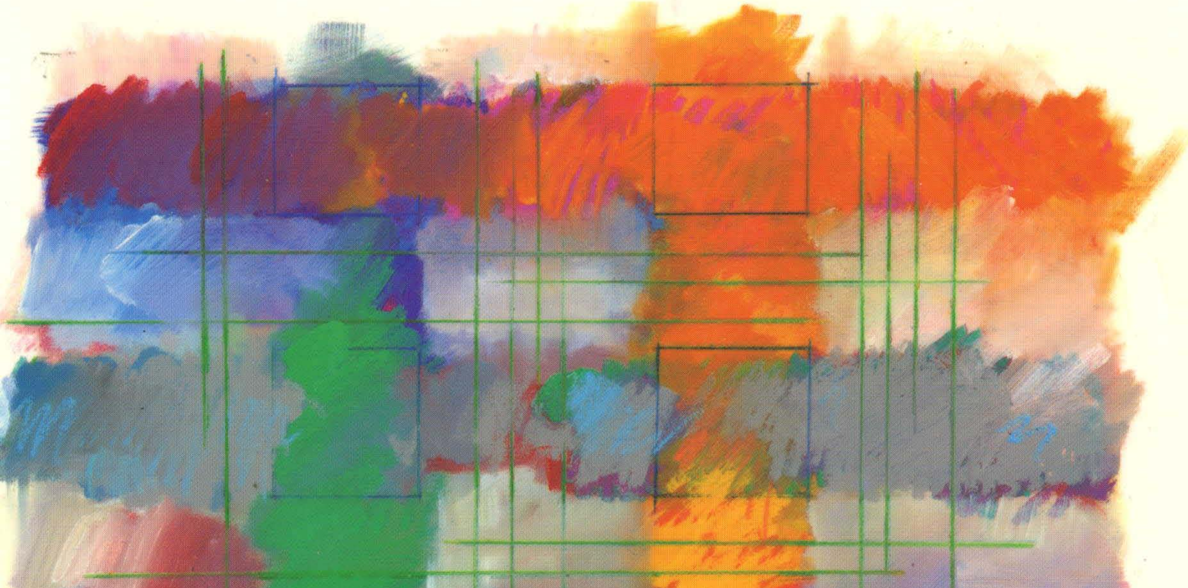
XILINX

The Programmable Gate Array Data Book



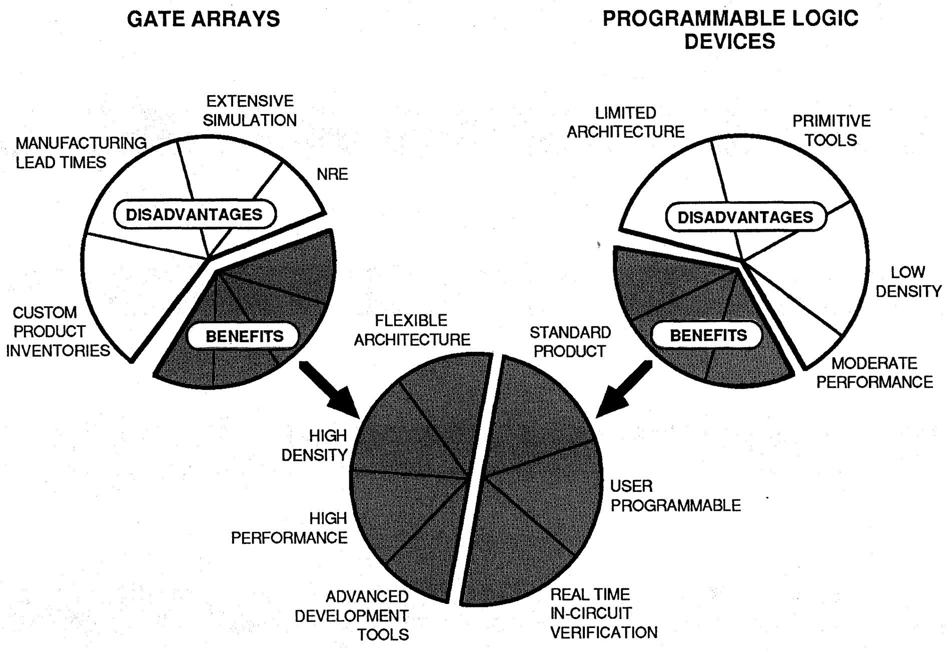
1989

XILINX

SECTION TITLES

XILINX

The Programmable Gate Array Company



1162 01

THE PROGRAMMABLE GATE ARRAY
(LOGIC CELL™ ARRAY)

XILINX

TABLE OF CONTENTS

TABLE OF CONTENTS

6 Applications *(Con't)*

7 Article Reprints

8 Index

XILINX

SECTION 1

Programmable Gate Arrays

XILINX

Programmable Gate Arrays

Contributors

Peter Alfke
Keath Armstrong
Jim Chumbley
Rick Dudley
Chuck Erickson
Lee Farrell
Brad Fawcett
Chuck Fox
Dave Galli
Jim Hsieh
Steve Knapp
Dave Lautzenheiser
Mark Markham
Bob McGrath
Bill O'Neill
Wes Patterson
Richard Ravel
Ed Resler
Steve Schreifels
Don Soderman
Robert Stransky
Thomas Waugh
Perry Wu
Pardner Wynn

Patents Pending

Xilinx, Logic Cell, LCA, XACT, XACTOR, Programmable Gate Array, and Logic Processor are trademarks of Xilinx, Inc. The Programmable Gate Array Company is a Service Mark of Xilinx, Inc.

IBM is a registered trademark and PC/AT, PC/XT, PS/2, and Micro Channel are trademarks of International Business Machines Corporation. ABEL is a trademark and Data I/O is a registered trademark of Data I/O Corporation. FutureNet is a registered trademark and DASH is a trademark of FutureNet Corporation, a Data I/O Company. SimuCad and Silos are registered trademarks and P-Silos and P/C-Silos are trademarks of SimuCad Corporation. Microsoft is a registered trademark and MS-DOS is a trademark of Microsoft Corporation. Logitech is a registered trademark of LOGITECH Inc. Lotus is a registered trademark of Lotus Development Corporation. AboveBoard and AboveBoard/PS are trademarks of Intel Corporation. RAMpage!, SixPakPlus and SixPakPremium are registered trademarks of AST Research, Inc. Mouse Systems is a trademark of Mouse Systems Corporation. Centronics is a registered trademark of Centronics Data Computer Corporation. PAL and PALASM are registered trademarks of Advanced Micro Devices, Inc., DAISY, Logician, and DNIX are registered tradmarks of Daisy Systems. UNIX is a trademark of AT&T Technologies, Inc. CUPL is a trademark of Logical Devices, Inc.. Apollo and AEGIS are registered trademarks of APOLLO Computer. Mentor and IDEA are registered trademarks and QuickSim, NETED, EXPAND are trademarks of Mentor Graphics, Inc. ValidGED and ValidSim are trademarks of Valid Logic Systems, Inc. Sun is a registered trademark of Sun Microsystems, Inc. SCHEMA II is a trademark of Omation Corporation. OrCAD is a registered trademark of OrCAD Systems Corporation. Viewlogic is a registered trademark of Viewlogic Systems, Inc. CASE Technology is a trademark of CASE Technology, a division of Teradyne's Electronic Design Automation Group. Xilinx, Inc. does not assume any liability arising out of the application or use of any product described herein; nor does it convey any license under its patent, copyright or maskwork rights or any rights of others. Xilinx, Inc. reserves the right to make changes, at any time, in order to improve reliability, function or design and to supply the best product possible. Xilinx, Inc. cannot assume responsibility for the use of any circuitry described other than circuitry entirely embodied in their product. No other circuit patent licenses are implied. Xilinx, Inc. cannot assume responsibility for any circuits shown or represent that they are free from patent infringement or of any other third party right. Xilinx, Inc. assumes no obligation to correct any errors contained herein or to advise any user of this text of any correction if such be made.

