042	96	288	192
XC3064	120	448	240
XC3090	144	640	288

Table 1. I/O and Storage Element Summary

If the desired LCA device has enough I/O pins, the next step is to count the required storage elements. Table 1 shows both logic-block storage elements and I/O-block storage elements. Logic-block storage elements should be considered first, since they are the most flexible. If the required number of storage elements is less than the number of logic storage elements, the desired functions can probably be performed in the chosen LCA device.

In some cases, the I/O-block storage elements can also be used to meet storage-element requirements. In particular, if the number of additional storage elements required beyond the available logic storage elements is less than the number of unused I/O pins, then the desired functions may still fit into the chosen device.

The following two examples illustrate the Step One quick estimation procedure:

Example 1. An 8-bit microprocessor peripheral.

Function	I/O requirements
8-bit data bus	8
5 bus-control signals	5
16 bits of output	16
4 bits of output control	4
2 internal control registers	–
Interrupt control logic	–
TOTAL	33

Even the smallest Logic Cell Array, the XC2064, passes the I/O test. It has 58 user I/O in its 68-pin PLCC package.

Function	Storage Elements
Control registers (assume 8 bits)	16
Buffered input shift register	16
Miscellaneous control logic	10
TOTAL	42

All of the storage elements can be put into logic storage in the XC2064. The XC2064 should fit this application, provided the desired performance can be achieved.

Example 2. A memory controller for a 32-bit high performance processor.

Function	I/O Requirements
32-bit processor data bus	32
32-bit processor memory bus	32
32-bit memory bus	16 (muxed)
32-bit control register	–
32-bit DMA control	–
Address multiplexing control	–
RAS/CAS/Refresh generation	3
Memory error check and correct	–
Processor and memory timing	10
TOTAL	93

Based on this I/O count, the XC3042 with 96 pins would be marginal. An XC3064 with up to 120 I/O pins may be required.

Function	Storage Elements
32-bit DMA (two 32-bit Counters)	64
Refresh generation (minimum)	10
32-bit control register	32
32-bit processor memory address	32
Error check and correct	44
Miscellaneous control	20
TOTAL	202

With two storage elements per logic block, the XC3042 can provide up to 288 storage elements. Based on this estimate, the desired functions should fit into the device. Some caution is indicated for two reasons. First, the I/O count is very near the limit of the device. This could cause some routing congestion in the I/O area, making a higher pin-count device a better choice. Second, high performance requires making the best use of device features. The 32-bit bus may impose critical performance requirements. Only the XC3064 and XC3090 permit a 32-bit internal bus, based on the number of available Long Lines. Choosing the XC3064 could address the I/O requirements as well as the performance needs.

STEP 2: Logic Block Requirements

After establishing design fit by counting I/O and storage elements, it may be necessary to make a more detailed analysis of the blocks required. The macro-library summary table in the Development System section of this data book may be used to determine specific CLB counts for each function to be implemented.

The macro list shows the various gates and functions available with each design library. Each entry in the list includes the required number of logic blocks to implement that function. The differences between XC2000 and XC3000 family block counts are noted. To develop a rough block count, the designer simply tabulates all of the blocks required by each of the functional elements in the design. Figure 1 shows a portion of a schematic and the block count from the macro list.

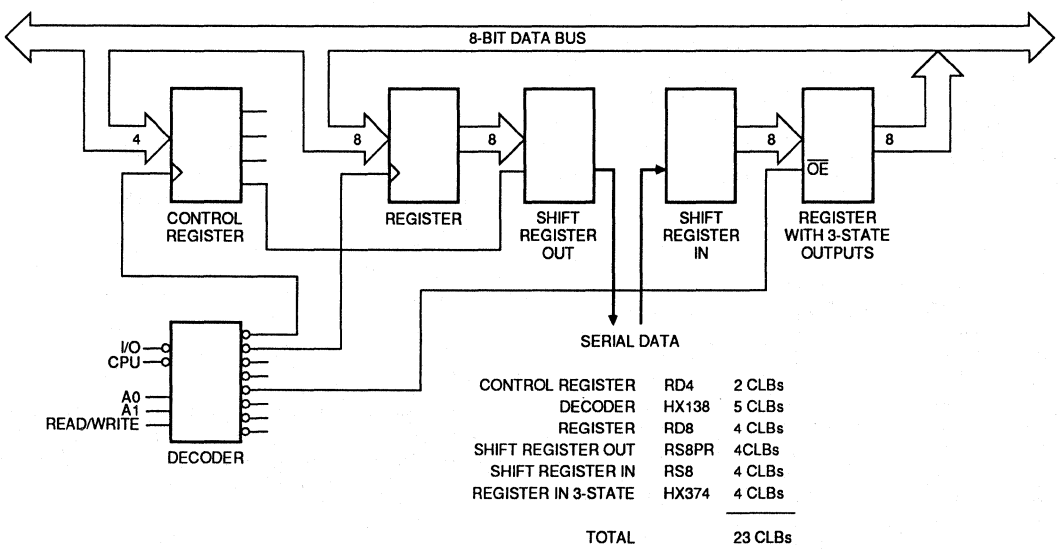


Figure 1. Portion of Schematic with Block Count

1133 01A

In many schematics there are collections of random gates that need to be considered, along with the higher level functions such as counters, decoders and multiplexers. The following technique can be used to estimate the logic blocks required for random logic. Begin at an output point and move back along the path collecting gates until the number of inputs is four for XC2000-family devices, or five for XC3000-family devices. These gates can be marked in some way to show that they occupy a single logic block. Blocks identified by this method are added to the block count from the macro list analysis. Figure 2 shows an example of this gate-collecting technique.

Estimating the block count for integrating PLD devices is more difficult. Each PLD output should be counted as at least one block. PLD devices using five or fewer of the inputs, will require only one block per output for the XC3000 family (four inputs for the XC2000 family). For complex equations using more than five (or four) inputs, a conservative estimate is to use three blocks per output pin.

Decisions about the appropriate device can be reviewed as more information is collected. Block count estimates which are near the limit of a device, either in block count or in I/O and storage element count, may suggest use of the next higher density device.

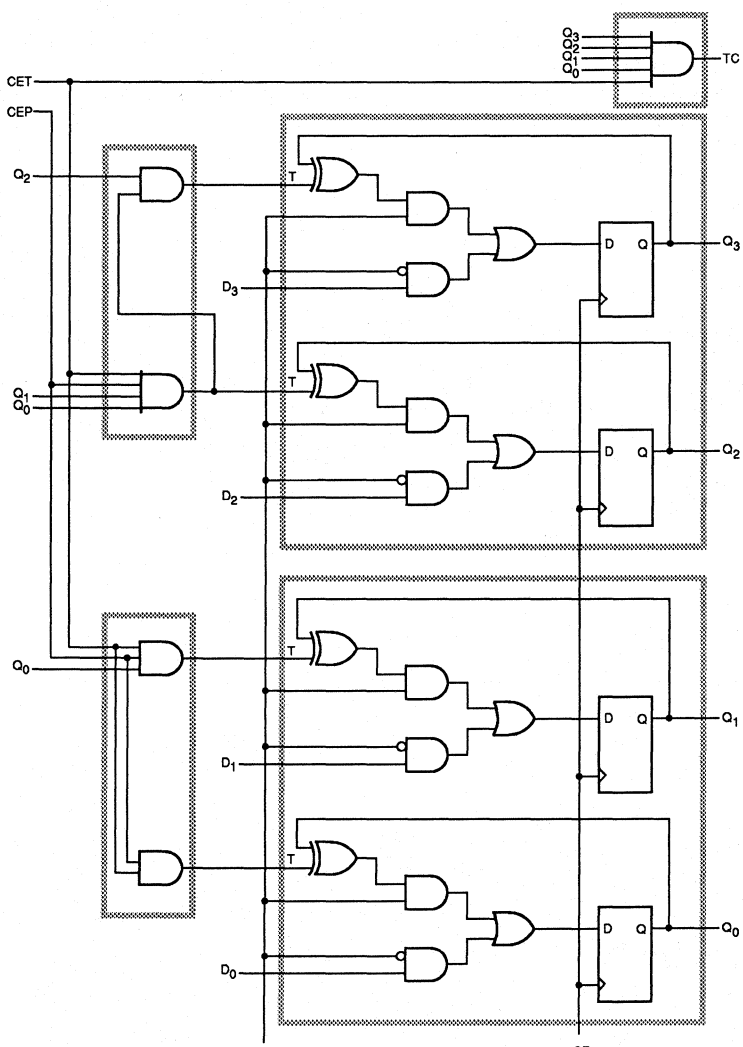


Figure 2. Five CLBs Are Required to Implement a 74161 Binary Counter

ESTIMATING PERFORMANCE

After selecting the right LCA device based on logic resources, an estimation of performance is often the next step. If the system clock rate is less than 20% of the flip-flop toggle rate of the selected device, then the performance goals can usually be met easily. In cases of higher system clock rates or very complex functions, a more detailed analysis may be required.

The macro library for each device family includes the number of logic-block levels used for each listed function; the LCA data sheet specifies the block delay for each level. Some routing delay, which can add 25 - 50%, must be added to the block delay.

As an example, a circuit might have three levels of blocks in the path from one clock edge to another. For a device with 10 ns block delays, this gives 30 ns delay from the first clock to the setup required for the next clock. Allowing 30% for routing (10 ns) and 8 ns for setup gives a total delay of 48 ns. This should permit operation at a system clock rate of up to 20 MHz.

SUMMARY

The final determination whether a logic device meets the goals for integration and performance can come only after the design has been completed. For Field Programmable Gate Arrays, estimating logic capacity and performance should precede device selection. If the design fits, the XACT development system and the simplicity of in-system design verification assures cost-effective and rapid design implementation.

Of course, specifications sometimes change during execution of a design. Logic changes may result in different requirements for I/O and logic blocks. In such cases, the Xilinx product line simplifies the migration to a compatible array that meets the new requirements.

Designing with the XC3000 Family

Application Brief BY THOMAS WAUGH

CLOCKING

Global and Alternate Clocks Buffers

There are two high-fan-out, low-skew clock resources. The global clock originates from the GCLK buffer in the upper left corner of the chip and the alternate clock originates from the ACLK buffer in the lower right corner of the chip.

These resources drive nothing but the K pins (clock pins) of every register in the device. They **cannot** drive logic inputs. In the rare case where this connection is required, tap a signal off the input to the clock buffer and route it to the logic inputs.

The global and alternate clocks each have fast CMOS inputs, called TCLKIN and BCLKIN respectively. Using these inputs provides the fastest path from the PC board to internal flip-flops and latches because the signal by-passes the input buffer. CMOS levels on the input clock signal must be guaranteed.

To specify the use of TCLKIN or BCLKIN in a schematic, connect an IPAD symbol directly to a GCLK or ACLK symbol. Placing an IBUF between the IPAD and GCLK or ACLK will prevent the TCLKIN and BCLKIN from being used.

Always use GCLK and ACLK for the highest fan-out clocks.

I/O Clocks

There are a total of eight different I/O clocks, two per edge on each of the four edges.

I/O storage elements can be configured to be latches or flip-flops. Clocking polarity is programmable per clock line, not per IOB. A clock line that triggers a flip-flop on the rising edge can be an active Low Latch Enable (latch transparent) and vice versa.

Crystal Oscillator

Connects to alternate clock buffer, ACLK, not to GCLK.

3-STATE BUFFERS

Active High 3-state is the same as active Low enable.

In other words: A one on the T pin of a TBUF or an OBUFZ 3-states the output, and a zero enables it.

Input/Output Blocks (IOBs)

Unused IOBs should be left unconfigured. They default to inputs pulled High with an internal resistor.

IOB pull-up resistors **cannot** be used with IOB outputs, only on pins that are inputs exclusively.

Configurable Logic Blocks (CLBs)

CLBs have two flip-flops (not latches). They share a common clock, a common reset, and a common clock-enable signal.

Asynchronous preset can be achieved by the asynchronous reset, by just inverting D and Q of the flip-flops.

ROUTING RESOURCES

Horizontal Long Lines

The number of Horizontal Long Lines (HLL) per device is double the number of rows of CLBs.

The number of TBUFs that drive each Horizontal Long Line is one higher than the number of columns on the device.

Part Name	Rows x Columns	CLBs	HLL	TBUFs per HLL
3020	8 x 8	64	16	9
3030	10 x 10	100	20	11
3042	12 x 12	144	24	13
3064	16 x 14	224	32	15
3090	20 x 16	320	40	17

Continued at the bottom of next page

Designing with the XC2000 Family

Application Brief BY THOMAS WAUGH

CLOCKING

Global and Alternate Clocks Buffers

There are two high-fan-out, low-skew clock resources. The global clock originates from the GCLK buffer in the upper left corner of the chip and the alternate clock originates from the ACLK buffer in the lower right corner of the chip.

The global clock buffer, GCLK, drives the B and K pins of the Configurable Logic Block (CLB).

The alternate clock buffer, ACLK, drives the B, C, and K pins of the Configurable Logic Block (CLB). The crystal oscillator drives the ACLK.

Always use GCLK and ACLK for the highest fan-out clocks.

I/O Clocks

There are four different I/O clocks, one per edge.

I/O flip-flops are positive-edged triggered.

INPUT/OUTPUT BLOCKS (IOBS)

Unconfigured IOB outputs must not be left floating. Configure them as outputs and drive them from internal logic or leave them unconfigured and pull them up with an external resistor.

CONFIGURABLE LOGIC BLOCKS (CLBS)

CLBs have one storage element that can be configured as a flip-flop or a latch.

CLB storage elements have both an asynchronous set and an asynchronous reset.

ROUTING RESOURCES

Horizontal Long Lines

There is one Horizontal Long Line per routing channel.

There are no internal 3-state buffers on the chip.

Vertical Long Lines

There are three Vertical Long Lines per routing channel, one general purpose, one for the global clock net and one for the alternate clock net.

CLB pins with Direct Access to Long Lines

- A- Horizontal Long Line above the CLB.
- B- Global clock buffer, Middle and Left Vertical Long Line.
- C- Middle and Left Vertical Long Line.
- D- Horizontal Long Line below the CLB.
- X- To Left Vertical Long Line.
- Y- To Middle Vertical Long Line.

Designing with the XC3000 Family (Continued)

T and I pins of TBUFs have limited interconnect resources.

Never use fewer than four TBUFs per Horizontal Long Line. When using TBUFs for multiplexing applications, using fewer than four wastes resources. Use CLBs for multiplexing instead.

Vertical Long Lines

There are four Vertical Long Lines per routing channel, two general purpose, one for the global clock net and one for the alternate clock net.

CLB Pins with Direct Access to Long Lines

- A- Lower Horizontal Long Line.
- EC- Left Middle Vertical Long Line.
- B- Left Middle Vertical Long Line.
- C- Right Middle Vertical Long Line
- K- Rightmost and Leftmost Vertical Long Lines (ACLK and GCLK).
- E- Right Middle Vertical Long Line.
- D- Upper Horizontal Long Line.
- RD- Left Middle Vertical and Lower Horizontal Long line.

1976

Additional Electrical Parameters

Application Brief

The LCA data sheets specify worst-case device parameters, 100% tested in production and guaranteed over the full range of supply voltage and temperature.

Some users may be interested in additional data that is not 100% tested and, therefore, not guaranteed. Here are results from recent bench measurements:

PULL-UP RESISTOR VALUES

IOB Pull-ups	40 to 150 kΩ
DONE Pull-up	2 to 8 kΩ
Long Line Pull-up (each)	3 to 10 kΩ

INPUTS

Hysteresis

All inputs, except PWRDN, and XTL2 when configured as the crystal oscillator input, have limited hysteresis, typically in excess of 200 mV for TTL input thresholds, in excess of 100 mV for CMOS thresholds.

Required Input Rise and Fall Times

For unambiguous operation, the input rise time should not exceed 200 ns; the input fall time should not exceed 80 ns.

These values were established through a worst-case test with internal ring oscillators driving all I/O pins except two, thus generating a maximum of on-chip noise. One of the remaining I/O pins was then tested as an input for single-edge response, the other one was the output monitoring the response. This specification may, therefore, be overly pessimistic, but, on the other hand, it assumes negligible PC board ground noise and good V_{CC} decoupling.

OUTPUTS

All XC2000/3000 LCA outputs are true CMOS with n-channel transistors pulling down, p-channel transistors pulling up. Unloaded, these outputs pull rail-to-rail.

DC Parameters	
Output Impedance	
Sinking, near ground:	25 Ω
Sourcing, near V_{CC}:	50 Ω
Output Short Circuit Current	
Sinking current by the LCA	96 mA
Sourcing current by the LCA	60 mA

The data sheets guarantee the outputs only for 4 mA at 320 mV in order to avoid problems when many outputs are sinking current simultaneously.

AC Parameters	**Fast**[1]	**Slow**[1]
Unloaded Output Slew Rate	2.8 V/ns	0.5 V/ns
Unloaded Transition Time	1.45 ns	7.9 ns
Additional rise time for 812 pF	100 ns	100 ns
normalized	0.12 ns/pF	0.12 ns/pF
Additional fall time for 812 pF	50 ns	64 ns
normalized	0.06 ns/pF	0.08 ns/pF

[1] "Fast" and "Slow" refer to the output programming option.

There is good agreement between output impedance and loaded output rise and fall time, since the rise and fall time is slightly longer than two time constants.

POWER DISSIPATION

LCA power dissipation is largely dynamic, due to the charging and discharging of internal capacitances. The dynamic power, expressed in mW per MHz of actual node or line activity is given below.

Clock line frequency is easy to specify, but the designer will usually have great difficulty estimating the average frequency on other nodes.

Two extreme cases are:

1. Binary counter, where half the total power is dissipated in the first flip-flop.
2. A shift register with alternating zeros and ones, where the whole circuit is excercised at the clocking speed.

	Dynamic Power (mW/MHz)
Output with 50 pF load*	1.9
Global Clock (XC3020)	1.7
Global Clock (XC3090)	3.6
CLB with Local Interconnect	0.36
Horizontal Long Line (XC3020)	0.09
Horizontal Long Line (XC3090)	0.15
Vertical Long Line (XC3020)	0.08
Vertical Long Line (XC3090)	0.19
Input without Pull-up	0.075

*Add 2.5 mW/MHz for every 100 pF of additional load

Example: *XC3020 with*	Dynamic Power (mW)
3 outputs at 5 MHz	28
20 outputs at 0.1 MHz	4
Global Clock at 20 MHz	34
10 CLBs at 5 MHz	18
40 CLBs at 0.2 MHz	3
16 Vertical Long Lines at 1 MHz	1
20 Inputs at 4 MHz	6
Total	94 mW

CCLK FREQUENCY VARIATION

Configuration Clock (CCLK) is the internally generated free-running clock that is responsible for shifting configuration data into and out of the device.

CCLK frequency is fairly stable over V_{CC}, varying only 0.6% for a 10% change in V_{CC}, but is very temperature dependent, increasing 40% when the temperature drops from 25°C to –30°C.

Vcc	T	Freq
4.5 V	25°C	687 kHz
5.0 V	25°C	691 kHz
5.5 V	25°C	695 kHz
4.5 V	–30°C	966 kHz
4.5 V	+130°C	457 kHz

CRYSTAL OSCILLATOR

The on-chip oscillator circuit consists of a high-speed, high gain inverting amplifier between two device pins, requiring an external biasing resistor R1 of 0.5 to 1 MΩ.

A **series-resonant** crystal Y1 and additional phase-shifting components R2, C1, C2 complete the circuit.

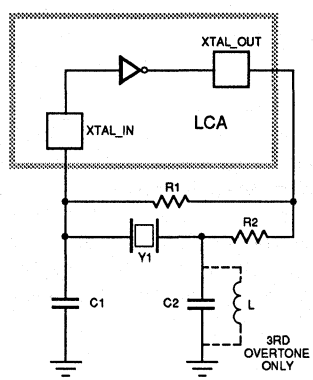
Fundamental Frequency Operation up to 24 MHz:

C1 = C2 = 34 pF
R2 = 1 kΩ up to 12 MHz, 800 Ω to 520 Ω for 15 to 24 MHz

Third Overtone Operation from 20 MHz to 72 MHz:

Replace C2 with a parallel resonant LC tank circuit tuned to ≈ 2/3 of the desired frequency, i.e., twice the crystal fundamental frequency.

Frequency (MHz)	L (µH)	C (pF)	LC Tank Freq (MHz)	R2 (Ω)	C1 (pF)
32	1	60	20.6	430	23
35	1	44	24.0	310	23
49	1	31	28.6	190	23
72	1	18	37.5	150	12



1158 02B

Crystal Oscillator

LCA Performance

Application Brief

ESTIMATING CLB PERFORMANCE

Since the delays in LCA-based designs are lay-out dependent, the data sheet cannot give all the answers needed to predict the worst-case guaranteed performance.

The timing calculator in XACT is a better tool, and a simulation using SILOS, after the design has been routed, will be the final arbiter for worst-case performance.

Still, most designer want to evaluate the possible performance, well before they have finished the design.

Here are some guidelines for XC3000 family devices:

1. A simple synchronous design-like a shift register, where a flip-flop feeds a flip-flop in the next vertical or horizontal CLB through the one level of combinatorial logic in front of the target flip-flop:

	-50	-70	-100	-125
clock-to-output	12 ns	8 ns	7 ns	6 ns
routing	1 ns	1 ns	1 ns	1 ns
logic set-up	12 ns	8 ns	7 ns	6 ns
clock period	25 ns	17 ns	15 ns	13 ns
clock frequency	40 MHz	59 MHz	67 MHz	77 MHz

2. A similar design with flip-flops several rows or columns apart would add routing delay:

	-50	-70	-100	-125
clock-to-output	12 ns	8 ns	7 ns	6 ns
routing	12 ns	8 ns	6 ns	6 ns
logic set-up	12 ns	8 ns	7 ns	6 ns
clock period	36 ns	24 ns	20 ns	18 ns
clock frequency	28 MHz	42 MHz	50 MHz	56 MHz

3. An additional level of combinatorial logic plus routing reduces performance further:

	-50	-70	-100	-125
clock-to-output	12 ns	8 ns	7 ns	6 ns
routing	12 ns	8 ns	6 ns	6 ns
logic delay	14 ns	9 ns	7 ns	6 ns
routing	1 ns	1 ns	1 ns	1 ns
logic set-up	12 ns	8 ns	7 ns	6 ns
clock period	51 ns	34 ns	28 ns	25 ns
clock frequency	20 MHz	29 MHz	36 MHz	40 MHz

Therefore, as a rule of thumb, the system clock rate should not exceed one third to one half of the specified toggle rate. Simple designs, like shift registers and simple counters, can run faster, approximately two thirds of the specified toggle rate.

These numbers assume synchronous clocking from the global clock lines. Remember, these are all worst-case numbers, guaranteed over temperature and supply voltage. Nobody should design with typical numbers.

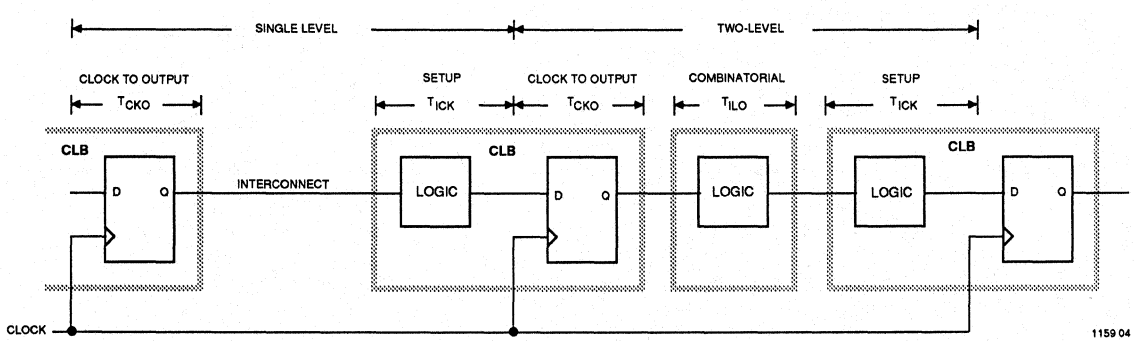


Figure 1. Critical Timing Parameters for Clocked CLB Driving Clocked CLB Directly (Single Level) and Driving it Through Additional Combinational Logic (Two-Level)

DESIGNING FOR HIGHEST DATA TRANSFER RATE BETWEEN XC3000-FAMILY LCAs

Worst-case analysis of a synchronous data transfer between XC3000-family devices postulates that the sum of clock-to-output propagation delay of the sending device, plus the input-to-clock set-up time of the receiving device, must be less than the clock period.

The inherent freedom in clock and signal routing makes it impossible to give exact values for an unprogrammed LCA without specifying certain restrictions:

On the transmitting LCA, the clock-pin to output-pin propagation delay is minimized if TCLKIN or BCLKIN are chosen as clock inputs. They are CMOS-level only, and offer the shortest on-chip clock delay.

The clock-pin to output delay is then

2 + 3.5 + 9 = 14.5 ns for the 3020-125
2 + 3.9 + 10 = 15.9 ns for the 3020-100
3 + 4.5 + 13 = 20.5 ns for the 3020-70

On the receiving LCA, the input-pin to clock-pin set-up time is the specified I/O pad input set-up time (parameter T_{PICK} in the IOB switching characteristic table of the XC3000 family data sheet) **minus** the actual delay for clock buffering and routing.

Assuming the same clock buffer choice on the receiver as on the transmitter, the longest input-pin to clock-pin set-up time is:

16 - 2 - 3.5 = 10.5 ns for the 3020-125
17 - 2 - 3.9 = 11.1 ns for the 3020-100
20 - 3 - 4.5 = 12.5 ns for the 3020-70

Under these assumptions, the worst case (shortest) value for the clock period is:

14.5 + 10.5 = 25 ns, i.e. max 40 MHz for the 3020-125
15.9 + 11.1 = 27 ns, i.e. max 37 MHz for the 3020-100
20.5 + 12.5 = 33 ns, i.e. max 30 MHz for the 3020-70

Bypassing the input flip-flop in the IOB and going directly to the DI input of the closest CLB is another, non-obvious, way of improving performance by 8 ns for the 3020-125 device, by 9 ns for the 3020-100 device and by 10 ns for the 3020-70 device.

If this is not fast enough, there are design methods that can improve the performance. Let us assume a -100 device. The easiest and safest method is to increase the clock delay on the receiving LCA, thus reducing the apparent input set-up time. Changing to a direct input (instead of TCLKIN) adds 2 ns to the clock delay and subtracts it from the input set-up time.

More aggressive methods of increasing clock delay inside or outside the receiving LCA must be used with care, since they might reduce the "best case" set-up time (fast process, low temperature, high V_{CC}) to a value of less than zero, i.e., make it a hold time requirement, which, in conjunction with a best case very fast transmitting device, can lead to problems.

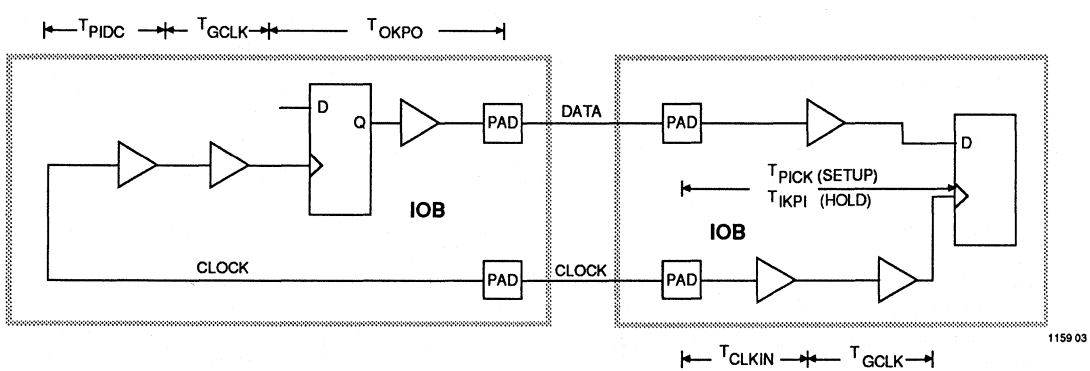


Figure 2. Critical Timing Parameters for Data Transfer Between LCAs

INPUT SET-UP TIME ON A XC3000-FAMILY LCA IS BETTER THAN THE SPECIFICATION.

The Xilinx XC3000-Family data sheet specifies a worst-case input set-up time of 17 ns for the -100 speed grade (parameter #1 on page 2-46), but this is the data input pad set-up time with respect to the **internal** IOB clock, not with respect to the clock input pad.

Any delay from clock pad to IOB clock must be subtracted from the specified set-up value in order to arrive at the true systems set-up time as seen on the device package pins (pads) for data and clock. Since the internal clock delay can be manipulated by the user, Xilinx cannot specify the systems set-up time.

The shortest possible clock delay from the package pin to the IOB clock is achieved by selecting the CMOS compatible clock inputs TCLK or BCLK. The guaranteed max value for their delay is 4.9 ns (XC3020-100), the sum of 2 ns for pad-to-CLKIN plus 2.9 ns for the clock buffer and clock distribution.

Xilinx does not guarantee any shortest values for all these parameters. An unrealistic worst-worst case analysis might, therefore, assume two extreme values:

> 17 ns set-up time for a slow data input with an infinitely fast clock path
>
> 4.9 ns hold time for an infinitely fast data input combined with a slow clock path.

That is a meaningless mathematical exercise. In reality, all these delays track very well over temperature, supply voltage and processing variations, never deviating more than 30% from each other's normalized value. When one parameter is at its absolute max value, any other parameter will be between 54% and 100% of its max value (54 = 100 x 0.7/ 1.3). The longest required set-up time for the data input with respect to the CMOS compatible clock input is, therefore, 14.4 ns (17 ns minus 54% of 4.9 ns).

What is the shortest set-up time, is there a danger of malfunction due to a positive hold time?

The fastest delay parameter is always longer than 10% of the specified guaranteed max value for the fastest available version of this device. The fastest value occurs at the lowest temperature and highest supply voltage.

The shortest data set-up time with respect to the CMOS compatible clock input is, therefore, 1.2 ns (10% of 17 ns minus 4.9 ns). This is still a positive value, sometimes called a negative hold time.

There will never be a hold time requirement if the user selects the CMOS-compatible clock-input option.

WHY ARE THERE NO GUARANTEED MIN DELAY SPECIFICATIONS?

IC manufacturers do not usually guarantee minimum propagation delay values, though some specify a token min delay of 1 ns. There are compelling reasons:

These short delays are extremely difficult to measure on a production tester. Even if it were possible, the necessary tester guard-banding might make the result meaningless.The spread between a conservative worst-case maximum value and a similarly conservative worst-case (best-case?) minimum value would be surprisingly large. There are five reasons:

1. *Temperature.* CMOS propagation delays decrease approximately 0.3% per degree C.

2. *Supply Voltage.* CMOS propagation delays are just about inversely proportional to V_{cc}.

3. *Test Guardband.* The max delay test is performed at a temperature well above TMAX and a supply voltage well below V_{cc} MIN. The accepted max delay is also less than the data sheet value. Equally conservative methods applied at the opposite extremes would give very short values.

4. *Process Variations.* LCAs are sorted into a few speed classes. A part marked -50 might have barely missed the -70 specification in only a few or perhaps only one parameter. IC manufacturers may sometimes mark down (call a -70 part a -50 part) in order to adjust production yield to market demand. This increases the spread even more.

5. *Process Evolution.* As IC technology improves, smaller geometries reduce not only device size and cost, but also propagation delay. Tight minimum specifications would be a hindrance to progress.

Finally, it can be argued that a proper synchronous design is insensitive to minimum propagation delay values. When the clock skew is small, (Xilinx clock networks guarantee extremely small clock skew values, less than 2 ns over a big chip like the XC3090) the designer can safely ignore the minimum delay issue. No Xilinx CLB or IOB input has a hold time requirement.

In the past, designers have faced far greater uncertainties when they populated PC boards with a variety of SSI, MSI and PAL devices, each from a different production run, each with different power dissipation and junction temperature. Such problems do not exist inside the LCA where delays track, and the temperature is the same for all elements.

Delay Tracking

Applications Brief BY PETER ALFKE

How much can delays on a chip vary, relative to each other?

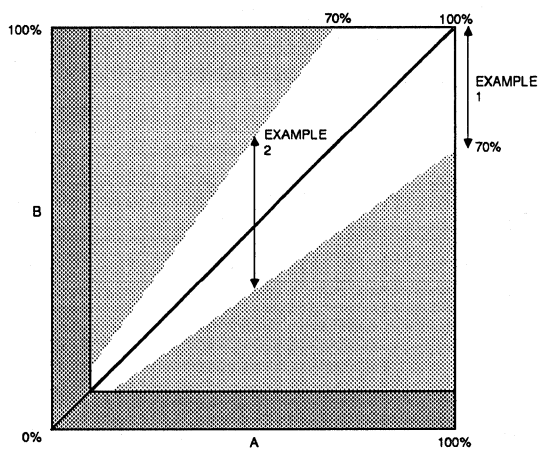
If I know one delay value exactly, how much tolerance must I assume for any other delay value on the same chip?

The figure explains delay tracking, i.e. the correlation between any two delays, called A and B, on the same chip. Delay A is represented on the horizontal axis, delay B on the vertical axis. 100% represents the guaranteed worst-case maximum value for either of the two delays.

No delay can ever be shorter than 10% of the specified max value; the horizontal and vertical strips parallel to both axes, therefore represent impossible combinations. If the correlation between delay parameters were perfect, all possible combinations would be on the diagonal line. If there were no correlation at all, the whole remaining square would represent valid combinations. Xilinx assumes a correlation factor of 70%; all possible combinations thus fall onto the unshaded part of the square.

Examples

If one delay is at its maximum value, the other is between 70% and 100% of its maximum value. If one delay is at 50% of its maximum value, the other one can be anywhere between 71% and 35% of its specified maximum value.



Warning: "Typicals" Are Hazardous to Your Designs

All Xilinx parameter specifications in this data book, shown on the screen and used in any simulator, are worst-case values, guaranteed over the full range of operating conditions. Delays in all CMOS devices are inevitably longest at the highest temperature and the lowest supply voltage, and delays are also affected by variations in the manufacturing process.

For a clear description of these delay variations, see Figure 29 on page 2-26. It shows that the "typical" delay, exhibited by an average device at 25°C and 5.0 V, is **slightly more than half** the worst case delay for a commercial temperature range product, and is **less than half** of the worst case military value.

Designers should regard "typical" values as meaningless averages, taken under favorable operating conditions. Nobody should base a design on "typical" values, but some manufacturers still use this misleading way to specify device performance. It should have died 25 years ago.

Start-Up and Reset

Application Brief

INTERNAL LOGIC DURING CONFIGURATION

During configuration, all I/O pins not used for configuration are 3-stated and all internal flip-flops and latches are held reset until the chip goes active. Even multiple LCAs hooked up in a daisy chain will go active simultaneously as a result of the same CCLK edge. This is well documented.

Not documented is how the internal combinatorial logic comes alive during configuration: As configuration data is shifted in and reaches its destination, it activates the logic and also "looks at" the inputs. Even the crystal oscillator starts operating as soon as it sees its configuration data. Since all flip-flops and latches are being held reset, and all outputs are being held 3-stated, there is no danger in this "staged awakening" of the chip. The user can take advantage of this to make sure that the chip comes to life with the internal output 3-state control signal on the output driver already active before the end of configuration, so that there is no chance of any output glitch.

FAST RECOVERY FROM RESET

Recovery from Reset is not specified in our data sheets because it is very difficult to measure in a production environment.

Here are benchmark values:

The CLB can be clocked immediately, i.e. within 0.2 ns, after the end of the internal direct reset (rd).

The CLB can be clocked no earlier than (worst case) 25 ns after the release (rising edge) of the externally applied Global Reset (acting Low) signal.

SYNCHRONOUS RESET AFTER CONFIGURATION

After configuration is completed, the LCA becomes active in response to a rising edge of CCLK. All outputs that go active will do so simultaneously, but they are obviously not synchronized to the system clock. Some designs might require a reset pulse synchronous with the system clock to avoid start-up problems due to asynchronous timing between the end of internal reset and the system clock.

The circuit below generates a short global reset pulse in response to the first system clock after the end of configuration. It consumes one CLB plus one output pin, and it also precludes the use of the $\overline{\text{LDC}}$ pin as I/O.

During Configuration:

$\overline{\text{LDC}}$ (Low) holds D High, but Q is held Low by internal reset.
$\overline{\text{RESET}}$ is pulled High by internal and external resistors.

End of Configuration before first System Clock:

$\overline{\text{LDC}}$ pin goes active High, Q stays Low, D stays High.
$\overline{\text{RESET}}$ is still pulled High by external resistor.

Result of first System Clock after end of Configuration:

Q is clocked High, which forces D Low.
Output driver goes active Low and forces $\overline{\text{RESET}}$ Low.
This resets the whole chip until the Low on Q causes $\overline{\text{RESET}}$ to be pulled High again
The whole chip has thus been reset by a short pulse instigated by System Clock.

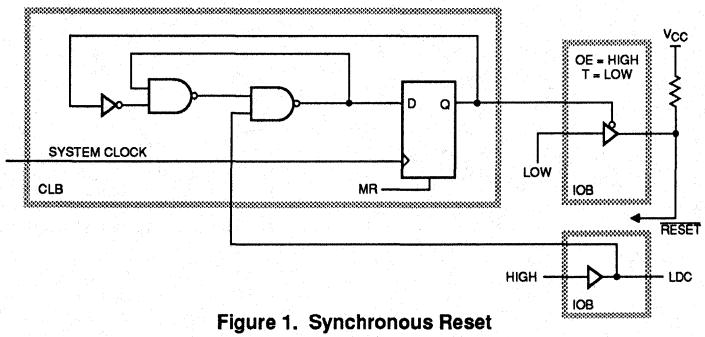


Figure 1. Synchronous Reset

1971 01

Metastable Recovery

Applications Brief BY PETER ALFKE & PERRY WU

CLB FLIP-FLOPS RECOVER SURPRISINGLY FAST FROM METASTABLE PROBLEMS

A specter is haunting digital design, the specter of metastability. From a poorly understood phenomenon in the seventies, it has developed into a scary subject for every designer of asynchronous interfaces. Now Xilinx offers data and a demonstration kit to help users analyze and predict the metastable behavior of LCAs.

Whenever a clocked flip-flop synchronizes a truly asynchronous input, there is a small but finite probability that the flip-flop output will exhibit an unpredictable delay. This happens when the input transition not only violates the setup and hold-time specification, but actually occurs within the tiny timing window where the flip-flop "decides" to accept the new input. Under these circumstances the flip-flop enters a symmetrically balanced state, called metastable, (meta = between) that is only conditionally stable. The slightest deviation from perfect balance will eventually cause the outputs to revert to one of the two stable states, but the delay in doing so depends not only on the gain bandwidth product of the circuit, but also on the original balance and the noise level of the circuit; it can, therefore, only be described in statistical terms.

The problem for the system designer is not the illegal logic level in the balanced state (it's easy enough to translate that to either a 0 or a 1), but the unpredictable timing of the final change to a valid logic state.

The basic phenomenon is as unavoidable as death and taxes, but the probability of erroneous operation can be determined, and the impact of various countermeasures can be evaluated quantitatively, if two fundamental flip-flop parameters are known, i.e., the metastability capture window, and the metastability recovery rate.

Xilinx has evaluated the XC3020 CLB flip-flop with the help of a mostly self-contained circuit on the Demonstration Board that is available to any Xilinx customer.

The result of this experimental evaluation shows the Xilinx CLB flip-flop superior in metastable performance to many popular MSI or PLD devices.

When an asynchronous event frequency of approximately 1 MHz is being synchronized by a 10 MHz clock, the CLB flip-flop will suffer an additional delay of

4.2 ns statistically once per hour

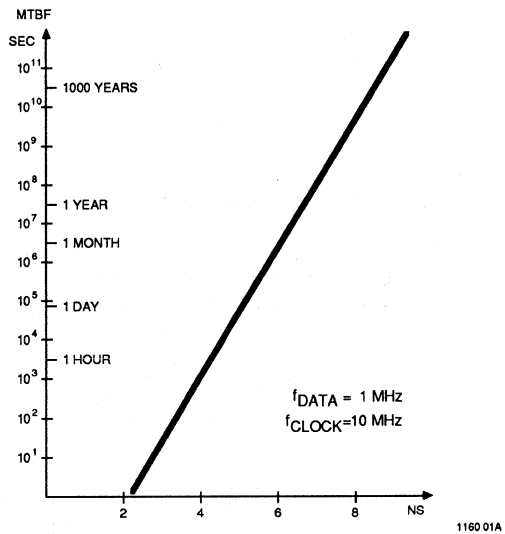
6.6 ns statistically once per year

8.4 ns statistically once per 1000 years

The frequency of occurrence of these metastable delays is proportional to the product of the asynchronous event frequency and the clock frequency.

If, for example, a 100 kHz event is synchronized by a 2 MHz clock, the above mentioned delays (besides being far more tolerable) will occur 50 times less often.

The mean time between metastable events lasting longer than a specified duration is an exponential function of that duration. Two points measured on that line, allow extrapolation to any desired MTBF (mean time between failure).



Metastable MTBF as a Function of Additional Acceptable Delay

MEASURING METASTABLE RECOVERY

The excellent metastable recovery rate of Xilinx LCA flip-flops was measured in a working XC3020-70 on the Xilinx evaluation board. Since metastability can only be measured as a statistical event, the device was configured with eight concurrent detectors:

Eight D flip-flops are clocked from a common high-speed source. Their D inputs are driven from a common, lower frequency asynchronous signal. Each flip-flop feeds the D inputs of two more flip-flops, one of them clocked on the opposite clock edge. This cuts the clock rate for the experiment in half, from 50 MHz to a more manageable 25 MHz. A comparator detects when Q1 differs from D = Q0 = Q2.

This can only be the result of Q0 having a clock-to-output delay in excess of a half clock period minus a set-up time. Varying the clock frequency and monitoring the pulse rate at the detector gives an indication of the probability of metastable delays.

The deliberate skew on the D-inputs of the eight flip-flops under test makes it extremely unlikely that more than one flip-flop will go metastable on any one clock edge. The eight detector outputs can, therefore, be ORed together and drive a counter with LED read-out. As expected, no metastable events were observed at clock rates below 25 MHz since a half clock period of 20 ns allows for propagation delay plus set-up time.

Increasing the clock rate brought a sudden burst of metastable events around a clock rate of 27 MHz. Careful adjustment of the clock frequency gave repeated, reliable measurements showing that a 500 ps decrease in the relevant half clock period increased the frequency of metastable occurrences by a factor of 41. In order to be conservative, to compensate for favorable conditions at room temperature, and to avoid any possibility of overstating a good case, the measurement was interpreted as follows:

Every ns in additional acceptable delay reduces the frequency of metastable events by a factor 40.

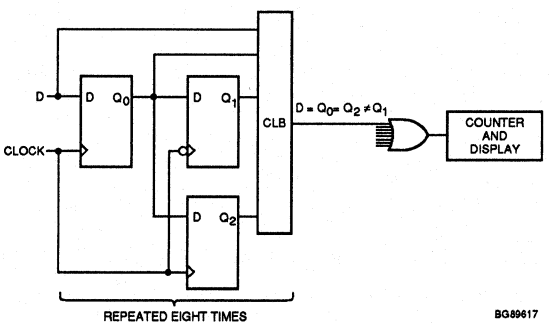
The MTBF curve was then normalized to a reasonable combination of clock and asynchronous data rates, using the generally accepted theory that, everything else being equal, the frequency of metastable events is proportioned to the product of the two frequencies at the D and CLK inputs of the flip-flop under test.

Assuming that the metastable window is 0.1 ns wide, and the clock is 10 MHz, one data change in 1000 will fall into the metastable window. A 1 MHz data rate gives an MTBF of 1 ms for an additional delay of zero. Each additional ns of acceptable delay increases the MTBF by a factor 40, see diagram on the previous page. It is difficult to measure the exact width of the window, but it hardly matters. If the assumption of 0.1 ns were wrong by a factor of 10, it would only move the graph horizontally by 0.624 ns.

Over the past fifteen years this writer has made many attempts to pin-point, demonstrate and quantify metastable behavior. Success came with the integration of multiple test circuits, multiple detectors and a common read-out, all in one Logic Cell Array.

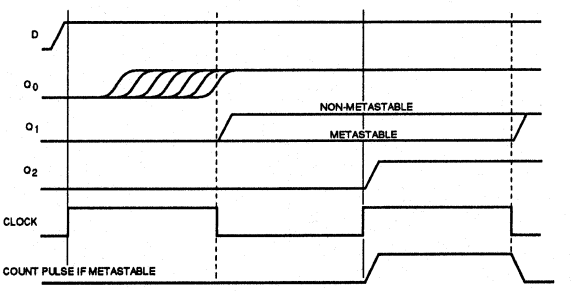
After so many inconclusive attempts, any repeatable results would have been appreciated, but these results also show the metastable response of LCA flip-flops to be superior to any other circuit documented so far. There is a reason. LCA flip-flops are dedicated circuits, small and very fast, with an extremely short loop delay. TTL flip-flops are bigger and slower, and gate arrays and gate-array-like structures that construct their flip-flops by interconnecting gates are bound to be far inferior.

Metastability is still a treacherous subject, but LCAs offer the closest thing to an acceptable solution.



1160 02

Metasable Delay Measuring Circuit



1160 03

Metasable Detection

Battery Backup for Logic Cell Arrays

Application Brief

Logic Cell Arrays use a high performance low power CMOS process. They can, therefore, preserve the program contents stored in the internal static memory cells even during a loss of primary power. This is accomplished by forcing the device into a low-power non-operational state while supplying Vcc from a battery.

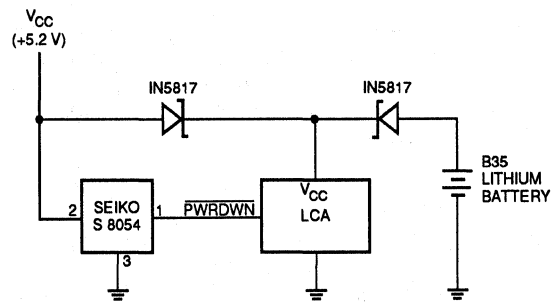
There are two primary considerations for battery backup which must be accomplished by external circuits:

- Control of the Power-Down ($\overline{PWRDWN}$) pin and
- Switching between primary V_{CC} supply and battery.

Important considerations are:

- Insure that $\overline{PWRDWN}$ is asserted logic Low prior to Vcc falling, held Low during the primary V_{CC} loss time, and returned High after Vcc has returned to a normal level. $\overline{PWRDWN}$ edges must not be slow rising or falling.
- Insure "glitch-free" switching of the power connections to the LCA from the primary Vcc to the battery voltage and back.
- Insure that during normal operation the LCA Vcc is maintained at an acceptable level, 5.0 V ± 5% (±10% for Industrial and Military).

Figure 1 shows a circuit developed by Shel Epstein of Epstein Associates in Wilmette, IL. Two 1N5817 Schottky diodes power the LCA from either the 5.2-V supply or a 3-V Lithium battery. A SEIKO S8054 3-terminal power monitor circuit measures V_{CC} and pulls $\overline{PWRDWN}$ Low



Compact Multiplexer and Barrel Shifter

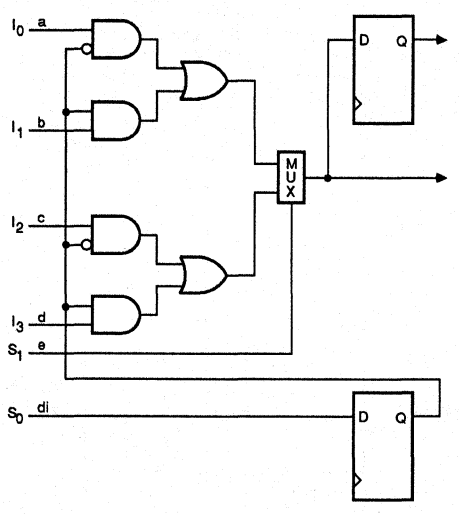
Application Brief

FOUR-INPUT MULTIPLEXER IN ONE CLB

Since the function generator in the XC3000 series CLB has only five inputs, it cannot directly implement a four-input multiplexer, which requires four data inputs and two select inputs.