Printed in U.S.A.

About the Company...

Xilinx is the first company to develop a user-programmable gate array. The invention of a high-density, general-purpose user-programmable logic device was the result of a number of breakthroughs, and several aspects of the device's array architecture have been patented. Since the introduction of its first product in 1985, Xilinx has continued to lead in the development of new programmable gate arrays with higher speeds, higher densities, and lower costs.

Due to their density and the convenience of user programmability, Xilinx's Programmable Gate Arrays represent an important new alternative in the market for Application Specific Integrated Circuits (ASICs). The company continues to concentrate its resources exclusively on expanding its growing family of Programmable Gate Arrays and associated development systems, and on providing technical support to a rapidly growing customer base.

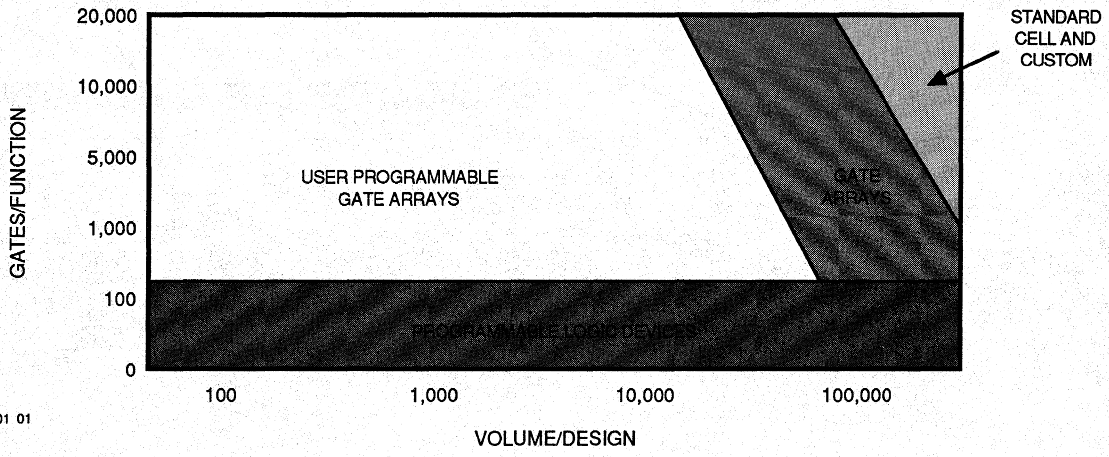
INTRODUCTION TO PROGRAMMABLE GATE ARRAYS

Steady advances in the level of integration 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).

Recent breakthroughs in logic architectures have resulted in the first high density ASICs that can be configured by the user. These user programmable gate arrays combine the logic integration benefits of custom VLSI with the design, production, and time to market advantages of standard products. The flexibility of user programming significantly reduces the risks of design changes and production rate changes.

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

□ USER PROGRAMMABLE GATE ARRAYS

Unlike conventional gate arrays, programmable gate arrays 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.

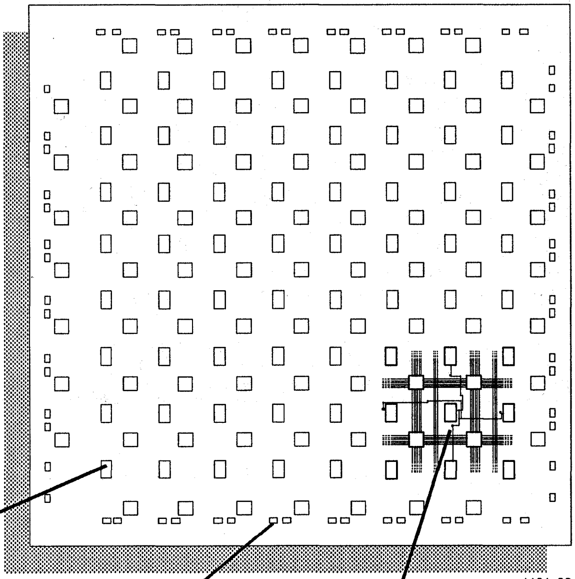
Unlike standard IC products, gate array costs include fixed costs as well as the production cost per unit. Gate arrays become cost effective when production volumes are high enough to provide a broad base for amortization of fixed costs.

PROGRAMMABLE GATE ARRAY ARCHITECTURE

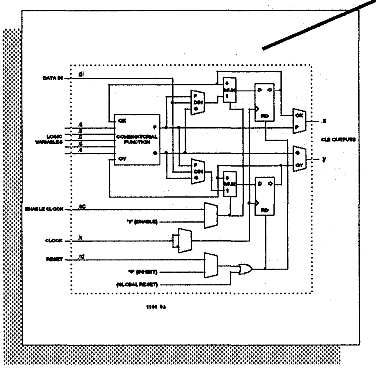
Xilinx's proprietary Logic Cell Array architecture is similar to that of other gate arrays, with an interior matrix of logic blocks and a surrounding ring of I/O interface blocks. Interconnect resources occupy the channels between the rows and columns of logic blocks, and between the logic blocks and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

Like a microprocessor, the Logic Cell Array (LCA) is a program-driven logic device. The functions of the LCA's 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.

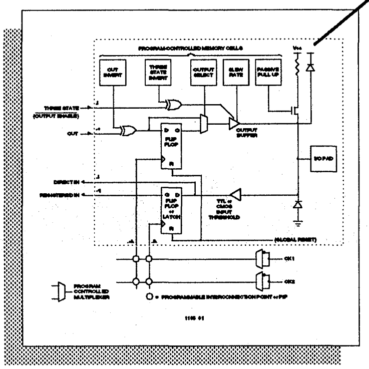
Logic Cell Array performance is determined by the speed of logic, storage elements, and programmable interconnect. LCA performance is specified by the maximum toggle rate for a logic block storage element configured as a toggle flip-flop. For typical applications, system clock rates are one-third to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



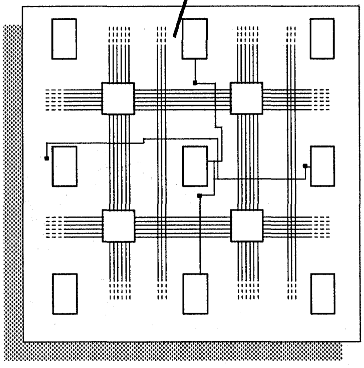
1101 02



1101 03



1101 04



1101 05

CONFIGURABLE LOGIC BLOCK

The core of the Logic Cell Array 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 be inputs to the combinatorial logic via an internal feedback path.

INPUT/OUTPUT BLOCK

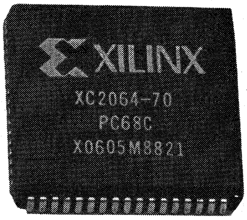
The periphery of the Logic Cell Array is made up of user programmable Input/Output Blocks (IOBs). Each block can be programmed independently to be an input, an output, or a bi-directional pin with three-state control. Inputs can be programmed to recognize either TTL or CMOS thresholds. Each IOB also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

INTERCONNECT

The flexibility of the LCA is due to resources that permit program control of the interconnection of any two points on the chip. Like other gate arrays, the LCA's interconnection resources include a two-layer metal network of lines that run horizontally and vertically in the rcws 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 allow signals to be switched 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 GATE ARRAY FAMILY



The XC2000 series of programmable gate arrays 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: 33, 50, and 70 MHz toggle rates
- ❑ Three package types: Dual In-line Package
  Plastic Leaded Chip Carrier
  Pin Grid Array
- ❑ TTL or CMOS input thresholds

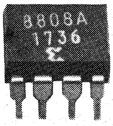
THE XC2000 FAMILY OF PROGRAMMABLE GATE ARRAYS

	XC2064	XC2018
Number of gates	1200	1800
Configurable Logic Blocks	64	100
Combinatorial Logic Functions	128	200
Latches and flip-flops	122	174
Input/Outputs	58	74

XC1736

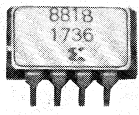
CMOS SERIAL CONFIGURATION PROM

The 1736 Serial Configuration PROM is a companion device that provides permanent storage of LCA configuration programs. It can be used whenever a dedicated device is preferable to sharing of a larger EPROM, or to loading from a microprocessor.



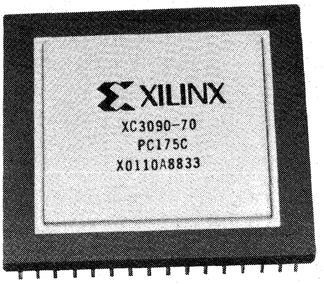
FEATURES

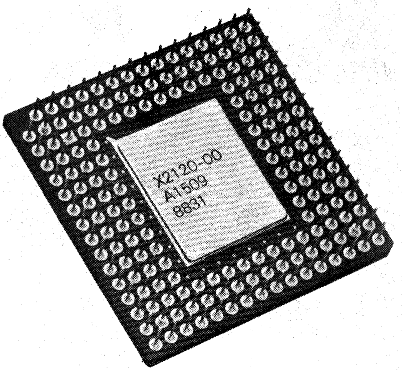
- ❑ One-Time Programmable (OTP) 36,288 bit serial memory designed to store configuration programs for Programmable Gate Arrays
- ❑ Simple interface to a Logic Cell Array (LCA) requires only two I/O pins
- ❑ Daisy chain support for multiple devices
- ❑ Cascadable for large arrays or many LCAs
- ❑ Storage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



XC3000
PROGRAMMABLE GATE ARRAY FAMILY

The XC3000 series is a second generation family of CMOS programmable gate 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

❑ Two performance options:
- 50 and 70 MHz toggle rates

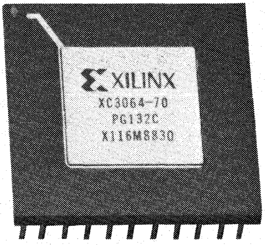
❑ Second generation architecture
- 5-input logic functions
- 2 flip-flops per CLB/IOB
- Enhanced routing resources
- Three-state drivers for wide ANDs

❑ Programmable voltage slew rates on outputs

❑ Three package types:
- Plastic Leaded Chip Carrier
- Pin Grid Array
- Quad Flat Package

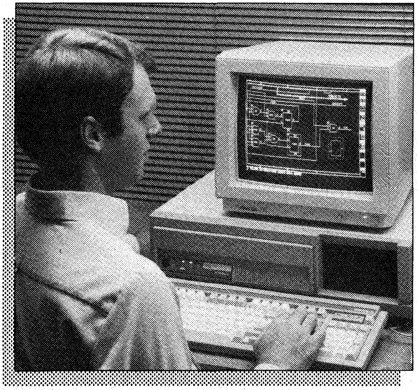
The XC3000 Family of Programmable Gate Arrays

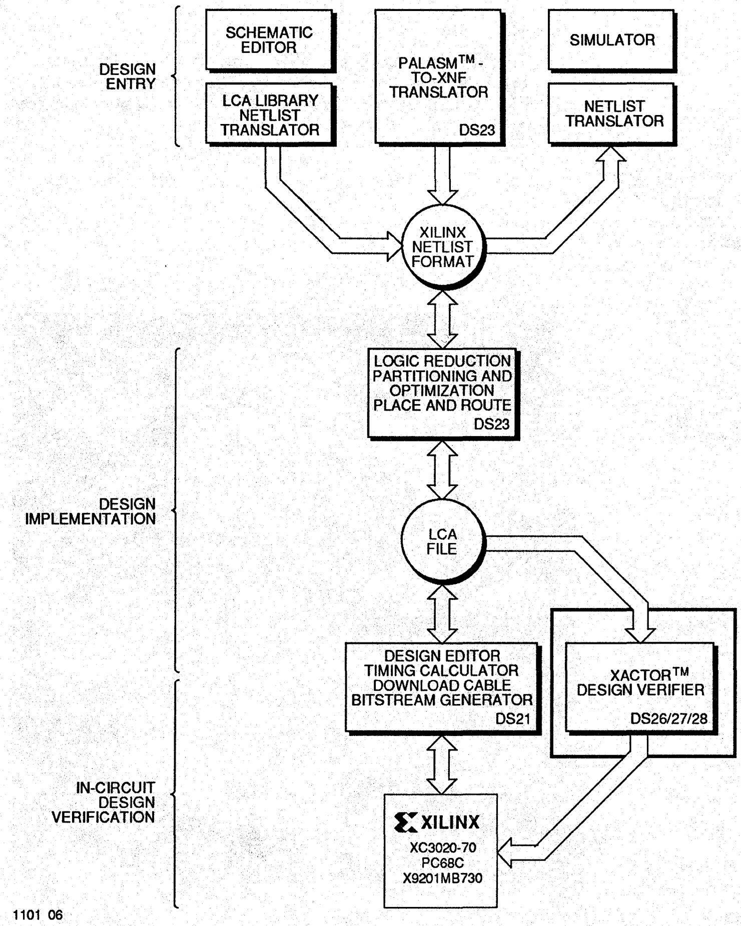
	XC3020	XC3030	XC3042	XC3064	XC3090
Number of gates	2000	3000	4200	6400	9000
Combinatorial Logic Functions	128	200	288	448	740
Latches and flip-flops	256	360	480	688	928
Input/Outputs	64	80	96	120	144