Registering one of the select inputs in the same CLB frees up one input and puts a complete four-input multiplexer into one CLB. It is even possible to register the multiplexer output.

This non-obvious trick increases the apparent delay of the registered select input, but that will be acceptable in the majority of applications. Since it reduces not only the size but also the through-delay of the four-input multiplexer by 50%, this approach is definitely worth considering.



1978 01

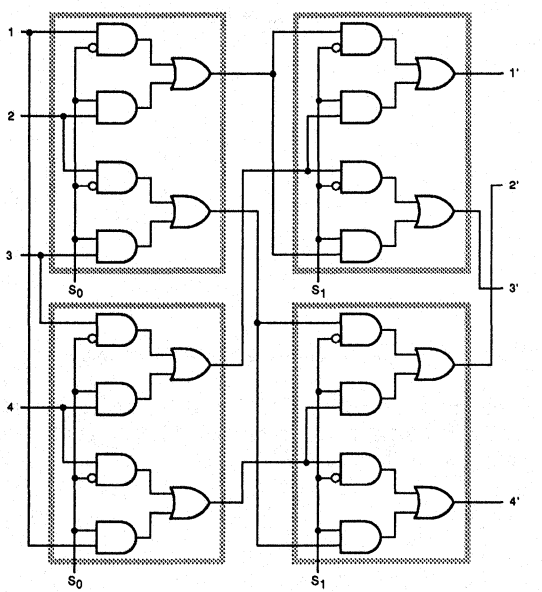
FOUR-BIT BARREL SHIFTER IN ONLY FOUR CLBs

A four-input barrel shifter has four data inputs, four data outputs and two control inputs that specify rotation by 0, 1, 2 or 3 positions. A brute force design would use four four-input multiplexers, since each output can receive data from any input. Each four-input multiplexer requires two XC3000 family CLBs, for a total of eight CLBs.

There is, however, a smarter method that reduces the design to only four CLBs. The key to this approach lies in the signal crossovers at the input and output of the second level CLBs.

Eight-Bit Barrel Shifter in 12 CLBs

The 4-Bit Barrel Shifter design can be extended to eight bits. A first-level shifter consisting of four CLBs rotates the eight inputs by one position, controlled by the least-significant control input. Two interleaved 4-Bit Barrel Shifters then take the eight outputs from the first level and rotate them by 0, 2, 4 or 6 positions.



1978 02A

Majority Logic, Parity

Application Brief

MAJORITY LOGIC, N-OF-X DECODING

Majority logic has interesting mathematical features, but has not become popular because its traditional logic implementation is quite complicated and expensive. Since LCAs can generate any function of five variables at the same cost and the same delay, they can easily decode majority logic. The output F, G is Low when none, one, or two inputs are High; it is High when three, four, or five inputs are High.

Majority logic is a special case of "N-of-X Decoding." An XC3000-series CLB can directly encode any "N of 5" inputs active. This concept can be cascaded so that three LCBs encode any "N of 7" inputs active.

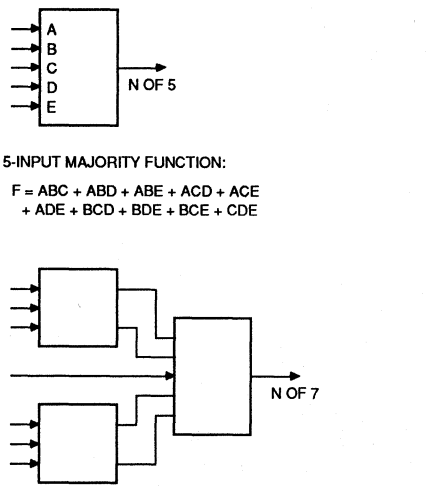


Figure 1. N of X, e.g., Majority Logic

The first-level blocks can only have three inputs, since the two outputs can only encode four different states: none, one, two, or three active.

PARITY

Two CLBs can generate the parity for nine inputs, or can check a 9-bit input for odd or even parity with a through-delay of two cascaded CLBs. Three CLBs can check 13 inputs; four CLBs can check 17 inputs; five CLBs can check 21 inputs; six CLBs can check 25 inputs; all with the same delay of two cascaded CLBs.

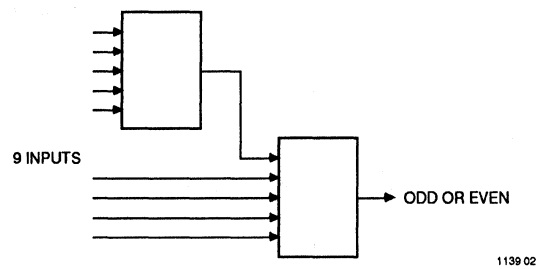


Figure 2. Nine-Input Parity

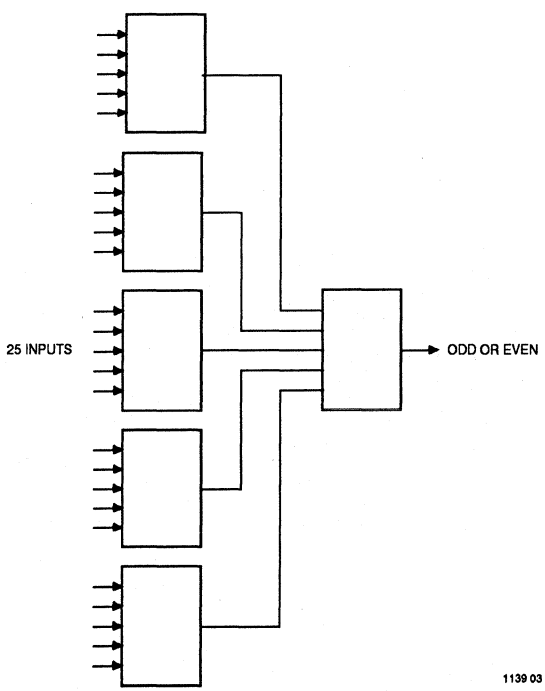


Figure 3. 25-Input Parity

Multiple Address Decoding

Application Brief BY PETER ALFKE

An XC3000-series CLB can decode a 5-bit address in any conceivable way, or it can decode a 4-bit address in two different ways, each without any restrictions.

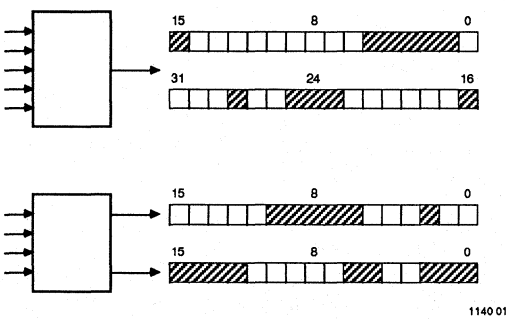


Figure 1. Multiple Address Decoding

Three XC3000-series CLBs can decode three distinct addresses in an 8-bit address field: One CLB decodes the lower four bits and encodes the result on its two outputs (00 = no match). The second CLB decodes and encodes the upper four bits in a similar way.

The third CLB encodes the four signals into two outputs (00 = no match). This works for any three distinct addresses, even when some share the same upper or lower nibble.

This scheme can be expanded to a 16-bit wide address, using seven CLBs.

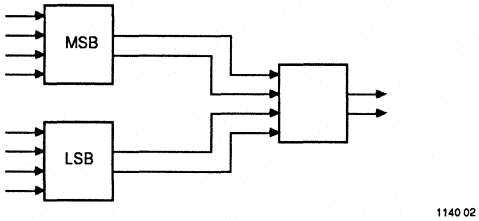


Figure 2.

Address Block Detection

The idea mentioned above is not restricted to detecting three specific addresses, it can also detect three groups of addresses, as long as none of them straddles the boundaries defined by the individual CLBs. If they do, this circuit cannot detect *three* address blocks, but can still detect any *one* address block in an 8-bit address.

Suppose we want to decode the block of 8-bit addresses starting at 24 and including 68 (hex).

With one CLB, we encode the least significant address nibble into a 2 bit output called LS:

inputs 0 through 3	generate LS = 1
inputs 4 through 8	generate LS = 2
inputs 9 through F	generate LS = 3

With another CLB we encode the most significant address nibble into a 2-bit output called MS:

inputs 0, 1, 7, 8 through F	generate MS = 0
input 2	generates MS = 1
input 3, 4, 5	generate MS = 2
input 6	generates MS = 3

The third CLB then encodes these signals

MS	LS	Output
0	x	0
1	2,3	1
2	x	1
3	1,2	1
3	3	0

The solution can be generalized:

Three CLBs can decode any one block of an 8-bit address.

Binary Adders, Subtractors and Accumulators

Application Brief

There are many different ways to implement binary adders, subtractors and accumulators with LCAs, using different trade-offs between size and speed.

Most compact, but slowest, is the bit-serial function that operates on one bit pair per clock cycle, generating sum and carry. The sum is fed back into the shift register, the carry is stored for the subsequent bit time.

The most compact combinatorial (parallel) adder, subtractor, or accumulator consists of cascaded CLBs. Each CLB (XC2000 or XC3000 family) is a full adder, accepting one operator bit pair (A, B) and an incoming carry. The CLB generates the sum and the outgoing carry. A 16-bit function requires 16 CLBs. It performs an operation in 16 combinatorial delays.

The 5-input function generator of the XC3000 family can add a carry to two operator pairs. Three CLBs can thus handle two input bit pairs, generating two sum outputs and the outgoing carry. A 16-bit function requires 24 CLBs. It performs an operation in eight combinatorial delays.

Carry Propagate and Carry Generate are intermediate signals that can speed up the operation as shown on pages 6-27 and 6-28. Such a 16-bit function requires 30 CLBs. It performs an operation in six combinatorial delays.

The concept of Carry Propagate and Carry Generate has been made popular by the 74181 ALU and its descendents. These signals can reduce the ripple carry delay. Both CP and CG are outputs from an arithmetic block (often of four bits). Both these outputs can be generated immediately since they are not affected by any incoming carry that might arrive late. As the names imply, Carry Generate is active if the block creates an overflow (carry), e.g. if the 4-bit sum, regardless of incoming carry, exceeds F. Carry Propagate is active if the block does not generate a carry by itself, but would generate a carry as a result of an incoming carry. In our 4-bit example this occurs when the sum is exactly F.

There is an even faster algorithm. As originally described by J. Sklansky in the June 1960 issue of the IRE Transaction on Electronic Computers, Conditional-Sum Addition can save time at the expense of higher logic complexity. Matt Klein of Hewlett Packard recently modified this algorithm to fit the XC3000 architecture. His design requires 41 CLBs to add or accumulate two 16-bit numbers in only three(!) combinatorial delays. With careful layout, such an adder/ accumulator can run at 30 MHz.

Note that all Xilinx adder structures can also be accumulators without any size or speed penalty. Conventional gate arrays and other gate-array-like structures usually configure flip-flops out of gates. The flip-flop set-up time must then be added to the combinatorial propagation delay. LCAs hide the flip-flop set-up time in the combinatorial propagation delay of the CLB. Adders and accumulators thus operate at the same speed.

Adders and Comparators

Application Brief BY PETER ALFKE

The LCA-structure accommodates 1-bit and 2-bit adders very efficiently. A 1-bit adder with three inputs (A ,B, C_{IN}) generating two outputs (S, C_{OUT}) fits exactly in one XC2000- series CLB, where the flip-flop might be used for storing the carry in a bit serial adder. A XC3000-series CLB can even include an additional control input, either ADD/SUBTRACT or ADD ENABLE.

A 2-bit adder requires three XC3000-series CLBs. The five inputs A0, B0, A1, B1, and C_{IN} are common to all three CLBs; the outputs are S0, S1, and C_{OUT}. The propagation delay is only one CLB combinatorial delay, as little as 10 ns. Two such adders can be cascaded to form a 4-bit adder in six CLBs with a through-delay of two CLBs, i.e., 25 ns (allowing for some interconnect delay).

Four 2-bit adders can be cascaded to form a byte-wide adder, using 12 CLBs with a through-delay of four CLBs, but there is also a slightly faster design using a carry look-ahead technique: The third and fourth di-bit adders are changed, they no longer generate Carry out, but now each generates two outputs as a function of the four A and B inputs (ignoring C_{IN}). These two outputs are called Carry Generate (when the addition exceeds a binary 3) and Carry Propagate (when the addition is exactly 3). These outputs from two di-bit adders are combined with C_{IN} and generate the Carry inputs to the fourth di-bit adder and its carry out. The whole 8-bit adder uses 14 CLBs and has a through-delay of four CLB delays from input to S6 and S7, only three CLB delays from input to C_{OUT}.

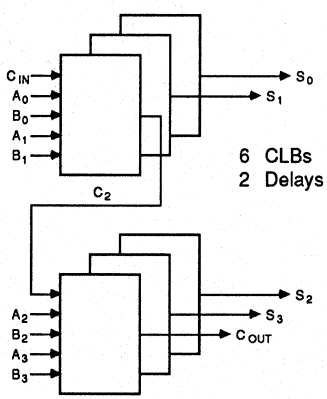


Figure 1. 4-Bit Adder

For eight bits, this look-ahead carry scheme is of marginal use, it reduces only the carry delay, and only by one CLB delay. For this small speed improvement it uses two additional CLBs (14 instead of 12). See truth table on page 6-25.

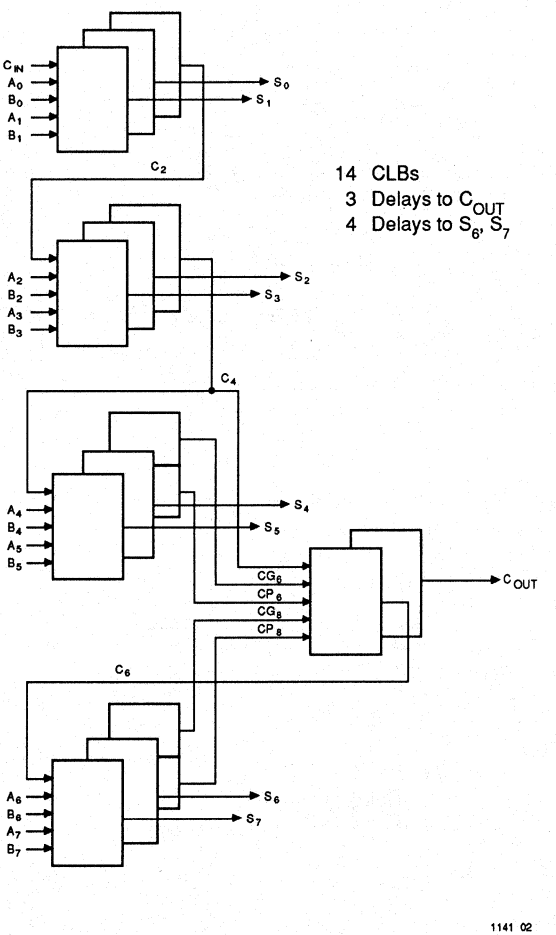


Figure 2. 8-Bit Adder with Carry Lookahead

A 16-bit adder benefits from carry-lookahead. Simply cascading di-bit adders uses 24 CLBs at a max propagation delay of eight CLBs from C_{IN} to C_{OUT} or to S14, 15.

A look-ahead carry scheme uses 30 CLB at a max prop delay of five CLBs from C_{IN} to C_{OUT} (six delays to S14, 15).This design is available from Xilinx. Call the applications hot line 408-559-7778 or 1-800-255-7778.

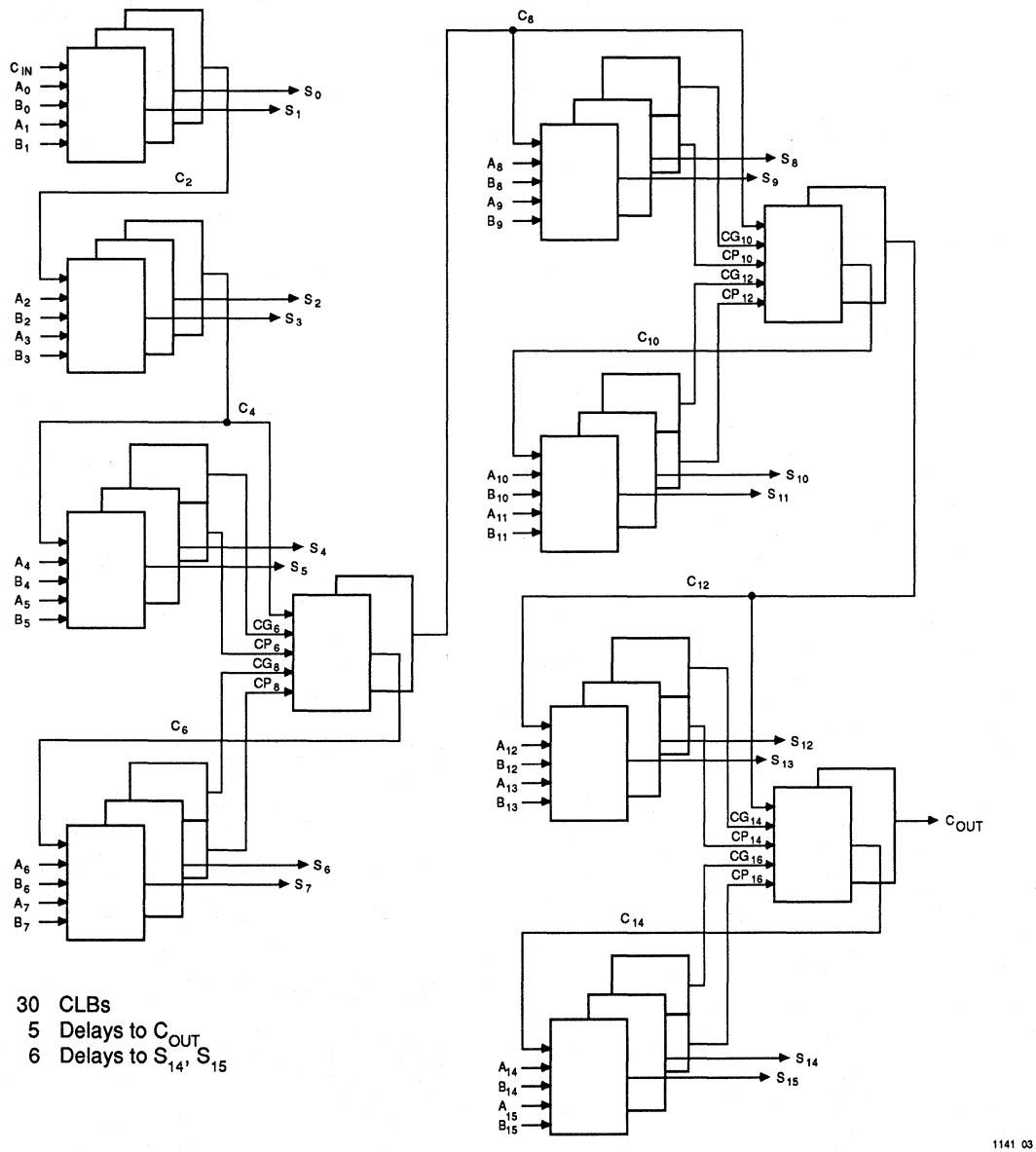


Figure 3. 16-Bit Adder with Carry Lookahead

Adder Logic Truth Tables

After adjusting the subscripts appropriately, the truth table for the three CLBs generating S_2 and S_3 is identical with that for the CLBs generating S_0 and S_1; and the truth table for the bottom three CLBs is identical to that of the three CLBs generating S_4 and S_5.

The 16-bit adder is a natural extension of the 8-bit adder.

Outputs	Inputs C_0	A_0	B_0	A_1	B_1
S_0=1	1	0	0	x	x
	0	1	0	x	x
	0	0	1	x	x
	1	1	1	x	x
S_1=1	x	1	1	0	0
	1	x	1	0	0
	1	1	x	0	0
	x	0	0	1	0
	0	x	0	1	0
	0	0	x	1	0
	x	0	0	0	1
	0	x	0	0	1
	0	0	x	0	1
	x	1	1	1	1
	1	x	1	1	1
	1	1	x	1	1
C_2=1	x	x	x	1	1
	x	1	1	1	0
	1	x	1	1	0
	1	1	x	1	0
	x	1	1	0	1
	1	x	1	0	1
	1	1	x	0	1

Outputs	Inputs C_4	A_4	B_4	A_5	B_5
S_4=1	1	0	0	x	x
	0	1	0	x	x
	0	0	1	x	x
	1	1	1	x	x
S_5=1	x	1	1	0	0
	1	x	1	0	0
	1	1	x	0	0
	x	0	0	1	0
	0	x	0	1	0
	0	0	x	1	0
	x	0	0	0	1
	0	x	0	0	1
	0	0	x	0	1
	x	1	1	1	1
	1	x	1	1	1
	1	1	x	1	1
CG_6=1	x	x	x	1	1
	x	1	1	1	0
	x	1	1	0	1
CP_6=1	x	0	1	1	0
	x	1	0	1	0
	x	0	1	0	1
	x	1	0	0	1

Outputs	Inputs C_{IN}	Lower CP_n	CG_n	Higher CP_{n+2}	CG_{n+2}
$Carry_n$	x	x	1	x	x
	1	1	x	x	x
$Carry_{n+2}$	x	x	x	x	1
	x	0	1	1	x
	1	1	0	1	x

Bit-Serial Adder, Subtractor, Comparator

The CLB architecture is ideally suited for bit-serial arithmetic, where the function generator performs the serial arithmetic (LSB first), and the associated flip-flop stores the carry or borrow.

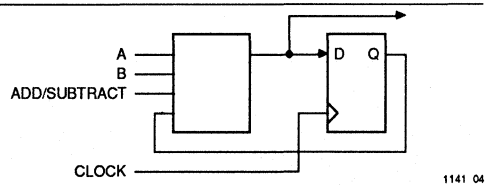


Figure 4. Serial Adder/Subtractor

A bit-serial **identity** comparator detects only whether the two operands are equal or not, without determining which one (if any) is larger. The bit stream can come in LSB or MSB first, the flip-flop gets set for any difference between A and B, and stays set until the end of the word, then gets reset before the beginning of the next word. This "difference detector" can also be implemented as a latch and folded into the combinatorial logic.

A bit-serial **magnitude** comparator distinguishes between A = B, A > B and A < B. It can operate LSB first or MSB first, if the logic is adjusted:

LSB first: Start with both flip-flops reset

if A > B set Qx, reset Qy

if A < B set Qy, reset Qx

MSB first: Start with both flip-flops reset

if A > B and Qy = 0: set Qx

if A < B and Qx = 0: set Qy

Result in both cases:

Qx	Qy	
0	0	A = B
0	1	A < B
1	0	A > B
1	1	Impossible

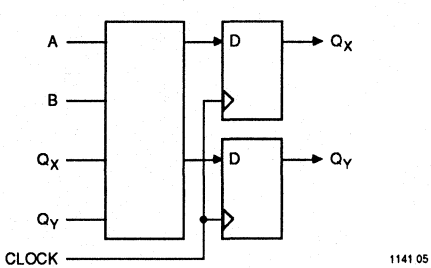


Figure 5. Serial Magnitude Comparator

Conditional Sum Adder Adds 16 Bits in 33 ns

Application Brief BY MATT KLEIN, HEWLETT-PACKARD

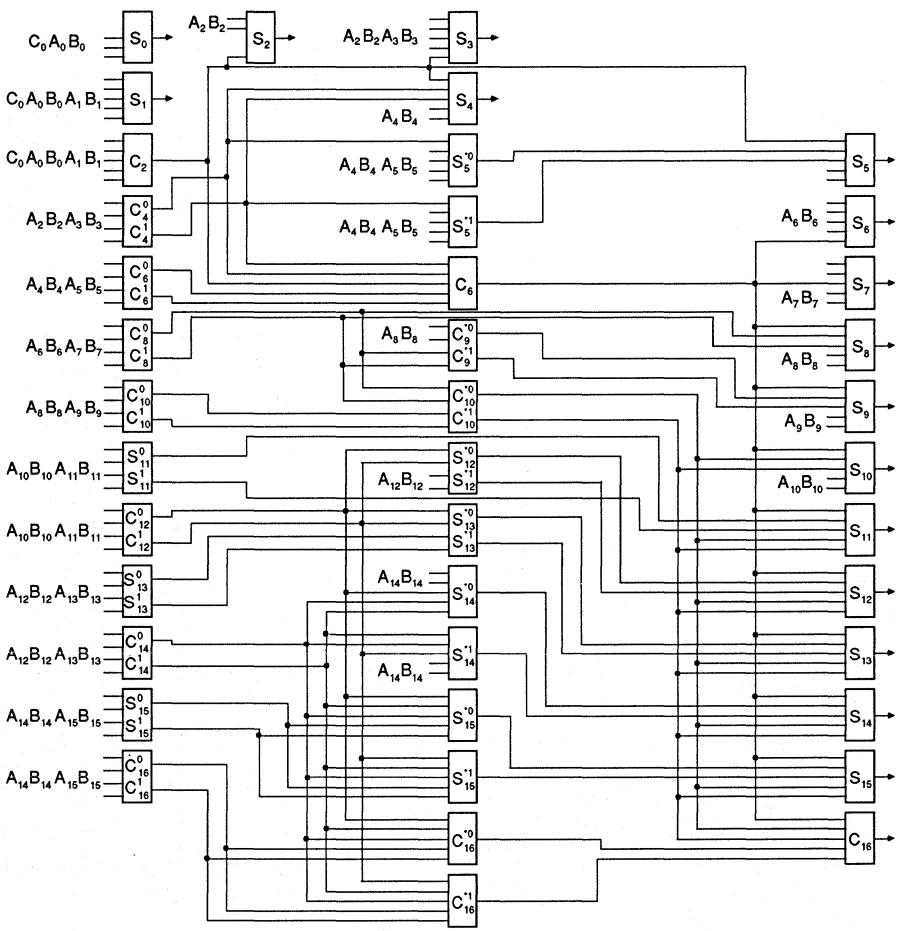
This circuit is based on a 1960 paper by J. Sklansky (see page 6-22). With careful placement and routing the total delay can be kept below 33 ns.

The block diagram below shows each CLB and its inputs and outputs.

27 of the CLBs each generate one function of up to five variables, 14 of the CLBs each generate two functions of four variables. In accordance with the original paper all subscripts denote the binary position (weight), and superscripts describe the assumed input condition:

0: carrry into this position is assumed inactive
*0: carry into the position one lower is assumed inactive
1: carrry into this position is assumed inactive
*1: carry into the position one lower is assumed inactive

This design is available from Xilinx. Call the applications hot line 408-559-7778 or 1-800-255-7778.



16-Bit Conditional Sum Adder

Building Latches Out of Logic

Application Brief

Since the XC3000-series, unlike the XC2000-series, cannot configure its CLB flip-flops into latches, there must be other ways to design latches. Obviously, the I/O block can be configured with latches on either the input, the output, or both. Beyond that, every CLB can form a latch.

The 5-input logic structure permits an amazing diversity of latch designs; here are several ideas:

With F fed back to close the feedback path, there are four control inputs left. They might be called Set, Reset, Data and Enable, defined such that S and R are independent of Enable, but D is activated by it. Any of these four inputs can be defined as active High or active Low. This results in 16 different latch designs, all with the same basic characteristics and the same timing.

We can also eliminate D and have two Enables, affecting S and R (again 16 different flavors) or we use multiple S and multiple R, either ORed, or ANDed, or XORed. We can also have two D inputs, each with its own Enable; or we can have two D inputs, a Select input and an Enable input; or we can have an Enable and three D inputs defined in any arbitrary way. Majority gating could be one way: if none or one is active, reset the latch; if two or three are active, set the latch. Or, if none is active, reset; if one or two are active, hold; if three are active; set. Or we can assign positive or negative weights to the D inputs.

Since there are 65,536 different functions of four variables, there are many different ways to define a latch, not counting pin rotations and active-High/active-Low variations.

All these latches have the same timing characteristics: propagation delay from input to output = 14/9 ns for the 50/70 MHz part. Set-up time to the end of Enable, or min.

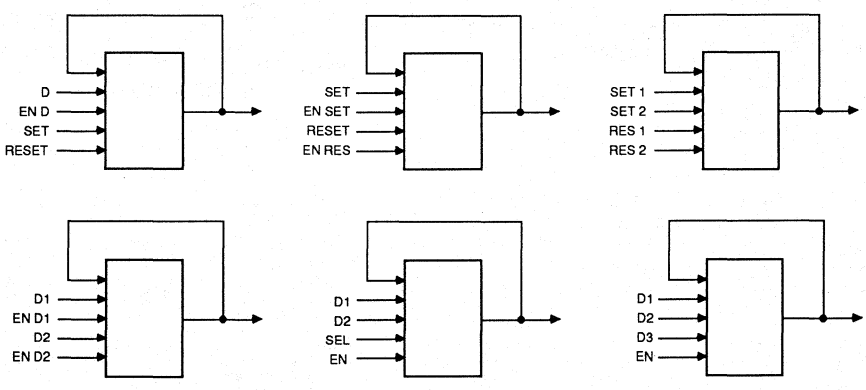


Figure 1. Latched Logic

Synchronous Counters, Fast and Compact

Application Brief BY PETER ALFKE

FULLY SYNCHRONOUS 4-BIT COUNTER USES ONLY TWO CLBS TO COUNT ANY CODE

This 4-bit counter operates synchronously and has a Count Enable (Clock Enable) input. Count length, count direction, and even the code sequence can be selected through configuration. There are 15!, i.e. more than 10^{12} different possible sequences. All four outputs are available. This counter cannot be preset to an arbitrary value, but it can be cleared by an asynchronous input.

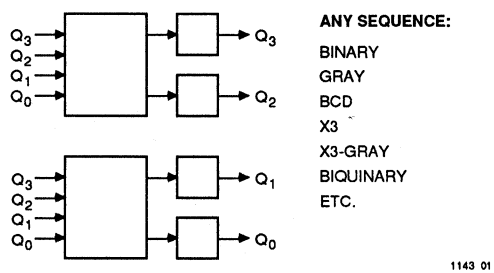


Figure 1. Synchronous 4-Bit Counter in 2 CLBs

The advantage of a Gray code is its glitch-less decoding, since only one bit changes on any code transition. A Gray counter can also be read "on-the-fly" without the well-known problems of reading a binary counter e.g., on its transition between 7 and 8, where any code might be read.

Decimal	Binary	Gray	X3 Binary	X3 Gray
0	0000	0000	0011	0010
1	0001	0001	0100	0110
2	0010	0011	0101	0111
3	0011	0010	0110	0101
4	0100	0110	0111	0100
5	0101	0111	1000	1100
6	0110	0101	1001	1101
7	0111	0100	1010	1111
8	1000	1100	1011	1110
9	1001	1101	1100	1010
10	1010	1111		
11	1011	1110		
12	1100	1010		
13	1101	1011		
14	1110	1001		
15	1111	1000		

FULLY SYNCHRONOUS 5-BIT COUNTER USES ONLY THREE CLBS

Three XC3000-series CLBs can implement a modified shift-register counter with the following features:

- Fully synchronous operation
- Count Enable Asynchronous clear
- Count-Modulus defined during configuration: 2...32
- Only one meaningful output, Q_5, but with complete freedom to define its waveform

Q_0 through Q_4 form a linear shift register counter. The 5-input combinatorial function F0 determines the modulus (there are no illegal or hang-up states). The 5-input combinatorial function F1 decodes the counter in any conceivable way, Q_5 synchronizes and de-glitches F1.

Examples:

+ 28 counter with output High at times
T2, 3, T10, T22 through T27

+ 19 counter with output Low at times
T9, T12, T15, T18.

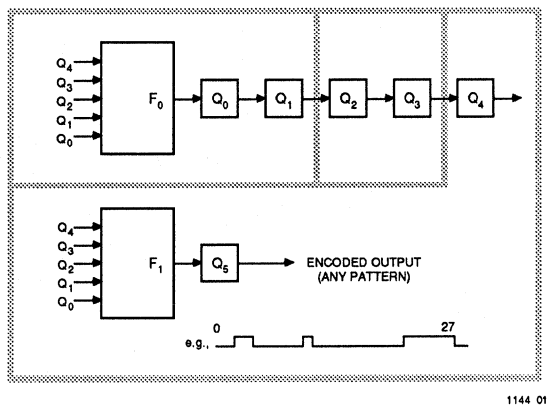


Figure 2. Synchronous 5-Bit Counter in 3 CLBs

30 MHz Binary Counter Uses Less Than One CLB per Bit

Application Brief BY PETER ALFKE

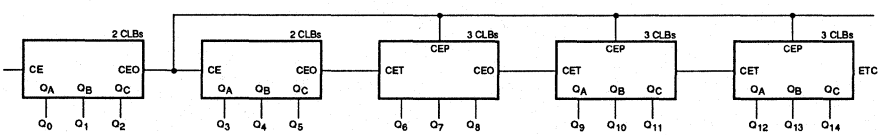
Borrowing the concept of Count-Enable Trickle/Count-Enable Parallel that was pioneered in the popular 74160 TTL-MSI counter, a fast non-loadable synchronous binary counter of arbitrary length can be implemented efficiently in the XC3000 series CLBs. For best partitioning into CLBs, the counter is segmented into a series of tri-bits.

The least significant, i.e. the fastest changing, tri-bit has a Count-Enable Output (CEO) that is routed to all the Count-Enable-Parallel (CEP) inputs of the whole counter.

Each Count-Enable Output from any other tri-bit drives the next more significant Count-Enable Trickle (CET) input. The clock causes any tri-bit to increment if all its Count-Enable (CE) inputs are active. CEO is active when all three bits are set **and** CET is High. CEP does not affect CEO.

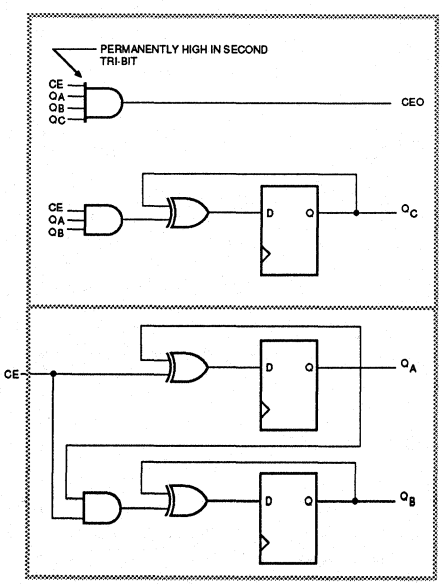
The least-significant tri-bit thus stops the remaining counter chain for seven out of eight incoming clock pulses, allowing ample time for the CEO-CET ripple-carry chain to stabilize. Max clock rate is determined by the first tri-bit's Clock-to-CEO delay ($T_{CKO} + T_{ILO}$), plus the CEP input set-up time for all other tri-bits (T_{ICK}), plus the routing delay of the CEP net. In a –70 device this sum can be below 32 ns. The higher tri-bits are not speed critical if they propagate the CET signal in less than eight clock periods, easily achievable for counters as long as 20 tri-bits, i.e. 60 bits.

The two least-significant tri-bits each have a single CE input; they fit, therefore, in only two CLBs each. The higher tri-bits have two Count-Enable inputs (CEP and CET) and require three CLBs.



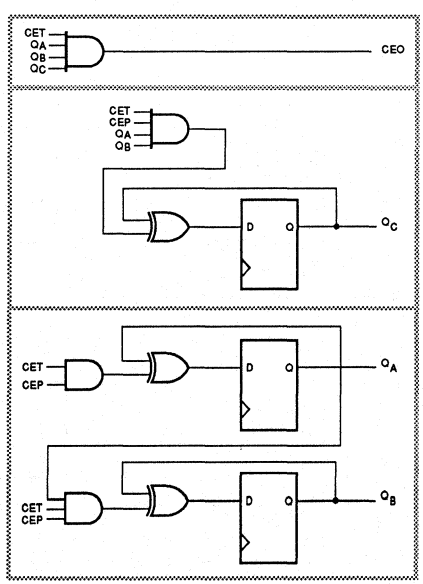
1980 01A

30-MHz Non-Loadable Binary Counter, Expandable up to 60 Bits



1980 02A

First and Second Tri-Bits Use Two CLBs Each



1980 03A

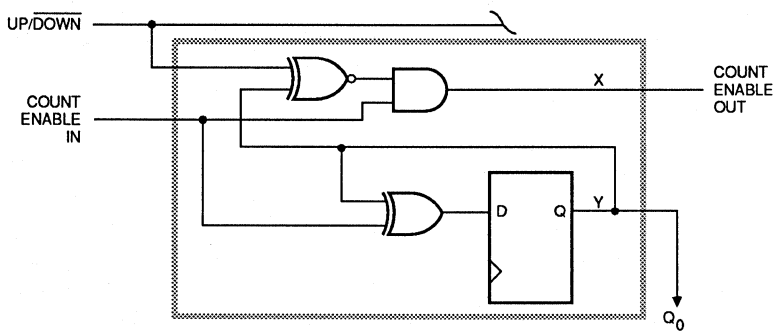
All More Significant Tri-Bits Use Three CLBs

Up/Down Counter Uses One CLB per Bit

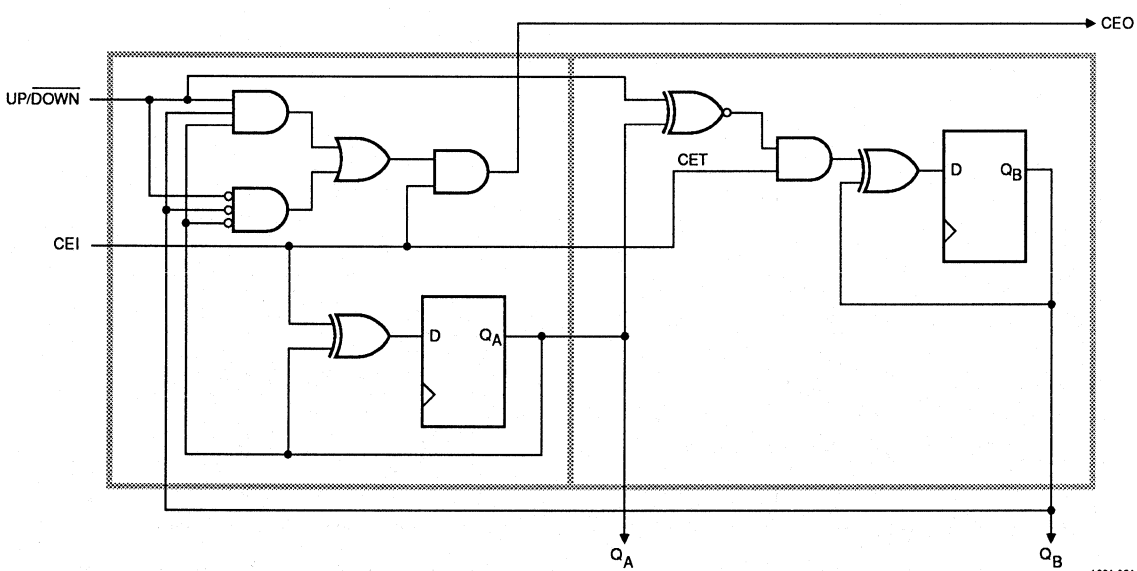
Application Brief BY PETER ALFKE

A fully synchronous resettable but non-loadable up/down counter of arbitrary length can be implemented with only one XC2000 CLB per bit. This design cascades the toggle information from the least-significant toward the most-significant position. Such an architecture reduces the maximum clock rate for longer counters, from 30 MHz for 2 bits, to 10 MHz for 8 bits, down to 5 MHz for 16 bits, assuming a -70 part. This simple design is, therefore, not suited for high-speed clocking, but it generates fully synchronous outputs, i.e., all flip-flops clock on the same edge.

The better functionality of the XC3000 CLBs can cut the cascaded toggle-control delay in half by looking at two counter bits in parallel. This doubles the max frequency for a given counter size. A 16-bit counter in a -70 part can count 10 MHz, guaranteed worst case.



1981 01A



1981 02A

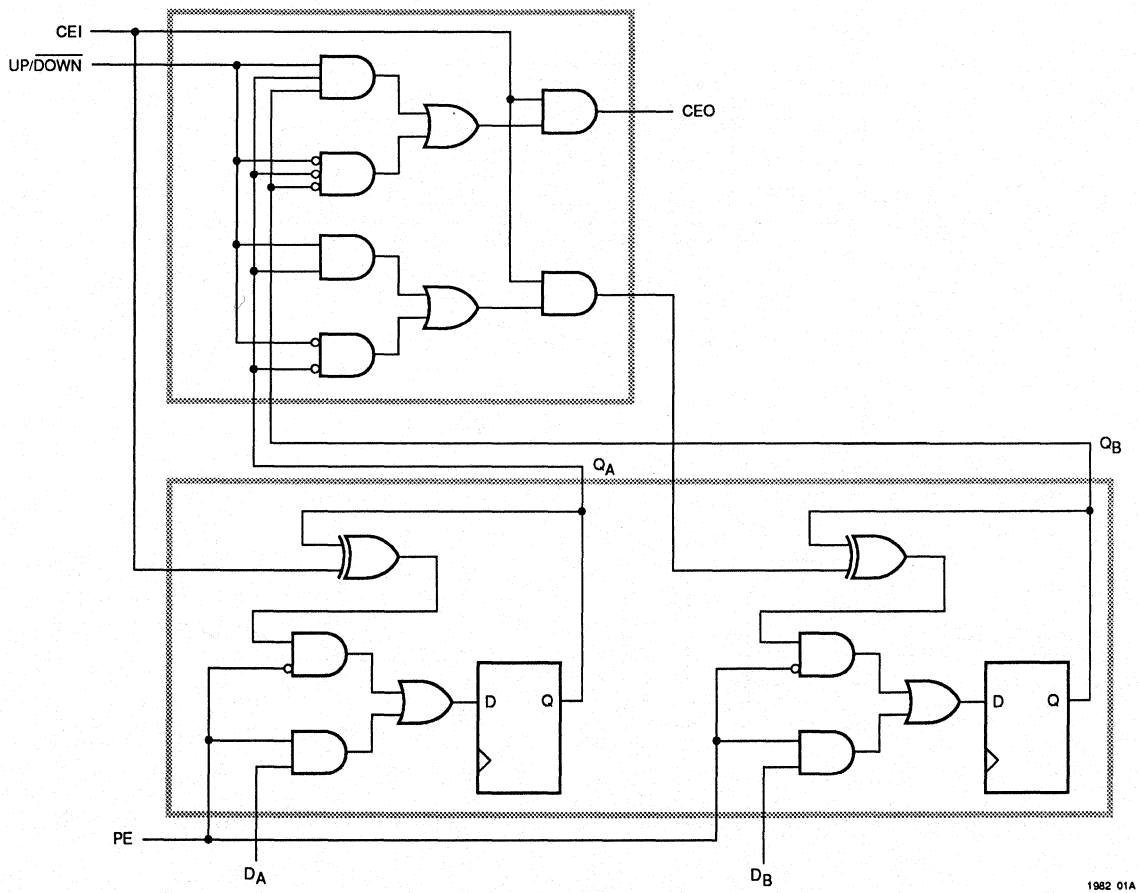
Loadable Up/Down Counter Uses One CLB per Bit

Application Brief BY PETER ALFKE

The 5-input function generator of the XC3000 family CLBs makes it possible to build expandable fully synchronous *loadable* up/down counters of arbitrary length using only two CLBs per two bits, i.e. one CLB per bit.

The basic concept is similar to the non-loadable up/down counter described on the previous page. The function generator driving the counter flip-flop has two additional inputs (Parallel Enable and Data). The cascaded toggle control circuit is moved to a separate CLB which serves two counter bits simultaneously. This cuts the effective ripple delay in half. A 16-bit counter in a -70 part can count 10 MHz, guaranteed worst case.

The CEP/CET speed enhancement cannot be used on up-down counters that might reverse their direction of count in any position. They can, therefore, not guarantee a defined number of clock periods for the ripple-carry chain to stabilize.



30 MHz Binary Counter with Synchronous Reset/Preset

Application Brief BY PETER ALFKE

In many applications, design modularity is more important than highest clock speed and best space efficiency. A counter design is described here that uses identical CLB primitives, one CLB per bit. The Count-Enable Trickle/Count-Enable Parallel concept, introduced by the 74160 family, is changed here to a 1-bit block size. Any block increments only if both Count Enables are High, but the outgoing Carry (C_{OUT}) is not a function of CEP. The CEP input thus prevents erroneous counts while the ripple carry chain is settling.

A shorter counter (six bits or less) drives the CEP net from the Q_0 output, achieving a 40-MHz speed. A longer counter generates a 1-in-4 duty cycle on CEP and runs at 30 MHz up to 12 bits long, or at 25 MHz up to 18 bits long as shown below. To achieve this performance, CEP and $\overline{R}$ must be driven by long lines.

Figure 3 shows a variation of the circuit in Figure 2, where the synchronous Reset input ($\overline{R}$) is changed to a synchronous Preset ($\overline{P}$). Any counter chain can use a mixture of these two circuits to preset the counter to an arbitrary predetermined value.

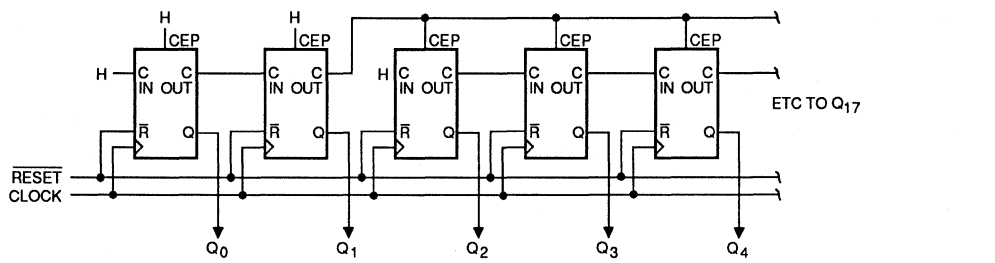


Figure 1. Long Counter (up to 18 bits)

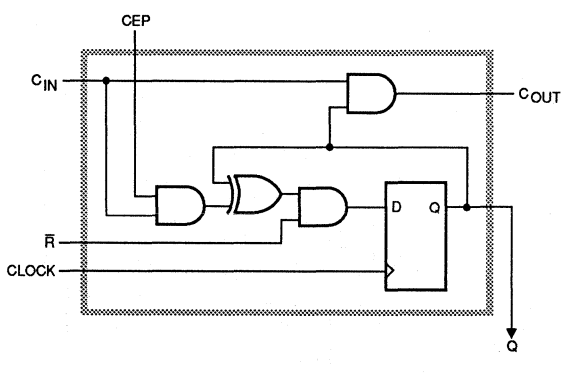


Figure 2. CLB Primitive with Reset, One per Bit

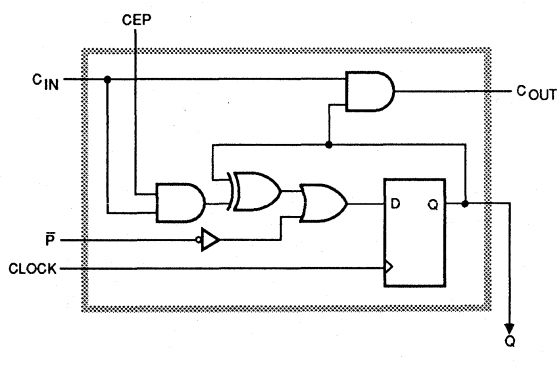


Figure 3. CLB Primitive with Preset, One per Bit

Fast Bidirectional Counters for Robotics

Application Brief BY PETER ALFKE

The position of a robotics arm is usually determined by three shaft encoders consisting of up/down pulse generators and counters. At a maximum speed of 5 meters per second and a resolution of 1 micron, these counters must resolve 0.2-µs pulses and should have a capacity of at least 2 million steps. The counters must have an easy interface to the microprocessor so that the count value can be read on-the-fly, without ambiguity.