DEVELOPMENT SYSTEMS

Designing with Xilinx Programmable Gate Arrays 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 Xilinx 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
SCHEMATIC EDITOR
LCA LIBRARY NETLIST TRANSLATOR
PALASM™-TO-XNF TRANSLATOR
DS23
SIMULATOR
NETLIST TRANSLATOR
XILINX NETLIST FORMAT
DESIGN IMPLEMENTATION
LOGIC REDUCTION PARTITIONING AND OPTIMIZATION PLACE AND ROUTE
DS23
LCA FILE
DESIGN EDITOR TIMING CALCULATOR DOWNLOAD CABLE BITSTREAM GENERATOR
DS21
XACTOR™ DESIGN VERIFIER
DS26/27/28
IN-CIRCUIT DESIGN VERIFICATION
XILINX
XC3020-70
PC68C
X9201MB730
1101 06

Design Entry Software

consists of libraries and netlist interfaces for standard CAE software such as FutureNet, Schema II, OrCAD, Viewlogic, Daisy, 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-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 so that they will continue to recieve software updates.

XILINX USER GROUP

Xilinx users are invited to attend training and information exchange sessions that are held two-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 8:00 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 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

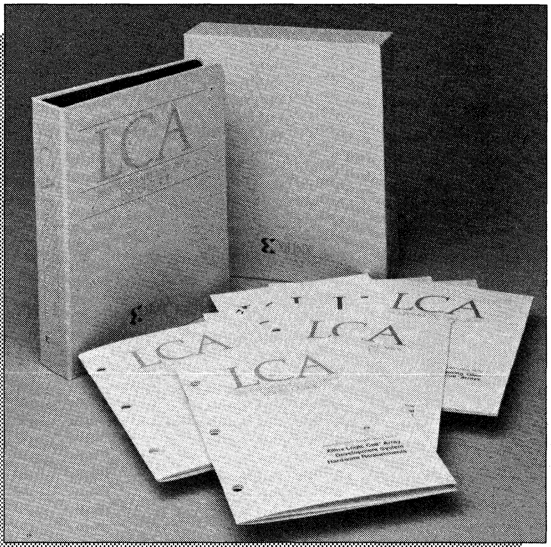
The User's Guide is a collection of "how to" applications notes on such subjects as getting started with an LCA design, Boolean equation design entry, use of the simulator, placement and routing optimization, and LCA configuration.

❑ 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.

User programmable gate arrays, 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.

Programmable Gate Arrays	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

Methodology

This analysis compares the total costs of custom gate arrays with those of user programmable gate arrays. It looks at the various categories of costs, both fixed and variable, for devices from 2000 to 9000 gates, 90% of the gate array market according to most studies.

Because the gate array has fixed or up-front development costs (NRE, extra simulation time, generating test vectors, etc.) that the programmable gate array 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 breakeven quantities—the volume below which it is more cost effective to use the programmable gate array (Breakeven Analysis). The overall objective is to determine the production volumes at which each product is most cost effective.

Executive Summary Conclusion

The choice between user 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: a programmable gate array. 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 breakeven graph for a 2000 gate device. The data are for 1990. 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 programmable gate array project cost starts at zero, but rises faster because of a higher cost per-unit. In this case the breakeven volume is between 10k and 20k units. This paper will discuss the various components of this analysis and show the user how to make a similar calculation for a specific situation.

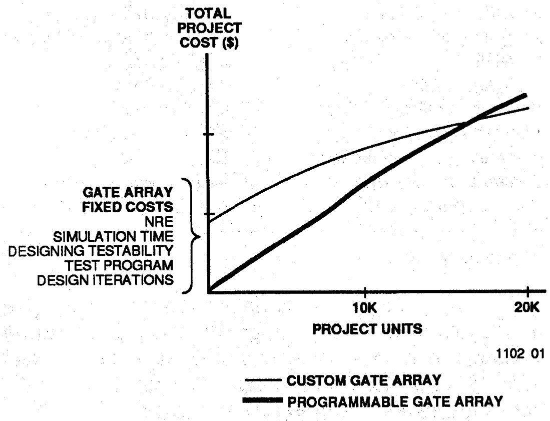
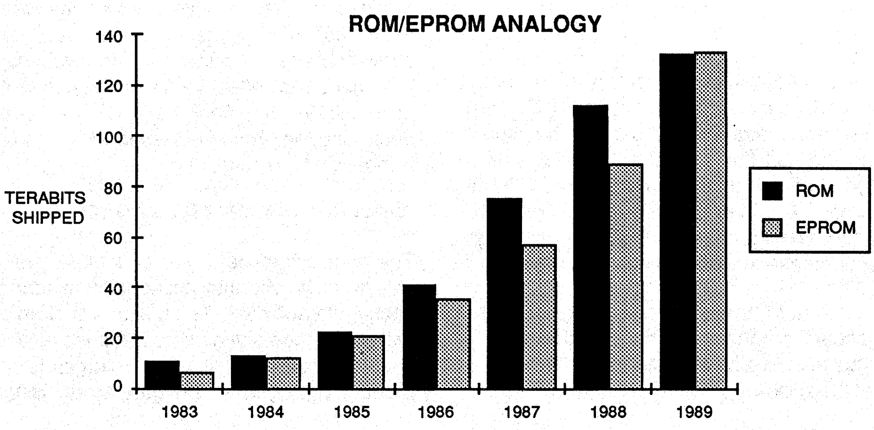


Figure 1. Typical Breakeven Analysis 2000 Gates—1990

Several significant factors are omitted from this graph. 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 Time to Market. Both of these factors would raise the custom gate array curve and increase the breakeven quantity. In other words, the programmable gate array 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 2 to 3 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,



	1983	1984	1985	1986	1987	1988	1989
EPROM MIL¢/BIT	10.9	9.2	4.0	2.5	1.8	1.3	0.9
ROM MIL¢/BIT	4.5	3.2	1.7	1.0	0.7	0.5	0.4
RATIO	2.4	2.9	2.4	2.5	2.6	2.6	2.3

1102 05

SOURCE: DATAQUEST

Figure 2. ROM/EPROM Analogy

lower inventory risk, 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, programmable gate arrays may be even more widely used versus gate arrays than are EPROMs versus ROMs.

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 programmable gate arrays 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: upfront 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 programmable gate arrays, 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 $5,000 and 2.5 manweeks of simulation effort for a 2000 gate array, and $10,000 and 7 manweeks for a 9000 gate array. This compares to 0.5 and 1 week for the programmable gate array, with no simulation charge.

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

	2000 Gates	9000 Gates
Gate Array		
Simulation Charge	$5K	$10K
Man Weeks	2.5 MW	7 MW
Programmable Gate Array		
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 1987 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 1 additional week for a 2000 gate array, and 2 additional weeks for a 9000 gate array.

The programmable gate array 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 online 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, $15,000 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. Programmable gate array 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
$15K–$20K	$30K–$40K

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, it is a risk 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	of	(original NRE time and cost + one half of original simulation time and cost)

Test Program Development

As noted in Time to Design for Testability, 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 2 weeks for a 2000 gate array, and 4 weeks for a 9000 gate array. Since the programmable gate array 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.

Programmable gate arrays are standard products that already have a second source.

Gate Array Incremental Cost

	2000 Gates	9000 Gates
NRE	$15K–$20K	$30K–$40K
Simulation		
Charge	$2.5K	$5K
Man Weeks	1 MW	3 MW

Summary of Gate Array Fixed Development Costs

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.

VARIABLE COSTS

Production Unit Cost (Cents/Gate)

Gate array prices are often quoted in terms of cents per gate. For 1.5 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 a programmable gate array is 2–3 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 programmable gate arrays 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 programmable gate arrays are newer products, their cost is declining at a steeper rate than gate arrays. They are in the introduction phase of their 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.

1990 Programmable Gate Array Unit Costs—Without Package

	2000 Gates	4000 Gates	9000 Gates
	20KU Qty	10KU Qty	10KU Qty
Programmable (Cents/Gate)	0.30–0.40	0.35–0.45	0.50–0.60

	Typical 2000 Gates	Typical 9000 Gates	Customer Application
1. Simulation			
NRE	$5,000	$10,000	
Man Weeks	2 MW	6 MW	
2. Design for Testability	1 MW	2 MW	
3. NRE Charges	$15K–$20K	$30K–$40K	
4. Design Iterations @ 50% probability			
NRE	$11,250	$22,500	
Man Weeks	0.5 MW	1.5 MW	
5. Test Program Development	2 MW	4 MW	
6. Second Source (NRE + 50% SIM)			
NRE	$20,000	$40,000	
Man Weeks	1 MW	3 MW	
Total Without Second Source			
NRE	$33,750	$67,500	
Man Weeks	5.5 MW	13.5 MW	
Total With Second Source			
NRE	$53,750	$107,500	
Man Weeks	6.5 MW	16.5 MW	
Total Fixed Costs @ $2K/MW			@ $__/MW
Without Second Source	$44,750	$94,500	
With Second Source	$66,750	$140,500	

Table 1. Typical Fixed Costs

Process Technology

There are also technology reasons for the steeper decline in cost of the programmable gate array. Figure 3 shows that the processes used for logic IC's, including gate arrays, typically lag behind those used for memory IC's. Since the programmable gate array 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 programmable gate array will move to successively more advanced processes, resulting in steadily decreasing costs. By the end of the production lifetime, the programmable gate array will be several processes ahead and the cost difference will be reduced significantly.

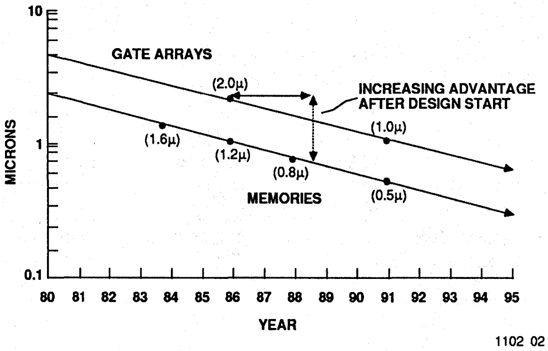


Figure 3. Process Evolution

Pad-Limited Die Sizes

As gate arrays and programmable gate arrays 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 programmable gate array and the gate array would be the same die size. As a result, the higher volume, standard product, programmable gate array could actually be less expensive on a per-unit basis than the custom product gate array. There would be no breakeven quantity—the programmable gate array would have a lower cost of ownership at all volumes.

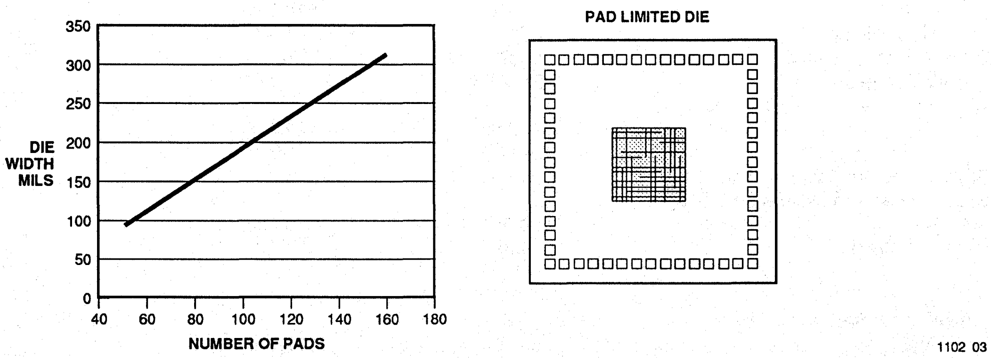


Figure 4. Minimum Die Size vs. I/O

Effect of Die Cost on Total Cost

Figure 5 illustrates a third point about the capability of programmable gate arrays 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 programmable gate array—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 programmable gate array in a more advanced process than the custom gate array, as discussed in Unit Cost (Cents/Gate).

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 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.

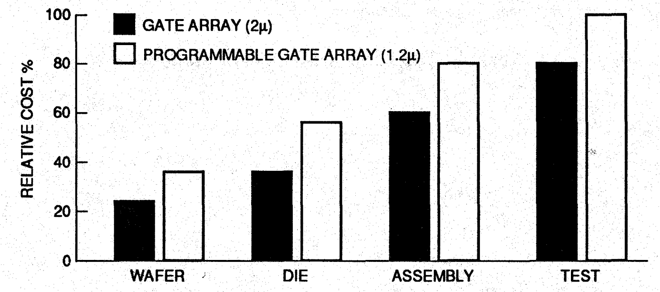
Although the safety stock reserve is a function of the cost of the product itself, a figure of 10% is reasonable for gate arrays that have unit costs under $25.00. In comparison, since changes to programmable gate arrays can be made in software in minutes, and since only one part type is widely stocked,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

The March, 1986 Technology Research Letter states that only 1/3 of gate array designs go into production. The 1987 Dataquest ASIC and Standard Logic Semiconductor Volume 1 reports 50% go into production. With 50%, 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 Summary of Fixed Development Costs, this would mean an additional ($5,000 + 3 MW + $17,500). For 9000 gates the number is ($10,000 + 8 MW + $35,000).