The established microprocessor peripheral counters have severe limitations. They are too short, lack up/down control or quadrature clock inputs, and cannot be read easily.

Now Xilinx suggests a design that packs three 22-bit counters into one LCA, the XC3020. Max count rate is 8 MHz, and the count values can easily be read on-the-fly. The counter architecture is somewhat unconventional. Each counter consists of two parts:

1. A conventional up/down 4-bit Grey-code counter with a capacity from -8 to +7. This counter is asynchronous to the system clock, affected only by the incoming clocks.
2. A 20-bit up/down counter in the form of a 20-bit recirculating shift register, a serial adder/subtractor, and a carry/borrow flip-flop. This shift register forms the most significant part of the counter. Synchronous with the LCA clock, it is easily synchronized to the microprocessor clock. At a 20-MHz clock rate, it recirculates once and can be incremented, decremented, and also read or preset, once per microsecond.

Communication between these two parts of the counter is through a carefully controlled mailbox. Whenever the 4-bit up/down counter reaches plus or minus 8, it sets a carry or a borrow flip-flop. The shift register counter accepts these inputs synchronously, with a max delay of 1 µs.

When the microprocessor wants to read the counter, it first disables the interaction between the two parts of the counter. Then both parts are transferred into 24 output registers and the counter interaction is enabled again. This mechanism insures reliable read-out, even if the counter is oscillating around certain critical values.

The problem of a traditional up/down counter is that it can oscillate between two values where all (or most) counter bits change at the incoming count rate. This makes a reliable microprocessor interface virtually impossible.

In this design, the most significant 20 bits of the counter do not have this problem, and the least-significant four bits count in a Grey code, where only one bit changes on any clock transition. Such counters can safely be read on-the-fly. This safe and compact design puts one additional burden on the microprocessor: The two parts of the counter must be added in software, since they have independent signs.

Speed can be increased to 20 MHz by changing the partitioning from 4/20 bits to 8/16 bits. The up/down count control can be implemented in several different ways.

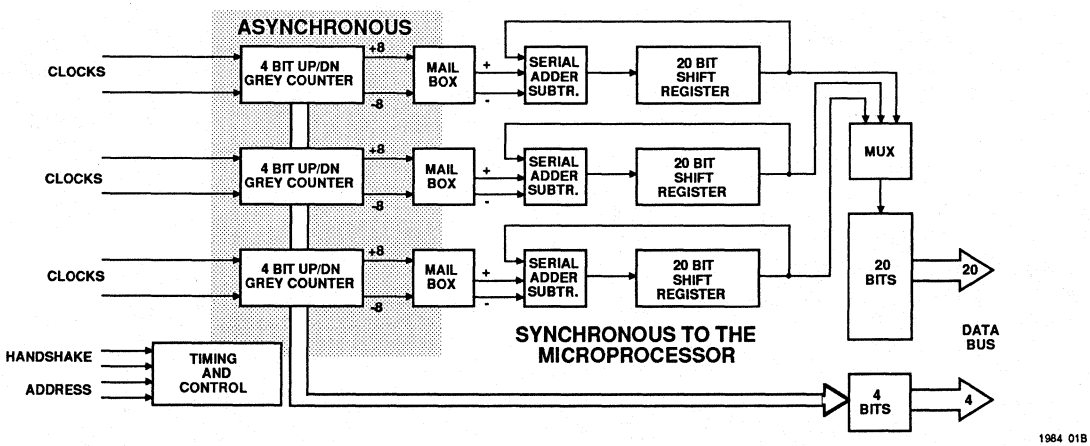


Figure 1. Triple 22-Bit Up/Down Counter with Microprocessor Interface

40 MHz Presettable Counter

Application Brief BY PETER ALFKE AND PERRY WU

A new counter architecture, described here, is used to implement a very high speed presettable, up-to-40-bit long binary counter in an XC3020 LCA devices. The design can easily be modified to implement two 20-bit counters or the equivalent BCD counters.

Traditional counter designs always represent a compromise between two conflicting goals: highest clock speed/event resolution on one hand, sophisticated features (like preset to any arbitrary value, or decode any state) on the other hand.

Asynchronous ripple counters offer highest speed, but cannot be decoded in one clock period, thus cannot be made programmable.

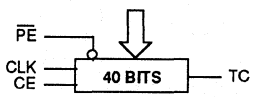
Synchronous counters permit decoding and presetting in one clock period, but pay for this with complex carry logic. Carry propagation is always the limiting factor in the traditional design of presettable synchronous counters, since the complete carry chain must reach a steady state before the next incoming clock edge. Brute force parallel decoding of all previous states becomes unmanageable beyond eight stages, but cascaded decoding introduces additional delays. Either approach reduces the inherent resolution of the counter.

Decoding Terminal Count (TC) to preset the counter again poses a similar problem. The design described here separates the two functions of the carry chain as follows:

- One propagates the carry signal from the less-significant to the more-significant bit positions, and causes the appropriate flip-flop to toggle.
- One cascades the decoding of the terminal count of the whole counter and generates a Parallel Enable signal

CASCADED TC DECODING

The TC decoder must receive inputs from all counter bits, but only the LSB timing is critical; the more-significant bits have been stable for many clock periods. TC can, therefore, be decoded in a slow gating chain that starts at the most-significant end of the counter.



1145 01

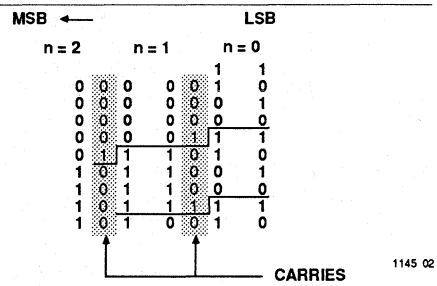
CARRY PROPAGATION

Since a presettable counter only decodes one state, TC, the decision to toggle any of the more-significant bits can be delayed and thus pipelined without any problem.

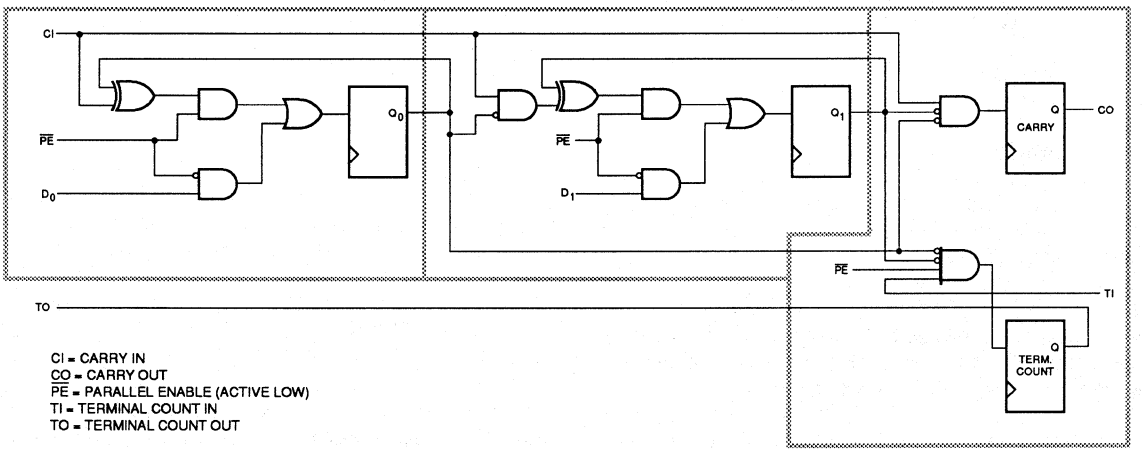
The counter is divided into a number of small sections, each two bits (a di-bit) long, implemented as a synchronous presettable down-counter, with carry-in (=count enable), parallel enable and two data inputs. Terminal count (0.0) is decoded with an additional input coming from the next higher section. The least-significant section decodes the state prior to TC; its output activates the parallel enable for all counters. The carry function between sections is pipelined. The carry flip-flop is set when carry-in is active and the di-bit is in state 00. The carry flip-flop stays set for only one clock period; its output drives the carry-in function of the next higher section. As a result of this pipelining, the counter can be made arbitrarily long without any speed penalty. Note that each di-bit, except the first, makes its transition n clock pulses later than required by the binary code sequence (n is the relative position of the di-bit, n=0 for the input di-bit). This code violation has no impact on TC decoding. This counter can be four times faster than presently available standard microprocessor peripherals like the 8254 and 9513. Typical applications are in instrumentation and communications, for example, as the frequency-determining counter in a phase-locked-loop frequency synthesizer.

SUMMARY

Unlike the speed of conventional synchronous counters, the speed of this design is independent of its length. All speed-critical paths are single level; their interconnect delay can be kept below 9 ns, which means that even a -70 device can count at a 40-MHz rate (worst case).

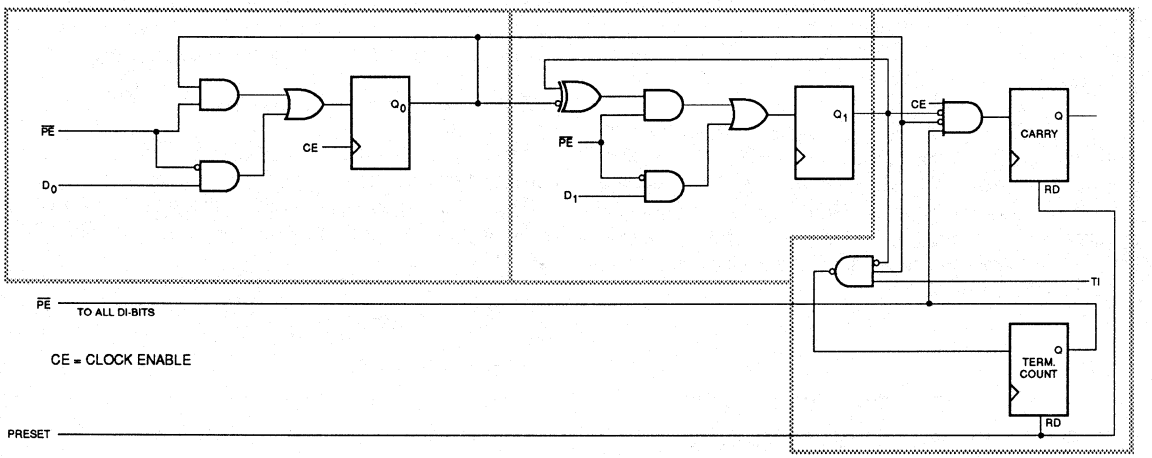


1145 02



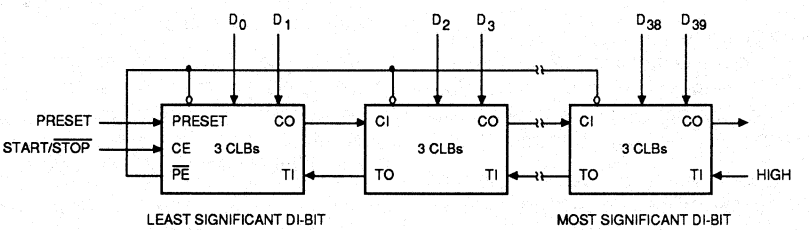
1145 04B

Any Di-bit Except the Least Significant



1145 05B

Least Significant Di-bit



1145 03A

Synchronous Presettable Counter — 40 Bits in 60 CLBs

Since this circuit was first published in mid 1988, several designers have used it to create fast counters.

What is the function of the TC pipeline flip-flop, formerly called Q3?

The unconventional idea behind this counter design is that Terminal Count decoding can be "rippled" from the MSB to the LSB, i.e. against the direction of carries. This is possible because the high order bits reached their TC long before the LSB does.

There is, however, a potential problem when the counter is being preset to a value with a string of LSB zeros. Let's assume the worst case where the preset value is all zeros except a single one in the MSB position:

When this counter reaches the all-zero Terminal Count, $\overline{PE}$ is activated and the counter is preset. This action should obviously de-activate the TC decoding, but in the given example a simple ripple decoder would have a very long delay. It might take 400 ns for the MSB =1 condition to ripple down through a 40-bit decoding chain. Such a delay would defeat the concept of the counter, reducing its max clock rate to 2.5 MHz. A better way must be found to de-activate TC within 25 ns.

The TC pipeline flip-flop and the inclusion of $\overline{PE}$ in the AND gate that detects TC, reliably de-activate TC and thus $\overline{PE}$ one clock after they have been activated. This has one side effect, however: It makes it illegal to preset the counter to very small numbers (less than 10 for a 20-bit counter), since the TC-pipeline takes that many clock pulses to become active again.

In the unlikely case where this might cause a problem, most TC pipeline flip-flops can be eliminated. They were inserted to simplify modeling and because they are available for free.

Why is the least significant di-bit different?

To achieve a 40-MHz clock rate, the $\overline{PE}$ signal must be made as fast as possible. It has to come directly from a flip-flop output so that the sum of clock-to-output delay, routing delay, and input set-up time is kept below 25 ns.

The position of the LSB TC pipeline flip-flop is, therefore, changed, so that it detects the TC-1 state (in a down-counter, that is state 1).

The flip-flop output is made active Low $\overline{PE}$ so that the asynchronous clear input can be used to force the counter into loading.

For operation below 30 MHz the least significant di-bit can be like all the other dibits, but $\overline{PE}$ must be excluded from the AND gate generating $\overline{PE}$, and the user may want to adjust the polarity of the last TC pipeline flip-flop to facilitate the preset function mentioned above.

Where should this design be used?

This counter design achieves high performance by using several logic "tricks". It generates incorrect outputs when undigested carries sit in the carry flip-flops. That makes this design useless for any parallel application like DMA counters.

For the intended application, timebase counters or frequency synthesizers, this design offers the highest possible count speed.

ASYNCHRONOUS PRESET IN XC3000 CLBS

The XC3000 CLB lacks the asynchronous preset capability available in the XC2000 CLB. Some designers are looking for this feature. Here are several solutions:

1. If asynchronous preset is needed, but no asynchronous clear:

   Turn the flip-flop upside down, i.e. invert the D input and the Q output and consider the asynchronous clear a preset. These inversions of D and Q come for free in a Xilinx LCA. Note that the flip-flop will now come out of configuration in the apparent preset state.

2. If the circuit needs both asynchronous preset and clear, chances are that the function can be performed by a latch. The XC3000 CLB can implement complex latches in its function generators (see page 6-27).

3. If the circuit really needs asynchronous preset and clear (or asynchronous data transfer) in a flip-flop, the problem must be solved on a system level.

The design can usually be transformed into a synchronous solution where all flip-flop changes occur as a result of the same clock edge.

Truly asynchronous parallel data transfer into several clocked flip-flops simultaneously is inherently unreliable and must be avoided. If, however, the transfer pulse is synchronized with the clock, it should not be too difficult to change the design to utilize the clock for loading.

Asynchronous data transfer was popular in early TTL MSI circuits designed in the late sixties, e.g., the 7494 and 7496. It is time to get away from the limitations of the past.

Frequency/Phase Comparator for Phase-Locked-Loops

Application Brief BY PETER ALFKE

A Phase-Locked-Loop (PLL) manipulates a local voltage-controlled oscillator (VCO) so that it is in phase with a reference signal. One popular application is a programmable frequency synthesizer for radio communications. Here a crystal oscillator is divided down to a low reference frequency of 5 kHz, for example.

A programmable divider scales the VCO frequency down to the same reference frequency. The two counter outputs are compared to generate a signal that, when required, modifies the VCO frequency up or down until the two comparator inputs are not only of the same frequency, but also in phase.

This frequency/phase comparator must have a wide capture range, i.e. it must generate the appropriate output, not only to pull in a small phase error, but also to correct a large frequency error. It may not generate false outputs when the input is at a multiple or fraction of the desired frequency. The well-known circuit shown in Figure 1 performs this function. It generates "pump-up" pulse when the VCO frequency is too low, "pump-down" when its too high. The multiple feedback network assures proper operation even at large frequency errors.

Figure 2 shows this circuit implemented in two CLBs plus two IOBs, directly driving the integrator (low pass filter) controlling the VCO. The LCA solution has been breadboarded at 10 MHz. It achieved a phase error of less than 2 ns.

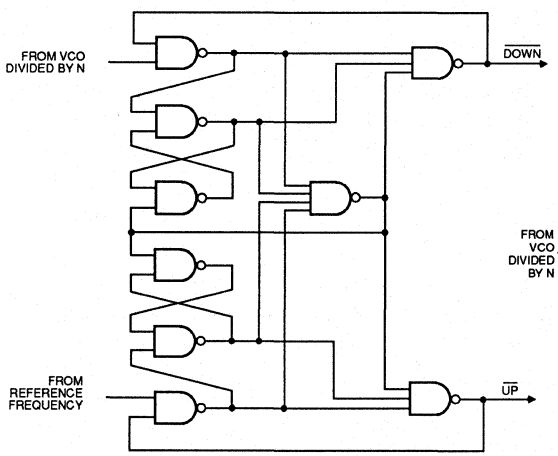


Figure 1. Digital Frequency/Phase Detector

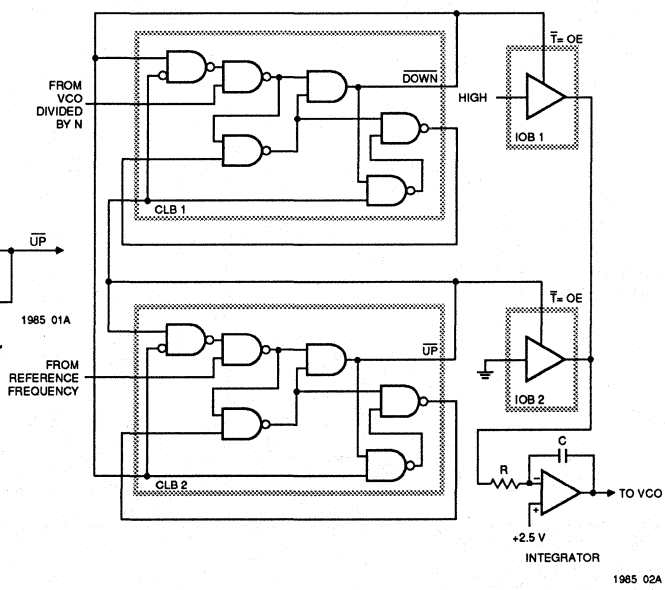


Figure 2. Frequency/Phase Detector Using Four Blocks

Gigahertz Presettable Counter

Application Brief BY PETER ALFKE

Some frequency synthesizers for communications, e.g., cellular telephone networks, require a clock frequency of hundreds of megahertz, up to a gigahertz. Obviously, the LCA cannot operate quite that fast, but with the help of a 2-modulus prescaler, the LCA can implement a fully presettable ultra-fast counter, resolving time in increments of one clock period, as small as 1 ns at 1 GHz.

Prescaling is the obvious method to adapt a slow device to a high clock rate. Simple prescaling by a fixed number, e.g. 8, 16, or 64, however, reduces not only the clock rate, but also the resolution. If, for example, the GHz clock of a phase-locked-loop synthesizer is first divided by 64, then the whole presettable counter is clocked at this lower rate. For a 25 kHz channel spacing, the PLL must, therefore, operate at 25 kHz + 64, i.e. less than 400 Hz. This results in slow response and might produce excessive phase jitter.

A "Pulse Swallowing" 2-modulus prescaler, originally described in 1970 by John Nichols of Fairchild Semiconductor Applications, avoids this drawback. Pulse swallowing combines a fast but dumb counter (the prescaler) with a smart but slow counter (in the LCA) to achieve the performance of a fast and smart, fully presettable counter.

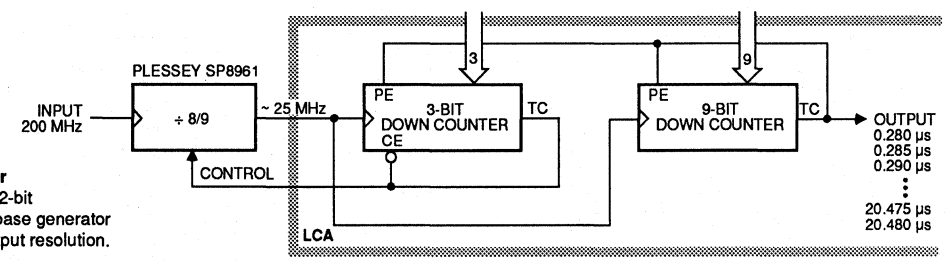
The prescaler divides by either n or n + 1, depending on the state of the control input. In other words, it "swallows" one additional clock pulse if told so by the control input. By keeping the control input active for the appropriate number of prescaler output periods, the LCA can fine tune the total divide ratio to any integer number.

Well, there are some impossible numbers:
When the prescaler divides by either n or n + 1, then the system cannot divide by certain numbers below n (n-1).

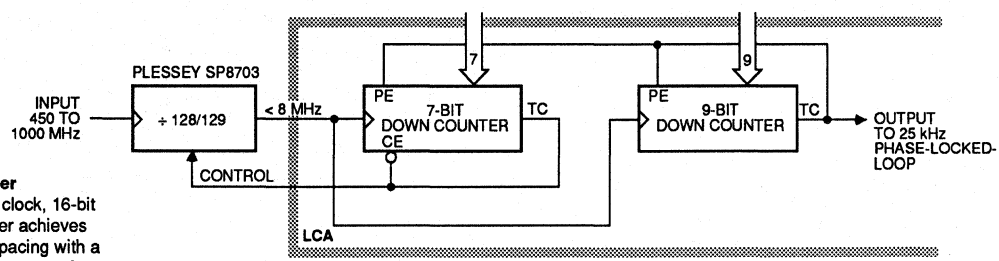
An 8/9 prescaler has blind spots below 56
A 64/65 prescaler has blind spots below 4,032
A 128/129 prescaler has blind spots below 16,256

This limitation is usually of no practical consequence in a real design.

The prescaler-LCA combination can divide by **any** integer number higher than the values above.

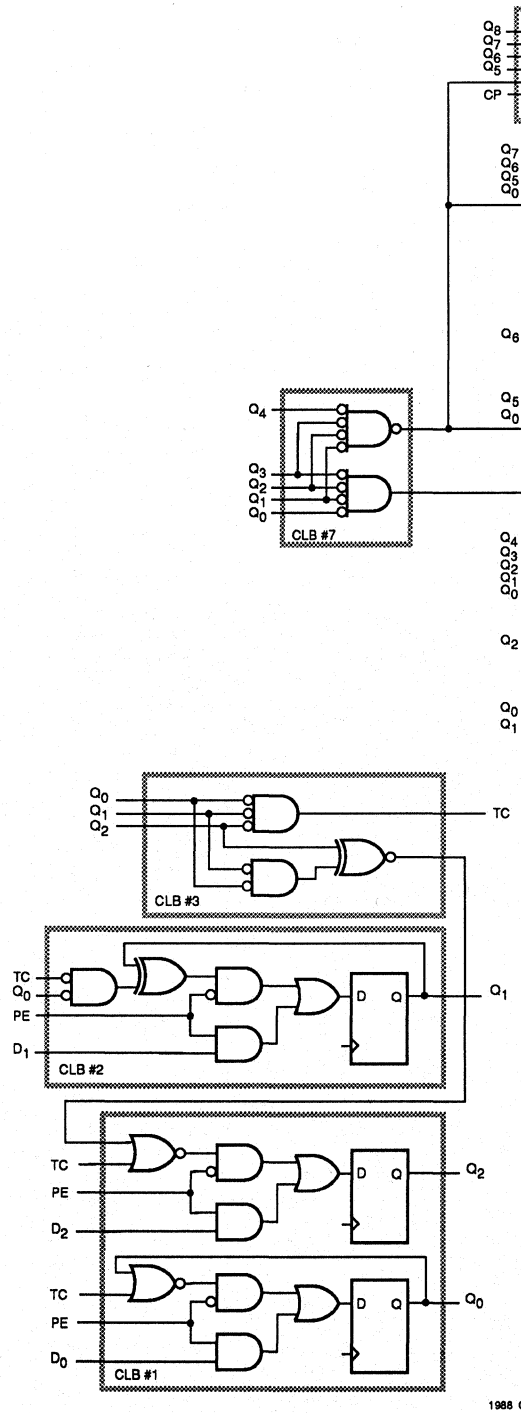


200-MHz Counter
200 MHz clock, 12-bit presettable time base generator achieves 5 ns output resolution.

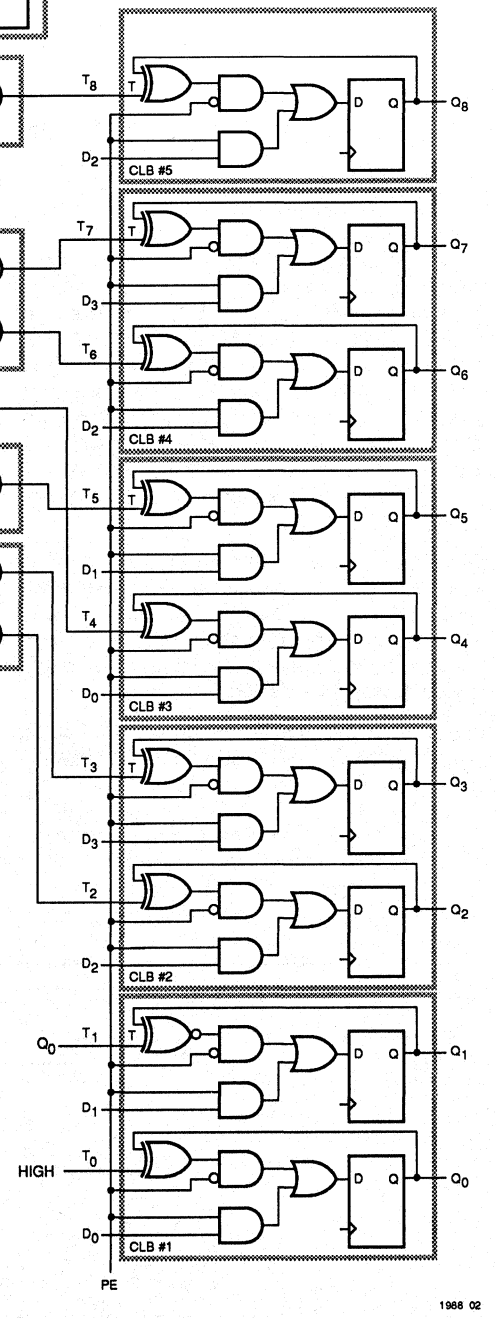


Gigahertz Counter
450 to 1000 MHz clock, 16-bit presettable counter achieves 25 kHz channel spacing with a 25 kHz phase comparator frequency.

1988 03A



3-Bit Presettable Down Counter with Pipelined Terminal Count, Locking Up on TC



9-Bit Presettable Down Counter with Decoded Terminal Count (TC)

75 MHz Presettable Counter or Programmable Delay

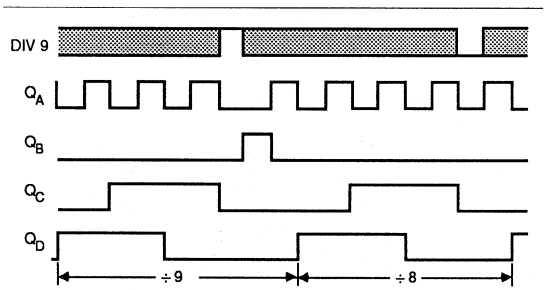
Product Brief BY PETER ALFKE

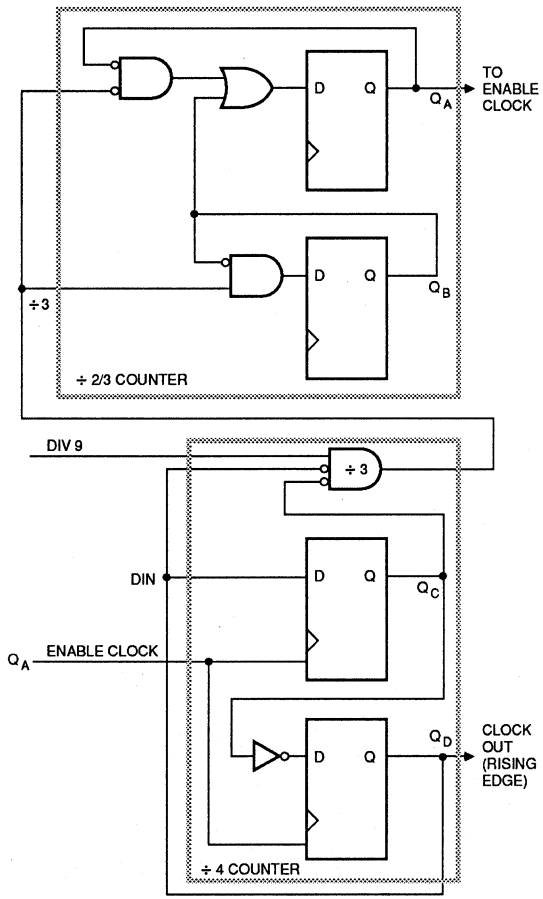
FEATURES

The ÷8/9 prescaler described on page 6-42 can also be implemented inside an LCA. The highest clock frequency for a -100 part is 75 MHz, i.e. the output delay can be programmed with a granularity of 13 ns. The ÷8/9 prescaler consists of a ÷2/3 counter followed by a ÷4 counter with one decoded state. Each of these counters fits into a single CLB. The ÷2/3 counter divides by 2 unless the 3-input AND is true, in which case it divides by 3. When the DIV9 input is Low, the two counters together divide by 2+2+2+2=8. When the DIV9 input is High, the two counters divide by 2+2+2+3=9. See page 2-42/43 for a more detailed description of such a pulse-swallowing counter.

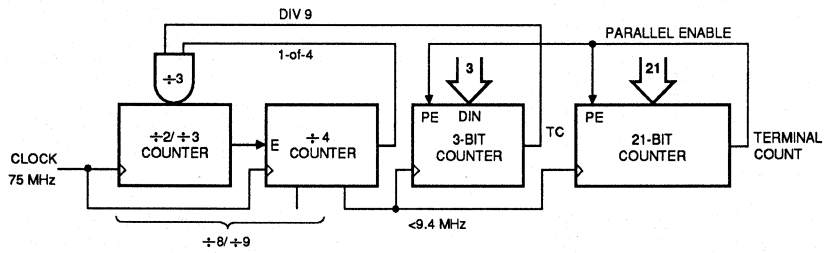
This design demonstrates the high performance possible with Xilinx LCAs when the user is willing to optimize the system design to fit the available logic. The high clock resolution of 75 MHz is partly due to a system "trick" (pulse-swallowing), partly due to the inherent flexibility, and high speed of the CLB function generators.

A conventional 24-bit presettable counter would be limited to a clock rate of 13 MHz. This pulse-swallowing design is six times faster.





Divide by Pulse-Swallowing Prescaler



24-Bit Frequency Division with Pulse-Swallowing Prescaler

Serial Pattern Detectors

Application Brief BY PETER ALFKE

FIXED PATTERN DETECTOR

This circuit compares a serial bit-stream against a predetermined (configured) pattern. Two bits are compared in each XC3000-series CLB. The outputs of the comparator are ANDed in with 3-state buffers on a long line.

Data is shifted through DIN into the Y-flip-flop, then shifted through the upper half of the combinatorial array into the X-flip-flop of the same CLB. From there it is routed to the DIN input of the next CLB.

The lower half of the combinatorial array compares the content of the two flip-flops against data supplied on the A and D inputs. A match is indicated on the G output and routed to a 3-state buffer driving a long line.

DYNAMIC PATTERN DETECTOR OR CORRELATOR

This circuit compares a serial bit stream against a previously shifted-in pattern, using only one XC3000-series-CLB per pattern bit. The output of the comparators are ANDed with 3-state buffers on a long line. The desired pattern is first shifted through the DIN input into the Y-flip-flop, and then routed to the DIN input of the next CLB.

When the complete pattern has been shifted in, it is transferred with one clock pulse to the X-flip-flops, using the lower half of the function generator. Data to be detected is then shifted in through the DIN input into the Y-flip-flop, and from there to the DIN input of the next CLB. The upper half of the function generator compares the content of Qx and Qy, and indicates a match on the CLB output. For identity comparison, these outputs are ANDed through 3-state buffers driving a long line.

This circuit can also be used as a correlator, in which case the outputs must be summed in a Wallace-type adder.

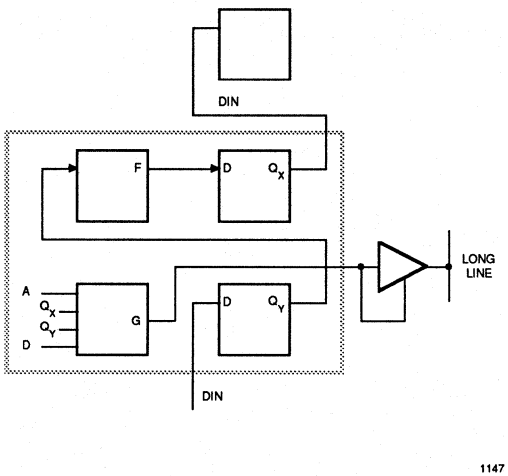


Figure 1. Fixed Pattern Detector

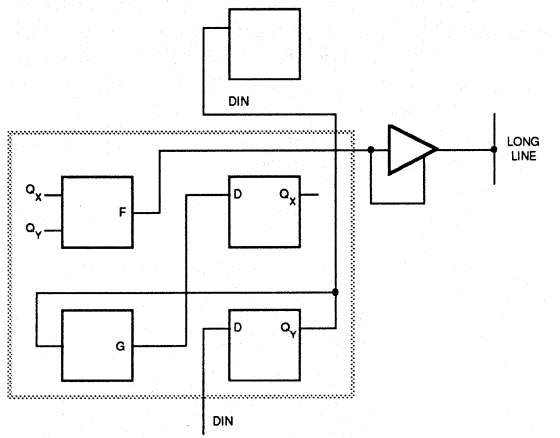
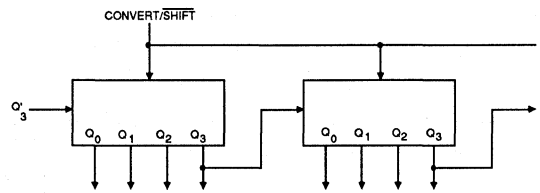


Figure 2. Serial Comparator Finds Pattern Match or Correlates Patterns

Serial Code Conversion Binary to BCD

Application Brief BY PETER ALFKE



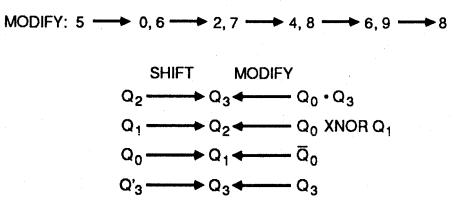
1146 03

Figure 1. Binary to BCD (MSB First)

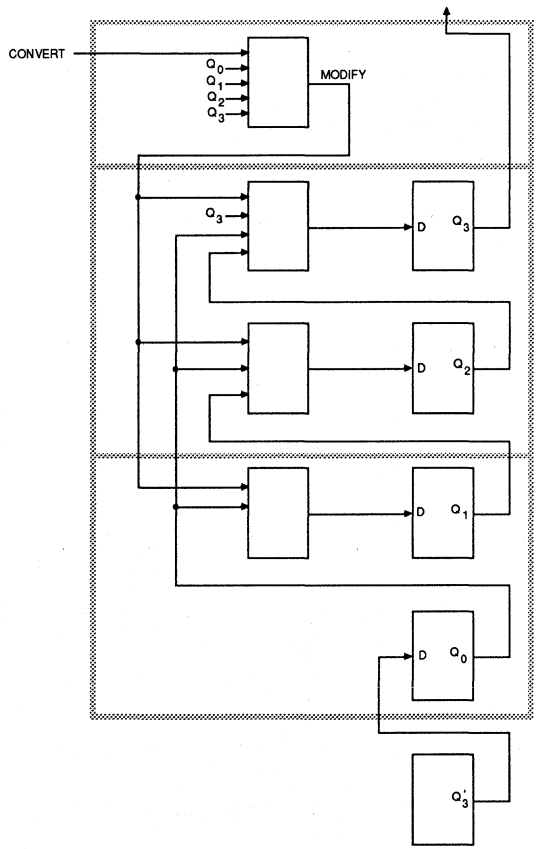
The LCA architecture with its powerful function generators evenly interspersed between flip-flops lends itself very well to serial code conversion, where data is shifted into a register in one format, and shifted out of the same register in a converted format.

A binary-to-BCD converter requires three CLBs for every four bits of BCD output i.e., for every digit. Data is shifted in serially, most significant bit first. Each shift thus doubles the content of the register.

To remain a valid BCD number, a 4-bit number of 5 or greater must not just be shifted, but must be converted into the proper BCD representation of its doubled value: A one is shifted into the next higher decade and the 5 is converted into a 0, a 6 into a 2, a 7 into a 4, an 8 into a 6, a 9 into an 8. When the binary LSB has been shifted in, BCD data is available in parallel form, or it can be shifted out serially with the conversion logic disabled.



X1246

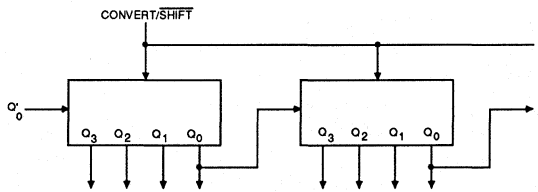


1146 01A

Figure 2. Binary to BCD converter. Three CLBs per Four Bits (MSB First)

Serial Code Conversion BCD to Binary

Application Brief BY PETER ALFKE



1146 04

Figure 1. BCD to Binary (LSB First)

The LCA architecture with its powerful function generators evenly interspersed between flip-flops lends itself very well to serial code conversion, where data is shifted into a register in one format, and shifted out of the same register in a converted format.

A BCD-to-binary converter requires three CLBs per digit. BCD data is shifted in, least significant bit first. Once the complete BCD word has been shifted in, the conversion process begins, shifting out binary data, LSB first.

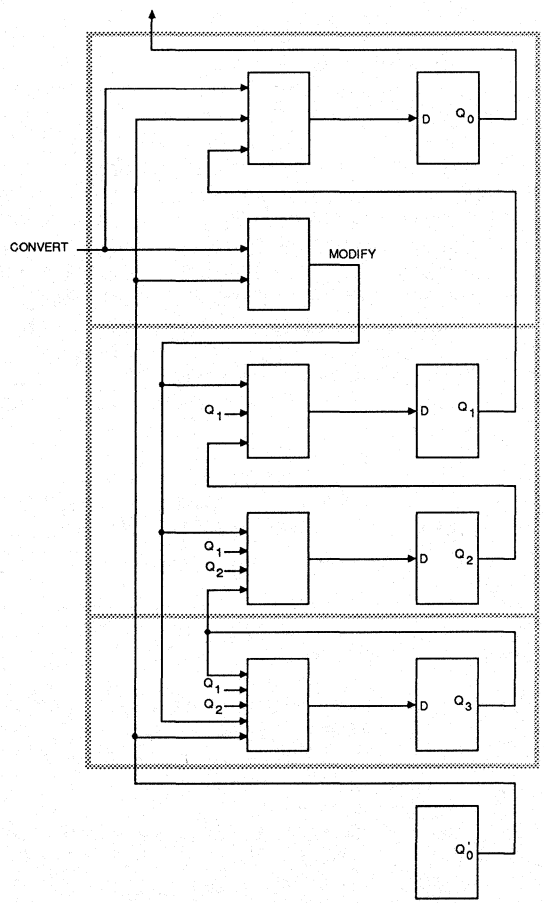
Each shift divides the content by two. When the LSB of a BCD digit is a one, shifting it one position down would give it a weight of 8 in the lower decade instead of the weight of 5 appropriate for a 10 divided by 2. A value of 3 is therefore subtracted from the content of the decade whenever a one is being shifted into it.

This design can be made smaller and faster by starting the conversion before the most significant BCD digit is being shifted in. Since these converters can be laid out with very short interconnect delays, they can operate at up to 60% of the specified toggle frequency, i.e. 42 MHz for the -70 parts.

MODIFY: 0 → 5, 2 → 6, 4 → 7, 6 → 8, 8 → 9

	SHIFT	MODIFY
Q_1 →	Q_0 ←	$\overline{Q}_1$
Q_2 →	Q_1 ←	Q_1 XOR Q_2
Q_3 →	Q_2 ←	$\overline{Q}_3$ AND ($\overline{Q}_1$ OR $\overline{Q}_2$)
Q'_0 →	Q_3 ←	Q_3 OR ($Q_1 \cdot Q_2$)

X1247



1146 02A

Figure 2. BCD to Binary converter. Three CLBs per Four Bits (LSB First)

"Corner Bender" or 8-Bit Format Converter

Application Brief BY PETER ALFKE

Pulse Code Modulation (PCM) has become the dominating encoding method in digital telephony. Analog signals are sampled at 8 kHz and represented by their 8-bit digital equivalent, using a logarithmic encoding scheme, μ-law in the US and Japan, A-Law in the rest of the world using the CCITT standard.

These eight bits are usually transmitted serially (the T1 standard time-multiplexes 24 channels on a single wire at 1.544 MHz. The CCITT standard time-multiplexes 32 channels at 2.048 MHz.

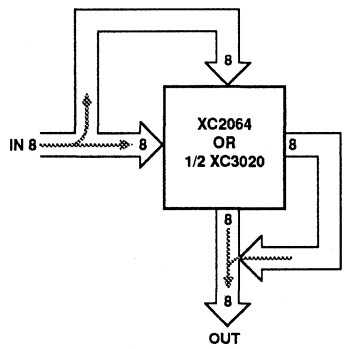
In the central office or PBX, however, the eight bits representing one particular sample must be routed together. The telephone system thus uses a large number of serial-to-parallel and parallel-to-serial converters, all operating on 8-bit words, all running synchronously. Eight S-P converters with eight data inputs and eight data outputs can easily be combined in one package. Eight serial data streams are shifted in simultaneously. After eight clock pulses the eight serial words can be shifted out in parallel, one word per clock pulse, and new serial bits can be shifted in simultaneously. It is interesting to note that the same circuit can also accept parallel words and shift them out in eight serial streams. The difference between S-P and P-S is not in the circuit, but in the mind of the beholder.

Such a "Corner Bender" is available as a standard part, the Plessey MJ 1410 8-Bit Format Converter. Its drawbacks are high power consumption (max 500 mW) and slow speed (2.4 MHz guaranteed worst case), a result of its NMOS heritage.

The LCA implementation of a 2-dimensional shift register is straightforward:

A common clock drives all flip-flops, organized in an 8 x 8 array. In mode A, each flip-flop receives data from its "left" neighbor; in mode B, each flip-flop receives data from its neighbor above.

For the first eight clock pulses, the array is in mode A, receiving eight bit streams and right-shifting them into the array. For the next eight clock pulses, the array is in mode B, down-shifting the previously received 64 bits.



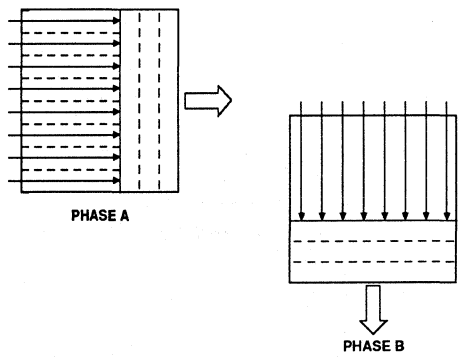
1122 03

New serial data can be shifted in from one side, while old parallel data is being shifted out at the opposite side. There is no need for any of the additional flip-flops required by the older designs.

After eight clock pulses, the mode control is again changed to A and old data is shifted out on the right side while new data is shifted in from the left.

This design uses only 64 flip-flops, and a mode-control signal derived from a divide-by-8 counter. The physical routing of the input signals can be done on-chip, but the eight bottom output pins can be externally combined with the eight right-hand outputs in a wired OR.

The design fits exactly into one XC2064 or into half of an XC3020 and can run at up to 50 MHz.



1122 02

100 MHz Frequency Counter

Application Brief BY PETER ALFKE AND NICK CAMILLERI

The block diagram below describes a complete 100-MHz frequency counter in an XC3020 PC84.

A 32,768-kHz crystal oscillator generates a time base of two seconds. The frequency to be measured clocks an 8-digit BCD counter. At the end of the measuring period of two seconds the counter content is transferred into four shift registers, and the counter is then reset before the beginning of the next measuring period. The shift register drives a 7-segment encoder that feeds into the LCD driving logic, which in turn drives seven 8-bit shift registers nestled in the IOBs.

The oscillator uses three IOBs, since the dedicated crystal oscillator input is already used as signal input.

The time base is generated by a 16-bit-binary counter consisting of four asynchronously cascaded 4-bit synchronous counters. The control unit eliminates the clock ripple delay by re-synchronizing the time base output. The eight counter decades are cascaded asynchronously, each decade consisting of a synchronous BCD counter.

The high resolution of 100 MHz or 10ns is achieved by using the divide-by-two flip flop driven by the alternate clock buffer. This is the simplest and therefore fastest flip-flop on the device.

The whole frequency counter uses 60 of the 64 CLBs in an XC3020:

Time Base	8 CLBs
BCD Counter	16 CLBs
4 Shift Registers	16 CLBs
7-Segment Encoder	4 CLBs
Leading Zero Suppressor	2 CLBs
Control	2 CLBs
Segment Conversion/ LCD Driving Logic	4 CLBs
Special Clock Generation	6 CLBs
Miscellaneous	2 CLBs

This design is available from Xilinx. Call the applications hot line 408-559-7778 or 1-800-255-7778.

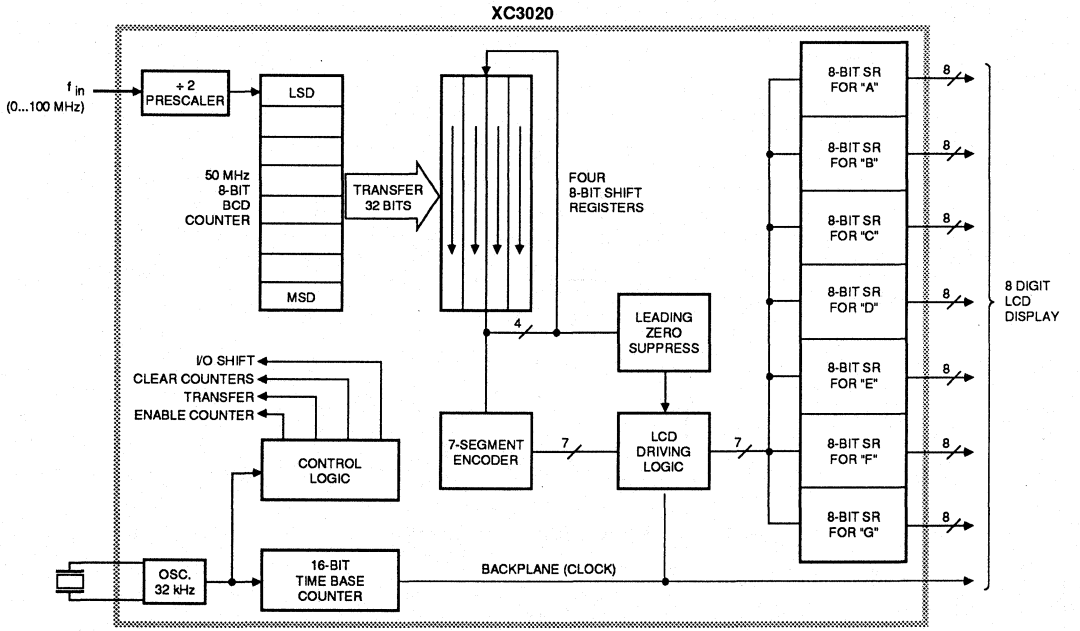


Figure 1. Block Diagram

Megabit FIFO in Two Chips: One LCA and One DRAM

Application Brief BY PETER ALFKE AND NICK CAMILLERI

A bit-serial FIFO buffer is a general-purpose tool to relieve system bottlenecks, e.g., in LANs, in communications, and in the interface between computers and peripherals. Small FIFOs are usually designed as asynchronous shift registers, but a larger FIFO with more than 256 locations is better implemented as a controller plus a two-port RAM, or as a controller plus a single-port RAM, either SRAM or DRAM.

SRAMs are fast and easy to use, but at least four times more expensive than DRAMs of equivalent size. Dynamic RAMs offer low-cost data storage, but require complex timing and address multiplexing, which makes them unattractive in small designs. For FIFOs with more than 256K capacity, a DRAM offers the lowest cost solution, if the controller can be implemented in a compact and cost-effective way. An XC3020 Logic Cell Array can easily perform all the control and addressing functions with many gates left over for additional features.

This FIFO DRAM controller consists of:

- An input/output buffer with synchronizing logic
- A 20-bit Write pointer (counter)
- A 20-bit Read pointer (counter)
- A 20-bit full/empty comparator
- A 4-to-1, 10-bit address multiplexer
- Control and arbitration logic

The Write pointer defines the memory location where the incoming data is being written, the Read pointer defines the memory location where the next data can be read. The identity comparator signals when the FIFO is full or empty.

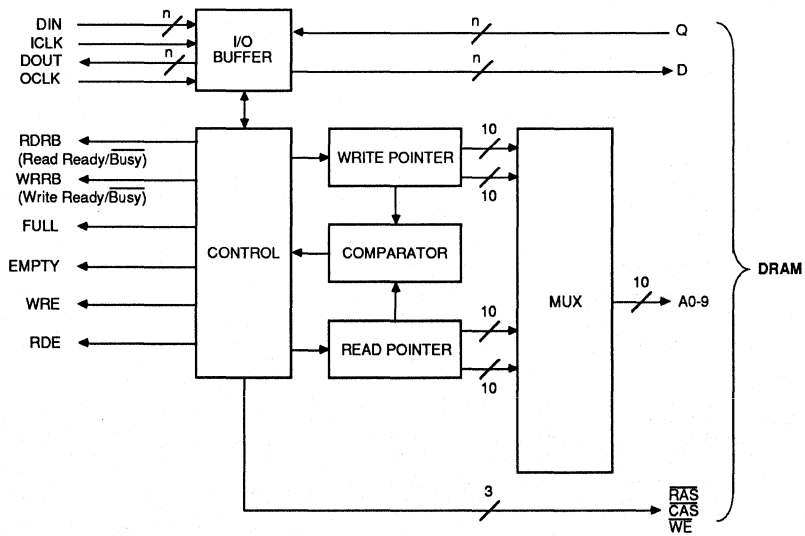


Figure 1. Megabit FIFO Controller in an XC3020

1130 01A

When the Write and Read pointers become identical as a result of a Write operation, the FIFO is full, and further Write operation must be prevented until data has been Read out. When the two pointers become identical as a result of a Read operation, the FIFO is empty and further Read operation must be prevented until new data has been written in. With a single-port RAM, Read and Write operations must be inherently sequential, and there is no danger of confusing the full and empty state, a problem that has plagued some two-port designs.

A straightforward design would use synchronous binary counters for the two pointers, but it is far more efficient to use linear shift-register (LSR) counters. Such counters require far less logic and are faster since they avoid the carry propagation problems of binary counters. LSR counters have two peculiarities: they count in a pseudo-random sequence and they usually skip one state, i.e., a 20-bit LSR counter repeats after $2^{20}-1$ clock pulses. In a FIFO Controller, both these features are irrelevant, the address sequence is arbitrary, provided both counter sequence identically.

This design fits two shift register counter bits in one XC3000-series CLB and the identity comparator uses the combinatorial portion of the same CLB.

The RAS/CAS multiplexing of the 20-bit address is performed without any logic by tapping every other bit of the shift register counter and using the 10 outputs before the incrementing shift as Row address, after the incrementing shift as Column address. (The Column address of any position is thus identical with the Row address of the following position, but since the binary sequence of a shift register counter is pseudo-random anyhow, this is no problem. It's an elegant and efficient trick).

The FIFO controller permits the user to perform totally asynchronous Read and Write operations, while it synchronizes communication with the DRAM. The design

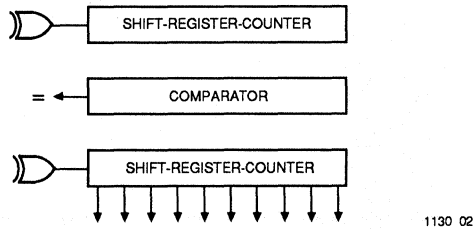


Figure 2. Shift-Register-Counter and Free Row-Column MUX

takes advantage of the DRAM internal refresh counter by using CAS-before-RAS refresh/address strobes.

Both 20-bit pointers, plus their 20-bit identity comparator, plus the Row/Column multiplexer thus fit into only 20 CLBs; refresh timer and address multiplexer use another 10 CLBs and the data buffer plus control and arbitration logic take another 23 CLBs, for a total of 53 CLBs, an easy fit in an XC3020.

This design can easily be modified for 256K DRAMs. Other variations are: multiple parallel bits, e.g., byte-parallel operation, interrupt-driven control, multiplexed data for multiple parallel-bit storage, and byte parallel storage with bit-serial I/O. The latter case requires special attention when the FIFO is emptied after a non-integer number of bytes had been entered, requiring direct communication between the input Serial-to-Parallel converter and the output P/S converter.

This applications brief shows that the XC3020 can be programmed to control one or a few DRAMs as a large FIFO of up to a megabyte, with data rates up to 16 Mbps serially or 2 Mbytes per second byte-parallel.

This design is available from Xilinx. Call the applications hot line 408-559-7778 or 1-800-255-7778.

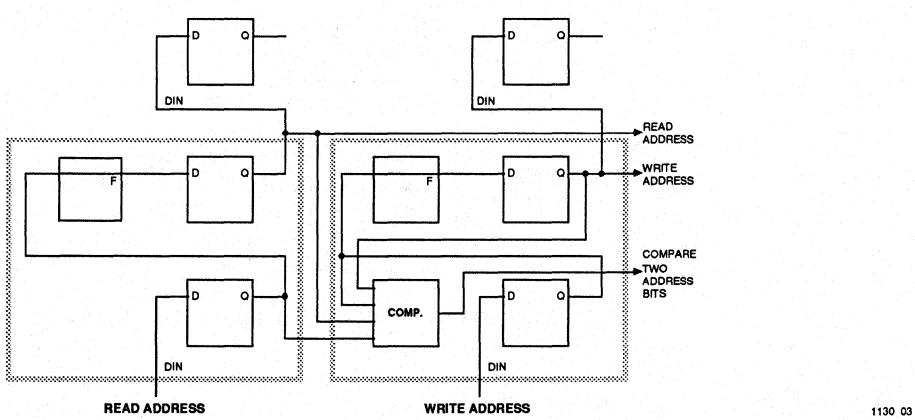


Figure 3. 2-Bit Slice of Two Counters and Comparator in Two CLBs

State Machines

Application Brief BY PETER ALFKE

State-machine design is a methodology that defines the contents of all flip-flops for any possible state of the design, then defines all possible paths that can cause the design to go from one state to another. In its simplest form this is just a rigorous way of designing synchronous logic, like 4-bit counters. For complex designs, the state-machine approach gives the designer a tool to investigate all possible operating conditions and avoid overlooked hang-up states or undesired transitions. LCA devices with their abundance of flip-flops lend themselves well to state-machine designs.

SIMPLE, FAST STATE MACHINES

Using the 5-input function generator of the XC3000-70 family devices as a 32 bit ROM, a state machine with up to 32 states without any conditional jumps uses only 5 CLBs and operates at up to 50 MHz.

The five registered CLB outputs drive the five function generator inputs of the 5 CLBs in parallel. This implements a fully programmable sequencer similar to the synchronous counter shown in the left column of page 6-28.

For a smaller number of states, some inputs can be used as conditional jump inputs. Encoding these condition codes may require an additional level of logic which reduces the maximum clock rate to 30 MHz.

SIMPLE STATE MACHINE RUNS AT 30 MHz

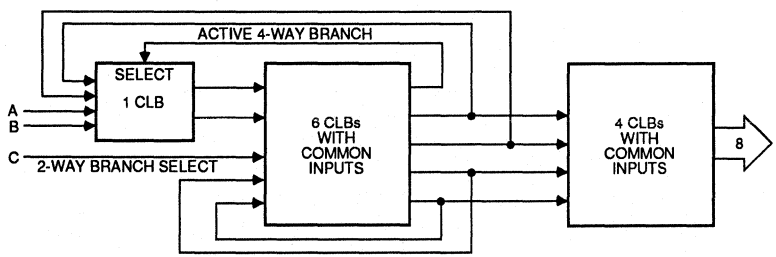
This simple state machine uses only 11 CLBs. It has up to 16 states, and eight outputs, each decoding/encoding any combination of states. It performs a 2-way branch from any state to any one of two freely assigned states, (possibly including the present state) determined by control input C. (Avoid the branch by making both destination states equal).

This design can also perform an 8-way branch from any state so programmed to either one of two selected quadrants (0..3, 4...7, 8...11 or 12...15). Control inputs A,B then determine the location within the quadrant.

Examples:

- From state ③, if C=High, go to ⑤ else go to ⑧
- From state ⑦, if C=High, go to ③ else stay in ⑦
- From state ⑨, unconditionally go to ②
- From state ⑥, execute the truth-table below

A B	C= Low	C=High
0 0	⑫	⓪
1 0	⑬	①
0 1	⑭	②
1 1	⑮	③



1986 01

30-MHz State Machine, 16 States, 2-Way/8-Way Branch, 8 Outputs

Complex State Machine in One LCA

Application Brief BY PETER ALFKE

Simple and fast state machines can easily be implemented in an LCA, as shown on the previous page. This page shows how an external EPROM can be the source of the next address in a complex state machine. This look-up table can easily be hidden in the EPROM required to store the LCA configuration data.

Assume that an XC3020 is configured in the Master Parallel mode, where it reads its configuration data out of a 256K (32K x 8) EPROM, starting at the top address location 7FFF (32K) through 77FF (about 30K). The remaining 94% of the EPROM can be used as a next-state look-up table with a capacity of 240 states.

The state address is read out of the EPROM, then manipulated (decoded, encoded, etc.) in the XC3020 LCA. The result is combined with incoming-control information to generate a new EPROM address. The EPROM can be considered as having 240 locations, each 128 bytes wide. Each byte is a potential next-state value, only one of which will be chosen by the 7-bit condition code.

In the simplest case, the EPROM output data is just latched in the LCA and is fed back as the most-significant part of the new EPROM address. Since the top 16 address locations are used for configuration data, the state codes are limited to 240 different values, 0...239.

The seven control inputs form the seven least-significant EPROM address bits. For reliable operation with asynchronous control inputs, they must be synchronized in an input register.

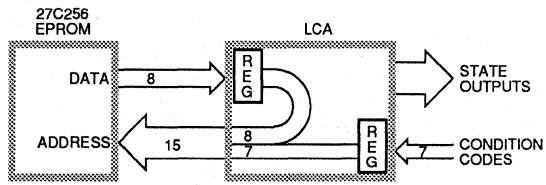
This rudimentary state machine can thus have 240 different states, and can jump from any state to any one of 128 arbitrarily defined next states, controlled by the 7-bit condition code.

This basic design uses no CLBs in the LCA, just IOBs; but it allows a number of states and a multi-way branch complexity far in excess of any normal need. The user will usually reduce the multi-way branch complexity by assigning identical values to many of the 128 possible next states.

The user has the logic resources of the LCA available to add features like:

- State decoding/encoding
- Stack registers
- Loop counters
- More sophisticated branch logic, etc.

This design is straightforward, inexpensive, compact and very flexible. Its speed is limited by the EPROM access time, which can be less than 100 ns. For higher speed—at a higher cost—the EPROM can be shadowed by fast SRAMS.



1987 01

PS/2 Micro Channel Interface

Application Brief BY ROB STRANSKY

IBM's general-purpose microcomputer, the Personal System 2, is available in several models, from the low-end Model 25 to the high-end Model 80. These third-generation PCs have several innovative features, including 3-1/2 inch floppy-disk drives, high-resolution VGA graphics, and a 20-MHz 80386 processor as the main engine for the Model 80. Among the most interesting features is the Micro Channel interface, the bus specification for the interface between the system and adapter cards. The Micro Channel's streamlined characteristics and flexibility provide PS/2 designers and users with many advantages over previous PC architectures.

One key aspect of this architecture is the ability to configure the system without the need for DIP switches on the bus adapter cards. Defined with System Configuration Utilities, an add-on card's addressing and other optional configuration data are established and stored in CMOS battery-backed memory on the main board. Upon power-up, this information is loaded into Programmable Option Select (POS) registers residing on the adapter cards.

Figure 1 indicates one way in which a Logic Cell Array can be used for the POS-register section of a Micro-Channel adapter card. The Micro-Channel interface includes logic to decode the address, status, and control signals associated with the bus to identify the appropriate POS register to be accessed. These signals determine if the card is being addressed, and whether the current operation is a Read or Write.

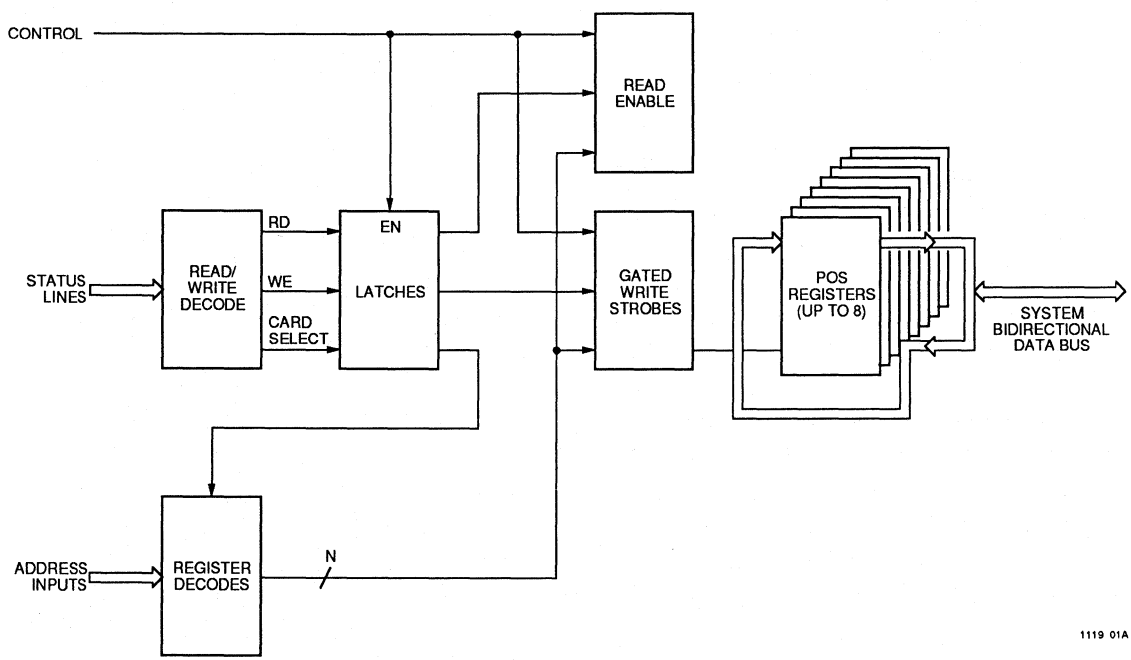


Figure 1. Micro-Channel-Interface Block Diagram

The Micro Channel specification reserves two POS registers for the upper and lower bytes of the Adapter Identification (ID). Six other byte-wide POS registers can hold additional configuration information; some of the bits within these are specifically dedicated to channel status information. Some applications will require the use of only portions of these six registers.

A second key aspect of the Micro Channel architecture is its ability to arbitrate the bus access of multiple adapters. The Micro Channel specification clearly defines the logic required for this arbitration. Each adapter in the system is assigned a priority level. These levels vary from the highest priority "-2" to the lowest priority "F". This "-2, -1, 0, 1, 2...A, B, ...F" scheme defines unique priority levels. The higher levels are primarily used for memory refresh or error recovery. The lower levels are reserved for the System Board processor and spares. The middle levels are used for DNA Ports 0-7, typically used for high speed transfers. The priority level assigned to any adapter is stored in one of its POS register nibbles. The arbitration logic must be very fast in order to grant control of the bus to the adapter with the highest priority.

As can be seen by the logic in Figure 2, this priority level (ARB ID 0-3) is driven onto the bus via an open-collector driver. The logic then turns around and accepts the driven bus as input. The cycle may repeat a few times before the adapter with the highest priority level actually gains control of the bus. For proper operation each half of the cycle must complete in 50 ns, a performance that can be achieved in the 70-MHz LCA devices.

Implementation of the POS registers, arbitration, logic and control sections typically requires only 1/3 to 2/3 of a single XC2018 or XC3020; the remainder of the LCA is available for implementing the unique functionality of the specific adapter card. Some Xilinx users have developed the standard interface and stored it as a recallable macro function in the Xilinx development system. Applications including hard disk controllers, communication controllers, and specialized memory controllers have been developed for the PS/2 using Xilinx FPGAs.

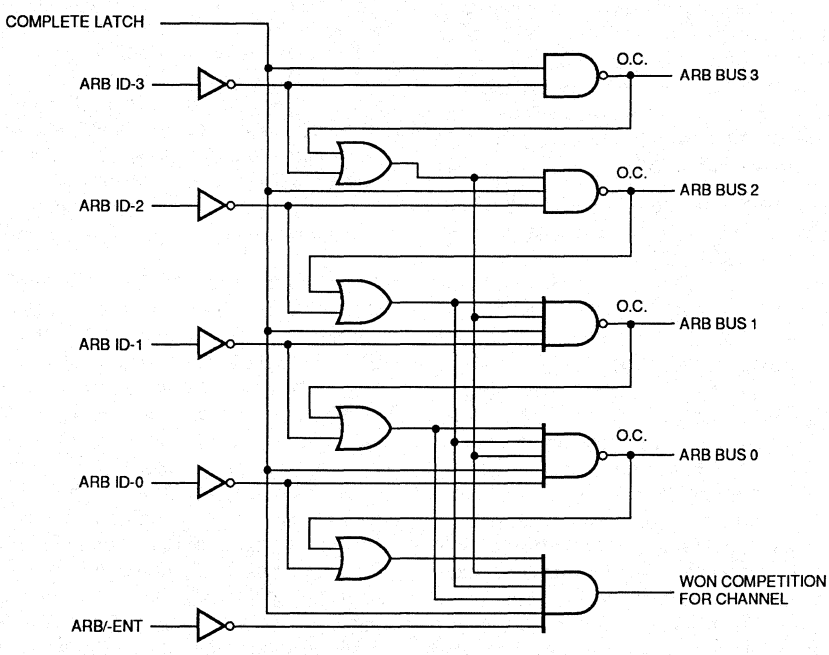


Figure 2. Local Arbiter Logic

DRAM Controller with Error Correction and Detection

Application Note BY TOM WAUGH

AN INTRODUCTION TO MEMORY CONTROL AND ERROR CORRECTION

The need to design memory controllers for systems that have a large amount of memory is a common design challenge that engineers must deal with today. Almost all large memory systems use dynamic random access memory (DRAM) because of its density and low cost. While designing large memory systems with static random access memory (SRAM), would make the design task easier, the drive to produce more cost effective products forces the engineer to design with DRAMs, despite their inherent drawbacks. The memory cell of a DRAM is a capacitor that holds a charge corresponding to the value of the data bit. Since all capacitors leak charge, a DRAM cell will gradually lose its charge, and its stored value, unless it is recharged. This recharging, known as refreshing, must typically be performed once every 2 to 4 ms depending on the DRAM. Refreshing is one of the DRAM controller's two primary functions. The other function is to arbitrate between requests for memory read and write accesses from the system's central processing unit and requirements for memory refreshes.

In addition to its need for periodic refreshing, the DRAM exhibits another problem that SRAM and other memory devices do not—greater susceptibility to soft errors. A *soft error* is the loss of a data bit in a memory cell in which the memory cell is not physically damaged. Rewriting the data in the cell corrects the error. This type of error is different from a *hard error* which is caused by a memory cell that has failed permanently. Soft errors in DRAMs are usually caused by alpha particles (helium nuclei), which are normally present in the atmosphere, but which are more often emitted by radioactive impurities in the IC packages of the DRAMs themselves. If an alpha particle hits a memory cell, it can corrupt the cell's charge, causing a data bit error. Most people believe that the likelihood of such an error is so low that it can be safely ignored. While this may have been true for the smaller memory systems of the past, it may no longer be so. The size of some memory systems today can make the likelihood of soft errors unacceptably high. The probability of a soft error can be reduced by device and packaging improvements and by reduction in signal noise. Another method of dealing with soft errors is to incorporate error detection and correction into the memory system. This solution decreases system performance and adds the cost of redundant memory, but prevents parity errors from causing system failures.

OPTIONS FOR DRAM CONTROLLER DESIGN

There are a number of options available to the engineer designing a memory system that requires DRAM control. (The following options apply to the design of error detection and correction circuits as well.) The simplest option is a standard off-the-shelf LSI memory controller. The manufacturers of these devices provide an integrated solution to DRAM control by combining CPU interface logic with the necessary memory access/memory refresh arbitration on a single chip. However, each memory system has unique timing and protocol requirements, and it is extremely difficult for these standard parts to accommodate the requirements of every system. This realization has driven many DRAM controller manufacturers to incorporate some degree of programmability into their parts to make them more flexible. Unfortunately, this has made the parts more complex, hungrier for power, and more expensive. Even so, they simply cannot meet every system's requirements without employing external "glue logic."

The need to match the DRAM controller to the specific requirements of the system has forced many engineers to consider two options for creating their own controllers: SSI/MSI packages or custom gate arrays. The use of SSI/MSI is low risk, but wastes space and power; while the use of the custom gate array provides a highly integrated solution, but at considerable risk and expense. Nonrecurring engineering costs (NRE), testing and simulation costs, inventory risk, and a long design cycle make the custom gate array option unattractive for most designs. Recent architectural advances in high-density Field Programmable Logic Arrays have created a third option. Xilinx's 3000 family of FPGAs brings unprecedented density to programmable logic, with devices containing as many as 9000 gates. The 3000-family architecture makes the devices particularly well-suited to memory-controller applications.

WHY IMPLEMENT A DRAM CONTROLLER WITH A FIELD PROGRAMMABLE GATE ARRAY?

There are several reasons why one would want to design a DRAM controller with a Logic Cell Array. First, the true programmability of the LCA device gives the designer the freedom to design the DRAM controller to the exact specifications of the memory system. There is no need for the external "glue logic" often necessary with standard solutions, because any necessary design tweaking is implemented internally. The LCA solution has the advantage of the SSI/MSI or custom gate array solution in that it can be configured to meet unique system requirements. There is no loss in integration as with the SSI/MSI solution, and the cost and risks of the custom gate array solution can be avoided. Second, the density of the 3000 family of LCAs makes it possible to implement DRAM control and error detection/correction in a single LCA device. This is traditionally a two-chip solution using standard parts: a DRAM controller and a separate error correction and detection unit. It can of course be implemented in a single custom gate array, but again with the earlier caveats. Finally, the CMOS LCA consumes less power than traditional standard "programmable" controllers which are typically implemented in NMOS or bipolar processes.

DESIGN EXAMPLE

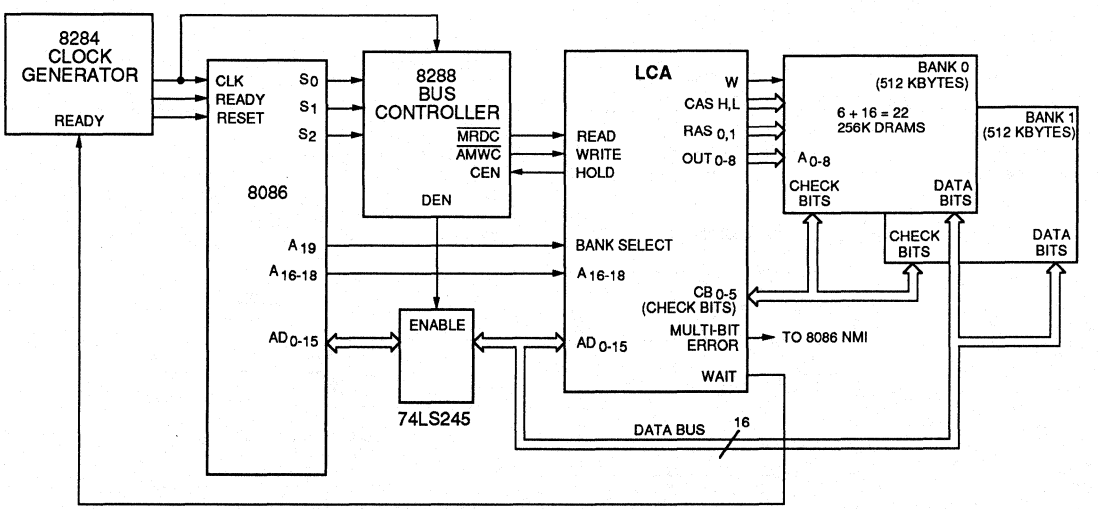
The following design example shows the implementation of a DRAM controller and an error checker/corrector (ECC) with an LCA. The example is an 8-MHz 8086-based system that directly addresses 1 Mbyte of memory comprising 44 256K DRAMs: 32 for data and 12 for the correction bits. A single LCA device can serve as both the DRAM controller and the ECC, which performs single- bit error correction and double-bit error detection. There are several features of the 3000-family architecture that make this design possible. These include five input-configurable logic blocks (CLBs) with two storage elements, internal buses, and flexible input/output blocks (IOBs).

DESIGN OVERVIEW

The DRAM Controller/ECC uses a 16- MHz clock synchronized with the processor clock, and sits between the 8086 microprocessor with its 8288 bus controller and the system memory (Figure 1). The 8288 decodes the processor status lines (S_2, S_1, S_0) and tells the DRAM Controller whether it is to perform a Read or Write access to the memory. (It is also possible to incorporate the bus controller logic into the larger LCAs). The DRAM Controller then performs the appropriate access issuing Row Address Strobe (RAS), Column Address Strobe (CAS), and Write, if necessary. The Error Checker and Corrector generates check bits on each Write, and checks for and corrects errors on each Read. The controller also signals the 8086 if the memory access requires a Wait state or if a non-correctable error is detected.

SYSTEM TIMING

Figures 2a—2c show the timing involved in some of the different memory cycles. The Word Write (Figure 2a) requires no wait states as shown. The check bits from the ECC are written to memory along with the data. The



1127 09A

Figure 1. System Overview of DRAM Controller with Error Correction and Detection.

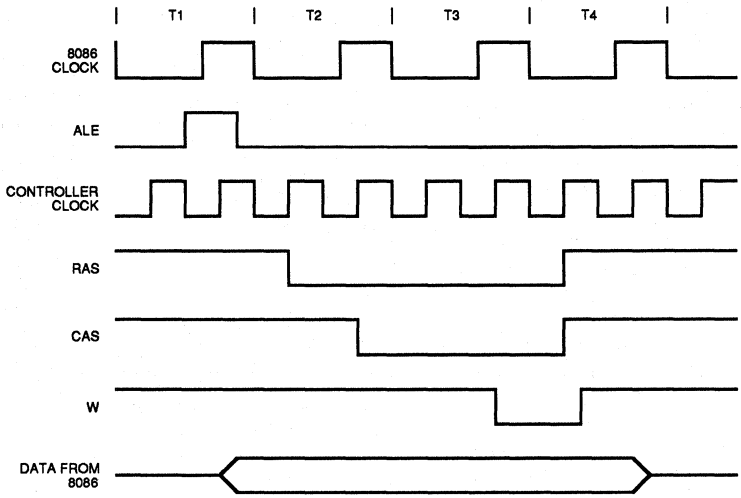


Figure 2a. Word Write Timing

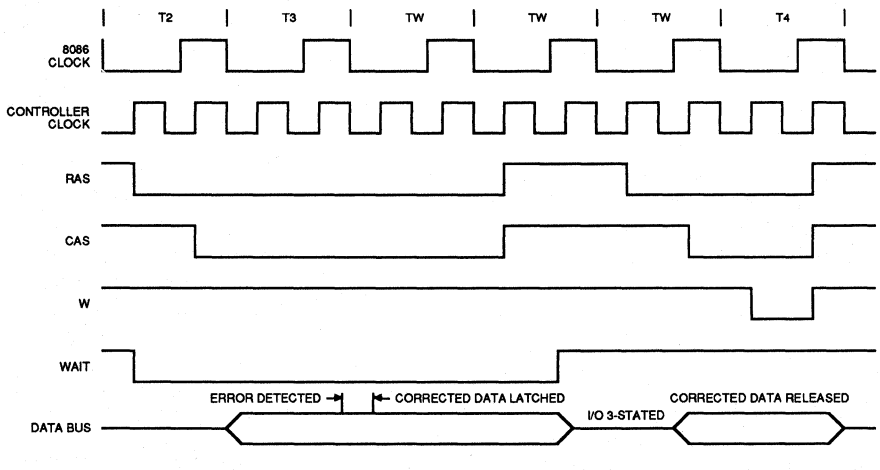


Figure 2b. Word Read Timing with Errors Detected.

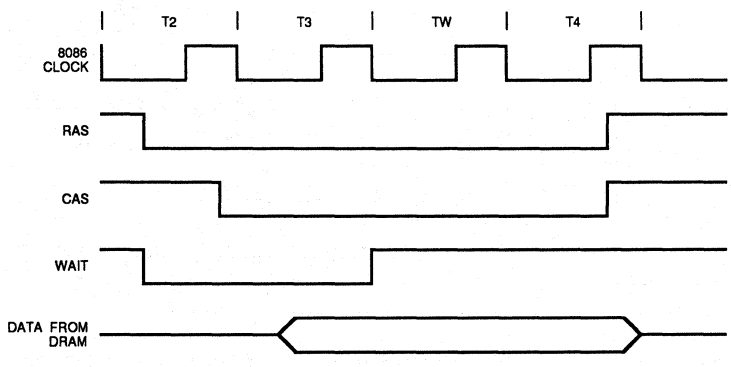


Figure 2c Word Read Timing with No Errors Detected.

Read cycle (Figures 2b & 2c) requires a minimum of one wait state. The insertion of a wait state is unavoidable because of the time it takes the 120-ns DRAMs to output the data. If the ECC detects no errors in the data, the WAIT signal is released and the Read operation is completed. If an error is detected, the insertion of two more wait states is required to provide time to correct the error. The insertion of the two additional wait states affects system performance, but this is the trade-off for having error correction, which avoids the fatal system errors that occur with parity-checking-only solutions.

DESIGN FOCUS

The 3000-family LCA architecture has a number of features that are essential to the DRAM controller design. The first such feature is the dual data input paths in the IOBs, one registered and one direct. This structure permits the address and data on a multiplexed bus to be latched from the same I/O pin. Figure 3 is a bit-sliced view of an IOB used to latch the multiplexed Address/Data bus. In this design, the address is latched into the IOB input flip-flop with the 8086 ALE. The data from the 8086 can enter the same input pin and go directly to the ECC circuit via the IOB direct input—there is no need for external latches.

Another feature of the IOBs is the output flip-flops with three-state buffer enables. This feature permits bit error correction using only one I/O pin. Figure 4 shows a bit-sliced view of how the ECC is accomplished. A memory Read cycle provides the best example for showing the capabilities of the IOB structure. During a Read, the IOB output is 3-stated, permitting the DRAM data on the data bus to enter the ECC via the IOB direct input. If the ECC detects a data bit error, it corrects the error and latches the corrected data word into the output flip-flops of the IOBs. The data bus is then 3-stated by turning off the DRAM outputs. The corrected word, latched in the outputs of the flip-flops, is then released onto the data bus by enabling the 3-state buffer. This permits the corrected data to be read by the 8086 at the same time it is being written back to the DRAM.

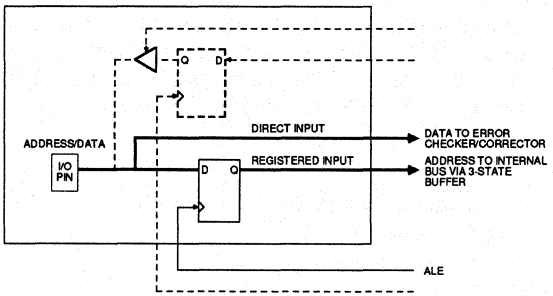


Figure 3. Address and Data Latching
Latching Data off a Multiplexed Address and Data Bus. The Input/Output block configuration shown above illustrates how the direct and registered inputs in the IOBs can be used to latch a multiplexed address/data bus into the LCA. The address is latched into the IOB flip-flop; the data flows directly into the ECC logic.

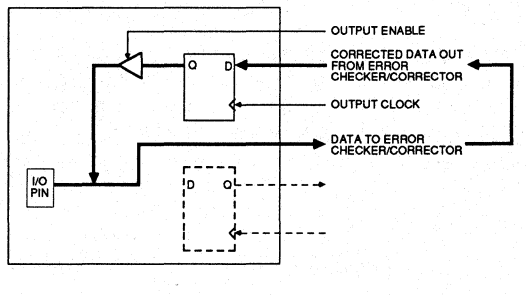


Figure 4. Data In and Out Through ECC
The data from the bus goes into the LCA, where it is corrected in the ECC. The corrected data is then put back onto the bus via the IOB output flip-flop.

Figure 5 is a block level diagram of the DRAM Controller and ECC that reside in the LCA. A functional description of each block follows:

The *refresh timer* is driven by the 16-MHz clock to provide a signal that tells the DRAM controller that the memory needs refreshing. Each of the 256 rows of memory in this system must be refreshed every 4 ms. The controller attempts to refresh eight rows every 125 µs, so that all 256 rows are refreshed in 4 ms. The refreshing technique employed in this design is a unique combination of burst and hidden refreshing to show the flexibility of the LCA-based solution. There is no need to force a system to conform to the constraints of an off-the-shelf part. The Hidden Refresh is performed when the 8086 is doing a Read from or Write to somewhere other than memory, like an I/O port. This involves giving the DRAM a refresh address from the *refresh address counter* via the *address selector* and a RAS pulse Low from the *timing generator.* The Burst Refresh is performed only if it has not received its eight required refreshes during the 125-µs refresh period. When a Burst Refresh is required, the controller will isolate the memory from the 8086, insert wait states, and provide the number of refreshes it needs to complete the eight refreshes required during the refresh period.

The *timing generator*, a state machine triggered by Address Latch Enable (ALE) at the beginning of the processor cycle, produces all the timing required to perform the memory accesses and refreshes. The signals generated by this block include the row address and column address strobes (RAS and CAS), the WRITE signal, the WAIT-state signal for the processor, the HOLD signal that isolates the processor from the memory, the clock for the *refresh address counter*, and the select control for the *address select.*

The *refresh address counter* is an 8-bit counter that provides the 8-bit addresses necessary to refresh the DRAMs.

The *address selector* selects which address is sent to the DRAM. During a Read or Write cycle, the *timing generator* select control signal tells the *address selector* to select the DRAM row address, strobe it with the RAS, and then select the column address and strobe it with the CAS. During a refresh, the *address selector* selects the address from the *refresh address selector* and strobes it into the DRAM with RAS.

During a Write cycle, the *error checker/corrector (ECC)* generates six check bits using a modified Hamming code for each 16-bit data word and writes them to memory along with the data. Use of a modified Hamming code permits single-bit data correction and double-bit error detection. During a Read cycle, the ECC compares the check bits read back from memory with new check bits generated from the data read back. If the comparison yields a correctable error, the ECC will correct it. If the error is not correctable, it will flag the NMI on the processor.

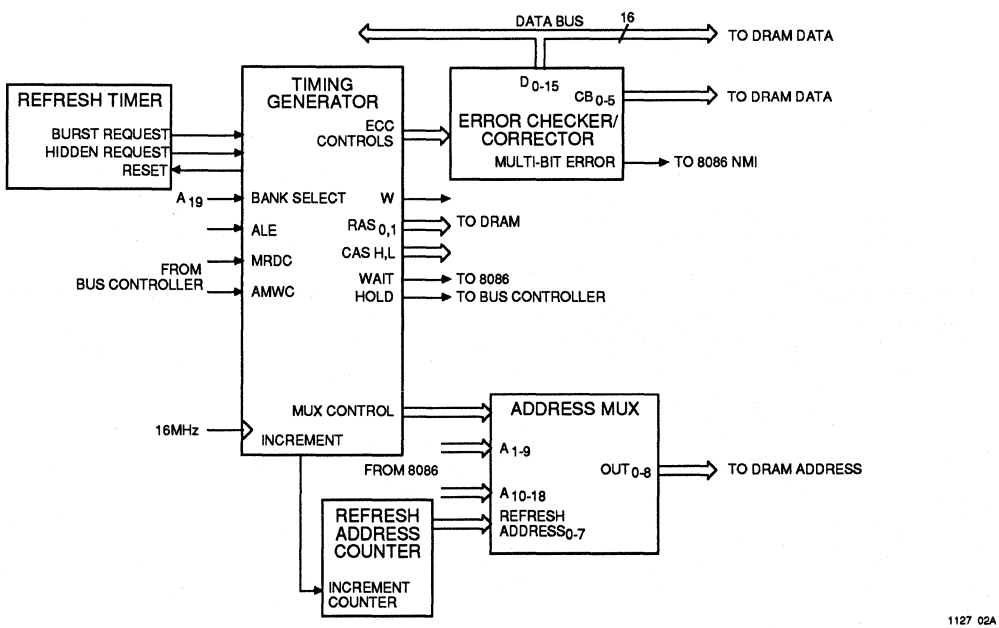


Figure 5. LCA Block Diagram
Block diagram of the DRAM controller functions implemented in the LCA.

Perhaps the most important feature of the LCA architecture for implementing a DRAM Controller is its internal three-state bus capability. The three-state buffer enables onto the horizontal longlines allow the designer to implement an internal bus in the LCA. This feature permits the implementation of the Address Selector without using any CLBs. Figure 6 shows a bit slice view. The row, column, and refresh addresses all have access onto the internal bus, and to the outputs that lead to the DRAMs. By controlling the three-state enables, only one address is allowed onto the bus at a time. This feature is essential to this design, and has many other applications including performing wired-AND functions and address decoding.

CONCLUSION

Although the bottom-up design of a DRAM controller is a complex task, it is necessary in cases in which off-the-shelf controllers do not meet the requirements of the system. SSI/MSI and custom gate array solutions involve trade-offs and compromises. Designing a DRAM controller and ECC with an LCA is a straightforward application and a good fit for the 3000 family architecture. The Field Programmable Gate Array offers the flexibility necessary to match the many different memory systems, the integration desirable for board level designs today, and the cost effectiveness required to make a competitive product.

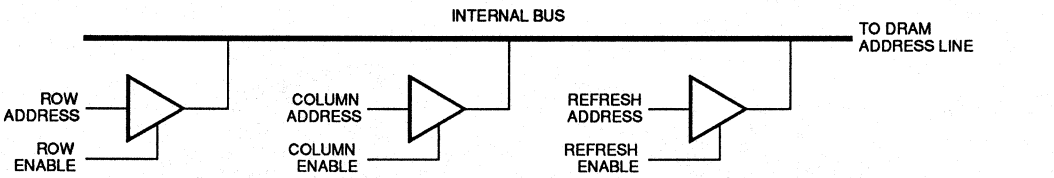


Figure 6. Address Multiplexing Using 3-State Enables onto Internal Buses

XILINX

The Programmable Gate Array Company

XILINX

Article Reprints

FOR ENGINEERS AND ENGINEERING MANAGERS — WORLDWIDE

ELECTRONIC DESIGN

A PENTON PUBLICATION U.S. $5.00

MARCH 8, 1990

DESIGN APPLICATIONS

BY INCLUDING A RAM-BASED PROGRAMMABLE LOGIC CIRCUIT, ONE CONTROL CARD CAN READILY HANDLE MULTIPLE INTERFACES.

BUILD RECONFIGURABLE PERIPHERAL CONTROLLERS

During the design of a computer peripheral, such as a printer, CRT terminal, disk drive, or other complex subsystem, decisions are often made regarding control logic partitioning. In some instances, the peripheral contains all of the control circuits, and the interface to a host system is over a standard port, such as an RS-232 or a Centronics link. However, the limited data-transfer speed and signal-control flexibility of those ports often causes a bottleneck when very large amounts of data must be transferred or complex operations must be controlled.

One solution is to keep the data-intensive portion of the logic in an adapter board that plugs into the host computer's bus and supply a custom high-speed link to the peripheral. But using hardwired logic to implement the adapter card limits the card's flexibility if updated versions of the peripherals are released, or if a second, relatively different model is developed. Ideally, one adapter card should be all that's needed. Simple software updates (device drivers) that can be loaded into the card for the specific model peripheral can provide an optimized interface.

Such a peripheral-control card can readily be developed by taking advantage of RAM-based programmable logic circuits, such as the logic cell arrays from Xilinx *(see "RAM-based Programmable Logic").* And the card's function can be altered in the field with just a new software driver that's loaded when the system boots.

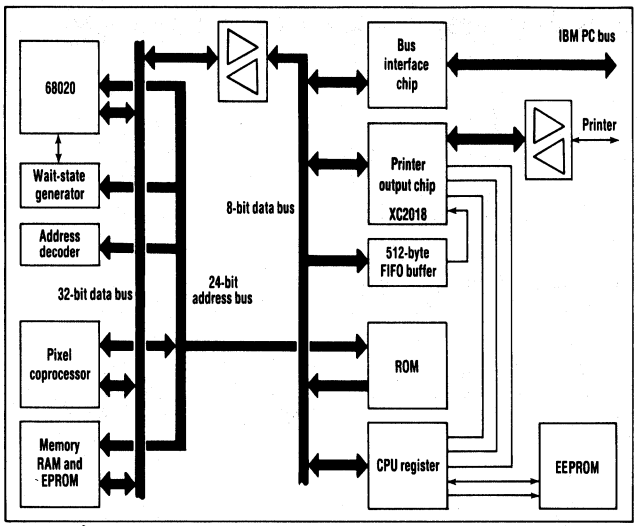
Furthermore, if the data is RAM-based, it can be altered during system operation as well. Consequently, if the same card must control multiple printer types, the card can be switched between printer drivers in just milliseconds by reloading the RAM-based programmable logic circuits to reconfigure the interface on-the-fly. Such a controller can be easily modified to support new peripheral devices, and it will never have to be removed from the computer system.

Programmable logic devices based on static RAM memory cells make it possible to implement "soft" hardware—that is, hardware whose functions can be

KENNETH K. HILLEN
Tektronix Inc., Mail Stop 63-356, P.O. Box 1000, Wilsonville, OR 97070;
(503) 685-3904.
BRADLY FAWCETT
Xilinx Inc., 2100 Logic Dr., San Jose, CA 95124; (408) 559-7778.

DESIGN APPLICATIONS

FLEXIBLE PRINTER CONTROLLER



1. IT'S ONLY A SMALL PORTION of the circuitry used on the 68020-based intelligent printer controller, but the reconfigurable printer output chip gives the board its flexibility. Two dedicated custom chips on the card handle the pixel processing and the host-system bus interface.

changed while it resides in the system. The configuration programs can be loaded automatically at power-up or on command at any time. Both users and manufacturers benefit from this flexibility. Microcomputer users can easily change or upgrade peripheral devices without purchasing a new controller or altering the internal hardware of the system. Equipment manufacturers don't need to design a new controller board for each new peripheral.

As a result of the improved flexibility, new peripherals can be brought to the marketplace sooner, and the cost of initially developing the flexible peripheral controller board is amortized over a larger number of units. The ability to support future peripherals extends the controller's product life. Field updates to the controller can be accomplished by distributing new program disks to update the logic configuration, as opposed to requiring hardware modifications to the board.

Advanced page printers and color thermal printers require huge amounts of data and some special control signals to keep their print engines busy and ensure optimum performance. Yet to have a custom adapter card for each printer would get very expensive, especially if two such printers were needed. By pulling out the common control circuits needed by either printer type and implementing those circuits in both custom and RAM-based programmable logic, system cost can be minimized and users save one card slot in the host system. In addition, because the printer has a minimal amount of logic in it, upgrades to the mechanism would cost less as well.

Controlling Printers

Consequently, by dividing the control and processing logic that a complex printer needs, the overall system can be made more flexible. The printer would contain only the basic machine-control circuits and the printing control logic, while the host computer's plug-in card would contain all of the data-processing logic. This approach was taken by the PhaserCard printer controller from the Graphics Printing and Imaging Division of Tektronix. By implementing the system architecture in that fashion, the card helps designers make printer changes or add new printers, still keeping the design cycle very short because there are less circuits in the printer to deal with.

A key ease-of-use feature for laser and color thermal printers would be to have them emulate Adobe System's Postscript and Hewlett-Packard's HPGL graphics-description languages. To do that, the host system plug-in board includes the emulation capability along with a 68000-family microprocessor and a custom chip that accelerates the computations needed to prepare an image for printing. Furthermore, the PC/AT/XT bus-compatible card uses one of the smaller Xilinx programmable gate arrays for the printer interface logic and a mask-programmed gate array to control the interface to the host PC's bus *(Fig. 1)*.

By loading the appropriate configuration program into the programmable logic chip, the controller card can be used with any of several printers. They include monochrome laser printers (such as the Canon LBP8), the Tektronix 4696 (a 120-dot-per-inch ink-jet color printer with a Centronics interface), the Tektronix 4693D/DX (a 300-dot-per-inch wax-transfer color printer with a Tektronix Parallel Interface—an enhanced version of the Centronics interface), and the new Tektronix Phaser CP (a 300-dot-per-inch wax-transfer color printer with a synchronous serial interface). The PhaserCard includes two printer interface ports so that two printers can be connected simultaneously.

Considerable processing power is needed in order to interpret the graphics-description languages, perform image processing, and run the printers. To handle the housekeeping and manage the host system interface, a 68020 32-bit processor, running at 12 MHz with one wait state, readily handles the control and leaves room for program growth if more complex tasks must be handled. To eliminate the bottleneck of

DESIGN APPLICATIONS

FLEXIBLE PRINTER CONTROLLER

converting the page description into the raster image, a custom graphics coprocessor tackles all of the computations. The chip is a full-custom IC, and assists in image generation during line drawing, area filling, and half-toning operations. Lines can be drawn at 6 million pixels/s.

During operation, the CPU accepts image description input streams from the host bus, rasterizes the image into bit-map memory (with assistance from the coprocessor), and then transfers the bit map to the printer interface logic. All timing signals are derived from a 24.23 MHz oscillator.

The processor's code and data, and the configuration programs for the programmable logic chip reside in on-board RAM. That RAM would typically be loaded from the host system's hard disk. Code updates could thus be distributed on diskettes, eliminating the need for a technician or service person to replace a nonvolatile memory chip from the control board to upgrade the card. To hold the control code and the page description, 3 Mbytes of dynamic RAM are included on the control card, and an additional 5 to 8 Mbytes can be added through a memory-expansion connector and a daughterboard.