1102 04

Figure 5. Relative Manufacturing Cost by Stage of Completion

Gate Array Incremental Cost

Simulation Cost + Time to Design for Testability + NRE Cost

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 programmable gate array is less expensive, until the gate array can amortize the upfront 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 "breakeven 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				
Programmable Gate Array				
13. Unit cost				

Table 2. Total Cost vs. Volume Purchased

BREAKEVEN ANALYSIS

Figure 6 is a graphical representation of the breakeven calculation for the case of 2000 gates, 1990 pricing, and no second source. Up to the breakeven 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 breakeven graph is merely line 5 or line 11 plotted versus quantity. For the programmable gate array, it is line 13 times the quantity plotted versus the quantity.

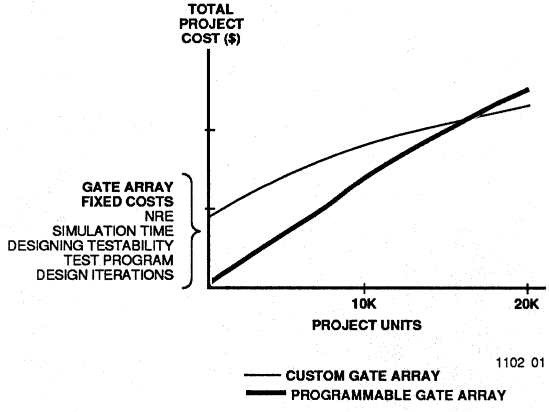


Figure 6. Typical Breakeven Analysis 2000 Gates—1990

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's lifetime. If there is any problem in simulation, or any iteration of the gate array design, then a gate array would easily add six additional 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

In the electronics industry the lifetimes of products are shrinking. In the personal computer industry it is not uncommon to find product upgrades within 6-12 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.

The numbers used for Figure 1 show that 2000 gate programmable gate arrays 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.

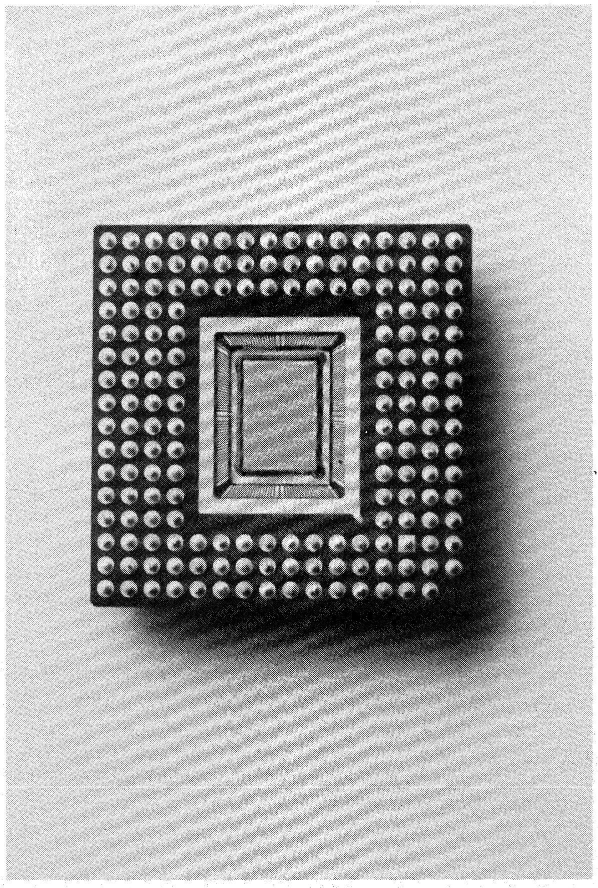
2. Dataquest Inc., *Gate Arrays—Product Analysis,* page 3, ASIC and Standard Logic Semiconductors, 1987.

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

The Programmable Gate Array Company



XILINX

SECTION 2

Product Specifications

XILINX

Product Specifications

XC3000
Logic Cell™ Array Family

Product Specification

FEATURES

- High Performance—50, 70 and 100 MHz Toggle Rates
- Second Generation User-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 three-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 I/O Blocks, a core array of Logic Blocks and resources for interconnection. The general structure of a Logic Cell Array 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 optimizaton, and to compile the data pattern which represents the configuration program.

The Logic Cell Array's 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 eight-pin DIP.

Basic Array	Logic Capacity (usable gates)	Configurable Logic Blocks	User I/Os	Program Data (bits)
XC3020	2000	64	64	14779
XC3030	3000	100	80	22176
XC3042	4200	144	96	30784
XC3064	6400	224	120	46064
XC3090	9000	320	144	64160

The XC3000 Logic Cell Arrays are an enhanced family of Programmable Gate Arrays, which 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 transistors. These functions of the Logic Cell Array 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 Logic Cell Array 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 bit-stream 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 full 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

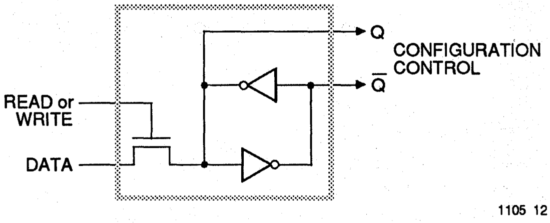


Figure 2. A static configuration memory cell is loaded with one bit of configuration program and controls one program selection in the Logic Cell Array.

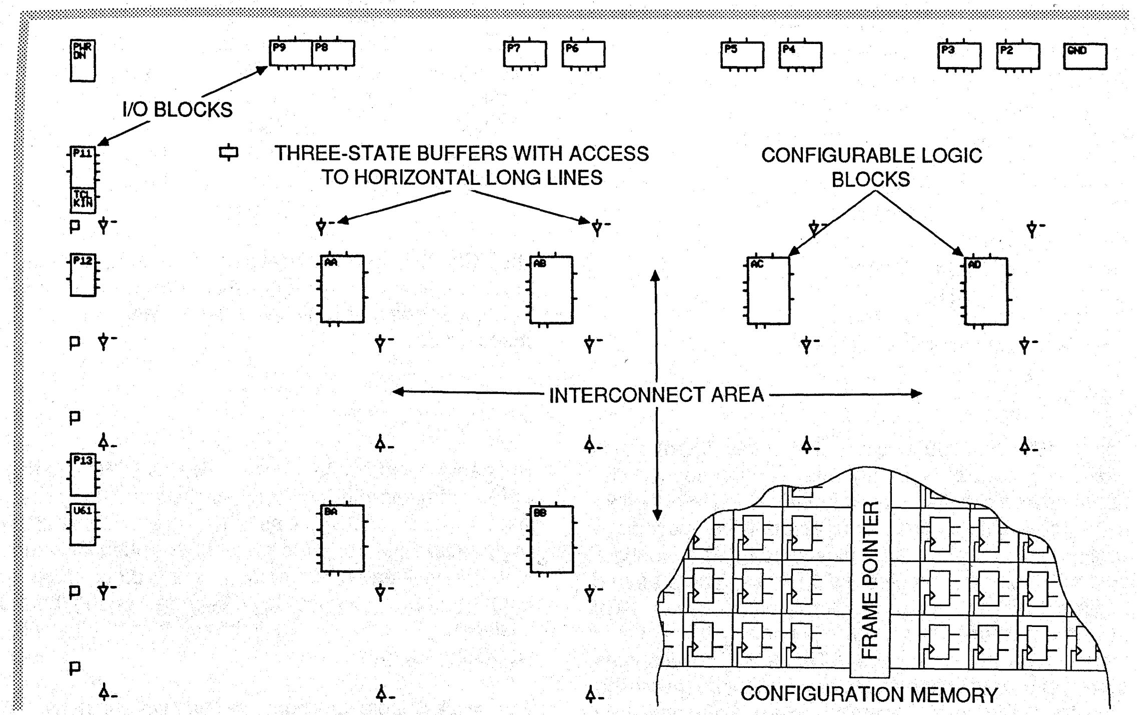


Figure 1. The structure of the Logic Cell Array consists of a perimeter of programmable I/O blocks, a core of configurable logic blocks and their interconnect resources. These are all controlled by the distributed array of configuration program memory cells.