The on-board memory is partitioned into three 1-Mbyte blocks—one holds program code, another serves as a data buffer, and the third holds the bit map for A-size, 150-dpi (dots per inch) printers. The additional memory is required for A- or B-size, 300-dpi printers. Power-up diagnostics and the bootstrap routine for downloading from the hard disk are held in 64 kbytes of EPROM, while a 512-byte electrically-erasable memory holds several parameters that process Postscript files and provide printer identification information.

A bus interface circuit (BIF), implemented with a 5000-gate mask-programmed gate array, connects the controller to the PC bus. The BIF chip emulates standard LPT (paral-

THE PROGRAMMABLE GATE ARRAYS

Based on static RAM cells that hold configuration data rather than metal wiring or some form of nonvolatile memory, the Xilinx programmable gate arrays are high-density CMOS chips that combine user-programmability with the flexibility and extensibility of a gate-array architecture.

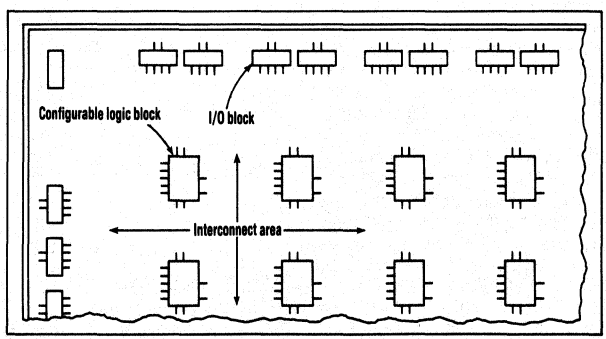
The general structure of the Xilinx programmable gate arrays, also known as logic cell arrays (LCAs), consists of a core area containing a matrix of configurable logic blocks *(see the figure)*. Interspersed with the logic blocks are channels with programmable interconnections, and surrounding the core area is a ring of programmable I/O cells. The I/O blocks supply an interface between the external package pin and the internal logic. Each of the configurable logic blocks includes a combinatorial section, storage elements, and internal routing and control logic. Programmable interconnection resources provide the routing paths that connect the I/O and logic blocks in the desired configuration.

Similar to a microprocessor, the LCA is a program-driven device. The configuration program is loaded automatically from an external memory on power-up or on command, or is programmed by a microprocessor as part of system initialization.

Designing with Xilinx programmable gate arrays is similar to designing with other gate arrays. Users can employ familiar CAE tools for design entry and simulation. The Xilinx-specific software for cell placement and circuit configuration runs on the PC/AT and compatibles, as well as on popular engineering workstations, such as the Apollo and Sun 3.

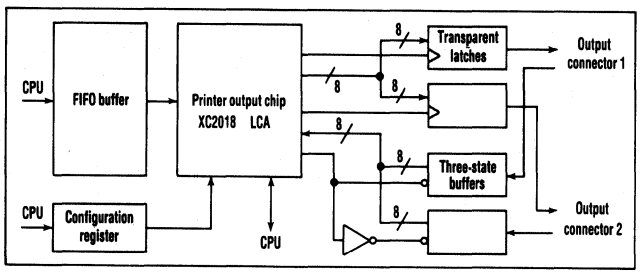
Currently, two families of compatible LCA devices are available. The original XC2000 series includes the 1200-gate XC2064 and 1800-gate XC2018. The second-generation XC3000 family has five members, ranging from the 2000-gate XC3020 to the 9000-gate XC3090. A third family, the XC4000, is now in development and should be sampled later this year. It will offer densities of up to 20,000 gates.

Because chips in the XC3000 series are now readily available, they would significantly simplify the design of a similar printer controller card if it were being done today. This is because they incorporate on-chip three-state buffers, offer more functionality in each configurable logic block, and have improved routing resources. However, when the controller's concept was initially started in 1988, only the XC2000 series was available.



DESIGN APPLICATIONS

FLEXIBLE PRINTER CONTROLLER



2. EVEN THOUGH THE PRINTER output-control chip is based on a programmable logic circuit, off-chip functions, such as three-state buffers, a FIFO-buffer memory, and a configuration register must be added. This is because the logic array can't efficiently implement such functions on chip.

lel) and COM (serial) ports on the PC side, making it possible for existing applications that drive those ports to use the printer control card as well.

One avenue to make the printer interface logic flexible enough to support various output devices while occupying a minimal amount of board space is by incorporating the 1800-gate Xilinx XC2018 logic cell array (LCA) chip. The chip offers enough gates and in-system malleability to make the logic flexible. The user-programmable aspect of the LCA also minimizes design risk and reduces the design cycle turnaround time. In addition, because each configuration program is independent of the others, each printer can be supported with a separate, optimal interface. In fact, with the on-the-fly reconfigurability of the LCA, two different printers—one monochrome and one color—could be supported simultaneously in near real time without duplicating much circuitry. The correct configuration program for each printer is downloaded to the LCA as that printer is accessed.

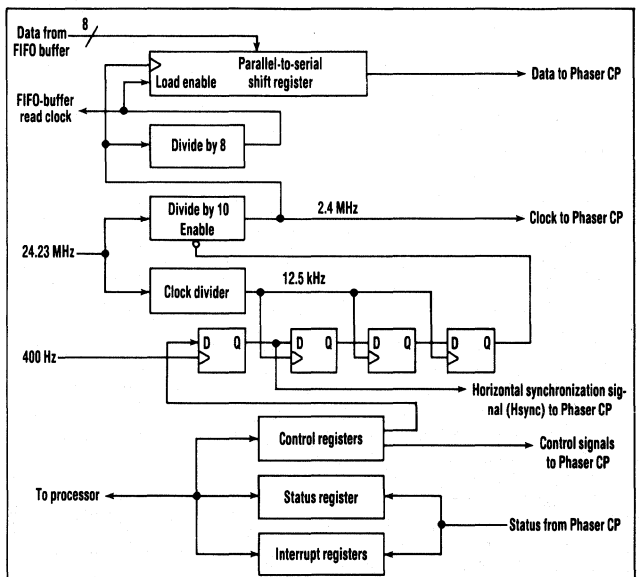
In addition to the actual printer interface implemented in the LCA chip, several other components are needed to flesh out the support *(Fig. 2)*. The largest component, a 512-word-by-9-bit first-in/first-out memory, buffers the data stream between the CPU and the printer. Each entry in the FIFO buffer includes 8 bits of data; the ninth bit marks special conditions, such as end-of-line (EOL).

The 68020 CPU writes data into the FIFO buffer in bursts to minimize the time its spends dealing with data-output operations. The LCA reads the data from the buffer, performs any formatting required by the particular type of printer being used, and sends the data to the printer. As a result, the CPU can perform other tasks while the printer receives the data at its own pace.

Signal buffers external to the LCA are used to isolate the LCA from the output connectors, as well as give additional drive capability and electrostatic protection for the CMOS chip. The output signals are buffered with dual byte-wide transparent latches. Typically, only one byte-wide set of latches will be transparent at one time; the other will be left holding a value that drives its attached printer to a quiescent state.

Also external to the LCA chip is a configuration register. It controls the downloading of configuration programs into the LCA. Directly accessible by the 68020 processor, this register is built with two devices: an octal flip-flop to drive signals to the LCA and a three-state buffer to send signals back to the processor.

The configuration process involves only three signals. The LCA's Done/Program input is driven low to initiate an LCA configuration cycle.



3. TO IMPLEMENT THE SERIAL interface required for a Phaser CP printer, blocks such as a parallel-to-serial shift register; various divide chains; a simple state machine composed of four flip-flops; and control, status, and interrupt registers must be configured in the LCA.

DESIGN APPLICATIONS

FLEXIBLE PRINTER CONTROLLER

The configuration program is then downloaded to the LCA as a serial bit stream using one clock and one data line. The LCA drives the Done/Program line back to the high state to signal the end of the configuration process. Downloading a configuration program takes less than 60 ms.

When two printers are connected to the card, selection of the correct configuration program is controlled by the processor; when a printer is to be accessed, the LCA is configured for that particular interface. If two different printers are connected, the LCA is reconfigured frequently during idle (non-printing) periods to check each printer's status.

In general, the operation of the configured LCA chip is similar for all printer types. The CPU writes data to the FIFO buffer, and a state machine created in the LCA reads the data from the buffer and sends it to the printer. Any necessary data formatting is performed in the LCA. For example, data bytes are converted to a serial data stream for the Phaser CP printer.

Logic in the LCA generates all of the required handshaking and timing signals. CPU interrupts are generated as needed, based on the status of the printers, FIFO buffer, and the LCA's internal state machine. Control and status registers that can be accessed by the CPU also are implemented in the LCA.

To show how the LCA can be configured for various printers, three specific configurations that each use about 2/3 of the 74 user-programmable I/O pins in the LCA must be examined. The first configuration looks at the interface to the Phaser CP, a 300-dpi wax-transfer color printer with a serial interface; the second examines a parallel interface to the 4693D/DX color printer; and the third shows a raw video interface to a bare-bones laser printer.

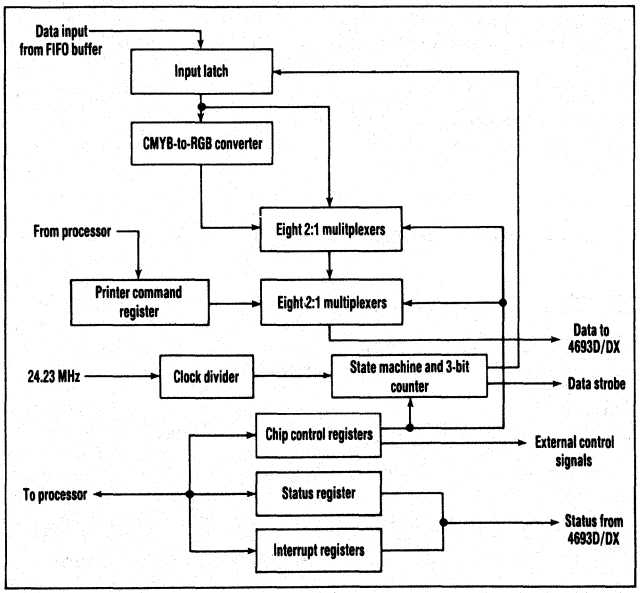
To control a serial-input printer, such as the Phaser CP, part of the LCA must implement a parallel-in/serial-out shift register that converts the byte-wide data from the FIFO buffer to a serial format. The remaining logic configured in the LCA includes a clock divider to control the shift register, a divider to generate the data clock to the printer, a clock divider and state machine to generate the horizontal synchronization (Hsync) signal, and the processor interface and interrupt registers *(Fig. 3)*. All of that logic employs 78 of the 100 logic blocks available in the XC2018.

The data-clock generator divides the 24.23-MHz board clock by 10 to create the 50%-duty-cycle 2.4-MHz clock that sends the serial data stream to the printer. This data clock is further divided by 8 to control the shift register. On every eighth data clock, the next byte in the FIFO buffer is loaded into the shift register.

A small state machine composed of four flip-flops generates the Hsync signal and enables clocks and data to be sent to the printer. After being enabled by the CPU, the state machine waits for a 400-Hz signal from the bus interface chip. A 12.5-kHz clock is used to sequence the state machine, which generates an 8 μs Hsync pulse, followed by an 8 μs delay. Then the state machine can start the data stream. The data stops when the EOL flag (the ninth bit in the FIFO memory word) is read from the FIFO buffer.

With six 8-bit processor interface registers, the CPU can write control information and read status information. These registers are mapped into the CPU's I/O address space; four are write-only and two are read-only *(see the table)*. The three interrupt registers share the same bit assignments; bits 4 and 5 of the chip-control register control the transparent latches for the two printer connectors.

Because both connectors are driven by the same pins of the LCA device, these bits force one connector's outputs to a static state (by disabling the transparent latch) while talking to the other printer. Bit 7 of the printer control register is toggled to control a serial-status-readback state machine in the printer. Using this



4. FOR A PARALLEL INTERFACE to talk to the 4693D/DX printer, the LCA must be configured to supply an input latch, a cyan-magenta-yellow-black to red-green-blue converter, several signal multiplexers, a simple clock divider and state machine, and various control, status, and interrupt registers.

DESIGN APPLICATIONS

FLEXIBLE PRINTER CONTROLLER

control, 16 bits of status information, identifying such conditions as paper jams, out-of-paper, and out-of-ribbon, can be read back one at a time through bit 2 of the printer status register.

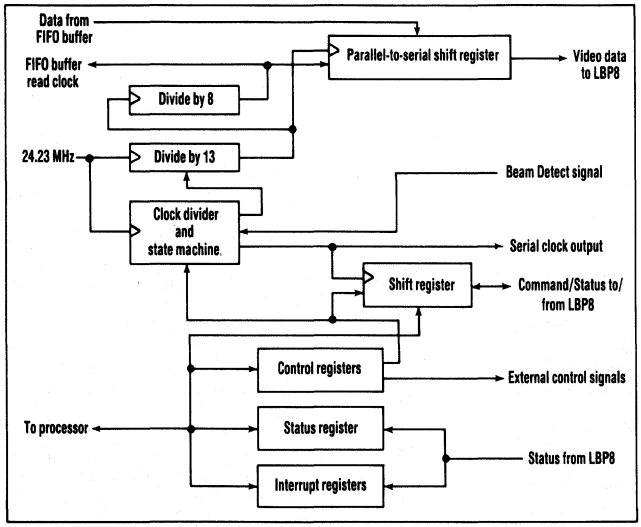
The parallel interface of the 4693D/DX printer requires 79 of the LCA's logic blocks. To implement that interface, the main functional blocks that must be configured in the LCA include an input-data register, a data converter, a state machine to control the data transfer between the LCA and the printer, and the processor interface and interrupt registers *(Fig. 4)*.

An 8-bit register is used to latch the data as it comes in from the FIFO buffer. The ninth bit, which indicates the EOL condition, goes to the state machine and interrupt registers. Optionally, the data can be converted from the cyan-magenta-yellow-black (CMYB) format generated by the controller card to the red-green-blue (RGB) format (some early versions of the 4693D/DX printer accept only the RGB format). This straightforward conversion requires only combinatorial logic.

A simple six-state state machine combined with a 3-bit counter controls the data flow and commands to the printer. The 24.23-MHz clock is divided to supply a clock of approximately 5 MHz to the state machine. The state machine sends data to the printer by reading it from the FIFO buffer, then stores it in the input register, converts it to the RGB format (if required, as determined by a bit in the chip control register), and asserts the Data Strobe signal.

Data is sent in a streaming mode—the machine will keep sending data to the printer until a byte marked with an EOL indicator is read from the FIFO buffer. The transfer of a command byte to the printer is triggered by loading the printer-command register and setting a bit in the chip-control register. The state machine controls the command sent to the printer and waits for the acknowledgement.

The six 8-bit processor interface registers are similar in function to those described for the Phaser CP printer. The chip-control register includes controls to clear the FIFO buffer, start the output state machine, enable the command mode, enable the transparent latches associated with the two printer connectors, and enable the CMYB-to-RGB converter. The printer-command register holds commands to be sent to the printer, and the printer-status register contains status information about the printer and the FIFO buffer. Interrupt-clear, mask, and read registers control the generation of interrupts to the CPU.



5. CONTROLLING A laser engine directly on the Canon LBP8 requires that the LCA supply precise data and control signals. Data input still requires a parallel-to-serial shift register, and several divider chains are needed to adjust the clock rate. A small state machine controls the data transfer and a simple bidirectional shift register.

HANDLING VIDEO DATA

When configuring the LCA chip to deliver the video data stream to the Canon LBP8 laser printer, the data from the controller card must be precisely synchronized with the laser mechanism to ensure image accuracy. Major circuit elements in this configuration include clock dividers that produce the data clock and control FIFO-buffer read operations, a parallel-to-serial shift register to serialize the data, a simple state machine to control data flow, a serial-to-parallel shift register to collect printer status information, and the processor interface and status registers *(Fig. 5)*. This configuration uses 97 of the 100 logic blocks in the LCA.

The 24.23-MHz clock is first divided by 13 to produce the 1.863-MHz data clock needed for the data transfer to the parallel-to-serial shift register. The data clock is further divided by eight to generate the signal that's used to read a new byte from the FIFO buffer and load it into the shift register as the transmission of the previous byte to the printer is completed.

To ensure that the data is properly positioned on the page, the data clock isn't started until a Beam Detect signal is received from the printer. This signal is sent at the start of each scan line. Data clocks will continue to be generated and data sent to the printer until an EOL mark is read from the FIFO buffer. The controller then waits for the next Beam Detect to start again. The state machine is just one flip-flop that's enabled by a bit in the chip-control register, set by a Beam Detect, and cleared when the

DESIGN APPLICATIONS

FLEXIBLE PRINTER CONTROLLER

REGISTER BIT ASSIGNMENTS FOR PHASER CP

Register	Read/write	Bit assignments
Chip control	write	bit 0 – Clear FIFO buffer bit 4 – Control transparent latch to color printer bit 5 – Control transparent latch to monochrome printer bit 6 – Reset the printer
Printer control	write	bit 1 – Start printer output state machine bit 2 – Output enable bit 3 – Command synchronization bit 4 – Double the vertical resolution bit 5 – Skip to next ribbon color bit 6 – Load sheet of paper bit 7 – Get next bit of printer status
Printer status	read	bit 0 – Printer status state machine at start bit 1 – Printer ready for current pass bit 2 – Serial status data readback bit 3 – On-line indicator bit 4 – Cable disconnected indicator bit 5 – Power-off indicator
Interrupt clear Interrupt mask Interrupt read	write write read	bit 0 – FIFO buffer empty bit 1 – FIFO buffer full bit 2 – Printer ready state change bit 3 – On-line state change bit 4 – Printer power off bit 5 – EOL (End-of-line)

EOL mark is encountered.

The LBP8 printer uses a serial path to receive certain commands and send back its status. These messages are clocked through a serial-to-parallel/parallel-to-serial shift register in the LCA, making it possible for the CPU to write commands and read status information as one byte.

The interface to the CPU consists of eight 8-bit control, status, and interrupt registers. The chip-control register and printer-control register include controls to clear the FIFO buffer, start the output state machine, control the transparent latches of the printer connectors, reset the printer, synchronize the image vertically, request the start of a new page, and notify the printer that the controller is ready.

One bidirectional shift register constitutes both the printer-command register and the printer-response register. The printer-command register holds commands to be sent to the printer serially; the printer responds by sending the requested information to the printer-response register. The printer-status register holds additional status information from the printer, such as the Beam Detect signal status, ready-to-print indicator, and a request for vertical synchronization. Interrupt-clear, mask, and read registers control the generation of interrupts to the CPU.

As with most logic technologies, the controller card's design was originally conceived as block diagrams and schematic drawings. Early in the system design cycle, the portions of the design to be implemented in the LCA were determined. State machine designs can be done with Mealy/Moore diagrams, which can be translated into a set of reduced next-state equations using the Abel software package from Data I/O Corp., Redmond, Wash.

Designing large state machines in the LCA can use up numerous internal configurable logic blocks (CLBs) very quickly because many state machines can have a dozen or more inputs to the next-state equations. However, by applying automata theory to find common terms that can be shared by more than one equation, the number of inputs to each CLB was kept to four or less.

Furthermore, because the XC2018 LCAs don't have three-state outputs, CLBs had to be used to implement 2:1 multiplexers to handle internal-register readback. The multiplexers would then select which of several registers would be connected to the processor's data bus during a read operation. In most cases, the CLB that contained the register had enough unused logic to implement the multiplexer. As a result, the need for the multiplexers doesn't have an impact on the number of CLBs available for the main control logic.

The design can now be mapped into the logic and I/O blocks of the LCA device using the XACT design editor from Xilinx, a development tool that allows users to manipulate a graphics image of the internal LCA architecture. About three weeks are usually needed for LCA newcomers to enter and debug the first circuit. Each subsequent configuration that's generated may typically require less than one week's development time.

The placement and routing of the circuitry can be a major area of concern because a limited amount of routing resources are available near each CLB. That was a key concern for the XC2018 because a large number of CLBs had to be interconnected to create the next-state equations. The net routings using the LCA's general-purpose interconnections sometimes result in surprisingly long propagation delays. To minimize the delays, the CLBs were clustered so that the time-critical paths were kept as short as possible. The reprogrammable nature of the array makes it possible for multiple what-if scenarios to be evaluated.

Because LCAs can be quickly and easily reconfigured, special configurations can be generated for test purposes during system development. During prototyping, a number of internal signals can be routed to the otherwise unused I/O pins to aid in the debugging process. A special diagnostic configuration can also be included in the final production design to ease the power-up diagnostic routines and servicing.□

Reprinted with permission from *Electronic Design* March 8, 1990.

DESIGN APPLICATIONS

WHEN DESIGNING STATE MACHINES, A TECHNIQUE CALLED ONE-HOT ENCODING CREATES EFFICIENT CIRCUITS FOR TOP-PERFORMING FPGA MACROS.

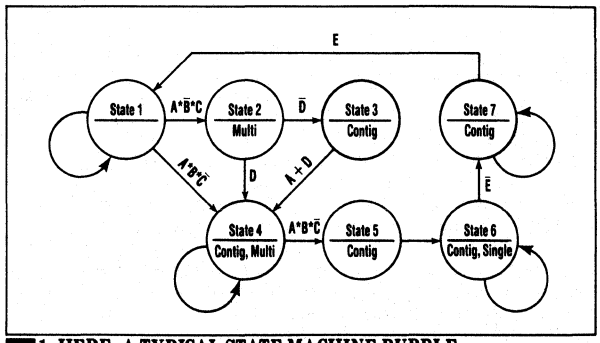
ACCELERATE FPGA MACROS WITH ONE-HOT APPROACH

STEVEN K. KNAPP
Xilinx Inc., 2100 Logic Dr.,
San Jose, CA 95124;
(408) 879-5172.

State machines—one of the most commonly implemented functions with programmable logic—are employed in various digital applications, particularly controllers. However, the limited number of flip-flops and the wide combinatorial logic of a PAL device favors state machines that are based on a highly encoded state sequence. For example, each state within a 16-state machine would be encoded using four flip-flops as the binary values between 0000 and 1111.

A more flexible scheme—called one-hot encoding (OHE)—employs one flip-flop per state for building state machines. Although it can be used with PAL-type programmable-logic devices (PLDs), OHE is better suited for use with the fan-in limited and flip-flop-rich architectures of the higher-gate-count field-programmable gate arrays (FPGAs), such as offered by Xilinx, Actel, and others. This is because OHE requires a larger number of flip-flops. It offers a simple and easy-to-use method of generating performance-optimized state-machine designs because there are few levels of logic between flip-flops.

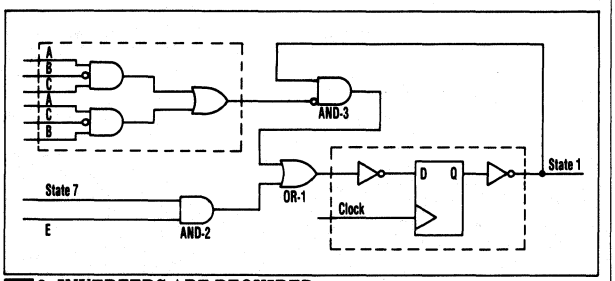
A state machine implemented with a highly encoded state sequence will



1. HERE, A TYPICAL STATE MACHINE BUBBLE diagram shows the operation of a seven-state state machine that reacts to inputs A through E as well as previous-state conditions.

DESIGN APPLICATIONS

STATE MACHINE DESIGN



2. INVERTERS ARE REQUIRED at the D input and the Q output of the state flip-flop to ensure that it powers on in the proper state. Combinatorial logic decodes the operations based on the input conditions and the state feedback signals. The flip-flop will remain in State 1 as long as the conditional paths out of the state are not valid.

generally have many, wide-input logic functions to interpret the inputs and decode the states. Furthermore, incorporating a highly encoded state machine in an FPGA requires several levels of logic between clock edges because multiple logic blocks will be needed for decoding the states. A better way to implement state machines in FPGAs is to match the state-machine architecture to the device architecture.

Limiting Fan-In

A good state-machine approach for FPGAs limits the amount of fan-in into one logic block. While the one-hot method is best for most FPGA applications, binary encoding is still more efficient in certain cases, such as for small state machines. It's up to the designer to evaluate all approaches before settling on one for a particular application.

FPGAs are high-density programmable chips that contain a large array of user-configurable logic blocks surrounded by user-programmable interconnects. Generally, the logic blocks in an FPGA have a limited number of inputs. The logic block in the Xilinx XC-3000 series, for instance, can implement any function of five or less inputs. In contrast, a PAL macrocell is fed by each input to the chip and all of the flip-flops. This difference in logic structure between PALs and FPGAs is important for functions with many inputs: Where a PAL could implement a many-input logic function in one level of logic, an FPGA might require multiple logic layers due to the limited number of inputs.

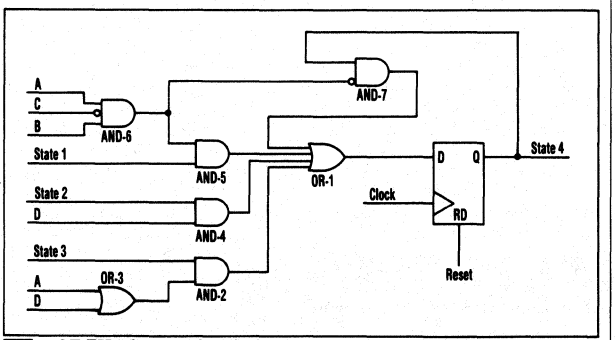
The OHE scheme is named so because only one state flip-flop is asserted, or "hot," at a time. Using the one-hot-encoding method for FPGAs was originally conceived by High-Gate Design—a Saratoga, Calif.-based consulting firm specializing in FPGA designs.

The OHE state machine's basic structure is simple—first assign an individual flip-flop to each state, and then permit only one state to be active at any time. A state machine with 16 states would require 16 flip-flops using the OHE approach; a highly encoded state machine would need just 4 flip-flops. At first glance, OHE may seem counter-intuitive. For designers accustomed to using PLDs, more flip-flops typically indicates either using a larger PLD or even multiple devices.

In an FPGA, however, OHE yields a state machine that generally requires fewer resources and has higher performance than a binary-encoded implementation. OHE has definite advantages for FPGA designs because it exploits the strengths of the FPGA architecture. It usually requires two or less levels of logic between clock edges than binary encoding. That translates into faster operation. Logic circuits are also simplified because OHE removes much of the state-decoding logic—a one-hot-encoded state machine is already fully decoded.

OHE requires only one input to decode a state, making the next-state logic simple and well-suited to the limited fan-in architecture of FPGAs. In addition, the resulting collection of flip-flops is similar to a shift-register-like structure, which can placed and routed efficiently inside an FPGA device. The speed of an OHE state machine remains fairly constant even as the number of states grows. In contrast, a highly encoded state machine's performance drops as the states grow because of the wider and deeper decoding logic that's required.

To build the next-state logic for



3. OF THE SEVEN STATES, the state-transition logic required for State 4 is the most complex, requiring inputs from three other state outputs as well as four of the five condition signals (A - D).

DESIGN APPLICATIONS

STATE MACHINE DESIGN

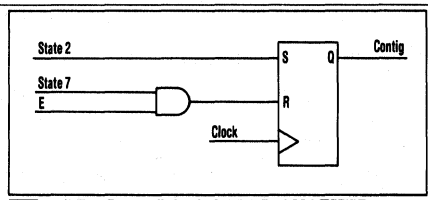
count the number of conditional paths leading into the state and add an extra path if the default condition is to remain in the same state. Second, build an OR-gate with the number of inputs equal to the number of conditional paths that were determined in the first step.

Third, for each input of the OR-gate, build an AND-gate of the previous state and its conditional logic. Finally, if the default should remain in the same state, build an AND-gate of the present state and the inverse of *all* possible conditional paths leaving the present state.

To determine the number of conditional paths feeding State 1, examine the state diagram—State 1 has one path from State 7 whenever the variable E is true. Another path is the default condition, which stays in State 1. As a result, there are two conditional paths feeding State 1. Next, build a 2-input OR-gate—one input for the conditional path from State 7, the other for the default path to stay in State 1 *(shown as OR-1 in Fig. 2).*

The next step is to build the conditional logic feeding the OR-gate. Each input into the OR-gate is the logical AND of the previous state and its conditional logic feeding into State 1. State 7, for example, feeds State 1 whenever E is true and is implemented using the gate called AND-2 *(Fig. 2, again).* The second input into the OR-gate is the default transition that's to remain in State 1. In other words, if the current state is State 1, and no conditional paths leaving State 1 are valid, then the state machine should remain in State 1. Note in the state diagram that two conditional paths are leaving State 1 *(Fig. 1, again).*

The first path is valid whenever ($A*\overline{B}*C$) is true, which leads into State 2. The second path is valid whenever ($A*B*\overline{C}$) is true, leading into State 4. To build the default logic, State 1 is ANDed with the inverse of all of the conditional paths leaving State 1. The logic to perform this function is implemented in the gate labeled AND-3 and the logic elements that feed into the inverting input of AND-3 *(Fig. 2, again).*



6. S-R FLIP-FLOPS OFFER ANOTHER approach to decoding the Contig output. They can also save logic blocks, especially when an output is asserted for a long sequence of contiguous states.

State 4 is the most complex state in the state-machine example. However, creating the logic for its next-state control follows the same basic method as described earlier. To begin with, State 4 isn't the initial state, so it uses a normal D-type flip-flop without the inverters. It does, however, have an asynchronous reset input, three paths into the state, and a default condition that stays in State 4. Therefore, a four-input OR-gate feeds the flip-flop *(OR-1 in Fig. 3).*

The first conditional path comes from State 3. Following the methods established earlier, an AND of State 3 and the conditional logic, which is A ORed with D, must be implemented *(AND-2 and OR-3 in Fig. 3).* The next conditional path is from State 2, which requires an AND of State 2 and variable D *(AND-4 in Fig. 3).* Lastly, the final conditional path leading into State 4 is from State 1. Again, the State-1 output must be ANDed with its conditional path logic—the logical product, $A*B*\overline{C}$ *(AND-5 and AND-6 in Fig. 3).*

Now, all that must be done is to build the logic that remains in State 4 when none of the conditional paths away from State 4 are true. The path leading away from State 4 is valid whenever the product, $A*B*\overline{C}$, is true. Consequently, State 4 must be ANDed with the inverse of the product, $A*B*\overline{C}$. In other words, "keep loading the flip-flop with a high until a valid transfer to the next state occurs." The default path logic uses AND-7 and shares the output of AND-6.

Configuring the logic to handle the remaining states is very simple. State 2, for example, has only one conditional path, which comes from State 1 whenever the product $A*\overline{B}*C$ is true. However, the state machine will immediately branch in one of two ways from State 2, depending on the value of D. There's no default logic to remain in State 2 *(Fig. 4, top).* State 3, like States 1 and 4, has a default state, and combines the A, D, State 2, and State-3 feedback to control the flip-flop's D input *(Fig. 4, bottom).*

State 5 feeds State 6 unconditionally. Note that the state machine waits until variable E is low in State 6 before proceeding to State 7. Again, while in State 7, the state machine waits for variable E to return to true before moving to State 1 *(Fig. 5).*

OUTPUT DEFINITIONS

After defining all of the state transition logic, the next step is to define the output logic. The three output signals—Single, Multi, and Contig—each fall into one of three primary output types:

1. Outputs asserted during one state, which is the simplest case. The output signal Single, asserted only during State 6, is an example.

2. Outputs asserted during multiple, contiguous states. This appears simple at first glance, but a few techniques exist that reduce logic complexity. One example is Contig. It's asserted from State 3 to State 7, even though there's a branch at State 2.

3. Outputs asserted during multiple, non-contiguous states. The best solution is usually brute-force decoding of the active states. One

ONE-STATE VS. BINARY ENCODING METHODS

Method	Number of logic blocks	Worst-case performance
One-hot	7.5	40 MHz
Binary encoding	7.0	34 MHz

DESIGN APPLICATIONS

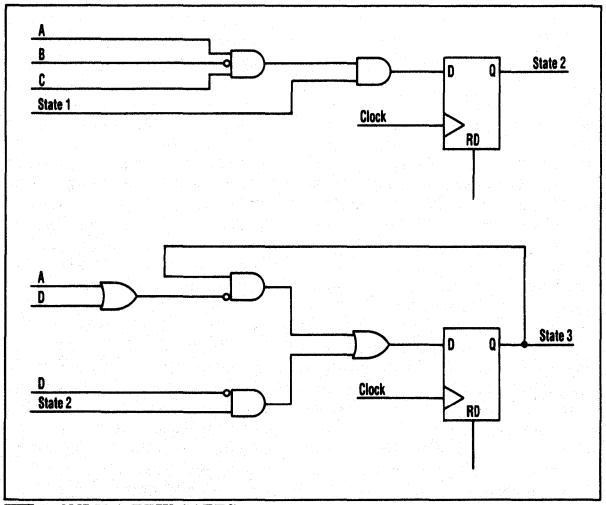
STATE MACHINE DESIGN

OHE state machines is simple, lending itself to a "cookbook" approach. At first glance, designers familiar with PAL-type devices may be concerned by the number of potential illegal states due to the sparse state encoding. This issue, to be discussed later, can be solved easily.

A typical, simple state machine might contain seven distinct states that can be described with the commonly used circle-and-arc bubble diagrams *(Fig. 1)*. The label above the line in each "bubble" is the state's name, the labels below the line are the outputs asserted while the state is active. In the example, there are seven states labeled State 1-7. The "arcs" that feed back into the same state are the default paths. These will be true only if no other conditional paths are true.

Each conditional path is labeled with the appropriate logical condition that must exist before moving to the next state. All of the logic inputs are labeled as variables A through E. The outputs from the state machine are called Single, Multi, and Contig. For this example, State 1, which must be asserted at power-on, has a doubly-inverted flip-flop structure *(shaded region of Fig. 2)*.

The state machine in the example was built twice, once using OHE and again with the highly encoded approach employed in most PAL designs. A Xilinx XC3020-100 2000-gate FPGA was the target for both implementations. Though the OHE circuit required slightly more logic than the highly-encoded state machine, the one-hot state machine operated 17% faster *(see the table)*. Intuitively, the one-hot method might seem to employ many more logic blocks than the highly encoded approach. But the highly encoded state machine needs more combinatorial logic to decode the encoded state values.



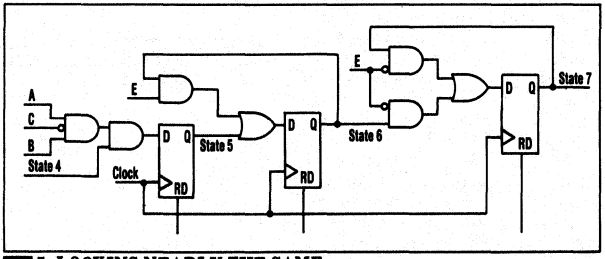
4. ONLY A FEW GATES are required by States 2 and 3 to form simple state-transition logic decoding. Just two gates are needed by State 2 (top), while four simple gates are used by State 3 (bottom).

The OHE approach produces a state machine with a shift-register structure that almost always outperforms a highly encoded state machine in FPGAs. The one-state design had only two layers of logic between flip-flops, while the highly encoded design had three. For other applications, the results can be far more dramatic. In many cases, the one-hot method yields a state machine with one layer of logic between clock edges. With one layer of logic, a one-hot state machine can operate at 50 to 60 MHz.

The initial or power-on condition in a state machine must be examined carefully. At power-on, a state machine should always enter an initial, known state. For the Xilinx FPGA family, all flip-flops are reset at power-on automatically. To assert an initial state at power-on, the output from the initial-state flip-flop is inverted. To maintain logical consistency, the input to flip-flop also is inverted.

All other states use a standard, D-type flip-flop with an asynchronous reset input. The purpose of the asynchronous reset input will be discussed later when illegal states are covered.

Once the start-up conditions are set up, the next-state transition logic can be configured. To do that, first examine an individual state. Then



5. LOOKING NEARLY THE SAME as a simple shift register, the logic for States 5, 6, and 7 is very simple. This is because the OHE scheme eliminates almost all decoding logic that precedes each flip-flop.

DESIGN APPLICATIONS

STATE MACHINE DESIGN

such example is Multi, which is asserted during State 2 and State 4.

OHE makes defining outputs easy. In many cases, the state flip-flop is the output. For example, the Single output also is the flip-flop output for State 6; no additional logic is required. The Contig output is asserted throughout States 3 through 7. Though the paths between these states may vary, the state machine will always traverse from State 2 to a point where Contig is active in either State 3 or State 4.

There are many ways to implement the output logic for the Contig output. The easiest method is to decode States 3, 4, 5, 6, and 7 with a 5-input OR gate. Any time the state machine is in one of these states, Contig will be active. Simple decoding works best for this state machine example. Decoding five states won't exceed the input capability of the FPGA logic block.

Additional Logic

However, when an output must be asserted over a longer sequence of states (six or more), additional layers of decoding logic would be required. Those additional logic layers reduce the state machine's performance.

Employing S-R flip-flops gives designers another option when decoding outputs over multiple, contiguous states. Though the basic FPGA architecture may not have physical S-R flip-flops, most macrocell libraries contain one built from logic and D-type flip-flops. Using S-R flip-flops is especially valuable when an output is active for six or more contiguous states.

The S-R flip-flop is set when entering the contiguous states, and reset when leaving. It usually requires extra logic to look at the state just prior to the beginning and ending state. This approach is handy when an output covers multiple, non-contiguous states, assuming there are enough logic savings to justify its use.

In the example, States 3 through 7 can be considered contiguous. Contig is set after leaving State 2 for either States 3 or 4, and is reset after leaving State 7 for State 1. There are no conditional jumps to states where Contig isn't asserted as it traverses from State 3 or 4 to State 7. Otherwise, these states would not be contiguous for the Contig output.

The Contig output logic, built from an S-R flip-flop, will be set with State 2 and reset when leaving State 7 *(Fig. 6)*. As an added benefit, the Contig output is synchronized to the master clock. Obvious logic reduction techniques shouldn't be overlooked either. For example, the Contig output is active in all states except for States 1 and 2. Decoding the states where Contig isn't true, and then asserting the inverse, is another way to specify Contig.

The Multi output is asserted during multiple, non-contiguous states—exclusively during States 2 and 4. Though States 2 and 4 are contiguous in some cases, the state machine may traverse from State 2 to State 4 via State 3, where the Multi output is unasserted. Simple decoding of the active states is generally best for non-contiguous states. If the output is active during multiple, non-contiguous states over long sequences, the S-R flip-flop approach described earlier may be useful.

One common issue in state-machine construction deals with preventing illegal states from corrupting system operation. Illegal states exist in areas where the state machine's functionality is undefined or invalid. For state machines implemented in PAL devices, the state-machine compiler software usually generates logic to prevent or to recover from illegal conditions.

In the OHE approach, an illegal condition will occur whenever two or more states are active simultaneously. By definition, the one-hot method makes it possible for the state machine to be in only one state at a time. The logic must either prevent multiple, simultaneous states or avoid the situation entirely.

Synchronizing all of the state-machine inputs to the master clock signal is one way to prevent illegal states. "Strange" transitions won't occur when an asynchronous input changes too closely to a clock edge. Though extra synchronization would be costly in PAL devices, the flip-flop-rich architecture of an FPGA is ideal.

Even off-chip inputs can be synchronized in the available input flip-flops. And internal signals can be synchronized using the logic block's flip-flops (in the case of the Xilinx LCAs). The extra synchronization logic is free, especially in the Xilinx FPGA family where every block has an optional flip-flop in the logic path.

Resetting State Bits

Resetting the state machine to a legal state, either periodically or when an illegal state is detected, gives designers yet another choice. The Reset Direct (RD) inputs to the flip-flops are useful in this case. Because only one state bit should be set at any time, the output of a state can reset other state bits. For example, State 4 can reset State 3.

If the state machine did fall into an illegal condition, eventually State 4 would be asserted, clearing State 3. However, State 4 can't be used to reset State 5, otherwise the state machine won't operate correctly. To be specific, it will never transfer to State 5; it will always be held reset by State 4. Likewise, State 3 can reset State 2, State 5 can reset State 4, etc.—as long as one state doesn't reset a state that it feeds.

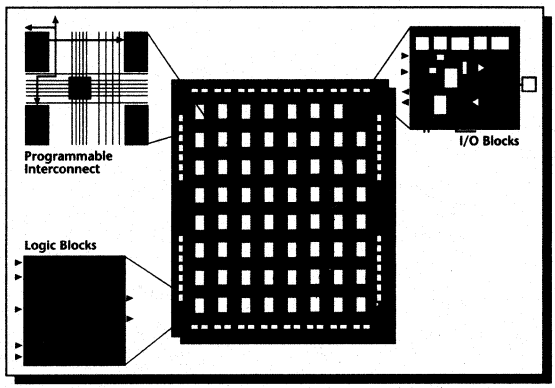
This technique guarantees a periodic, valid condition for the state machine with little additional overhead. Notice, however, that State 1 is never reset. If State 1 were "reset," it would force the output of State 1 high, causing two states to be active simultaneously (which, by definition, is illegal).□

Design In

Reprogrammable Missile:

How an FPGA Adds Flexibility to the Navy's Tomahawk

By Kent Tallyn
Design Engineer
McDonnell Douglas
Electronic Systems Co.
St. Louis, Mo.



PROGRAMMABLE GATES: Xilinx's field programmable architecture breaks down into three categories — I/O Blocks, Logic Blocks, and Programmable Interconnects.

When the McDonnell Douglas Missile Systems Co. set out to design the Digital Scene Matching Area Correlator IIA — an upgraded guidance subsystem for the Navy's conventional land-attack variant of the Tomahawk Cruise Missile — we at the Electronic Systems Co., who were given the task, planned to integrate the unit's logic functions in a conventional gate array.

But the development cost of gate arrays and the lack of hard design specs changed our minds. We needed rapid turn-around from the time we made design changes to the time we had a working part. So we decided to switch.

The conventional gate array plan was scrapped in favor of using a field programmable gate array, or FPGA, and the payoff was immediate: McDonnell's engineers could process the design from the schematic level to a working part themselves, without having to let any of the design data leave their sight. What's more, if initial prototypes didn't work or requirements changed, they wouldn't have to worry about the expense and time involved with redoing the chip's mask layer — changes could be made in software alone.

The conventional land-attack Tomahawk is a long-range Navy cruise missile designed to perform a variety of missions. Flying at low altitudes and high, subsonic speeds, the missile's range — in any weather, day or night — is 500 to 700 miles, and it can be launched from either surface ships or submarines.

Key to the system's ability to complete its missions is the Digital Scene Matching Area Correlator — the DSMAC IIA. That subsystem receives video input from an onboard camera, digitizes it, and compares it to pictures previously stored in memory. Once a match is found, the missile can determine its exact location relative to its "on-course" position and make adjustments accordingly.

The DSMAC IIA is based on a Performance Semiconductor Corp. Mil-Std-1750A microprocessor, which first determines the proper scan rate and passes this information to a set of counters which generate the timing signals for the digitizer. The video image is passed through a set of digital filters, stored in memory, and then compared by the processor to selections from a library of existing pictures to match the new data to a known location. Based on this information, control signals are generated to guide the course of the missile.

That's where the FPGA, a 4,200-gate Logic Cell Array from Xilinx Inc., San Jose, Calif. comes into the picture.

McDonnell programmed the part to generate the timing signals for the digitizer and the address bits for storage. The DSMAC IIA was designed to operate in either of two modes, depending on the mission at hand. But rather than designing separate logic for each mode, McDonnell engineers drew on the programmable gate array technology and designed the system so the operating software for each mode would be kept on-board in read-only memory.

The DSMAC IIA was designed to operate in two modes, depending on the mission at hand.

Depending on the mode of operation, then, the FPGA can be configured in mid-flight — according to the needs of the system software. The concept will have other payoffs in the future. Five years down the line, if the Navy wants to add new features, they'll be able to because it's just a matter of loading new flight software. Hardware need not be changed.

That bonus is what led McDonnell Douglas to Xilinx's LCA. Unlike some other FPGAs, which can't be reprogrammed, these static-random-access-memory-based parts permit changes to be made to a system's logic functions simply by reconfiguring the programmable logic in the system.

Like a microprocessor, the LCA is a program-driven device. The architecture features three types of user-configurable elements: an interior array of logic blocks, a perimeter of I/O blocks, and programmable interconnection resources. Configuration is established by programming internal static memory cells that determine the logic functions and interconnections. The configu-

ration programs can be loaded automatically at power-up or upon command at any time.

The functions of the LCA's configurable logic and input/output blocks — CLBs and IOBs, for short — and their interconnections, are controlled by a program stored in an on-chip memory. Each CLB contains combinatorial logic and storage registers. The logic section of the block uses its inputs, outputs, or bi-directional pins. Inputs can be programmed to either TTL or CMOS thresholds. The programmable interconnect switches connect the inputs and outputs of CLBs and IOBs to nearby interconnect and long lines run the length of the part to provide low skew paths for critical signals.

A knowledge of the architecture, although helpful, is not necessary for designing with LCAs. Design implementation software provided with the system automatically translates a design into a working part, enabling the engineer to work at a PC or workstation.

The first step in designing with programmable gate arrays is schematic entry. Interfaces and libraries are available for the most popular schematic capture systems. Entry through Boolean equations or from a variety of state machine languages is also supported.

Simulation

Once the design is entered, unit delay simulation can be performed to verify the design's functionality. Next, the design gets partitioned into CLBs and IOBs using a translation program that lets the user select the way the part is mapped: designs can be partitioned for performance — so that only related logic is put together on a given cell — or for density — so that the maximum level of integration can be achieved.

After the design has been mapped, an automatic placement and routing program determines the optimal placement for the logic blocks and routes the interconnecting nets. A manual design editor can be used here to pre-reroute or reroute critical signals to ensure that timing specifications are met. When that's done, the part can be simulated to show potential performance data and then implemented in a system.

Solution

Critical issues facing the DSMAC II design team included size, development cost, and rapid turn around. The field programmable gate array, selected in a 25-mil ceramic gull-wing surface-mount package, solved these problems. In addition, the Xilinx LCA reconfigurable architecture allows for future upgrades with no hardware impact. ▲

Reprinted with permission from *Military and Aerospace Electronics* April 1990.

Pivoting Monitor Increases Versatility Of Workstations

Creating a pivoting display monitor requires the seamless integration of mechanical design, hardware and software.

by **Julien Tan-Nguyen,**
Terry Oyama,
and **Nick Moss,**
Radius Inc.

Personal computer users spend most of their time with two application categories: word processing and spreadsheet. When working with a word processing application a portrait display is preferred. When working with a spreadsheet application a landscape display is preferred. To be able to effectively use both types of applications, one can buy a larger display. However, a larger display requires a larger tube which is more expensive and also requires more desk space. Another solution would be to create a compact display which can be used in either portrait or landscape mode—a pivoting monitor.

The goals for creating such a monitor were:

- **High quality display.** The display has to be very stable (with a refresh rate of 69 Hz), very sharp (with advanced focus control circuitry), and gray scale capable (1, 2 or 4 bit/pixel modes).
- **Effortless rotation of the monitor in real time.** The user can easily rotate the monitor without having to restart the computer or leave the application being used.
- **Compatibility with major Macintosh applications.**

Mechanics Of The Pivot Monitor

The mechanics of the display allow the monitor to change orientation in a few seconds effortlessly without any complicated manipulation.

Similar to the mixer of a cement truck, the rotation of the pivot monitor is accomplished on the rear housing. The front and rear cylindrical sections rest on a rotation cylinder on the inside of the monitor which turns on teflon rollers. This rotation cylinder is fixed to the monitor case and screen. The monitor is balanced because the center of gravity barely moves during the rotation operation. Using teflon on the rollers creates a smooth silent surface which wears evenly, outlasting the life of the machine. The base assembly is a tilt-swivel type that attaches to the rear housing.

While the monitor is in portrait position, the left hand corner of the bezel has a 45° chamfer that allows the display to be rotated in the complete forward tilt position.

The ventilation of the monitor has to work in both orientations. Fans were not used in the monitor so as to reduce the noise factor. In order to ventilate the monitor, the electronics were laid out taking advantage of the convection cooling in both orientations. All four sides of the monitor have ventilation slots and the bottom part of the case is lifted off the base allowing air to come through.

Interface Hardware Design

The image on a CRT is displayed by an electron beam scanning quickly from left to right (fast scan direction), and slowly from top to bottom, similar to the way a TV screen works. When a monitor pivots, the left and right, and top and bottom references are no longer fixed. The top side in a landscape mode becomes the left side in the portrait mode.

If the electron beams always scanned from left to right, the picture tube would rotate while the deflection circuitry remained fixed. Enormous efforts would be required to control the pixel aspect ratio, the picture dimension and position, and the beam focusing. The advantage would be that the standard interface board would work without any major modifications because the consecutive pixels are also arranged from left to right in both orientations.

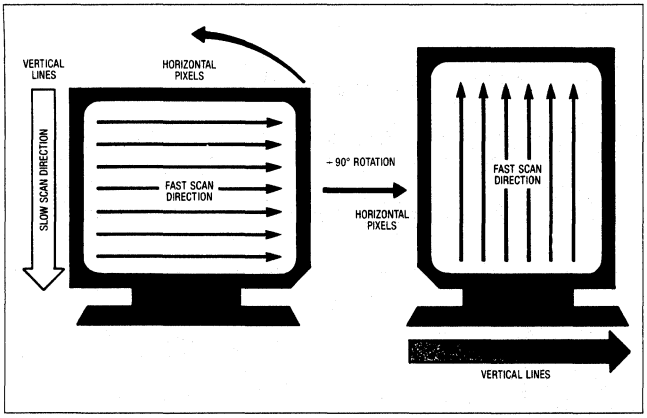


Fig 1 In the Macintosh software graphics format the pixels are accessible along a horizontal line and are numbered from left to right.

However, the better solution would be to rotate the entire display housing. The display electronics do not change from one orientation to another. The electron beam scans from left to right in landscape orientation and bottom to top in portrait orientation. The deflection circuitry and the picture tube rotate as a single component causing the picture geometry to remain the same. This ensures the best possible picture quality in both orientations.

The pixel data is stored in Video RAM on the display controller in the computer. The display controller sends the data through the serial port in increasing address from left to right for the landscape mode. The desired effect is to keep the serial port along the fastest scan direction, meaning left to right in the landscape mode and bottom to top in the portrait mode.

In the Macintosh software graphics interface the pixels are accessible along a horizontal line and are numbered from left to right 0, 1, 2, 3, etc. (Fig 1). In the landscape mode the pixels are arranged in the same direction as the beam scans. In the portrait mode the pixels are also accessed along the horizontal lines and numbered from left to right 0, 1, 2, 3— however, the fast scan direction is now changed to bottom to top.

Therefore, pixel addresses need to be renumbered in the portrait mode. This renumbering corresponds to a 90° rotation of the pixel addresses. To be compatible with the Macintosh applications this rotation must be transparent to the software graphics interface and performed in real time. In a 90° rotation the top left and bottom right pixel addresses (e.g. 0 and nw – 1) are preserved while the rest of the pixel addresses are rotated. In other words the pixel address n – 1 becomes w – 1 and (w – 1)n become (n – 1)w.

The rotation circuitry is part of the interface board which contains the memory to store the frame buffers (Video RAM). It also contains the control circuitry to the interface with the data bus (1st Translation Xilinx PGA), the control circuitry to the interface with the video circuitry which builds up the video stream (2nd Translation Xilinx PGA), and the digital to analog converter which sends the signal to the monitor (VDAC) (Fig 2). Part of the rotation is performed when the pixels are sent to memory, and the rotation is completed when the pixels are sent to the monitor. The final bit stream sent to the monitor is at 50 MHz. The bitstream is split into two 25 MHz substreams in the Translation circuitry.

There are six modes of operation—1, 2 and 4 bits/pixels for the various shades of gray in a portrait orientation and 1, 2 and 4 bits/pixels in a landscape orientation. The pixel addressing for each mode is different. The interface board must contain the circuitry required to translate each of the six modes. This circuitry, if implemented in a classical gate array, would require around 6000 gates.

The solution to high density and high speed requirements was a Field Programmable Gate Array (FPGA). The logic for each of the six modes fits into an 1800-gate device. The monitor operates in one mode at a time and only 1800 gates of logic were needed for each mode. The reprogrammability feature of the Xilinx Logic Cell Array (LCA) allowed the monitor to operate in these six different modes while using only one logic device. The LCA being a reprogrammable logic device, configuration bitstream determines the

Misconvergence In Color Monitors

In a monochrome monitor, the DC component from the earth's magnetic field can cause the picture to shift and tilt slightly when the monitor is turned. The solution is to build a shield around the front face of the picture tube. However, in a color monitor the problems caused by the earth's magnetic field are more complicated, since there are three electron beams and they are traveling at a higher energy.

All beams have to land on the face of the display in such a way that the geometry of the image remains the same. This means that the borders of the display are straight and the image does not shift or tilt. In a monochrome display there is only one beam to worry about. In color display three beams are used to create one color pixel with all three beams accurately converged on the same spot of the display phosphors. If any of the three beams become misaligned, the color will smear. An error as small as .1 mm in the beams' convergence will cause color smearing.

The electron beam in a monochrome display can withstand a larger shift in comparison to a color monitor. Therefore the shield used in the monochrome monitor may not be enough to prevent color misconvergence for a rotating color monitor.

The electrons in a color monitor have a much higher energy level. The acceleration voltage is around 20-30,000V as compared to 10-15,000V in a monochrome monitor. Therefore the shield must account for the increase in the energy of the electrons. In addition, color monitors are heavier and the mechanics of the rotation will be more difficult. ■

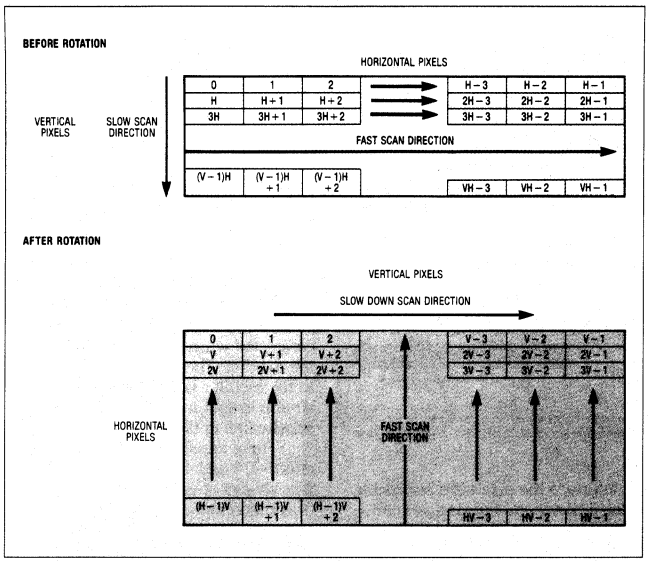


Fig 2 The rotation circuitry is part of the interface board which contains memory to store the video RAM.

functionality of the device. A position sensitive device (somewhat like a mercury switch) is used to detect which direction the monitor is in, either portrait or landscape. An interrupt is sent to the system software to tell it that the monitor is changing direction. The system software determines which of the six configuration bitstreams should be loaded and then automatically loads it into the FPGA. The FPGA is reprogrammed in one msec while the software is updated to the new mode.

The LCA reduced the amount of gates which needed to be used. In addition the FPGA made the developing of the hardware much easier. Each of the six modes of operation could be developed separately eliminating fringe effects from the clock loading from the other modes which allowed the circuit to run at the required 25 MHz.

Earth's Magnetic Field

When designing a monitor, the earth's magnetic field has an influence on the quality of the picture. The deflection inside the picture tube is based on an AC magnetic field. On top of the AC magnetic field is the earth's DC magnetic field. When the monitor is turned on its side the DC noise can cause the picture to shift and tilt slightly. The DC component is normally compensated for before the monitor leaves the factory. To alleviate this problem a silicon steel magnetic shield is placed around the front face of the picture tube. This shield also contains the deflection field inside the display, therefore minimizing radiation leakage. As a consequence, interference from other displays will also be reduced.

In color monitors the problems caused by the earth's magnetic field are more complicated because there are more electron beams traveling with a higher energy.

Software Development

Once the monitor is rotated, the software needs to tell the system that the shape of the desktop has changed. The desktop is the graphical metaphor used by Apple to represent the computer display. On the desktop there are windows with applications, and icons representing these applications windows which are closed. Any application running on the desktop can query the system as to the size of the new desktop.

In addition, a cleanup is performed moving the icons on the desktop to a location where they can be accessed. When the display is rotated the shape of the desktop is changed. The system icons in the common area of rotation remain the same. However, the system icons in the removed area must be relocated. In Fig 3 the system icon is illustrated with the Macintosh Trashcan icon. For example, if the monitor is in the portrait mode and a system icon is located in the bottom right of the desktop, it will be moved to the upper right hand of the screen when the monitor is rotated to

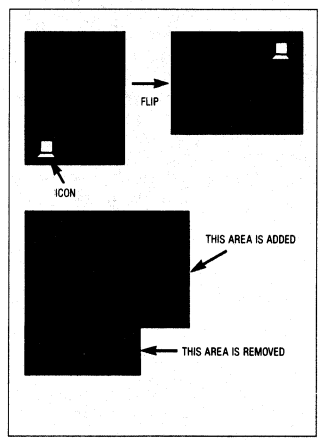


Fig 3 When the display is rotated, the system icons in the common area of rotation remain the same.

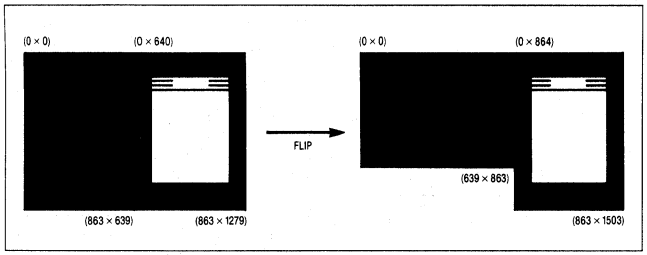


Fig 4 Global coordinates of the other displays may also need to be updated to maintain overlapping, non-contiguous characters.

the landscape position allowing it to be accessed.

Multiple Screen Support

Another complication to the software is that the Macintosh supports a multiple monitor display mode. This is where two or more monitors can be placed side by side in order to provide a contiguous drawing area (i.e. there are no gaps between the monitors allowing the user to move the mouse freely between the displays). Each screen has its own local coordinate system which is translated into global coordinates for drawing. When the operating system builds the desktop from the available display, it assigns their global coordinates such that the displays do not overlap and the drawing space is continuous.

When the monitor is rotated, some desktop area is added and some is removed. Not only do the local coordinate systems of the display change, the assignment of the global coordinates of the other displays may also need to be updated to insure that the display maintains their contiguous, non-overlapping nature (Fig 4).

The rotation of the left-hand display causes the local and global coordinates to change from (top: 0, left: 0) and (bottom: 863, right: 639) to (0,0) and (639,863). The right-hand display's global coordinates change from (0,640) and (863,1279) to (0,864) and (863,1503).

The window on the desktop's coordinates change from (100,800) and (700,1100) to (100,1024) and (700,1324) after the flip.

Software Application Support

When the monitor pivots, most applications will truncate the portion of the window that no longer exists in the new orientation. A small number of applications which bypass the systems software graphics interface, QuickDraw, and directly manipulate the display's video frame buffer, render a scrambled mess on the screen.

The goal is to work with the software development community to take advantage of the new orientation by automatically adjusting the window. This means that when the display is rotated the application detects the pivot, moves and resizes the windows to take advantage of the new screen size.

Utilizing hooks installed in the operating system, applications can query the status of the display to detect a pivot. For each of its windows, the application can then request a hint, a message returned to the application containing a location and size for the window which makes optimum use of the new desktop shape.

However, a vast majority of the applications work with the pivot monitor without any modifications. The hooks available from the operating system will allow these applications to have additional functionalities.

Creating a pivot display monitor was made possible only through the tight and seamless integration of the mechanical design, the hardware and the software. The mechanics of the monitor make rotation a simple one-hand operation. The reprogrammability feature of the Xilinx LCA allowed the monitor to operate in six different modes using only one reprogrammable logic device. The software made the desktop dynamic, allowing applications to take full advantage of the display space. ■

Reprinted with permission from the Publisher from the Fall 1990 Computer Technology Review.

Electronic Engineering
TIMES

Two, Two, Two Chips in One

By Tom Liehe, Principal Design Engineer, Test Instrument Division, Honeywell Inc., Denver, Colo.

Designers at Honeywell picked the RAM-based Xilinx LCA for its short development cycle, and realized savings in board real estate through its dynamic reprogrammability.

Advances in one technology often lead to improvements in other, more dated design and manufacturing practices. A recent example of this occurred at Honeywell during the development of a high-capacity digital tape recorder.

Honeywell's original objective was to design the VLDS (very large data storage) recorder, taking maximum advantage of the available analog technology currently being used in standard VCRs for home use. The recorder developed under this program uses digital rotary technology to record large amounts of data on a standard VHS video cassette. It transfers data at a rate of 4 Mbauds, and is able to store 5.2 Gbauds of information on a single BHS tape. Its major planned application is in capturing and storing digital medical images, such as those produced by a CAT scanner.

When this recorder was in the prototype stage, it became apparent that the addition of an error-correction circuit would significantly enhance system performance. This requirement dictated the design of an entirely new and major logic circuit to accomplish the desired error correction.

Design of this circuitry would not normally be a problem, but at this stage of development, there were several challenges. First, the design allowed almost no circuit board space for addition of the error correction code (ECC) circuitry. Second, very tight deadlines were being faced if the promised delivery date was to be met.

The entire system was housed in a 19-inch-wide by 20-inch-deep rack-mounted cabinet. The cabinet already contained eight separate circuit boards, and there was room enough for only one additional board to incorporate the ECC circuitry. Space was at a premium. The goal was to design and manufacture a 10^{-12} corrected bit error rate circuit that could be contained on one circuit card. The targeted time for completion of this work was three months.

Errors on tape typically are caused by tape defects, dirt, head clogs, etc. Because these error bursts can be thousands of bits long, sophisticated ECC techniques are required. Initially, two basic circuits, using Reed-Solomon algorithms and TTL technology, were designed. These were the ECC encoder and decoder.

The write portion of the circuitry (the encoder) uses a byte-wide linear feed-back shift register (LFSR) to create a 68-byte code word form each 64-byte incoming message block. During operation, parity check bits are computed based on the data within a block of the message to be encoded. These check bits are appended to the block to create the code word.

During decoding, the code words are checked for errors by regenerating the parity bits which are then compared with the check bits. If they match, it is assumed that no errors have occurred. If they do not, the pattern of mis-matches (called the syndrome of the error) is used to compute the corrected form of the message block.

The ECC decoder (the read circuit) required a partial syndrome generator and the solution of a set of simultaneous non-linear equations to determine error locations and values. This error-determination step is performed by a special-purpose processor with a microinstruction sequencer, a finite field arithmetic unit, two discrete registers and an eight-word memory. The correction step is then accomplished in circuitry whereby the error values are exclusive-ORed with the message at the address given by the previously computed error locations.

Using wrapped-wire techniques, a working prototype of the ECC circuitry was developed. However, it was quickly recognized that the long lead time required to design and fabricate a factory-programmed gate array to replace this prototype TTL circuit was not practical with the tight delivery schedule.

An option considered, but not chosen, was to design and fabricate a conventional gate array. The considerable design time required, together with the inherent risks associated with masking and manufacturing a custom logic circuit, made this an unattractive alternative.

XILINX'S LCA

Finally, the search for an alternative solution led to the discovery of a programmable gate array known as the logic cell array (LCA), designed and manufactured by Xilinx Inc. (San Jose, Calif.). The LCA is a standard, off-the-shelf device that is custom configured to the customer's requirements by means of the Xilinx development system. This development package consists of a personal-computer-based software system combined with an in-circuit emulator.

Use of the LCA seemed to be the ideal solution to the time constraints. So, a Xilinx XC2064 LCA was sected. In this device, any logic function having up to four variables can be implemented in any one of the 64 configurable logic blocks (CLBs). Optionally, results can be stored in either a latch or a flip-flop. Thus, implementation of the design can be constrained by a fixed set of standard logic elements.

The I/O pins of this device also can be configured as registered inputs. The large number of flip-flops, plus the ability of each CLB to function as four-input exclusive-ORs, made this LCA ideal for ECC circuit implementation.

MULTIFUNCTION CAPABILITY

One of the real benefits of this LCA is its multifunction capability. The capability of performing a number of functions with the same device provides optimum utilization of circuit board space. This was a real bonus with the VLDS recorder. At any given time, the VLDS operates in only the read or the write mode—it is never required to do both simultaneously. Consequently, the same LCA could be reconfigured electronically to perform one function in the write mode, and a completely different function in the read mode. This versatility eliminated the need for two separate circuits, and thereby conserved space.

The LCA has a usable density of 1,000- to 1,500-gate equivalents, and is capable of replacing up to 75 SSI/MSI devices, five to 15 PAL-type devices, or some combination of both. In the VLDS, the entire ECC encoder and the partial syndrome generator portion of the ECC decoder were replaced by the LCA. The initial encoder circuit used eight identical PALs, each of which implemented a 1-bit slice of the shift register, and four PROMs. The original partial syndrome generator design used six PALs and four 74LS374 tri-state octal flip-flops to store the four syndromes. Thus, the LCA replaced a total of 14 PALs, four 256k x 8 PROMs and four 74LS374s, or a total of 22 20-pin DIPs.

ANOTHER BENEFIT

Another significant benefit derived from the use of the Xilinx LCA was reduced power consumption. The original bipolar IC design consumed approximately 12 W of power. Through the use of CMOS technology, the replacement LCA consumes only 50 mW of power. It should be pointed out that the bipolar version was capable of operating at a much higher clock rate than the LCA. However, the clock rate used this particular design was only 2 MHz. The speed of the LCA was, therefore, adequate for the VLDS application.

Because the required logic circuitry was already designed and tested, the development of the configuration program for the LCA went very smoothly. It took only two days to configure the circuit using the Xilinx XACT LCA development system running on a standard, IBM-compatible personal computer. The primary effort involved was the partitioning of the logic to match the capabilities of the LCA.

For a regular, repetitive design, a small portion of the logic was defined. This portion was then copied and minor modifications were made to complete the design. The byte-oriented nature of the RS ECC circuitry lent itself to easy entry. Starting with tables showing the bits to be exclusive-ORed, the entire circuit was entered in a few hours.

The software simulation capability, which enabled the modeling of physical delays and logic functions, resulted in a very high design confidence factor before the first hardware checkout. The simulator provided both tabular output and logic analyzer style waveforms, which aided considerably in the visualization of the circuit performance. A high-level language program was used to generate expected results of the encoder, and to perform partial syndrome generator simulations. This greatly aided the evaluation of the simulation output.

By using the in-system emulation feature, configurations were loaded directly from the PC to an LCA mounted in the target system. Thus, the usual step of programming an EPROM from which the LCA can boot itself was eliminated. Initial design checkout of the ECC circuitry was performed using the emulator connected to the wrapped-wire board containing the discrete IC version.

There was a problem with the encoder circuit that was delaying data for an extra byte. Correcting this problem required removing the input flip-flops on the LCA. The entire process of reentering the LCA editor, removing the mouse and reloading the new configuration took no more than five minutes.

Compared with the time required to rework any other type of hardware, the LCA is the only way to go. Also, taking into consideration the high costs associated with reworking a factory-programmed gate array, or even a semi-custom PLD, the LCA technology is an extremely cost-effective alternative.

In summary, the Xilinx part was well suited for our application because of its high flip-flop count and its ability to be configured in exclusive-OR trees. Additionally, its capability of being electronically reconfigured while in the system (when switching from write to read) offered significant savings.

Further, power consumption was much lower than when using equivalent discrete ICs. And finally, the ability to perform design entry, simulation, emulation and in-system testing through the software development system facilitated quick and easy implementation of the user's ideas.

Today, the Honeywell VLDS offer error correction as powerful as most major computer tape subsystems. It is ideally suited for the newly developing imaging technologies used in electronic office documents, advanced geophysical analysis and computer-aided graphic arts. Without the Xilinx logic cell array however, Honeywell could still be waiting for a custom gate array.

Reprinted with permission from *Electronic Engineering Times.*

ESD:
THE Electronic System Design Magazine

LCA Stars in Video

by Rusty Woodbury, Interactive Educational Video, Salt Lake City, UT.

Reprinted with permission from *ESD: The Electrical System Design Magazine.*

The market for tools and overlay products for video pictures generated from laser disks is in its infancy. Applications for this emerging video-based technology can require high resolution and high performance, and the wide variety of video disk players employed means that problems associated with varied noise characteristics must be overcome. What works with one particular brand and model in the factory may falter with another brand in the field.

The Xilinx Logic Cell Array (LCA) helps to solve the problem of meeting different system requirements because the device elevates hardware to the same level of programmability as software. Before the LCA, once a design had been committed to hardware, revisions to the design could only be implemented via software changes.

Interactive Educational Video (IEV) has implemented three separate designs and logic replacements with the LCA. These functions reside on IBM PC expansion cards, where space limitations would ordinarily preclude such a design. Although application-specific video ICs could perform similar functions, they cost more than the LCA and offer lower performance.

The first application is a graphics engine that uses four LCAs. Here, the LCAs replace over 50 SSI/MSI chips, including four traditional programmable logic devices (PLDs).

One LCA functions as the address generator for the video memory. By relying on a pair of high-speed counters to locate horizontal and vertical coordinates, memory write functions (which implement line drawing) can perform at high rates. Given the slope, starting point and length of a line, the logic simply increments counters that point to video memory locations. Scanning and writing to the screen are interleaved. The data written to memory corresponds to a particular color and, by simple incremental additions to the slope of the address pointer counters, powerful line drawing functions are easily implemented.

Important to the design is the decision logic, which determines how to increment the counter. All of these functions, plus logic to generate the read/modify/write cycle timing, are implemented in a single LCA that replaces nine MSI parts, four of which are PLDs.

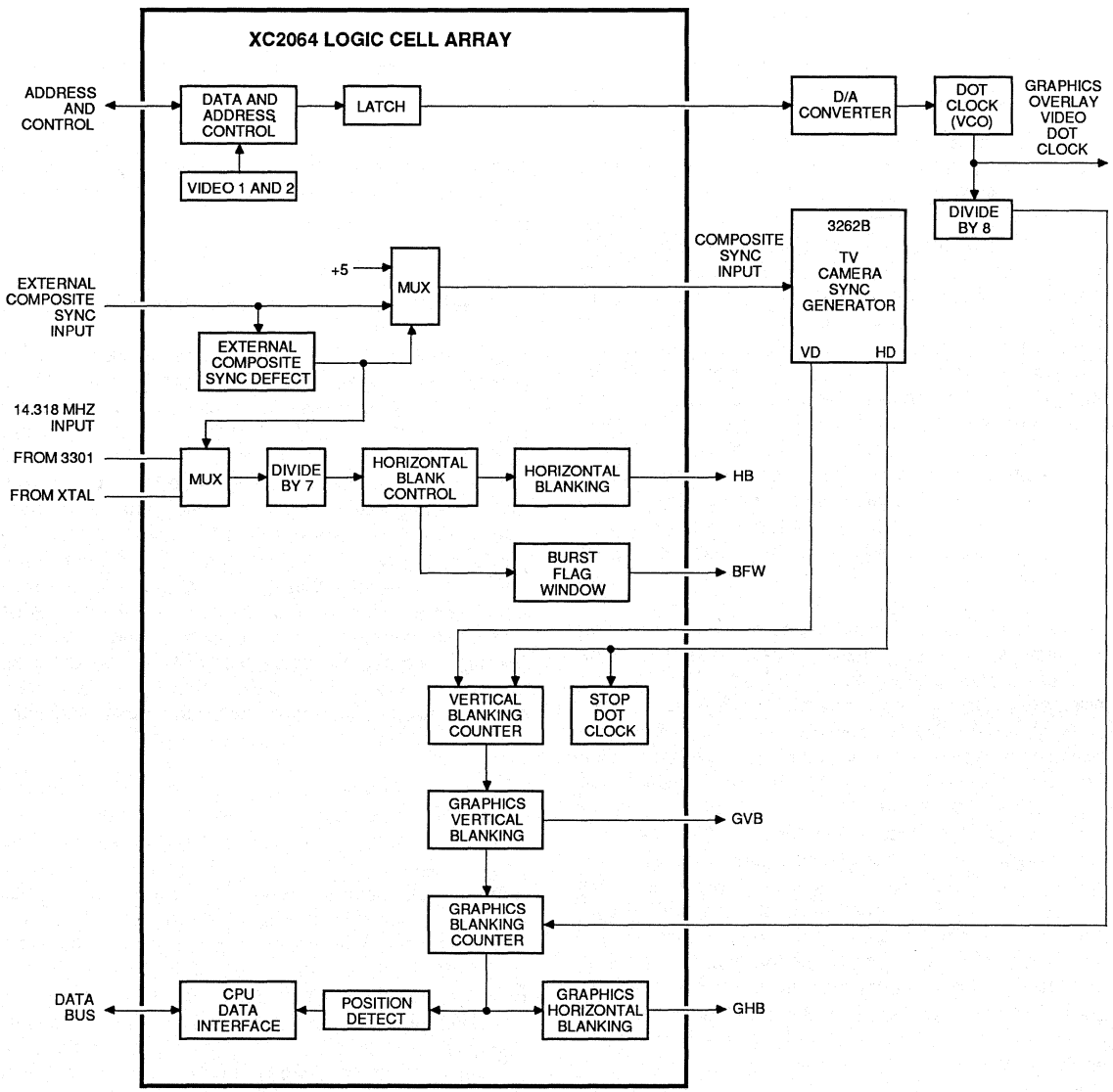
Two more LCAs implement a three-bit ALU. This technique achieves ultra-high-speed screen writes for both horizontal and vertical lines. For many applications, these are the most common lines drawn, so a special control bit is used to simultaneously modify pixels. Horizontal lines can be written at 14 Mpixels/sec instead of the normal 2 Mpixels/sec—a seven-fold improvement. Though more logic could be placed in these two devices, a bit-sliced logic approach permits continuous enhancements. Moreover, a board layout can be defined at the beginning of the product cycle while logic enhancements are made internally in the LCA. Nearly 30 SSI/MSI devices were integrated into the LCAs.

A fourth LCA fully implements the graphics engine. To read out data to the screen, scan counters point to memory. A shift register serializes at a rate of 14 Mpixels/sec. Using traditional MSI devices, these functions require about 10 chips.

The second design fabricated by IEV is a graphics controller (Figure 6). Using an external genlock IC, the LCA relies on an NTSC composite sync signal to generate timing signals for the CRT display. Instead of using PLDs, IEV uses the LCA to implement digital counter and timers. The result is higher performance and reduced complexity. The previous generation board has only half the functionality and demands four times the board space. To further reduce complexity, the same hardware can be used with a different configuration program to match a particular video disk player's noise characteristics. Without the LCA, this design needs eight PALs plus 12 to 15 MSI devices.

In another IEV design, a PC serial port emulator integrates a subset of the IBM PC serial port functions onto the graphics card, making an IBM serial card unnecessary. With the given space restrictions, this implementation proves particularly cost-effective. Seven PLDs are required to match this design.

Reprinted with permission from *ESD: The Electronic System Design Magazine.*



1148 13

Figure 6. IEV implementated an itelligent Graphics Overlay Controller microprocessor peripheral with one XC2064 Logic Cell Array, replacing eight PALs and 12 MSI devices. The controller generates all timing for a video graphics overlay by deriving the necessary timing from the underlying video disk signal.

Taking Advantage of Reconfigurable Logic

by Bradly K. Fawcett, Xilinx Inc., San Jose, CA

An abbreviated version of this paper was published in the High Performance Systems Programmable Logic Guide, 1989.

The availability of programmable logic devices based on static memory cells now allows the implementation of "soft" hardware—hardware whose functions can be changed while resident in the system. When using most current IC component technologies, hardware is indeed "hard"; once a given logic function is implemented in hardware, changing that logic is difficult, requiring modifications to printed circuit board traces, the addition or replacement of components, and other costly measures. However, with static-memory-based programmable logic, changes can be made to a system's logic functions simply by reconfiguring the programmable logic in the system. This capability can lead to significant advantages for the system designer. These include both product-related benefits, in the form of smaller, less expensive, and more reliable systems, and design-related benefits, such as increased design flexibility, decreased risk, and faster design cycles.

Programmable logic devices capable of being reconfigured in the system are available to system designers in the form of programmable gate arrays from Xilinx, Inc. The Xilinx Logic Cell Array (LCA) architecture features three types of user-configurable elements: an interior array of logic blocks, a perimeter of I/O blocks, and programmable interconnection resources. Configuration is established by programming internal static memory cells that determine the logic functions and interconnections. The configuration programs can be loaded automatically at power-up or upon command at any time. Several available configuration loading modes accommodate various system requirements. The benefits of a static-memory-based device include high density, high performance, testability, and the flexibility inherent to a device that can be programmed while resident in a system. Designers have taken advantage of this capability in a wide range of applications.

The flexibility inherent in reconfigurable Logic Cell Arrays (LCAs) can be used to create systems that are also more flexible and, therefore, more powerful. Often systems will include multiple configuration programs for their LCAs, allowing varying operations to be efficiently performed with a minimal amount of hardware. For example, 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. The result is smaller, more powerful, less expensive, and more reliable systems. As an added benefit, use of reconfigurable LCAs simplifies hardware design and shortens a product's time-to-market.

RECONFIGURING FOR SYSTEM DIAGNOSTICS

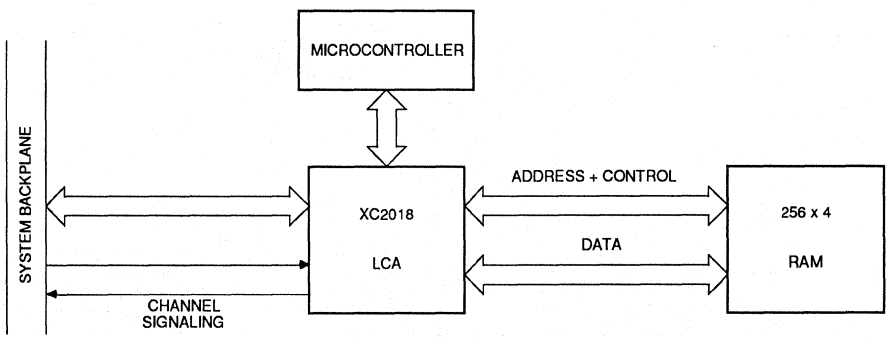
System self-diagnostics can be implemented by using programmable gate array configurations dedicated to testing functions. When the system is powered-up or placed in a test mode, its programmable gate arrays are configured with logic functions dedicated to testing other circuitry in the system. Once the testing is successfully completed, another configuration program is loaded into the programmable gate array to implement the actual logic of the particular end application intended for that system. Typically, very little additional logic is required to add self-diagnostic functions in this manner (usually just some additional memory to hold the extra configuration programs). Such self-diagnostic capabilities make products easier to manufacture, increase system reliability, and simplify system maintenance, with little, if any, additional cost.

Designers at Tellabs Inc. (Lisle, IL) used this strategy in a voice compression module, an optional unit for the Crossnet 440 T1 multiplexer. The design includes two XC2018 devices, 1800-gate programmable gate arrays (Figure 1). During normal operation, one LCA provides all the interface logic for the board's microcontroller, RAM, and system backplane, arbitrating accesses to the RAM from the controller and the main system. The second LCA contains most of the "glue logic" for the data compression operation. However, both LCAs can be loaded with special diagnostic configurations. In the test mode, the first LCA connects the microcontroller to the RAM for memory testing, and monitors controls on the system backplane. The second LCA can receive timing information from the microcontroller instead of the system backplane, verify the data paths, and check the contents of the 32K-bit EPROM used to implement the code converter's companding algorithm. Actually, two different test configurations have been generated, and other diagnostic LCA configurations are planned for a future upgrade. All the configurations are present in memory on the board; the microcontroller handles the downloading of LCA configuration programs.

ADAPTABLE SYSTEM DESIGNS

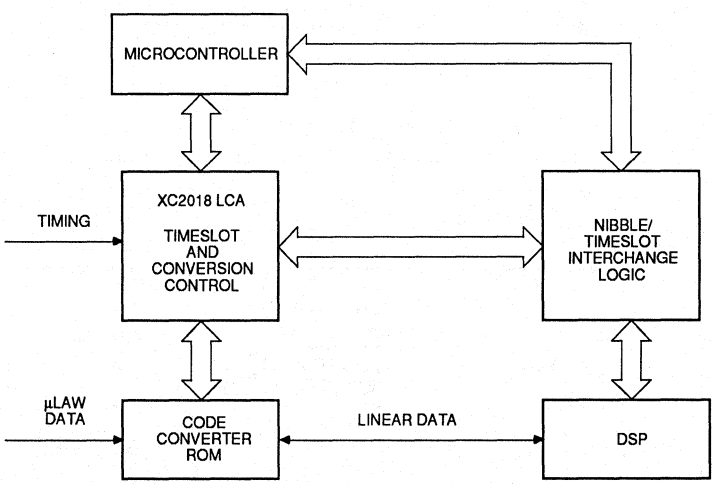
A similar use of reconfigurable logic is the implementation of a single hardware design that can be adapted for varying tasks or environments. In such systems, any of a number of potential configuration programs can be downloaded into a system's LCAs to alter the logic for particular applications or operations as needed. Hence, more functions are implemented with fewer components, hardware design costs can be amortized over a greater number of systems, and design cycle times are greatly reduced. The manufacturer could select the configuration program to be included in the system dependent on the intended end application or customer, or, alternatively, all the different LCA configuration programs could be included in the system with logic that selects the appropriate configuration at the appropriate time. Many different types of applications benefit from this approach.

The Freeland Medical Division of Good Technologies Inc. (Indianapolis, IN) used reconfigurable LCAs in this manner when designing a "frame grabber" board for the Cine' View family of digital imaging systems. A mix of seven XC2064, XC2018, and XC3020 LCAs are used on this AT-format board, providing graphics control and interfacing a PC-compatible computer to the video output of medical equipment such as ultrasound scanners and magnetic resonance imaging systems. In order to support different video formats from the varying types of medical instruments, several different LCA configuration programs



1953 01

An LCA contains interface logic for the micro-controller, memory, and system backplane.



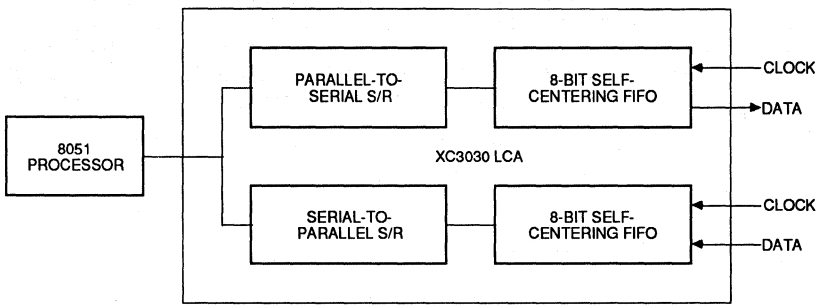
1953 02

A second LCA implements the glue logic for the data compression circuit.

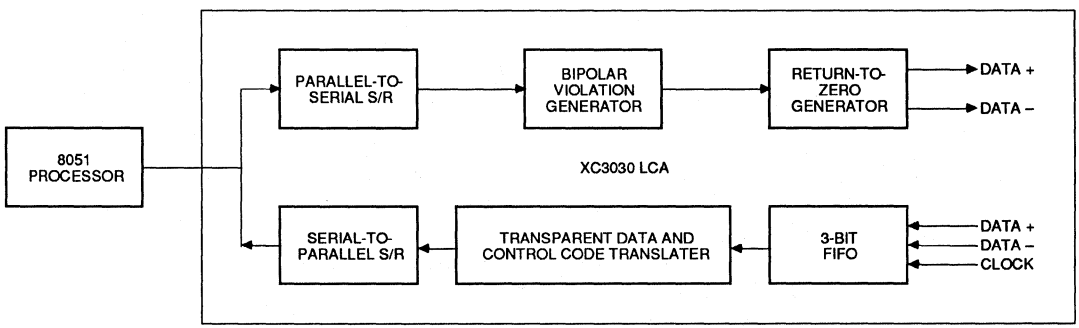
Figure 1. LCAs in a voice compression system can be reconfigured to implement internal system diagnostics.

are available for the LCA devices in the system. When system operation begins, the user selects the desired video format (monochrome or RGB color, for example); the appropriate LCA configuration program is then loaded to match that format. Thus, one hardware design can support virtually any video format, without having to include customized hardware for each one.

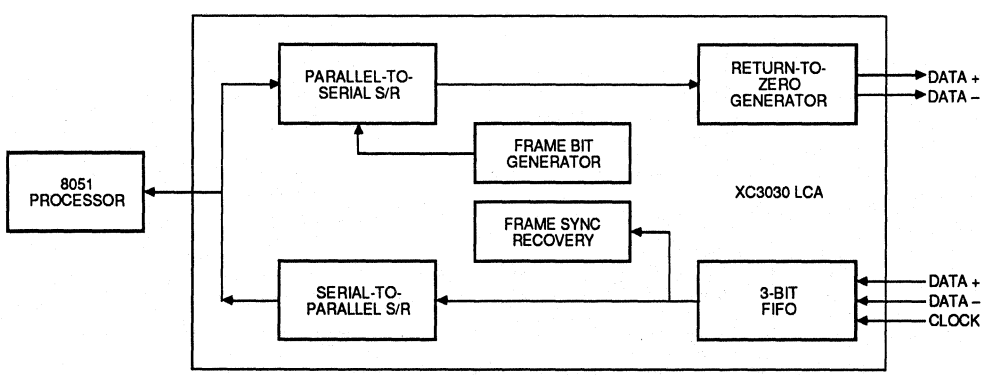
A similar scheme was used on Tellabs' channel interface cards for the Crossnet 440 T1 multiplexer. Each channel's logic consists of an 8051 microcontroller and a 3000-gate XC3030 LCA; four channels are implemented on each card. Using a keyboard, the user can select from among three communication protocols for each channel: a Data Service Unit (DSU) interface, an Office Channel Unit (OCU) interface, or a secondary-mode OCU interface (Figure 2). A fifth 8051 processor controls the user interface and the downloading of the appropriate LCA configuration programs.



DSU mode block diagram



OCU mode block diagram



Secondary OCU block diagram

Figure 2. By reconfiguring an XC3030 LCA, each channel in a multiplexer from Tellabs can implement any of three communication protocols.

Reconfigurable logic can be used to adapt add-in circuit boards to the environment of a particular computer. In such systems, configuration programs can be downloaded by the host processor (from a floppy disk or modem, for example), allowing simple installation procedures and easy field upgrades. Several recently announced personal computer products illustrate this capability. Buffalo Product's (Salem, OR) More Memory memory expansion card for PC/XT or PC/AT compatible systems employs a 1200-gate XC2064 LCA for the bus and memory interface and control logic. An installation program analyzes system parameters (bus width, type of card slot, available address spaces, etc.) and then loads the appropriate configuration program to match the system's requirements. Similarly, the Mach II/SE (Figure 3), an accelerator board for the Macintosh II from Dove Computers (Wilmington, NC), uses an XC2018 LCA for all its interface logic; different LCA configurations are used to support different memory sizes and speeds. The MultiScreen card from Mobius Technologies Inc. (Oakland, CA), a monitor interface board, includes an XC2018 LCA for controlling the video output. Different LCA configurations support different monitor types, allowing for variations in timing requirements and screen resolution. As new monitors are introduced in the market, additional LCA configuration programs will be developed and distributed on floppy disks.

Several other applications involving the use of Xilinx LCAs to implement adaptable hardware have been described in recent articles:

- Tektronix Inc. (Wilsonville, OR) employed an XC2018 LCA for the printer interface logic in their Phaser Card printer controller.[1] Interfaces to several different types of printers can be implemented through reconfiguration of the LCA.

- The FASTPACKET data multiplexer from Stratacom Inc. (Campbell, CA) uses LCAs to incorporate its four serial channel interfaces.[2] Different communication protocols can be accommodated through reconfiguration of the LCAs. A special configuration of the LCAs also provides for bit error rate testing without the use of external test equipment.

- Reconfiguring an LCA in a graphics controller for a laser disk system from Interactive Educational Video (Salt Lake City, UT) allowed a single hardware design to be matched with various video disk players' noise characteristics.[3]

- GTECH Corp. (Providence, RI) designed a lottery bet-slip reader using LCA technology that can be reconfigured to accommodate variations in bet-slip size and format without hardware alterations.[4]

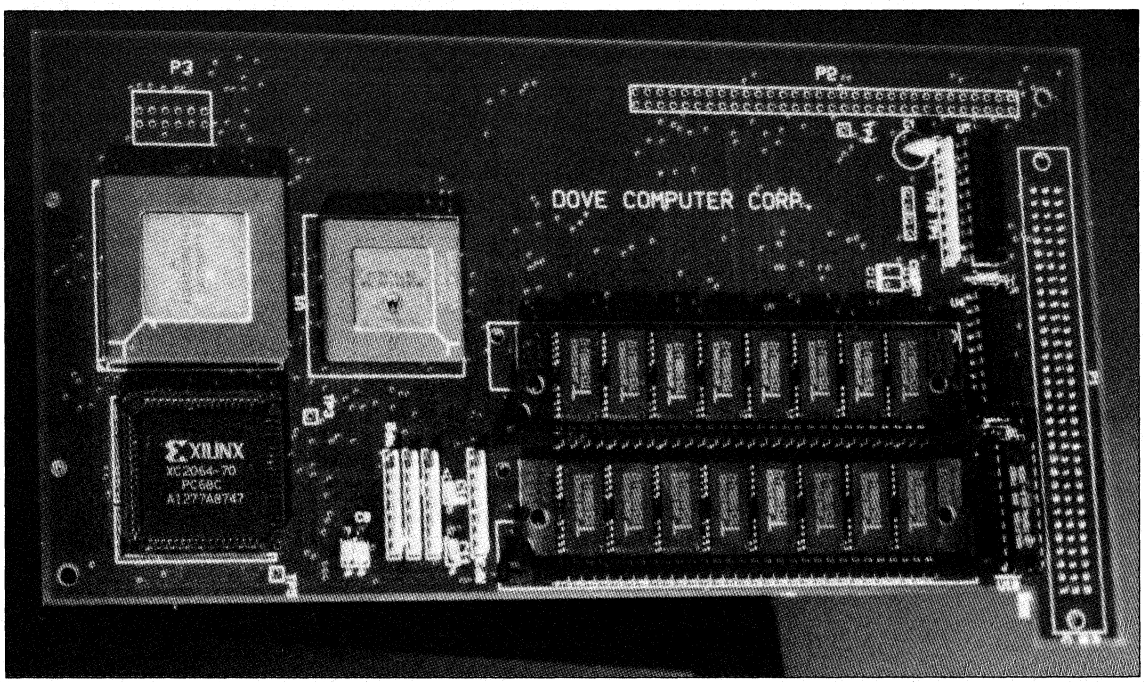


Figure 3. The Dove Computer Mach II/SE includes a micro-processor, floating-point co-processor, memory, bus drivers, and an LCA that holds all the interface logic.

CONFIGURABLE TEST EQUIPMENT

In a similar manner, programmable gate arrays often are used to implement configurable test equipment, wherein different LCA configurations are used to program the same hardware to perform varying types of tests.

Innovage Microsystems (Calgary, Alberta) chose programmable gate arrays for test circuitry used in the Fluke 90 Series (John Fluke Mfg. Co., Everett, WA) and Innovage Microsystems' own Tracer-4 series of microprocessor board testers. These portable test instruments facilitate the trouble-shooting of microprocessor-based boards; testers are available for a number of popular microprocessor types (Z80, 8086, etc.). As shown in Figure 4, an LCA provides interface and control logic between a resident microcontroller and the unit-under-test interface card. An 1800-gate XC2018 LCA is used in the 8-bit series, and a 2000-gate XC3020 is used in the 16/32-bit series of testers. Different configuration programs are stored in the system's ROM during production, dependent on the type of microprocessor targeted for that tester, allowing the same basic hardware configuration for all tester types. A keypad allows the user to choose from a variety of pre-programmed trouble-shooting modes; the microcontroller downloads one of seven different available configuration programs to the LCA, dependent on the type of test selected. Use of the LCA allowed Innovage Microsystems to increase the functionality of their testers while reducing the number of components by 49%, as compared to previous models.

Semiconductor Test Solutions (Santa Clara, CA) included reconfigurable logic in several optional units for their STS 6000 and 8000 series of Sentry-compatible IC testers. For example, an optional memory test unit uses the XC2018 LCA to interface between the internal memory that holds the test patterns and the pins of the memory device being tested. Different LCA configurations are used for testing different types of memory devices. An extended vector memory option uses an XC2018 LCA as a FIFO buffer between the extended memory and the pattern control logic. Upon command, this LCA can be reconfigured to create a cyclic redundancy code (CRC) checker used to verify the test patterns stored in the extended memory.

Designers of telecommunications test equipment have also discovered the advantages of reconfigurable logic. Three LCAs are used in the PC-based TC2000-B1 T1/PCM tester from LP Com, a Tektronix subsidiary (Mountain View, CA). The LCAs provide clock and timing generation for the receiver/transmitter, interface logic, and bit error generation logic. The logic can be altered by downloading different LCA configuration programs to support several user-selected operating modes. When analyzing DS1 lines, any standard framing mode can be selected (D1D, D2, D3/4, or ESF). In DS1 bit error testing (BERT) mode, any AT&T standard or user-defined test bit pattern can be specified. The use of reconfigurable LCAs allowed the logic to be packed into just two boards; LP Com engineers estimate that the design would be at least twice as complex with traditional logic devices.

Sage Instruments (Freedom, CA) used a similar strategy in their Model 930A Communication Test Set, a general purpose channel access test system. Four LCAs are used to implement data interface, channel signalling, diagnostic, and microprocessor interface functions, respectively. The LCA that handles channel signalling has two possible configurations to support two different signaling formats, RBS (robbed-bit signalling) and DMI (digital multiplex interface). The data interface and channel signalling LCAs are both reconfigured to support bit error rate testing.

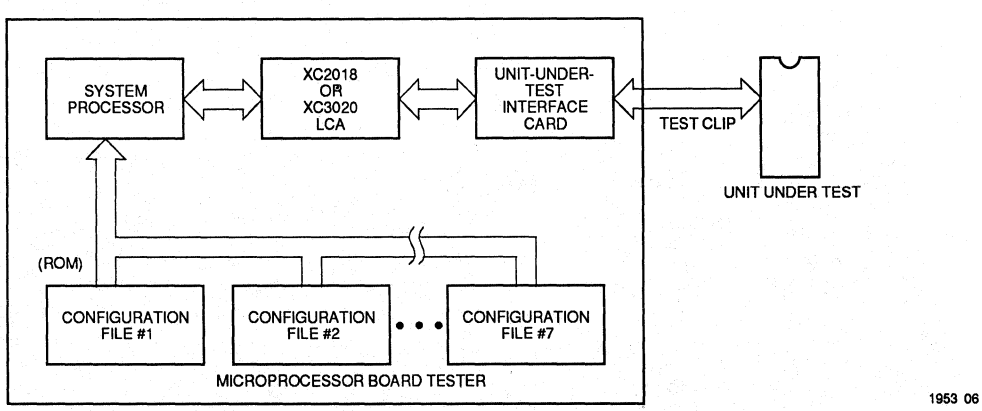


Figure 4. In Innovage Microsystem's microprocessor board tester, an LCA is configured for the appropriate microprocessor type and selected diagnostic test.