cell. Due to the structure of the configuration memory cells, they are not affected by extreme power supply excursions or very high levels of alpha particle radiation. In reliability testing no soft errors have been observed, even in the presence of very high doses of alpha radiation.

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

I/O BLOCK

Each user-configurable I/O Block (IOB), shown in Figure 3, provides an interface between the external package pin of the device and the internal user logic. Each I/O Block includes both registered and direct input paths. Each IOB provides a programmable three-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.

The input buffer portion of each I/O Block provides thresh-

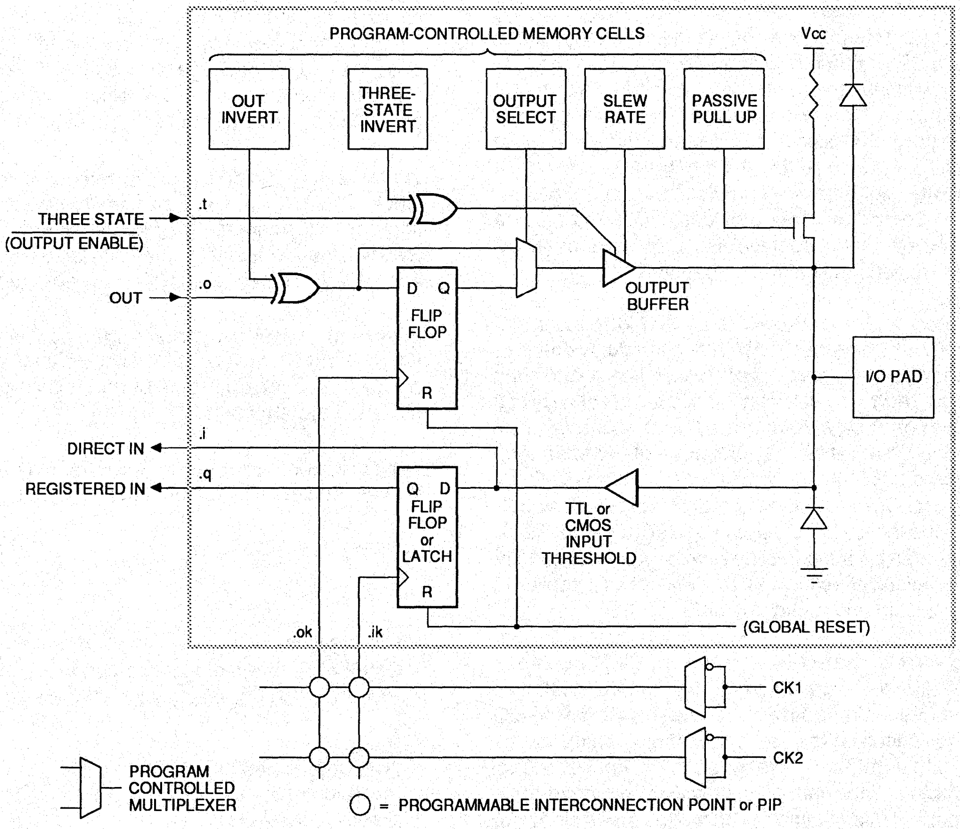


Figure 3. The Input/Output Block includes input and output storage elements and I/O options selected by configuration memory cells.
A choice of two clocks is available on each die edge.
The polarity of each clock line (not each flip-flop or latch) is programmable. A clock line that triggers the flip-flop 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.

old detection to translate external signals applied to the package pin to internal logic levels. The global input-buffer threshold of the I/O Blocks 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 a positive edge-triggered "D" flip-flop or a low level-transparent latch. The sense of the clock can be inverted (negative edge/high transparent) as long as all IOBs on the same clock net use the same clock sense. Clock/load signals (I/O Block pins *.ik* and *.ok*) can be selected from either of two die edge metal lines. I/O storage elements are reset during configuration or by the active low chip $\overline{\text{RESET}}$ input. Both direct input [from I/O Block pin *.i*] and registered input [from I/O Block 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 I/O Block 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 I/O Block and logic block flip-flops are about 3 nanoseconds. This short delay provides good performance under asynchronous clock and data conditions. Short loop delays minimize the probability of a metastable condition which can result from assertion of the clock during data transitions. Because of the short loop delay characteristic in the Logic Cell Array, the I/O Block flip-flops can be used 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.

Output buffers of the I/O Blocks provide CMOS-compatible 4 mA source-or-sink drive for high fan-out CMOS or TTL compatible signal levels. The network driving I/O Block pin .o becomes the registered or direct data source for the output buffer. The three-state control signal [I/O Block pin *.t*] can control output activity. An open-drain type output may be obtained by using the same signal for driving the output and three-state signal nets so that the buffer output is enabled only for a LOW.

Configuration program bits for each I/O Block control features such as optional output register, logical signal inversion, and three-state and slew rate control of the output.

The program-controlled memory cells of Figure 3 control the following options:

- Logical **inversion of the output** is controlled by one configuration program bit per I/O Block.

- Logical **three-state control** of each I/O Block output buffer is determined by the states of configuration program bits which turn the buffer on, or off, or select the output buffer three-state control interconnection [I/O Block pin *.t*]. When this I/O Block output control signal is HIGH, a logic "1", the buffer is **disabled** and the package pin is high impedance. When this I/O block output control signal is LOW, a logic "0", the buffer is **enabled** and the package pin is active. Inversion of the buffer three-state control logic sense (output enable) is controlled by an additional configuration program bit.