By reconfiguring a 3000-gate XC3030 LCA, an error-correction channel designed by Wiltron Co. (Morgan Hill, CA) can support either of two error checking and correction (ECC) formats, one for Digital Data System (DDS) and one for Adaptive Data Port (ADP) network configurations. The circuit is incorporated into several products, including Wiltron's Model 9966 Digital Services Test Unit for testing DDS-like services. Use of the LCA also provides insurance against evolving standards; new LCA configuration programs can be developed if standards for ECC formats and network configurations change.

DUAL-PURPOSE HARDWARE

In the examples sited above, programmable gate arrays are reconfigured to implement internal system diagnostics, adapt a circuit to the external environment, or completely change the functions of a system. Some logic designers have taken this concept one step further—programmable gate arrays are reconfigured as part of the normal operation of the system.

For example, at any given time, a tape recorder can either read or write, but it never does both simultaneously. Consequently, a programmable gate array within a digital tape recorder could be configured to perform one function when writing data to the tape, and then reprogrammed to perform a different function when reading from the tape. Honeywell's Test Instruments Division (Denver, CO) incorporated this scheme in their VLDS (Very Large Data Storage) recorder.[5] An XC2064 LCA is configured to perform error code generation in write mode, and then reconfigured to perform error code checking and correction in read mode. This type of application is especially cost-effective; about twice the logic would be required to implement the same functions with traditional logic devices.

A similar strategy can be used in the design of most logic analyzers, microprocessor in-circuit emulators, and similar test equipment. Each involves the monitoring and control of nodes within the system being tested. In the "acquisition mode", the target system is active and a record of the target's activity is stored in a memory buffer called trace memory. Trigger and breakpoint logic specifies when tracing begins and ends. A history of the system's operation can then be read from trace memory and displayed to the user, the "analysis mode". In an LCA-based system, programmable gate arrays could be used to implement the multiplexer, registers, and comparators of the trigger and breakpoint logic, interface to the system under test, and control the writes to the trace memory while in acquisition mode. Those same LCAs could be reconfig-

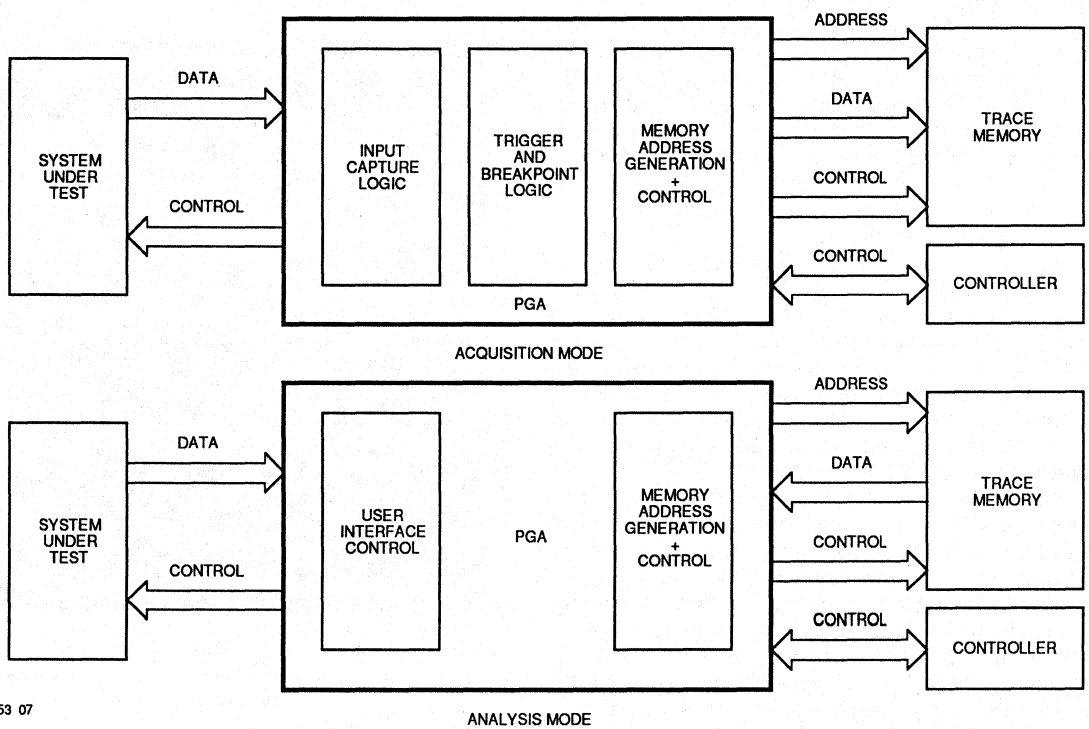


Figure 5. An LCA can be reconfigured to support both acquisition mode and analysis mode operations in a logic analyzer.

ured to control reading trace memory and displaying its contents when in the analysis mode (Figure 5). For example, Data I/O's MESA-1, an in-circuit verifier for LCA designs, uses LCAs exclusively to implement its logic (Figure 6).

Intel's Development Tools Operation (Hillsboro, OR) used a slightly different tactic when designing a series of in-circuit emulators for derivatives of the 80386 processor. The emulators contain six LCAs. Four of them comprise the bus event recognition circuitry used to define and detect triggers and breakpoints; three of these are largely filled with comparators, and the fourth holds the breakpoint state machine. When preparing for an emulation, these four LCAs can be reconfigured in the system, dependent on the type of breakpoints and triggers being specified. A DMA channel is used to download the LCA configuration programs. A fifth LCA holds the bus interface state machines; as a future product upgrade, Intel designers may generate another optional configuration program for that LCA to add additional tracing capabilities.

THE ULTIMATE RECONFIGURABLE SYSTEM

A system composed entirely of programmable gate arrays could be configured to implement any given logic functions. This concept has been incorporated into a new ASIC design tool that provides real-time in-circuit emulation of complex ASIC designs. The RPM Emulation System, from Quickturn Systems Inc. (Mountain View, CA), is a workstation-based design verification tool that combines automatic ASIC netlist conversion software with emulation hardware based on 9000-gate XC3090 LCAs (See Figure 6). The RPM Emulation System can be configured with up to four emulation modules with over thirty XC3090 LCAs each, allowing emulation of ASIC designs of up to 100,000 gates. Once the ASIC design is converted for emulation, existing complex VLSI devices may be internally connected to the emulation logic with Component Adapter boards, or the design may be plugged into a target system with an In-Circuit Interface consisting of cables, an active Pod, and ASIC Plug Adapters. The netlist conversion software reads the netlist (a variety of popular formats and libraries are supported), partitions the design for programming each XC3090 LCA, places and routes the design into the matrix of XC3090 LCAs, and checks the timing to determine the maximum speed of correct functional operation. The Control Panel user interface on the workstation guides the designer through the emulation set-up and provides the controls for the integral Logic Analyzer and Stimulus Generator, allowing quick access to any node in the design during debugging. Thus, using the RPM Emulation System, a designer can emulate and debug the logic operation of any large digital design before committing to a custom implementation.

RECONFIGURABLE LOGIC EASES DESIGN

While not every system requires reconfigurable logic to implement its digital functions, the design-related benefits of static-memory-based programmable logic apply to all designs. The ability to reconfigure programmable gate arrays resident in the target system significantly eases the debugging process, reducing overall development time and shortening the product's time-to-market. A download cable provided with the basic development system allows configuration programs to be downloaded directly from a PC to an LCA device resident in the target system; the actual download operation requires less than 100 milliseconds. Thus, the designer can immediately check the results of design changes in the target system. Often, design changes can be implemented and tested in just a few minutes time.

In essence, Xilinx programmable gate arrays provide a flexible means of "breadboarding" logic designs, as well as a cost-effective means of implementing the logic in the final product. Temporary modifications to the logic, such as routing an internal node to an otherwise unused I/O pad, can be quickly implemented for debugging purposes and then removed from the production design. Devices are reusable simply by downloading a new configuration. There is no lengthy wait for a custom device to be manufactured, and no waste of components as with one-time-

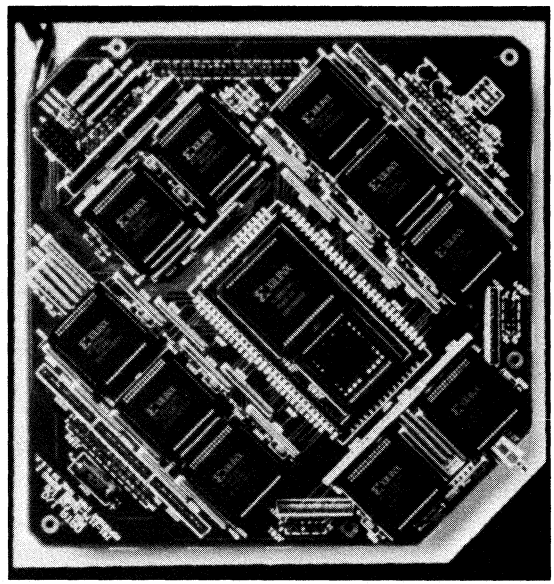


Figure 6. The internal logic of Data I/O's MESA-1 in-circuit debugger is implemented entirely in Xilinx programmable gate arrays.

programmable solutions; there is not even the inconvenience of long erase times using ultraviolet lights, as with EPROM-based logic. The designer receives nearly instantaneous feedback on the effects of design modifications. Furthermore, since the LCA's configuration can be verified in the target system, extensive simulation is not required; typically, simulation is used only for critical timing path analysis under worst-case conditions.

The ability to implement easily modifications to the logic enables and encourages experimentation during the design cycle, resulting in better designs. For example, the use of Xilinx LCAs allowed GTECH Corp. to evaluate different image sensors during the design of a bet-slip reader for the lottery industry.[4] Since there are no standard architectures or interfaces for image sensors, different interface logic was required for each sensor type. By incorporating the sensor interface logic in LCAs, a single hardware implementation could be reconfigured for each sensor type, allowing the sensitivity and resolution of each to be measured under identical conditions.

The flexibility of in-circuit reconfiguration greatly reduces design risks. The inevitable last-minute bug fixes and specification changes can be implemented by changing an LCA's configuration program rather than altering the hardware. M/A-Com Telecommunications (Germantown, MD), for example, was able to correct an error in the PCB layout without changing the board by reconfiguring an LCA used to implement the channel interface logic within a satellite earthstation.[7] This flexibility proved critical during Buffalo Products' design of the More Memory board mentioned above. During testing of the board using various manufacturers' PC clones, problems caused by incompatibilities in some PC models were corrected as they were found through reconfiguration of the LCA device.

FIELD UPGRADES SIMPLIFIED

Similarly, field upgrades can be easily implemented through changes to LCA configuration programs. Andromeda Systems (Canoga Park, CA) took full advantage of this capability in their Storage Module Device Controller, a disk controller for LSI-11 and Micro/VAX systems.[8] The configuration programs for three XC2064 devices are stored in EEPROM that can be altered using a service port that connects directly to terminals or modems. The interfaces to the disk, processor bus, service port, and cache memory are implemented in the LCAs (Figure 7). Modifications to the logic, such as adjusting the caching algorithm to match the requirements of a particular application, can be made without removing the disk controller from the system; new LCA configuration programs can be sent to the controller using a modem.

In many cases, compatible programmable gate arrays with a range of densities are available in identical packages. (For example, the 2000-gate XC3020, 3000-gate XC3030, and 4200-gate XC3042 are all available in 84-pin PLCC and PGA packages.) So if logic needs exceed the current LCA device, during either initial design or a product

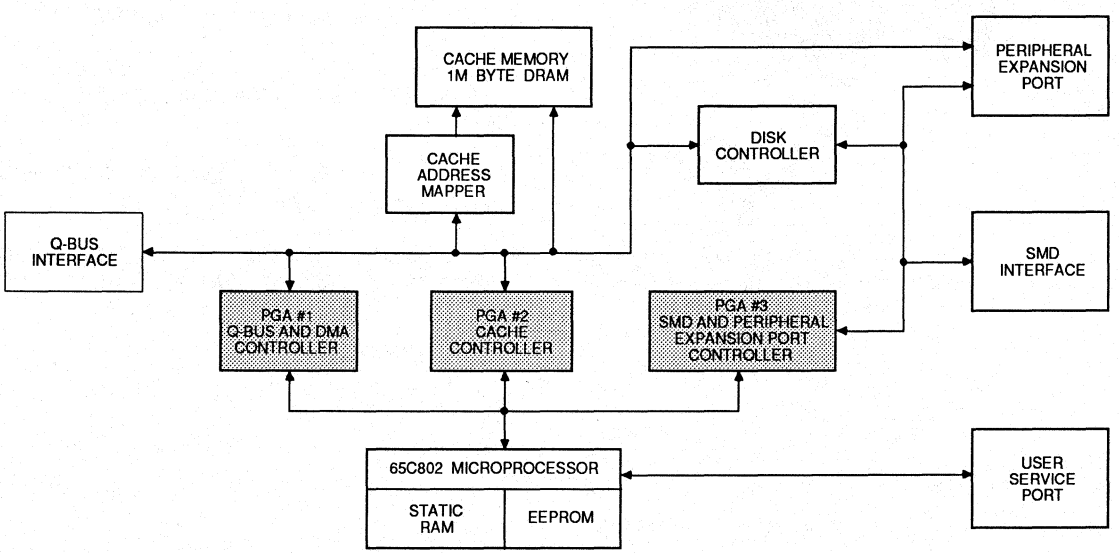


Figure 7. In Andromeda Systems' SMDC disk controller, LCA configurations can be downloaded to EEPROM through a modem port for easy field upgrades.

upgrade (due to the addition of new product features, for example), a higher-density device can be placed in the same PCB location, with no modifications required to the circuit board.

The reconfigurable nature of the programmable gate array also allows for the design of its own in-circuit debugging tools, such as Xilinx's XACTOR and Data I/O's MESA-1 (Figure 8).[9] Similar in many ways to microprocessor in-circuit emulators, these sophisticated verification tools provide for easy, fast debugging and testing. Since configuration programs can be downloaded into an LCA at will, LCA devices in the target system can be replaced or functionally duplicated by an LCA device in an in-circuit debugger; LCA activity can then be controlled and monitored by the user.

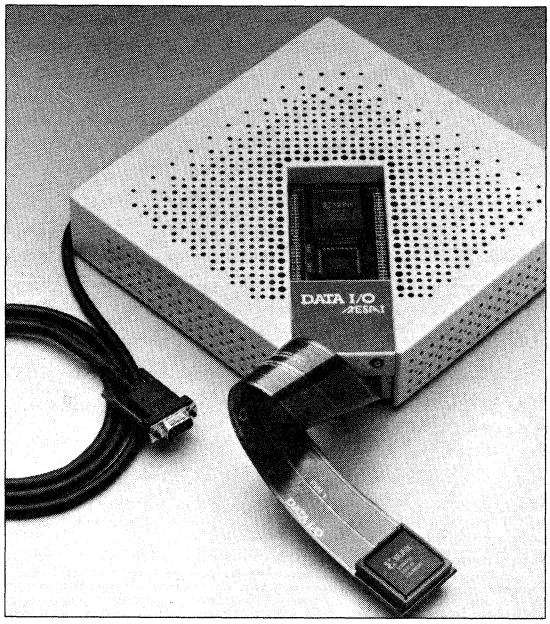


Figure 8. The reconfigurability of LCAs allows for the design of their own in-circuit verification tools, such as the MESA-1 from Data I/O.

SUMMARY

The advent of programmable logic that can be reconfigured while resident in a system has freed the designer from the "hard" nature of traditional logic ICs. With programmable gate arrays, adaptable systems that adjust to changing environments or varying tasks can be created, and hardware design is simplified. New system architectures that take advantage of reconfigurable logic will continue to emerge as programmable gate array densities and performance levels continue to increase.

REFERENCES

1. Loring Wirbel, "Tek Takes Color Printer to the Office," *Electronic Engineering Times*, Nov. 14, 1988.

2. David Smith, "User-Programmable Chips Take on a Broader Range of Applications," *VLSI Systems Design*, July, 1988.

3. Rusty Woodbury, "LCA Stars in Video," *ESD: The Electronic System Design Magazine*, Feb., 1987.

4. Cliff Dutton, "Programmable Logic Betters the Odds for Bet-Slip Readers," ESD: *The Electronic System Design Magazine*, Oct., 1987.

5. Tom Liehe, "Two, Two, Two Chips in One," *Electronic Engineering Times*, Nov. 17, 1986.

6. Loring Wirbel, "Quickturn Offers ASIC Emulator," *Electronic Engineering Times*, Nov. 14, 1988.

7. Dave Farrow, "Using Programmable Logic Cell Arrays in a Satellite Earthstation," *VLSI Systems Design*, April, 1987.

8. Jim Reynolds, "Building Tomorrow's Disk Controller Today," *Electronic Products*, Dec. 15, 1987.

9. John Novellino, "Development Tool Trouble-Shoots PGAs in the Target System," *Electronic Design*, Jan. 26, 1989.

ESD:
THE Electronic System Design Magazine

Faster Turnaround for a T1 Interface

by Carl Erite, Teltrend Inc., St. Charles, IL

Important design considerations for an interface system to a digital T1 network (which carries voice, data, video and fax traffic at rates up to 56 Kbytes/sec) include conserving board space, improving throughput and reducing power consumption. The user interface is achieved via a conventional four-wire loop providing independent transmit and receive capabilities. In designs that Teletrend Inc. initially considered for a single-user T1 interface, 5000 gates of conventional SSI/MSI glue logic were to be integrated using two custom gate arrays. However, a short development cycle and low market risks were also desired. This led to a search for an alternative to th time-consuming process of casting two gate arrays.

Upon completing the initial circuit design, a breadboard was built using CMOS SSI/MSI logic components. After the breadboard was working, integration path decisions were needed. Instead of hard-tooling two custom gate arrays, designers determined that three standard, programmable Xilinx Logic Cell Arrays (LCAs) met all of the design requirements—high integration, high density, high performance, low cost, low risk and quick time-to-market.

The Xilinx devices implement a digital phase-locked loop, as well as the T1 transmitter and receiver. A Hitachi microprocessor provides overall intelligence to handle T1 controls, network code manipulation and other tasks.

The dual digital phase-lock loop provides the key function of the system. Data on the user interface is encoded with the clock signals, a process that may occur at various send/receive data rates. Data extraction from the user interface must be phase-locked and, at the same time, data must by synchronized with the T1 network clock. A Xilinx LCA implements the phase-locked loop that synchronizes both the interface and the T1 network.

The second LCA transmits data onto the T1 network. Here, data transmits serially at 1.544 Mbits/sec in one of the 24 assigned time slots. A unique data word to be

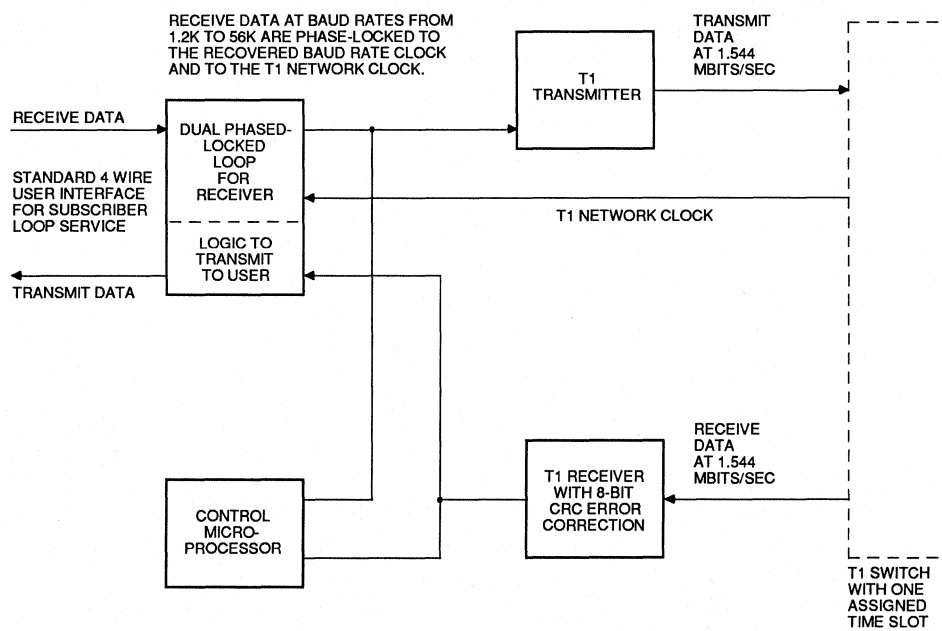


Figure 7. Teltrend's digital TI interface is built around three user-programmable Xilinx Logic Cell Arrays in lieu of two conventional gate arrays. One LCA implements a dual digital phase-lock loop around four-wire loop; other LCAs form both the transmitter and reciever logic circuits, including error correction.

transmitted is held in the LCA while logic synchronization determines the start of the first time slot or the beginning of the data frame. The assigned time slot is found by counting time slots from the start of a complete frame. After locating the assigned time slot, data is transmitted onto the T1 network.

A third LCA, complementary to the transmitter function, receives data. It also furnishes complete error correction for incoming data. Time-slot detection logic determines the start of data for the assigned channel. Serial data comes from the T1 network. After the LCA performs 8-bit error correction, the data passes to the processor and user interface.

The first iteration of the design was extracted directly form the CMOS breadboard schematics using the Xilinx XACT system running on an IBM PC/AT. The working design for the first device was completed in two weeks, with some time-critical elements moved off the chip. Designs for the second and third parts took about the same time, but additional interaction during the design process resulted in higher performance in critical timing paths and higher overall device utilization. In all three designs, LCA logic resource utilization exceeded 95%.

All three designs are flip-flop intensive, involving multiple counters, shifters, registers and other memory-oriented functions. The LCAs provide more flip-flops per device than any other programmable logic alternative. Only a few simple 8-bit registers were implemented externally with octal devices. Next-generation designs will use Xilinx's compatible higher density devices to achieve greater logic density in the same socket.

Overall, the ability to enter the original design using the Xilinx LCA XACT design system ensured that all the integrated logic functioned as desired before the part was placed in the system. With a conventional gate array, the design might still be waiting for silicon, since turnaround times for production quantity gate arrays typically range from 8 to 16 weeks (production quantities).

Reprinted with permission from *ESD: The Electronic System Design Magazine.*

Using Programmable Logic Cell Arrays In a Satellite Earthstation

Dave Farrow, M/A-Com Telecommunications, Germantown, MD

Conventional programmable logic devices (PLDs) include several interesting variations of latch-based AND-OR plane architectures in various technologies, all of which are useful for low-gate-density applications. Typically, a PLD can replace five to ten SSI/MSI parts.

A newer digital logic technology with an array architecture and flexible interconnection offers the programming flexibility of PLDs plus the gate density of low-end gate arrays. Architecturally, these devices have some similarities to gate arrays: they contain an internal matrix of logic blocks and a ring of configurable I/O interface blocks. Unlike conventional gate arrays, each part is a standard off-the-shelf unit that can be programmed by the user. The configuration program is automatically loaded into an on-chip static memory at power-up from either an on-board EPROM or an external source such as a floppy disk.

THE EARTHSTATION SYSTEM

M/A-Com recently employed one of these "programmable gate arrays" in the design of a satellite earthstation, intended to network commercial facsimile operations. The network handles traffic at 56 kb/s, multiplexed into 26 channels and convolutionally encoded, yielding an overall 3 Mb/s transmission rate. The earthstation product, called an OPT (for On-Premises Terminal) is a "small-aperture" satellite earthstation, permitting efficient employment in a large number of remote locations, as illustrated in Figure 1.

Two main components comprise the OPT: an indoor unit and an outdoor unit. The outdoor unit includes the antenna and associated radio-frequency equipment.

At the outset of the design process, the indoor unit was intended to be contained in a small chassis that could support three standard-size boards. The boards originally planned for the system included one board each for controlling data traffic, transmit functions, receive functions, and demodulation. However, the chassis provided space for only three boards.

Project goals included the use of an existing proprietary custom chip design from a previous application. M/A-Com also investigated whether the design could be fit on only two boards, by using a gate array. Board design itself was driven by three primary factors: resource availability, cost, and schedule. Since reducing the number of required boards would reduce design time and keep product costs lower, M/A-Com decided to go with the gate array.

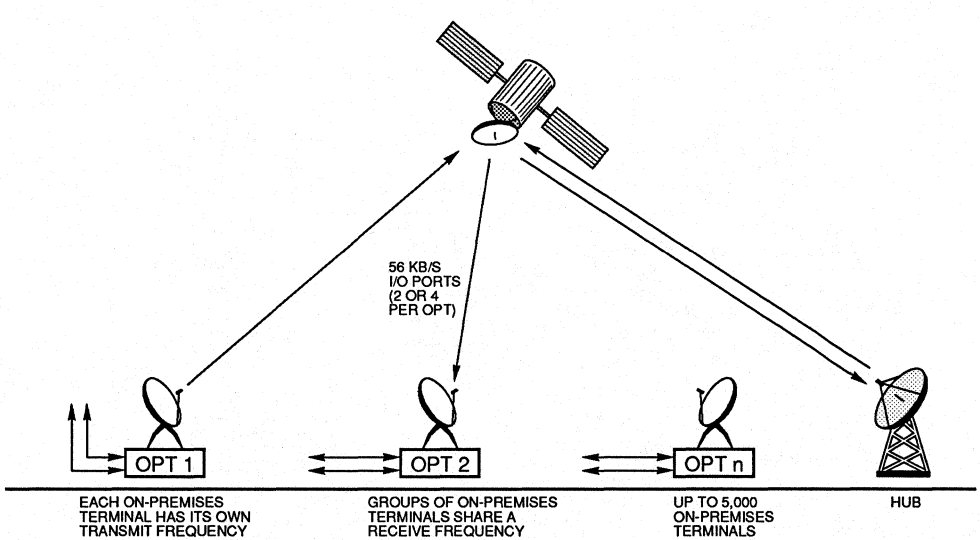


Figure 1. Satellite System

1148 07

QTY.	DESCRIPTION	ITEM
3	8-BIT SHIFT REGISTER	74HCT164
6	4-BIT COUNTERS	74HCT163
4	DUAL D FLIP-FLOP	74HCT74
2	QUAD 2:1 MULTIPLEXER	74HCT157
1	QUAD XOR	74HCT86
1	HEX INVERTER	74HCT04
1	QUAD NOR	74HCT02
2	QUAD OR	74HCT32
3	QUAD AND	74HCT08
1	OCTAL LATCH	74HCT374
1	OCTAL BUFFER	74HCT244
25 ICs		

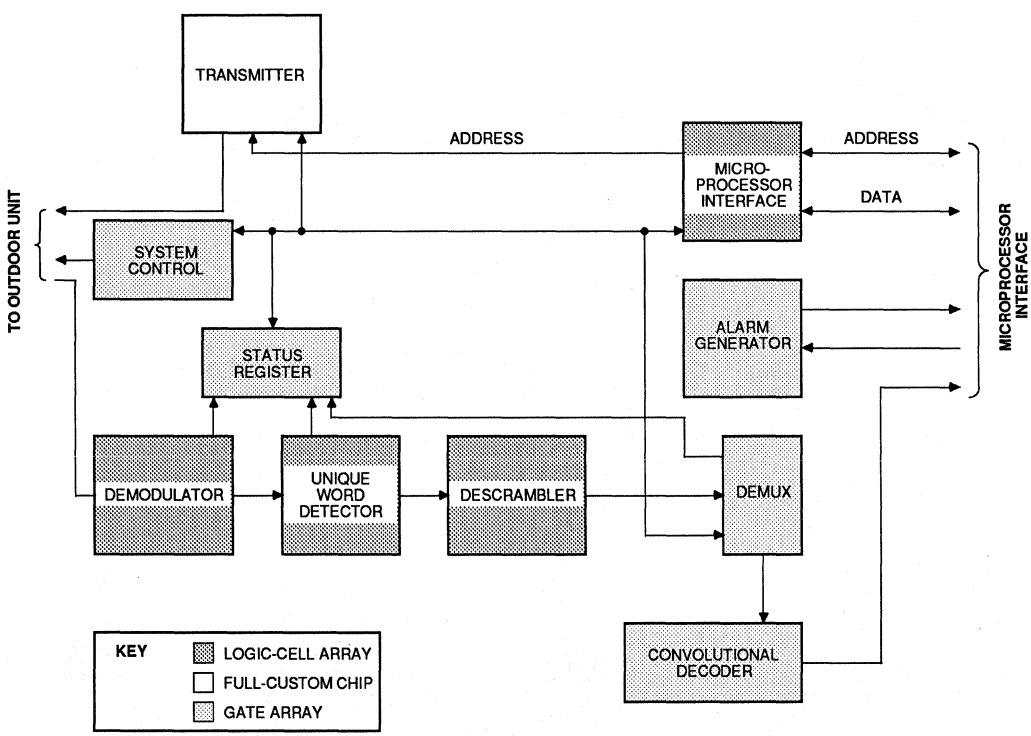
Table 1. Standard Off-the-Shelf Equivalents to the Logic Contained in the LCA.

1148 08

The completed design employs a full custom IC, a gate array, and programmable logic, and subsists on only two boards. On one board, an Intel processor acts as a traffic-port controller and handles base-band X.25 data. Due to the use of semicustom and programmable technology, the remaining three functions were all merged onto the other board, which we call a "satellite channel interface" (see Figure 2).

We used a gate array for the transmit function, which otherwise would have required about 70 chips. For the receive function, we originally planned to use an existing full-custom ASIC (previously designed by M/A-Com) for forward error correction, and an additional 25 SSI/MSI parts for the receive logic. However, due to chassis constraints, the high density of components would have necessitated a multi-layer board for the initial design. Furthermore, based on previous experience, the likelihood of changes in the design specification was too high to risk a custom or semicustom solution for the initial design. Therefore, we originally planned to produce the high-density boards in quantity and to reduce the cost of the system at a later date, by first transferring the receive logic into a gate array and then replacing the expensive high-density four-layer board with a two-layer board.

While the design criteria were being prescribed and board-level functionality was being determined, we also investigated the newer programmable gate-array technology. The programmable part, the Xilinx Logic Cell Array (LCA),



1148 09

Figure 2. Block Diagram of Satellite Channel Interface.

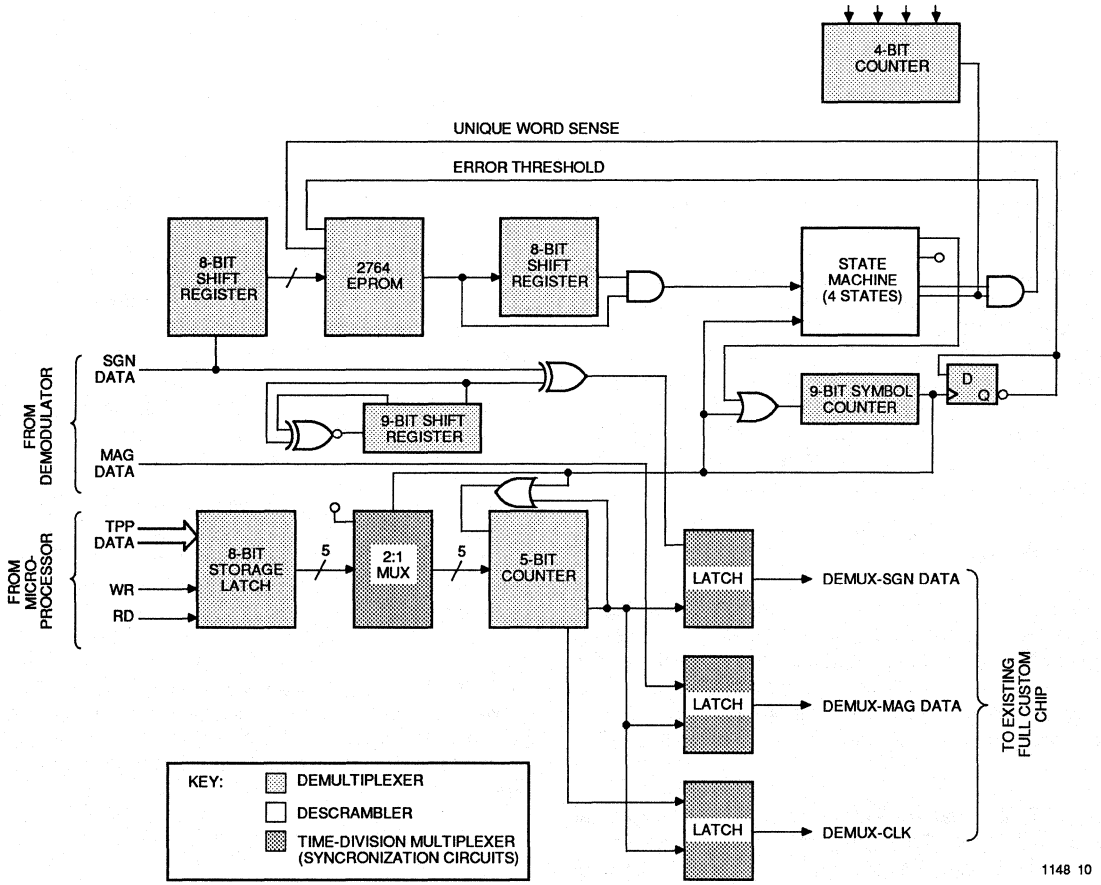


Figure 3. A Schematic of the Digital Systems Incorporated into the LCA.

is architecturally similar to a gate array and is supported by a PC/AT-based workstation.

We determined that the internal organization of the LCA fitted the design requirements of the receive function. Specifically, the LCA provides many more flip-flops than other programmable logic devices, so that one chip contained enough functionality for our needs. Further, the LCA provided the required density savings, and its reprogrammability obviated the risks associated with late engineering changes. When engineering management was presented with the design alternatives, we decided to prototype a reduced portion of the receive circuit and thus evaluate the reconfigurable chip.

To implement the design, M/A-Com acquired the Xilinx XACT PC-based LCA development system. The system includes a macro library, with some of the required logic already defined. After several days of experimenting with the design tools, it took us one day to enter and only two hours to debug the design. We uses Xilinx's XACTOR in-circuit emulator for debugging.

Our original schematic was based on conventional LS and HCT parts; it included JK flip-flops and large counters (implemented by cascading common 4-bit counters), rather than gate-level elements. Since that method of design was inefficient for the LCA, we redesigned the receive circuit at the gate level and then implemented it in software via the cell array editor.

Using an LCA reduced the amount of hardware overhead normally associated with LKS and HCT technology. It was not necessary to waste control inputs, to cascade counters, or to determine what to do with unused bits of multiplexers. In our design, 25 SSI/MSI gate-equivalents did not even use up all the resources available in one LCA. Table 1 indicates the parts that we actually employed in the present design. Putting these functions in the LCA resulted in an 88% utilization of the internal cells, and a 60% utilization of the I/O cells. Thus it still remains feasible to add further functionality to the system, with no PCB

changes. We plan to do so in the future. Figure 3 is a schematic of the circuit placed in the LCA. Since the design is not I/O limited, there was no necessity to multiplex any of the input or output lines; but additional logic could have been added, should I/O multiplexing been needed. Note also that the descrambling circuit can easily be reconfigured, or made more complex. Changing the descrambler can be achieved merely by reprogramming the LCA.

One criticism leveled against the LCA is that it requires 12K bits of storage space to program the part during power-up. However, in our design, a 27C64 EPROM (used for a look-up table) was already on the board. A portion of this EPROM was available to store the LCA configuration program at no additional cost. Since the 12K bits of storage space are used to program all the RAM cell locations in the LCA, adding further functionality to the LCA would not require more storage space.

ARCHITECTURE

From the OPT, transmission is executed in the SCPC (single channel per carrier) mode. All scrambling, encoding, and error-code generation are performed by M/A-Com's proprietary transmit gate array. The gate array contains registers, allowing it to be programmed to transmit in different schemes and protocols, including SCPC mode.

The OPT receives a TDM (time division multiplexed) bitstream composed of 56 kb/s data channels in a modulated 3-MHz carrier. The bitstream contains a UW (unique word), and data and parity bits for each channel in each frame. The received carrier is demodulated by analog circuitry on the SCI, which passes the digital bitstream to the LCA.

To isolate the UW and lock onto the data, the LCA contains several counters and a state machine, configured in TDM synchronizer. The state machine controls he synchroniztion algorithm, which manipulates the frames.

The TDM synchronizer moves between four states (see Figure 4). The first state entails acquiring "sync" by recognizing the unique word in the unsynchronized data stream. Once the unique word is acquired without errors, the second state occurs. The circuit verifies "sync" by detecting the unique word again one frame later in the bitstream. Upon second detection, the circuit is considered in sync, and the synchronizer shifts to the third state—the sync state—where data are allowed to proceed as long as the system detects at least one unique word in every 11 frames.

The fourth state is entered every time a unique word is missed; the system stays in the fourth state until the unique word is found or is missed 11 consecutive times. If the unique word is found, the system returns to state three; if it is not found after 11 attempts, then the first state (the search mode) is initiated again. This method of operation ensures that the demultiplexer will remain locked even in the presence of random bit errors in the data stream.

After the unique word is detected, the receiver locks onto the data. The LCA chip then descrambles the data stream. The data is originally scrambled by the transmitter to place a fairly equal number of ones and zeros into the transmitted carrier. If this is not done, the transmitted carrier may not contain an even distribution of spectral components, which makes it difficult for a demodulator to acquire the carrier. The descrambling process is merely the reverse of the 9-bit scrambling procedure.

A single channel is isolated from the others by demultiplexing the descrambled data stream. The demultiplexing function is performed through a pair of counters that count the bits between unique words and tell the demultiplexer when data is available.

Once the incoming data stream has been descrambled and demultiplexed, it moves on to the M/A-Com proprietary convolutional decoder, a custom chip where error detection and correction is done on a per-channel basis. Decoded data is passed on to a microprocessor for data extraction.

TESTING THE LCA

To test the TDM synchronizer, the LCA was loaded via the Xilinx in-circuit emulator and set into the test bed. We tested with a satellite simulator and found one design error. Both isolation and remedy of the fault were simple to perform, due to the reconfigurability of the part. Fault location was eased by choosing internal test nodes and connecting them to I/O pads. This technique made it possible to find the fault very quickly.

By using a satellite simulator we were able to insert errors into the datastream. We measured the time to lose sync and the time to acquire sync, and determined that the ripple counter was a little too slow for the required function. Since we were using an in-circuit emulator, it was very easy to reprogram the device. After the design was debugged, we left the simulator on-line for a week to ensure a thorough test of the Xilinx part under operational conditions. Our concern was how well the LCA would retain its configuration, since this information is stored by RAM cells. However, in our environment, it performed flawlessly.

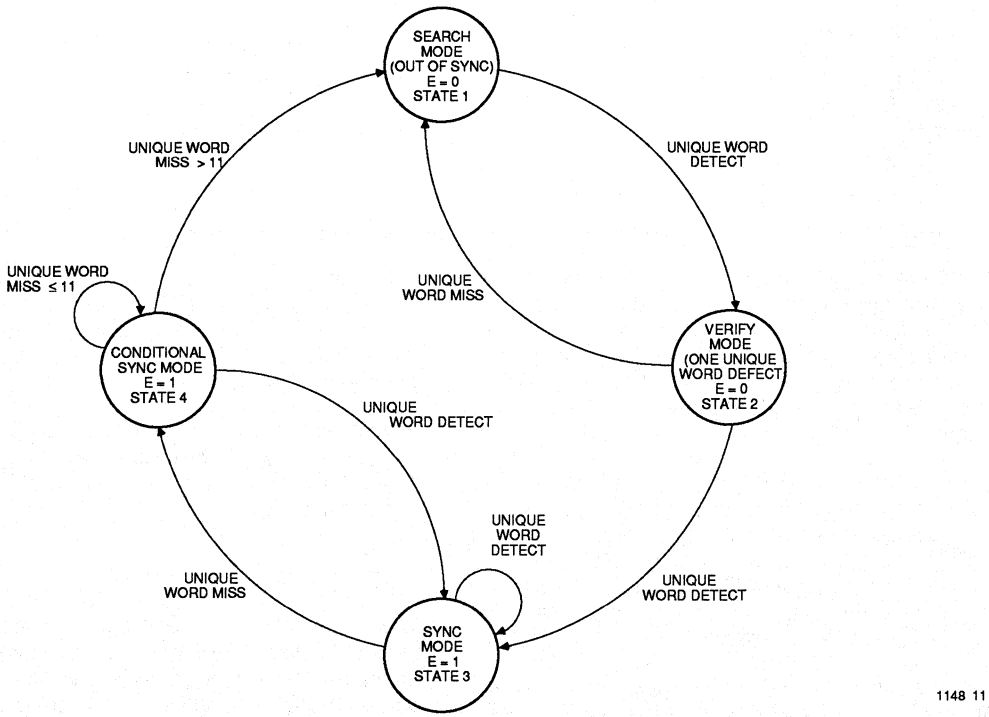


Figure 4. State Machine for the Time-division Multiplexer.

OPEN-END DEVELOPMENT

Late into the design cycle we began to add additional planned functions to the LCA. Because we knew we could add these extra features, we finished the PCB layout and ordered PC boards without waiting for the final design. Then the process of adding putting functions into the LCA was begun.

Normally this time would have been used to design a test fixture. Instead, another LCA design was created to support a test implementation. Before the PCB was delivered, the test fixture simulating the system was built, primarily around the second Xilinx part. In the process of building the fixture, we discovered an error in the PCB layout, even before it was delivered. It was possible to fix the error by reconfiguring the LCA.

When the board was delivered, a new version of our logic design had been implemented in the Xilinx LCA, including the demultiplexing and descrambling functions.

CONCLUSIONS

The flexibility of the Xilinx LCA lowers design costs, reduces project schedule risks, and reduces inventory risks. Using the LCA does not require much design sophistication, but rather a good general knowledge of basic digital circuitry. For example, designers must be able to recognize the worst-case timing scenarios of their networks. Delay and system-speed considerations can now be checked with the Xilinx simulator, but at the time of our design, the simulator was still in beta test; we calculated the circuit behavior with preliminary timing software. Since then the simulator has been revised and its present version would have spotted our timing error.

Rather than packing complete design into the front end of an ASIC development, as is required for conventional gate arrays, the LCA offers the flexibility to indicate roles for the part. Designers can specify the I/O pins for the LCA then send the PC board to fabrication. While the board is in fabrication, designers can build into the LCA the gate-level logic they want and continue to make changes up until, and even after, the PCB is delivered.

After final product delivery, the on-board logic can still be reconfigured to match specific customer needs—without having to cast custom silicon for a few dozen units or changing the PC artwork. Great NRE savings are passed back to the customer. In summary, the LCA has proved to be an extremely efficient, useful, and cost-effective extension to our semicustom design capabilities.

Reprinted with permission from *VLSI System Design.*

ESD:
THE Electronic System Design Magazine

Programmable Logic Betters the Odds for Bet-Slip Readers

by Cliff Dutton, GTECH Corp., Providence, RI

In countries throughout the world, the vitality of the on-line lottery industry is enhanced by seasonal and special promotional games. But new games require new bet-slips, and bet-slip readers must be able to accommodate frequent changes in format. To accomplish this, programmable gate arrays are replacing older, less flexible architectures.

In the development of GTECH's Solid State Reader, many existing technologies were evaluated, but they imposed unacceptable limitations on bet-slip processing, restricting bet-slip formats to rows and columns. Moreover, the process of reading the coupons was dependent on complex moving parts, and the reading elements were exposed to the external environment.

To maximize flexibility and minimize board space, Xilinx's (San Jose, CA) Logic Cell Array (LCA) was chosen for the Solid State Reader. The LCA, touted by the company as a "programmable gate array," represents a novel programmable logic device that is notable for its reprogrammable architecture. This architecture provides flexibility throughout the product's life span, which allows on-line bet-slips to be produced with marks in any arrangement. Each bet-slip reader at every terminal can be configured on-line to read any bet-slip from an active suite of eight different bet-slips.

Figure 1 shows three lottery bet-slips. Some of the pertinent features of the European Lotto game slip (a) include strobe marks along the top edge, the OCRB-3 characters (bottom center), and the name and address field (bottom right). In the sample bet-slip from a lottery in the U.S. (b), there are no OCR characters or name and address information. However, there is an area from which handwritten information must be extracted. Apart from the different features, the aspect ratios of bet-slips are not standard. Modern bet-slip processing systems must be able to read all of the different formats in many aspect ratios. A format that forgoes the usual row and column arrangement (c) is also depicted.

As there are no standard architectures or interfaces for image sensors, GTECH evaluated many image sensor approaches. However, direct comparison of sensor performance could not be made in the application environment. For example, comparisons of sensor sensitivity at the pixel level were impossible due to the differences in sensor-interface electronics. If degraded sensitivities were evident, they could derive from either the sensor or the sensor interface. Similar difficulties hindered direct comparison of achieved resolution. To accurately evaluate these parameters, each sensor had to be designed into prototype readers. This involved driver and frame acquisition clock signal generation.

Because lotteries have no standard bet-slip size, as many "standards" as possible need to be accommodated. Thus, it was necessary to maintain flexibility in the format of the target image.

PROTOTYPING A SYSTEM

The implementation of a prototype system had one goal: to prove the feasibility of recognizing handmade marks in an imaging system. Because the volume of readers is potentially high, component costs were a serious issue.

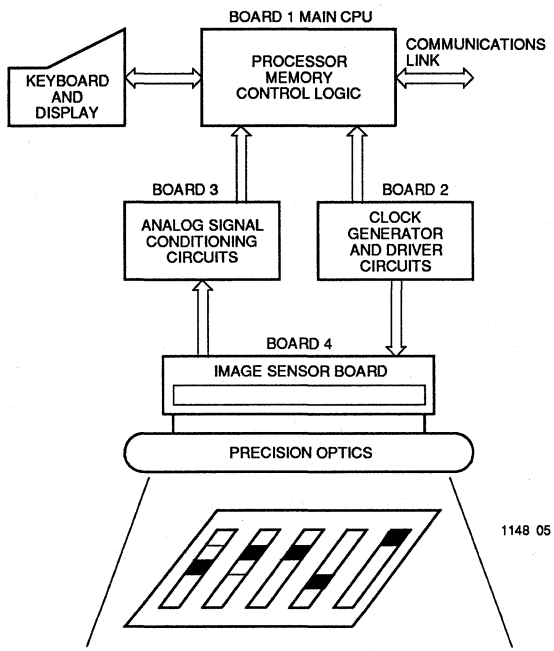
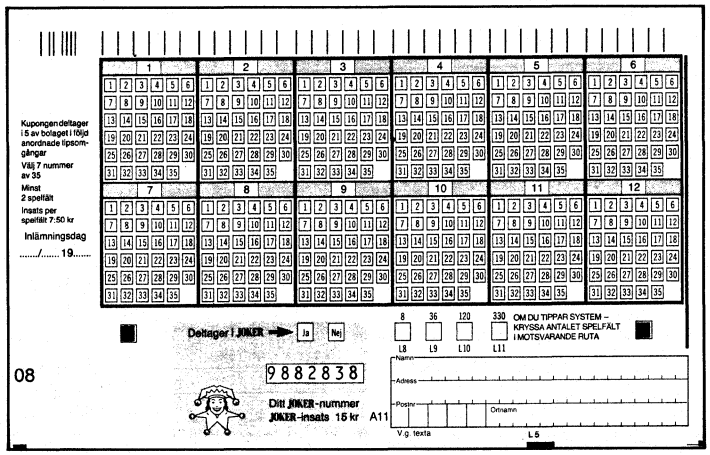


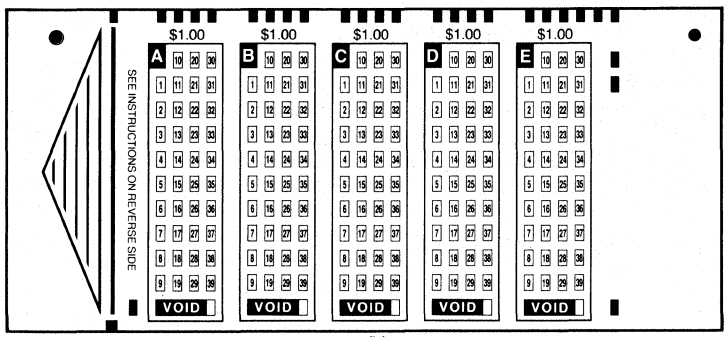
Figure 2. The goal of developing a prototype bet-slip processor (shown above) was to prove that handmade marks could be recognized in an imaging system. Four boards were initially developed for this modular design: CPU/memory, clock-driver, analog amplifier, and sensor mounting.

First, a working model was developed. To balance development costs, a set of printed circuit boards based on TTL logic devices was manufactured. Partitioned functionally, the board set supported modular design changes. Four pc boards were initially developed: a CPU/memory board, a clock-driver board, an analog amplifier board, and a sensor mounting board (Figure 2).

In the initial design, flexibility did not exist. Even though modularity protected the design from becoming obsolete, significant design alterations were required to accommodate different sensors. Because sensor clock signals are multiphase, new clock generators would be needed for new sensors. Also, bugs were difficult to find, and circuit board modifications were required to eradicate such bugs.



(a)



(b)

(c)

Figure 1. Betting slips for lotteries come in varied shapes and sizes. (a) Shown here are lotto slips from Europe and (b and c) the United States. Such variety in slip design must be accommodated in the developement of bet-slip readers.

Finally, the target image aspect ratio was fixed because the clock generation circuits were implemented in hardware.

Aspect ratios of target images are important because only necessary information on the image needs to be processed. If the target image is 2:1 and the imaging format is 1:1, for example, then half the image is useless. A better solution would mirror the aspect ratio of the target image in the image format.

To overcome the limitations of hardwired logic and reduce board space, several technologies were evaluated. These included programmable logic arrays (PLAs), field programmable logic devices (FPLDs), semicustom and fullcustom devices, and Xilinx's Logic Cell Array (LCA).

Size constraints indicated the necessity for semicustom of full-custom integration, but traditional LSI technologies violated the flexibility constraint. Although full-custom was attractive, design costs were prohibitive and did not permit iterative development. Standard PLDs did not allow for the variety of register-like functions that the clock generation logic required.

Programmable logic arrays were attractive for some logic functions and would have been the least costly. However, PLAs did not allow the multiple register implementation necessary for clock generation. Thus, the counting algorithms would have remained external to any integration of the combinatorial logic. Also, although the PLA architecture would have saved board space, it would not have preserved the functional modularity achieved in the first implementation. Thus, it would have been impossible to evolve a PLA-based system in response to changes in sensor technology. Finally, any required changes would have to be performed by field replacement. With over 35,000 lottery terminals installed on five continents, this was unacceptable.

Field programmable logic devices, an update of the PLA-style architecture allowing limited reprogrammability, appeared to provide some of the flexibility needed. If the problem were merely a straight combinatorial one, FPLDs could have been used. However, the difficulty in supporting both registers and counting algorithms ruled out their use.

Semicustom and full-custom technologies would have solved all the functional problems, but they lack flexibility. Because the development of the reader was ongoing, the commitment to custom implementations was out of the question. In addition, nonrecurring engineering (NRE) costs were prohibitive and the devices could not be adapted to changing sensor technologies or changing bet-slip reading requirements.

Xilinx's LCAs permit a two-board set to be designed without sacrificing functional modularity. In addition, counting algorithms can be implemented in the LCAs. Finally, LCAs allow for a multiple-iteration development cycle.

PUTTING A BUG TO REST

Initially, the TTL-based system was implemented in four pc boards. However, it contained a bug. For every horizontal line, an extra pixel pulse was being supplied. Although this was confusing to the eye, it was compensated for in firmware. Because the redesign of the clock driver board was a significant task, the bug was allowed to live through many iterations of the development cycle. When the design of the clock generation circuit was translated into the LCA, it was a trivial matter to delete a single horizontal clock pulse and put the bug to rest in an afternoon.

Using the LCA also provided the ability to vary the clock generation circuitry to evaluate different sensors. Because there is no standard architecture for solid-state digital imaging devices, clock requirements vary for different sensors. In a standard imaging application, it might be possible to source the appropriate support chips for each sensor from the manufacturer. But because development of the reader involved nonstandard video speeds in a noninterlaced mode, it was impossible to use standard support chips. If it had been necessary to develop a clock driver pc board for every sensor evaluated, it would have been impossible to evaluate more than one sensor in the development time. Because LCAs were used, varying multiphase clocks could be generated for different sensors under evaluation. Thus, the turnaround time for a design change in the clock generation circuits was reduced from one to six weeks to one day.

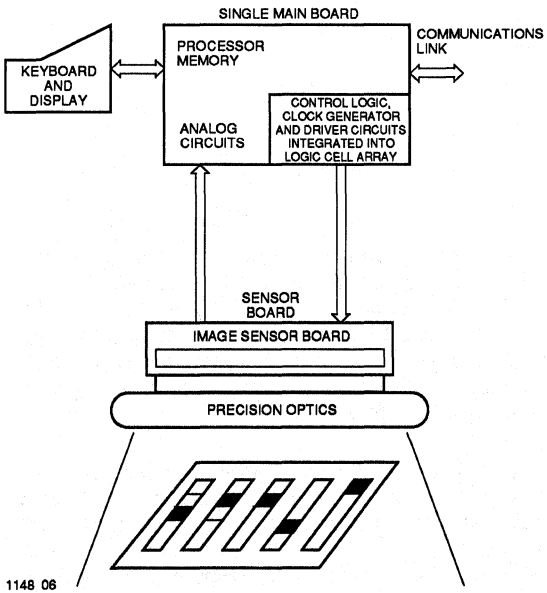


Figure 3. GTECH's Solid State Reader uses Logic Cell Arrays (LCAs) to maximize flexibility and minimize board space. Frame-grabber, memory addressing, and sensor clock driver functions are consolidated in the LCA. By reducing the number of chips, the required number of boards shrinks from four to two.

The Solid State Reader does not rely on standard video output. Thus, the 4:3 standard aspect ratio for broadcast television is not a requirement. All image processing is internal to the system. Real-time display of the image is never required. Therefore, only those areas of the sensor that may contain relevant information need to be required. Information-bearing areas of a bet-slip vary with the bet-slip design, so it is helpful to redefine the area of the sensor that is acquired for processing.

Because the clock driver circuitry, the memory addressing logic, and the frame-grabber logic are all implemented in the reconfigurable LCA, it is possible to acquire only certain areas of the image. As each sensor has different horizontal and vertical clock pulses, this flexibility cannot be achieved in hardwired logic.

Figure 3 illustrates the current architecture of the Solid State Reader. Because of the functions consolidated in the LCA, the system was reduced from four pc boards to two. This could have been done using other technologies, but they would not have preserved the functional modularity of the system. The LCA-based design provides both size reduction and functional modularity.

Reprinted with permission from *ESD: The Electronic System Design Magazine.*

Electronic Products
The engineer's magazine of product technology

Building Tomorrow's Disk Controller Today

Jim Reynolds, President, Dave Randall, Chief Engineer, Andromeda Systems, Canoga Park, CA

Reprogrammable logic with a flexible architecture enables a controller to keep up with today's high-capacity, high-speed disk drives

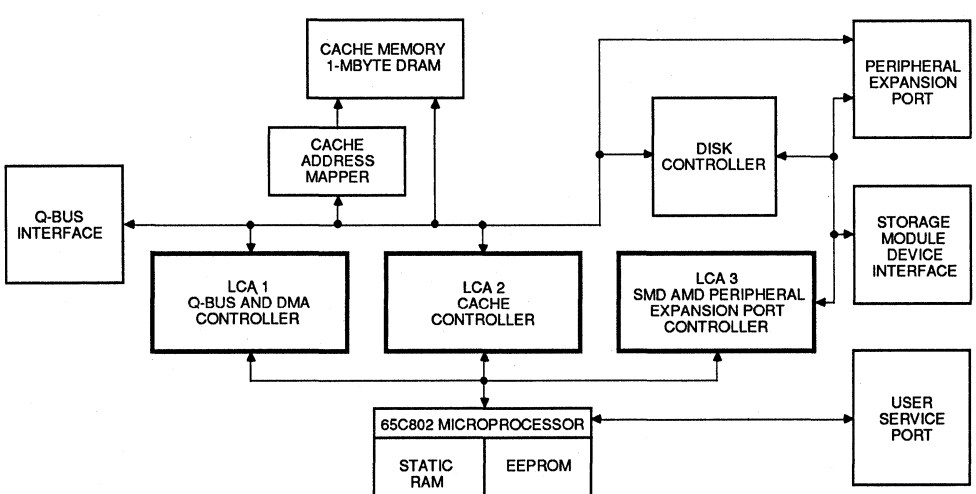
Computer manufacturers historically have relied on advances in CPU and semiconductor memory technology for increasing system throughput. At the same time, they accepted as inevitable the hardware-bound I/O bottleneck. This position is becoming untenable with recent advances in magnetic disk technologies, which have led to a proliferation of high-capacity, high-speed drives.

Full performance from these drives needs sophisticated controllers like Andromeda Systems' new Storage Module Device Controller (SMDC). With a 1-Mbyte data cache and dynamic read-ahead algorithms, the SMDC dramatically reduces average disk access time and significantly improves overall system performance (see box, "The Storage Module Device Controller"). The design and performance benefitted greatly from using Xilinx's Logic Cell Arrays (LCAs).

Very early in the design, it was clear that its high-performance caching scheme needed more SSI/MSI logic than could be surface mounted onto a 35-in.^{2} dual-width board. The only answer appeared to be VLSI custom or semicustom devices like gate arrays. But gate array definition requires absolute design accuracy, and so a prototype must be constructed long before custom-tooled ICs can be specified and manufactured. Paradoxically, the prototype itself required highly integrated logic.

To break that frustrating circle, it was necessary to convert directly from schematic capture to a silicon breadboard of multiple *electrically programmable logic devices* (EPLDs). Because many logic functions would be added to the prototype after the initial test, EPROM-based PALs were considered, like the EP1200 from Altera, which licenses the technology from Monolithic Memories.

The EP1200 could provide the minimum functionality on the silicon breadboard, but not the level of device integration for the production circuit board. To implement the various state machines and other logic of the design, each target gate array would need three EP1200s. The resulting schematic capture and simulation would then be used to fabricate the gate arrays for the final product.



1148 01

Figure 1. On Andromeda Systems' new Storage Module Device Controller, Xilinx Logic Cell Arrays handle the Q-bus interface and direct memory access (DMA) Control, RAM/data-cache control, and SMD and peripheral expansion port control.

Fortunately, this circuitous design path was bypassed by using Xilinx's LCA (see box, "Xilinx's programmable gate array"). There are two basic differences between LCAs and other EPLDs. First, the LCA has the flexible architecture of a gate array. Second, LCAs employ static memory to hold the logic configuration data.

The LCAs brought several significant advantages to the controller design. Since the Xilinx 2064 LCA has 64 configurable logic blocks and the EP1200 only 20, a single LCA could replace the three target gate arrays, elimination the fabrication delays and costs of custom tooling.

Furthermore, the position of the LCAs on the board could be determined before their internal logic configuration was designed. Other than dedication input and output pins, only a general idea of the function of each LCA was needed. The board layout and the internal LCA logic design could proceed in parallel, greatly reducing development time. Most design changes could be implemented merely by reprogramming the LCAs. Thus, use of the LCAs allowed the design to go directly from schematic capture to a production board, skipping the wire-wrapped prototype.

The first LCA on the SMDC is the Q-bus interface and direct memory access (DMA) controller (see Fig. 1). All but 5 of the 64 internal logic blocks were used. The LCA holds the DMA addressing logic, the bus registers, and the interrupt logic.

RAM/data-cache control is the job of the second LCA. It controls the cache and has the interface between the disk controller IC and the DMA logic. It signals cache-write enables, multiplexes memory addresses, and enable DMA reads and writes.

The third LCA controls the SMD port and peripheral expansion port. The expansion port is just a group of programmable I/O connections. Since the LCA is programmable, the control logic for the expansion port can be reconfigured for any desired I/O interface. Thus, this port provides for future expansions (like adding a tape drive, optical disk, or extra cache memory) at a fraction of the cost of a separate controller. Unused logic in this LCA will permit on-board functions to be added in future microcode revisions to the controller.

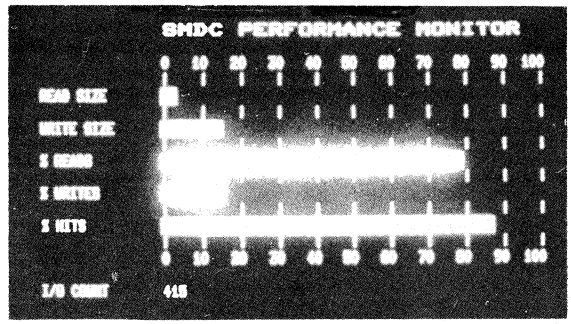


Figure 2. The user service port can create color bar graphs that dynamically show various attributes of the data cache, such as read times, forward block reads, and I/O completion rates.

Aside from the LSI circuitry, the only other logic on the SMDC board are TTL bus transceivers, SMD interface drivers, and a few PALs.

The RAM of the data cache is in ZIPs. Most of the interface logic was surface mounted to the board. Despite the board's small size, these VLSI devices permit several advanced features.

The SMDC's user service port connects directly to terminals or modems. No special test programs for specific system environments are needed to communicate with the controller. Users can define drives, assign logical units, format drives, and do other more esoteric functions.

This port can monitor the operation of the controller while the drive is in operation. The user can display color bar graphs that dynamically show various attributes of the data cache, such as read times, forward block reads, and I/O completion rates. Caching parameters can be adjusted, letting the user tune the system for optimum performance.

Firmware can alter the configuration data for the LCAs, modifying the circuit schematic and not the board. Since the firmware is in EEPROMs, the service port can accept microcode upgrades in the field via modem. PROM set replacement and on-shelf obsolescence are avoided.

THE STORAGE MODULE DEVICE CONTROLLER

Designed for LSI-11 and Micro/VAX II systems, Andromeda Systems' Storage Module Device Controller (SMDC) for Winchester drives supports two SMD or SMDE drives at data rates up to 25 Mbits/s. Another Andromeda controller, the ESDC, works with the Enhanced Small Device Interface, the ESDI, for Winchesters or floppy-disk drives. Both controllers use the standard DU device driver and work with such operation systems as RT-11, TSX+, RSX, RSX-11M, MicroRSX, RSTS, MicroRSTS, Ultrix, DSM, Unix, and MicroVMS.

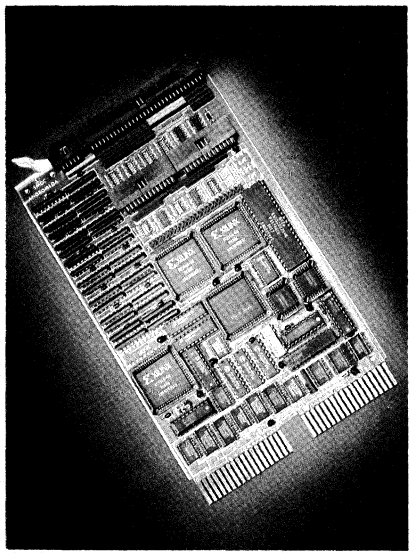
The SMDC achieves more performance and flexibility than did previous generations of disk controllers. It includes data caching, high data-transfer rates, a peripheral expansion port, field-loadable microcode, and a user service port. State-of-the-art VLSI components and packaging techniques fit the entire controller within the 35 sq in. of a dual-width Q-bus board (see figure).

Using Digital Equipment's Mass Storage Control Protocol (MSCP), the SMDC can partition two drives into as many as 16 logical units with up to 32 Gbytes each. Onboard intelligence comes from a 65C802 microprocessor, and all the processor's code resides in just two EEPROMs. The majority of the remaining logic is implemented with Xilinx programmable Logic Cell Arrays (LCAs). Data integrity is ensured by 48-bit error detection and correction logic. An expansion port can be connected to accessory modules, allowing control of devices like tape drives, optical disks, or extra cache memory.

The performance of the SMDC is greatly enhanced with a 1-Mbyte data cache and unique caching algorithms. Andromeda divides the cache into 1,024 granules. The information kept for each 1-Kbyte granule depends on select criteria, which include:

The time data is first accessed
The number of times data is read
The time of the most recent read
The size of the read.

This information is then entered into an equation that approximates how probable it is that the granule will be requested again soon. Those granules with low probabilities are designated to be overwritten by the next disk-read operation. During cache accesses, a memory mapper translates logical memory addresses into the physical addresses of the appropriate granule in much the same way that the Micro-Vax II memory management unit would.



PREDICTIVE CACHING

In a novel departure from most caching schemes, the SMDC caching mechanism not only looks at the past, but tries to gaze into the future as well. As the system requests the data that has been pre-fetched into the cache, the controller retrieves not only the requested data, but also preemptively reads extra sequential blocks when specific probability conditions are met. As a result, the on-board cache's typical hit rate is over 80%. In other words, the data being sought by the application will be ready and waiting in the cache over 80% of the time.

Approximately 90% of the disk access time is due more to average seek times and rotational latency than to the actual data transfer rate. However, when a cache hit occurs, the access time depends only on the speed of the DMA channel responsible for sending the data to the Q-bus.

That DMA channel operates as fast as Q-bus specifications allow—to be specific, at a rate of up to 4 Mbytes/s. Consequently, with the SMDC cache, seek time and rotational latency are reduced to zero over 80% of the time. This reduces the average time for a four-block read from 27 ms to less than 6 ms.

In the majority of computer systems, mass-storage access time is undoubtedly the largest component of throughput. In this situation, use of the SMDC enormously improves total system performance.

$$$$$

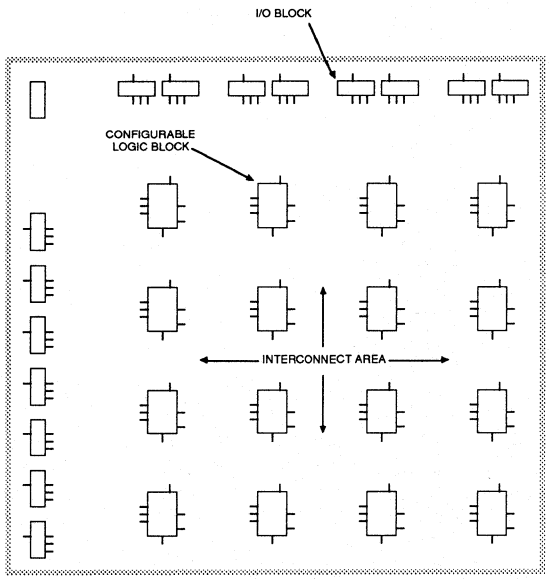
Andromeda Systems' Storage Module Device Controller is available now for $2,195. (The company's ESDI controller is available for $1,995.) For more information, call Don Talmadge at 818-709-7600, or circle 336 for the SMDC and 337 for the ESDC.

XILINX'S PROGRAMMABLE GATE ARRAY

The Xilinx programmable gate array, known as a Logic Cell Array (LCA), is a high-density CMOS IC that combines user programmability with the flexibility of a gate array architecture and the economy and testability of standard products. Elements of the array include three categories of configurable elements: I/O blocks, configurable logic blocks, and programmable interconnections (see figure).

I/O blocks provide an interface between the external package pin and the internal logic. Each block includes a programmable input path and output buffer. The array of configurable logic blocks contains the functional elements from which the user's logic is constructed. Each array includes a combinatorial section, storage elements, and internal routing and control logic. Programmable interconnection resources connect the inputs and outputs of the I/O blocks and configurable logic blocks into the desire networks.

An LCA is configured by programming static memory cells that determine the logic functions and interconnections. On-chip logic provides for automatic loading of the configuration program at power-up or upon command. A personal computer-based development software package generates the configuration program. Other tools include a simulator, in-circuit, and schematic capture package.



1148 02

Reprinted with permission of *Electronic Products.*

Index

Index

Sales Offices

HEADQUARTERS

XILINX, INC.
2100 Logic Drive
San Jose, CA 95124
(408) 559-7778
TWX: 510-600-8750
FAX: 408-559-7114

XILINX SALES OFFICES

NORTH AMERICA

XILINX, INC.
3235 Kifer Road
Suite 320
Santa Clara, CA 95051
(408) 245-1361
FAX: 408-245-0517

XILINX, INC.
15615 Alton Parkway
Suite 310
Irvine, CA 92718
(714) 727-0780
FAX: 714-727-3128

XILINX, INC.
61 Spit Brook Rd.
Suite 403
Nashua, NH 03060
(603) 891-1096
FAX: 603-891-0890

XILINX, INC.
65 Valley Stream Parkway
Suite 140
Malvern, PA 19355
(215) 296-8302
FAX: 215-296-8378

XILINX, INC.
919 North Plum Grove Road
Suite A
Schaumburg, IL 60173
(708) 605-1972
TLX: 510-601-5973
FAX: 312-605-1985

EUROPE

XILINX, Ltd.
Suite 1B, Cobb House
Oyster Lane
Byfleet
Surrey KT14 7DU
United Kingdom
Tel: (44) 932-349401
FAX: (44) 932-349499

XILINX, G.m.b.H.
Leonhardsweg, 2
8025 Unterhaching
Munchen, West Germany
Tel: (49) 89-6110851
Fax: (49) 89-6112246

JAPAN

XILINX K. K.
Kybashi No. 8
Nagaoka Bldg. 8F
20-9 Hatchobori Nichome
Chuo-ku, Tokyo 104, Japan
Tel: (03) 3297-9191
FAX: (03) 3297-9189

U.S. SALES REPRESENTATIVES

ALABAMA

Novus Group - Huntsville
2905 Westcorp Boulevard
Suite 120
Huntsville, AL 35805
(205) 534-0044
FAX: 534-0186

ARIZONA

Quatra Associates
4645 S. Lakeshore Dr.,
Suite 1
Tempe, AZ 85282
(602) 820-7050
TWX: 910-950-1153
FAX: 602-820-7054

ARKANSAS

Bonser-Philhower Sales
689 W. Renner Road
Suite 101
Richardson, TX 75080
(214) 234-8438
TWX: 214-437-0897

CALIFORNIA

SC Cubed
468 Pennsfield Place
Suite 101A
Thousand Oaks, CA 91360
(805) 496-7307
FAX: 805-495-3601

SC Cubed
17862 17th. St. #207
Tustin, CA 92680
(714) 731-9206
FAX: 714-731-7801

Quest-Rep Inc.
9444 Farnham St., Suite 107
San Diego, CA 92123
(619) 565-8797
FAX: 619-565-8990

Norcomp
3350 Scott Blvd., Suite 24
Santa Clara, CA 95054
(408) 727-7707
TWX: 510-600-1477
FAX: 408-986-1947

Norcomp
2140 Professional Drive
Suite 200
Roseville, CA 95661
(916) 782-8070
FAX 916-782-8073

COLORADO

Front Range Marketing
3100 Arapahoe Rd.,
Suite 404
Boulder, CO 80303
(303) 443-4780
TWX: 910-940-3442
FAX: 303-447-0371

CONNECTICUT

Lindco Associates, Inc.
Cornerstone Professional Park
Suite C-101
Woodbury CT, 06798
(203) 266-0728
FAX: 203-266-0784

DELAWARE

Delta Technical Sales, Inc.
122 N. York Road
Suite 9
Hatboro, PA 19040
(215) 957-0600
FAX: 215-957-0920

Micro Comp, Inc.
1421 S. Caton Avenue
Baltimore, MD 21227-1082
(301) 644-5700
TWX: 510-600-9460
FAX: 301-644-5707

FLORIDA

Semtronic Assoc., Inc.
657 Maitland Avenue
Altamonte Springs, FL 32701
(407) 831-8233
FAX: 407-831-2844

Semtronic Assoc., Inc.
3471 N. W. 55th Street
Ft. Lauderdale, FL 33309
(305) 731-2484
FAX: 305-731-1019

Semtronic Assoc., Inc.
1467 South Missouri Avenue
Clearwater, FL 34616
(813) 461-4675
FAX: 813-442-2234

GEORGIA

Novus Group - GA
6115-A Oakbrook Parkway
Norcross, GA 30093
(404) 263-0320
404-263-8946

IDAHO (Southwest)

Thorson Company Northwest
12340 N.E. 8th Street
Suite 201
Bellevue, WA 98005
(206) 455-9180
FAX: 206-455-4185

Wasatch-Pro Rep Marketing
360 East 4500 South, Suite 6
Salt Lake City, UT 84107
(801) 268-3434
FAX: 801-266-9021

ILLINOIS

Beta Technology Sales, Inc.
1009 Hawthorne Drive
Itasca, IL 60143
(708) 250-9586
TWX: 62885853
FAX: (312) 250-9592

Advanced Technical Sales
1810 Craig Road, Suite 213
St. Louis, MO 63145
(314) 878-2921
FAX: 314-878-1994

INDIANA

Arete Sales Inc.
2260 Lake Ave Suite 250
Fort Wayne, IN 46805
(219) 423-1478
FAX: 219-420-1440

Arete Sales Inc.
918 Fry Road Suite B
Greenwood, IN 46142
(317) 882-4407
FAX: 317-888-8416

IOWA

Advanced Technical Sales
375 Collins Road N.E.
Cedar Rapids, IA 52402
(319) 393-8280
FAX: 319-393-7258

KANSAS

Advanced Technical Sales
610 N. Mur-Len, Suite 8
Olathe, KS 66062
(913) 782-8702
TWX: 910-350-6002
FAX: 913-782-8641

KENTUCKY

Arete Sales, Inc.
918 Fry Road
Suite B
Greenwood, IN 46142
(317) 882-4407
FAX: 317-888-8416

LOUISIANA (Northern)

Bonser-Philhower Sales
689 W. Renner Rd., Suite 101
Richardson, TX 75080
(214) 234-8438
TWX: 910-867-4752
FAX: 214-437-0897

LOUISIANA (Southern)

Bonser-Philhower Sales
10700 Richmond, Suite 150
Houston, TX 77042
(713) 782-4144
TWX: 910-350-3451
FAX: 713-789-3072

MAINE

Mill-Bern Associates, Inc.
2 Mac Road
Woburn, MA 01801
(617) 932-3311
TWX: 710-332-0077
FAX: 617-932-0511

MARYLAND

Micro Comp, Inc.
1421 S. Caton Avenue
Baltimore, MD 21227-1082
(301) 644-5700
TWX: 510-600-9460
FAX: 301-644-5707

MASSACHUSETTS

Mill-Bern Associates, Inc.
2 Mack Road
Woburn, MA 01801
(617) 932-3311
TWX: 710-332-0077
FAX: 617-932-0511

MICHIGAN

A.P. Associates
810 E. Grand River
Brighton, MI 48116
(313) 229-6550
FAX: 313-229-9356

MINNESOTA

Com-Tek
6525 City West Parkway
Eden Prairie, MN 55344
(612) 941-7181
TWX: 310-431-0122
FAX: 612-941-4322

MISSOURI

Advanced Technical Sales
601 N. Mur-Len, Suite 8
Olathe, KS 66062
(913) 782-8702
TWX: 910-350-6002
FAX: 913-782-8641

Advanced Technical Sales
1810 Craig Road, Suite 213
St. Louis, MO 63146
(314) 878-2921
FAX: 314-878-1994

NEBRASKA

Advanced Technical Sales
601 N. Mur-Len, Suite 8
Olathe, KS 66062
(913) 782-8702
TWX: 910-350-6002
FAX: 913-782-8641

NEVADA

Norcomp
(Excluding Las Vegas)
3350 Scott Blvd., Suite 24
Santa Clara, CA 95054
(408) 727-7707
TWX: 510-600-1477

Quatra Associates
(Las Vegas)
4645 S. Lakeshore Dr., Suite 1
Tempe, AZ 85282
(602) 820-7050
FAX: 602-820-7054

NEW HAMPSHIRE

Mill-Bern Associates, Inc.
2 Mack Road
Woburn, MA 01801
(617) 932-3311
TWX: 710-332-0077
FAX: 617-932-0511

NEW JERSEY (Northern)

Parallax
734 Walt Whitman Road
Mellville, NY 11747
(516) 351-1000
FAX: 516-351-1606

NEW JERSEY (Southern)

Delta Technical Sales, Inc.
122 N. York Road
Suite 9
Hatboro, PA 19040
(215) 957-0600
FAX: 215-957-0920

NEW MEXICO

Quatra Associates
600 Autumwood Place, S. E.
Albuquerque, NM 87123
(505) 296-6781

NEW YORK (Metro)

Parallax
734 Walt Whitman Road
Mellville, NY 11747
(516) 351-1000
FAX: 516-351-1606

NEW YORK

Gen-Tech Electronics
4855 Executive Drive
Liverpool, NY 13088
(315) 451-3480
TWX: 710-545-0250
FAX: 315-451-0988

Gen-Tech Electronics
41 Burning Tree Lane
Rochester, NY 14526
(716) 381-5159
FAX: 716-381-5159*

Gen-Tech Electronics
70 Sandoris Circle
Rochester, NY 14622
(716) 467-5016

Gen-Tech Electronics
5 Arbutus Lane
Binghampton, NY 13901
(607) 648-8833

NORTH CAROLINA

The Novus Group - NC & SC
102L Commonwealth Court
Cary, NC 27511
(919) 460-7771
FAX: 919-460-5703

NORTH DAKOTA

Com-Tek
6525 City West Parkway
Eden Prairie, MN 55344
(612) 941-7181
TWX: 310-431-0122
FAX: 612-941-4322

OHIO

Bear Marketing, Inc.
P.O. Box 427
3554 Brecksville Road
Richfield, OH 44286-0427
(216) 659-3131
FAX: 216-659-4823
TWX: 810-427-9100

Bear Marketing, Inc.
240 W. Elmwood Drive
Suite 1012
Centerville, OH 45459-4248
(513) 436-2061
FAX: 513-436-9137

OKLAHOMA

Bonser-Philhower Sales
689 W. Renner Rd., Suite 101
Richardson, TX 75080
(214) 234-8438
TWX: 910-867-4752
FAX: 214-437-0897

OREGON

Thorson Company Northwest
6700 S.W. 105th Ave.,
Suite 104
Beaverton, OR 97005
(503) 644-5900
FAX: 503-644-5919

PENNSYLVANIA (Eastern)

Delta Technical Sales, Inc.
122 New York Road
Suite 9
Hatboro, PA 19040
(215) 957-0600
FAX: 215-957-0920

PENNSYLVANIA (Western)

Bear Marketing, Inc.
300 Mt. Lebanon Blvd.
Pittsburg, PA 15234
(412) 531-2002
FAX: 412-531-2008

PUERTO RICO

Mill-Bern Associates, Inc.
2 Mac Road
Woburn, MA 01801
(617) 932-3311
TWX: 710-332-0077
FAX: 617-932-0511

Semtronic Assoc., Inc.
Mercantile Plaza Building
Suite 816
Hato Rey, PR 00918
(809) 766-0700/0701
FAX: 809-763-0701

RHODE ISLAND

Mill-Bern Associates, Inc.
2 Mack Road
Woburn, MA 01801
(617) 932-3311
TWX: 710-332-0077
FAX: 617-932-0511

SOUTH CAROLINA

The Novus Group - NC & SC
102L Commonwealth Court
Cary, NC 27511
(919) 460-7771
FAX: 919-460-5703

SOUTH DAKOTA

Com-Tek
6525 City West Parkway
Eden Prairie, MN 55344
(612) 941-7181
TWX: 310-431-0122
FAX: 612-941-4322

TEXAS

Bonser-Philhower Sales
8240 MoPac Expwy.,
Suite 135
Austin, TX 78759
(512) 346-9186
TWX: 910-997-8141
FAX: 512-346-2393

Bonser-Philhower Sales
10700 Richmond, Suite 150
Houston, TX 77042
(713) 782-4144
TWX: 910-350-3451
FAX: 713-789-3072

Bonser-Philhower Sales
689 W. Renner Rd., Suite 101
Richardson, TX 75080
(214) 234-8438
TWX: 910-867-4752
FAX: 214-437-0897

TEXAS (El Paso County)

Quatra Associates
600 Autumwood Place SE
Albuquerque, NM 87123
(505) 296-6781

UTAH

Wasatch Pro Rep Marketing
360 East 4500 South, Suite 6
Salt Lake City, UT 84107
(801) 268-3434
FAX: 801-266-9021

VERMONT

Mill-Bern Associates, Inc.
2 Mac Road
Woburn, MA 01801
(617) 932-3311
TWX: 710-332-0077
FAX: 617-932-0511

VIRGINA

Micro Comp, Inc.
Rt. 2, Box 390
Huddleston, VA 24104
(703) 297-6295

WASHINGTON

Thorson Company Northwest
12340 N.E. 8th Place
Suite 201
Bellevue, WA 98005
(206) 455-9180
FAX: 206-455-9185
TWX: 910-443-2300

WASHINGTON
(Vancouver, WA only)

Thorson Company Northwest
6700 S.W. 105th Ave.,
Suite 104
Beaverton, OR 97005
(503) 644-5900
FAX: 503-644-5919

WASHINGTON D.C.

Micro Comp, Inc.
1421 S. Caton Avenue
Baltimore, MD 21227
(301) 644-5700
TWX: 510-600-9460
FAX: 301-644-5707

WISCONSIN (Western)

Com-Tek
6525 City West Parkway
Eden Prairie, MN 55344
(612) 941-7181
TWX: 310-431-0122
FAX: 612-941-4322

WISCONSIN (Eastern)

Beta Technology Sales, Inc.
9401 Beloit, Suite 409
Mllwaukee, WI 53227
(414) 543-6609

WYOMING

Wasatch Pro Rep Marketing
360 East 4500 South, Suite 6
Salt Lake City, UT 84107
(801) 268-3434
FAX: 801-266-9021

INTERNATIONAL SALES REPRESENTATIVES

AUSTRALIA

ACD/ITRONICS
106 Belmore Rd. North
Riverwood, N.S.W. 2210
Tel: (61) 2-534-6200
FAX: (61) 2-534-4910

ACD/ITRONICS
Unit 2, 17-19 Melrich Road
Bayswater VIC 3153
P.O. Box 139
Tel: (61) 3-762 7644
FAX: (61) 3-762 5446

ACD/ITRONICS
55 Noreen Street
Chapel Hill QLD 4069
Tel: QLD 878 1488
FAX: QLD 878 1490

ACD/INTRONICS
261 Sturt Street
Adelaide
S.A. 5000
Tel: (61) 8-213-6505
FAX: (61) 8-231-8245

AUSTRIA

Eljapex G.m.b.H
Eitnergasse 6
A-1232 Wien
Austria
Tel: (43) 1-861531
FAX: (43) 1-861531300

BELGIUM & LUXEMBURG

Lemarie Rodelco
Limburg Stirum 243
1810 Wemmel
Belgium
Tel: (32) 2-460-0560
FAX: (32) 2-460-0271

CANADA
(BRITISH COLUMBIA)

Thorson Company Northwest
12340 N.E. 8th Street
Suite 201
Bellevue, WA 98005
(206) 455-9180

Electro Source
3665 Kingsway, Suite 300
Vancouver, B.C. V5R 5W2
(604) 435-8066
FAX: 604-435-8181

CANADA (ONTARIO)

Electro Source, Inc.
340 March Road, Suite 503
Kanata, Ontario K2K 2E3
(613) 592-3214
FAX: 613-592-4256

Electro Source, Inc.
230 Galaxy Boulevard
Rexdale, Ontario M9W 5R8
(416) 675-4490
FAX: 416-675-6871

*Activates Fax

CANADA (QUEBEC)

Electro Source
6600 TransCanada Hwy
Suite 420 Point Claire
Quebec H9R 4S2
(514) 630-7486
FAX: 514-630-7421

DENMARK

Dana Tech KS
PO Box 1361
Smedeland 8
2600 Glostrup
Denmark
Tel: (45) 2-4345 47
FAX: (45) 2-4345 67

FINLAND

Field OY Instrumentarium
Nilittylanpolku 10
SF -00620
Helsinki, Finland
Tel: (358) 0-7571011
FAX: (358) 0-798853

FRANCE

Reptronic
1 Bis, rue Marcel Paul
Batimen R
Z.I. de la Bonde
F-91300 Massy
Tel: (33) 1-60139300
FAX: (33) 1-60139198

R.T.F. Composant
81, Rue Pierre Semard
92320 Chatillon s/s Bagneux
France
Tel: (33) 1-49 65 27 00
FAX: (33) 1-49 65 27 38

R.T.F. Sud-Ouest
Avenue de la Mairie
31320 Escalquens, France
Tel: (33) 61-81 51 57
FAX: (33) 61-81 51 57

R.T.F. Aquitaine
13, Rue l'Hote
33000 Bordeaux, France
Tel: (33) 56-52 99 59
FAX: (33) 56-48 17 83

R.T.F. Rhône-Auvergne
Parc Club du Moulin à Vent
Bât G.33, Rue du Docteur
Levy
69200 Venissieux, France
Tel: (33) 78-00 07 26
FAX: (33) 78-01 20 57

R.T.F Provence Côte D'Azur
Résidence du Petit Bosquet
Bât. C-18
Avenue du Petit Bosquet
13012 Marseilles, France
Tel: (33) 91-06 02 18
FAX: (33) 91-06 47 82

R.T.F. Ouest
3, rue de Paris
35510 Cesson Sevigne,
France
Tel: (33) 99-83 84 85
FAX: (33) 99-83 80 83

R.T.F. Rhône-Alpes
Mini Parc - Zac des Bealières
23, Avenue de Granier
38240 Ebbens
Tel: (33) 76-90 11 88
FAX: (33) 76-41 04 09

GERMANY

Metronik
Leonhardsweg 2
8025 Unterhaching
München, Germany
Tel: (49) 89-611080
FAX: (49) 89-6116468

Metronik
Semerteichstrasse 92
4600 Dortmund 30
Dortmund, Germany
Tel: (49) 231-423037/38
TLX: (49) 8227082

Metronik
Osterbrooksweg 61
2000 Schenefeld
Hamburg, Germany
Tel: (49) 40-8304061
TLX: (49) 2162488

Metronik
Siemensstrasse 4-6
6805 Heddesheim
Mannheim, Germany
Tel: (49) 62-034701-03
TLX: (49) 465053

Metronik
Laufamholzstr. 118
8500 Nürnberg 30
Nürnberg, Germany
Tel: (49) 911-590061/62
TLX: (49) 626205

Metronik
Löwenstr. 37
7000 Stuttgart 70
Stuttgart, Germany
Tel: (49) 711-764033/35
TLX: (49) 7255228

GREECE

Peter Caritato and Assoc. S. A.
Ilia Iliot, 31
Athens 11743 Greece
Tel: (30) 1-9020165
FAX: (30) 1-9017024

HONG KONG

Excel Associates, Ltd.
1502 Austin Tower
22-26A Austin Avenue
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 3-7210900
FAX: (852) 3-696826

INDIA

Malhar Corporation
924 County Line Road
Bryn Mawr, PA 19010
(215) 527-5020
FAX: (215) 525-7805

IRELAND

Memec Ireland Ltd.
Innovation Centre
Enterprise House
Plassey Technological Park
Limerick
Ireland
Tel: (353) 61-330742/5
FAX: (353) 61-3318-88

ISRAEL

E.I.M International Ltd.
8 Emil Zola St.
P.O. Box 4000
Petach Tiqva
Israel 49130
Tel: (972) 3-92 33257
FAX: (972) 3-924 4857

ITALY

ACSIS S.R.L.
Via Alberto Mario. 26
20149 Milano, Italy
Tel: (39) 2-4390832
FAX: (39) 2-4697607

Celdis Italiana S.P.A.
Via F.ill Gracchi N36
20092 Cinisello Balsamo
Milano, Italy
Tel: (39) 2-61 839 1
FAX: (39) 2-61 735 13

Celdis Italiana S.P.A.
Via Massarenti 219/4
40138 Bologna, Italy
Tel: (39) 51-53 333 6

Celdis Italiana S.P.A.
Via Savelli 15
35100 Padova, Italy
Tel: (39) 49-77 209 9

Celdis Italiana S.P.A.
Via G. Pitre' 11
00162 Roma, Italy
Tel: (39) 6-42 897 1

Celdis Italiana S.P.A.
Via Mombarcaro 96
10136 Torino, Italy
Tel: (39) 11-32 993 88

JAPAN

Okura & Co., Ltd.
6-12, Ginza Nichome
Chuo-Ku
Tokyo, 104 Japan
Tel: (03) 3566 6361
FAX: (03) 3563 5447

Dia Semicon Systems, Inc.
Flower-Hill Shin-Machi
East Bldg.
1-23-9 Shin-machi,
Setagaya-ku,
Tokyo, 154 Japan
Tel: (03) 3439-2700
FAX: (03) 3439-2701

New Japan Radio Trading
Co., Ltd.
Shiba-Eitaro Bldg.
4-14 Shiba-Daimon 1-chome,
Minato-ku,
Tokyo, 105 Japan
Tel: (03) 3459-1521
FAX: (03) 3459-1520

Okura Electronics Co., Ltd.
3-6, Ginza Nichome,
Chuo-ku, Tokyo, 104 Japan
Tel: (03) 3564-6871
FAX: (03) 3564-6870

Okura Electronics Service
Co., Ltd.
Kyoei Bldg.
5-3, Kyobashi 3-chome,
Chuo-ku, Tokyo, 104 Japan
Tel: (03) 3567-6501
FAX: (03) 3567-7800

Tokyo Electron Limited
P. O. Box 7006
Shinjuku Monolith
3-1 Nishi-Shinjuku 2-chome,
Shinjuku-ku,
Tokyo, 163 Japan
Tel: (03) 3340-8193
FAX: (03) 3340-8408

Towa Elex Co., Ltd.
Lapore Shinjuku
2-15-2 Yoyogi,
Shibuya-ku, Tokyo, 151
Japan
Tel: (03) 5371-3411
FAX: (03) 5371-4760

Varex Co., Ltd.
Nippo Shin-Osaka No. 2 Bldg.
1-8-33, Nishimiyahara,
Yodogawa-ku,
Osaka, 532 Japan
Tel: (06) 394-5201
FAX: (06) 394-5449

KOREA

Excel-Tech
410-5 Hapjeong-Dong
Mapo-Gu
Seoul, Korea
Tel: (82) 2-3357823
FAX: (82) 2-3357825

THE NETHERLANDS

Rodelco BV Electronics
Takkebijsters 2
P.O. Box 6824
4802 HV Breda
The Netherlands
Tel: (31) 76-784911
FAX: (31) 76-710029

NORWAY

B.I.T. Elektronikk SA
P.O. Box 36 Lerbyen
N-3401 Lier, Norway
Tel: (47) 3-84 70 99
FAX: (47) 3-84 55 10

SOUTHEAST ASIA

Excel Associates, Ltd.
1502 Austin Tower
22-26A Austin Avenue
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 3-7210900
FAX: (852) 3-696826

SPAIN

ADM Electronica SA
Menorea No. 3 Entreplants
Madrid 28009
Spain
Tel: (34) 1-4094725
FAX: (34) 1-4096903

SWEDEN

D.I.P. Electronics AB
Danvik Centre
P.O. Box 15046
S-104 65 Stockholm, Sweden
Tel: (46) 8-44 91 90
FAX: (46) 8-43 00 47

SWITZERLAND

Data Comp AG
Silbernstrasse 10
CH-8953 Dietikon
Zurich, Switzerland
Tel: (41) 1-7405140
FAX: (41) 1-7413423

TAIWAN

Molecatex, Inc.
21F 258 Sec 3
Nanking East Road
Taipei, Taiwan R.O.C.
Tel: (886) 2-7410400
FAX: (886) 2-7217461

Jeritron Ltd.
5/F Fu San Building
1182 Cheng-Teh Road
Taipei, Taiwan, R.O.C.
Tel: (886) 2-8823154
FAX: (886) 2-8820710

UK

Memec
17 Thame Park Road
Thame
Oxon OX9 3XD
England
Tel: (44) 84-4261939
FAX: (44) 84-4261678

Distributed in North America By

Hamilton/Avnet
locations throughout
the U.S. and Canada.
1-800-HAM-ASIC
FAX: 408-743-3003

Western Microtechnology
12900 Saratoga Ave.
Saratoga, CA 95070
(408) 725-1660
TWX: 910-338-0013
FAX: 408-255-6491

Insight Electronics
6885 Flanders Drive
San Diego, CA 92121
(619) 587-9757
TWX: 183035-UD
FAX: 619-587-1380

Phase 1 Technology
Corporation
1110 Rte. 109
N. Lindenhurst, NY 11757
(516) 957-4900
FAX: 516-957-4909

Nu Horizons
Electronics Corp.
6000 New Horizons Blvd.
Amityville, New York 11701
(516) 226-6000
FAX: 516-226-6262

Marshall Industries
locations throughout
the U.S. and Canada.
(818) 459-5500
FAX: 818-459-5660

For Further Information... *Please check the appropriate box*

- ❑ Please have a Sales Representative call me.
- ❑ I would like to borrow a copy of your Logic Cell Array Technical Demonstration Video.
- ❑ Please add my name to your mailing list.

My application is _____

I have a new design starting in ____ weeks ____ months

Name _____ Title _____

Company _____ M/S _____

Street Address _____

City _____

State _____ Zip _____

Phone (_____) _____

The Programmable Gate Array Company

For Further Information... *Please check the appropriate box*

- ❑ Please have a Sales Representative call me.
- ❑ I would like to borrow a copy of your Logic Cell Array Technical Demonstration Video.
- ❑ Please add my name to your mailing list.

My application is _____

I have a new design starting in ____ weeks ____ months

Name _____ Title _____

Company _____ M/S _____

Street Address _____

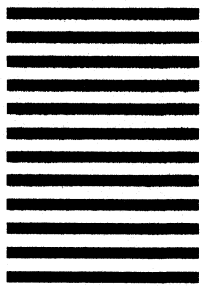
City _____

State _____ Zip _____

Phone (_____) _____

The Programmable Gate Array Company

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES



BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 8051 SAN JOSE, CA

POSTAGE PAID BY ADDRESSEE

XILINX

2100 Logic Drive
San Jose, CA 95124-9920

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

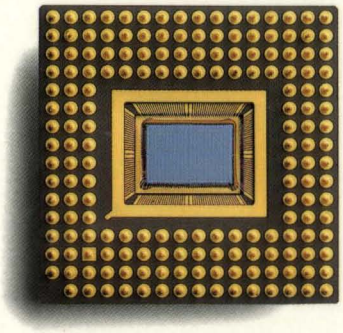
BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 8051 SAN JOSE, CA

POSTAGE PAID BY ADDRESSEE

XILINX

2100 Logic Drive
San Jose, CA 95124-9920



XILINX

The Programmable Gate Array Company.

2100 Logic Drive, San Jose, CA 95124.

Tel: (408) 559-7778 EasyLink 62916309 TWX: 5106008750 XILINX UQ FAX: (408) 559-7114

Printed in U.
P/N 0010048