- **Direct or registered output** is selectable for each I/O block. The register uses a positive-edge, clocked flip-flop. The clock source may be supplied [I/O Block 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

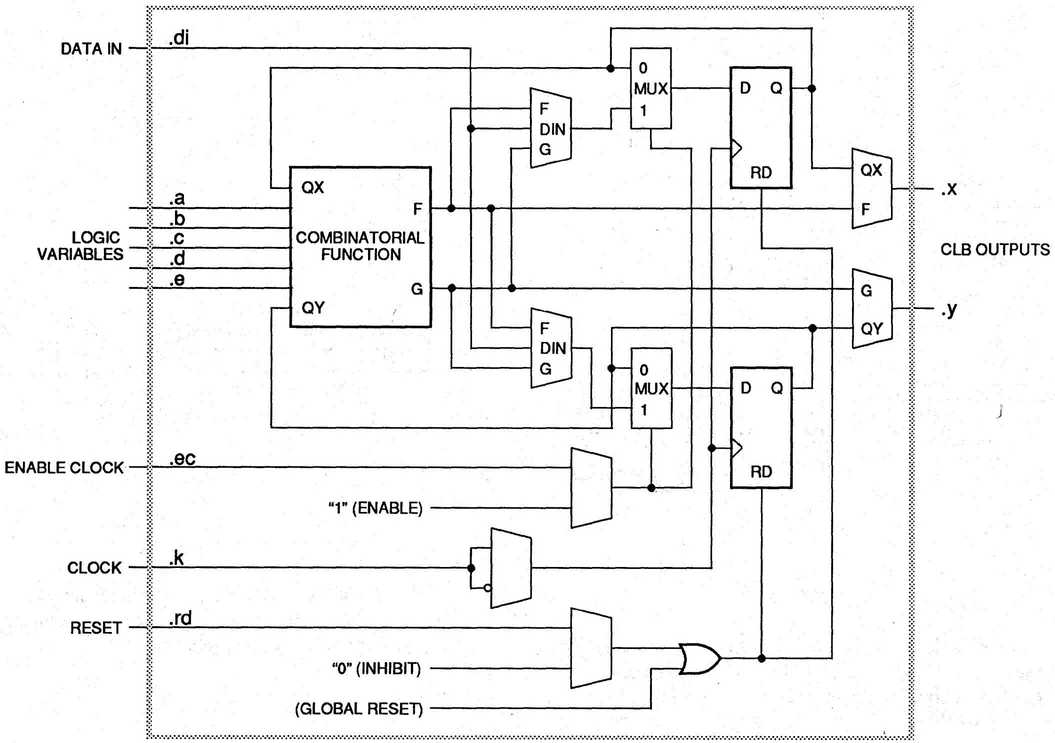
- Outputs
  - Direct/registered
  - Inverted/not
  - Three-state/on/off
  - Full speed/slew limited
  - Three-state/output enable (inverse)

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 I/O Blocks. 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.

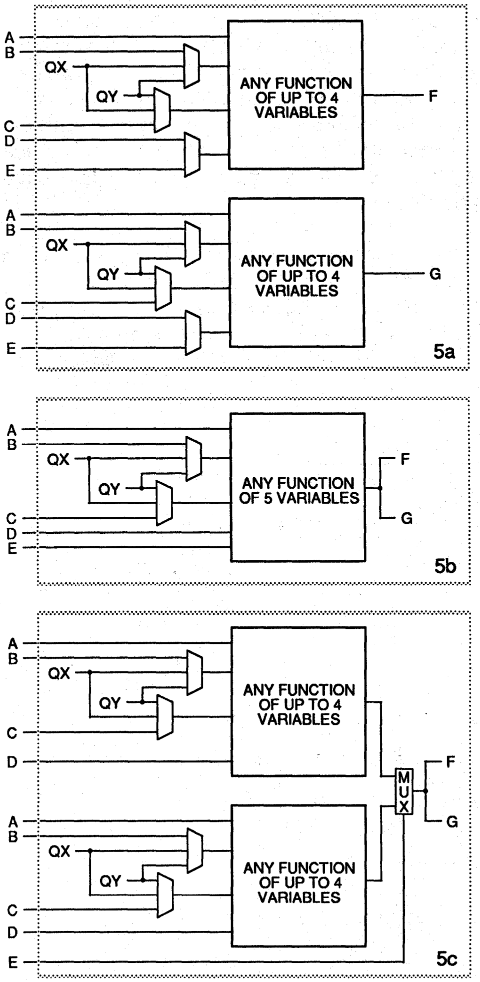
Data input for either flip-flop within a CLB is supplied from the function F or G outputs of the combinatorial logic, or the block input, data-in [*.di*]. Both flip-flops in each CLB share the asynchronous reset [*.rd*] which, when enabled and HIGH, is dominant over clocked inputs. All flip-flops are



1105 02

Figure 4. Each Configurable Logic Block 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


A
B
QX
QY
C
D
E
ANY FUNCTION OF UP TO 4 VARIABLES
F
G
5a
ANY FUNCTION OF 5 VARIABLES
5b
MUX
5c

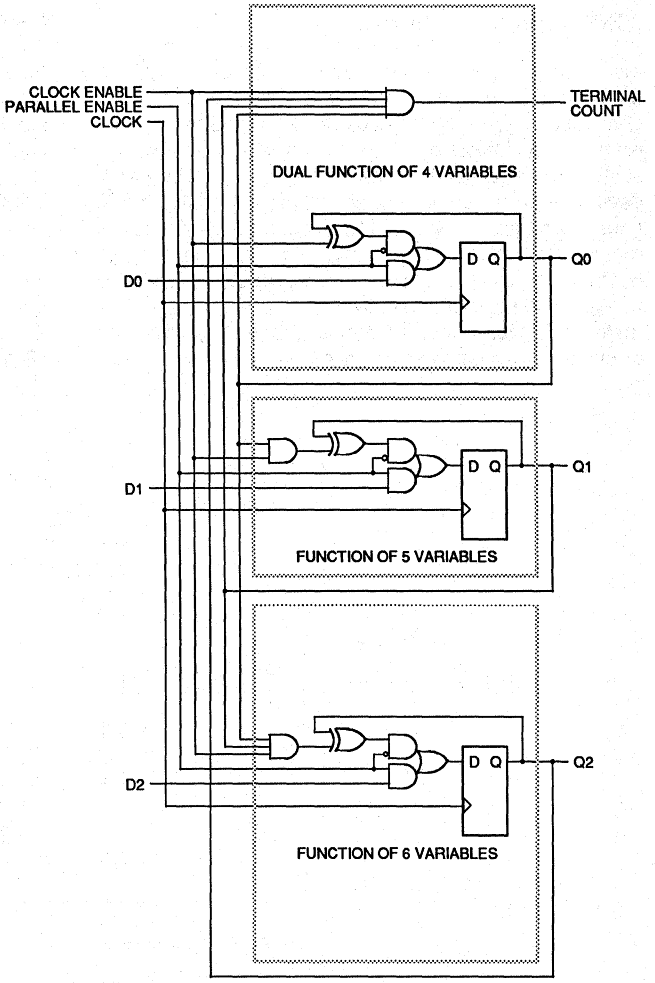


Figure 5

5a. Combinatorial Logic Option 1 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 2 generates any function of five variables: A, D, E and and two choices out of B, C, Qx, Qy.

5c. Combinatorial Logic Option 3 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. The C8BCP 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, 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 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 logical 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 I/O Blocks 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 I/O Blocks. 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.

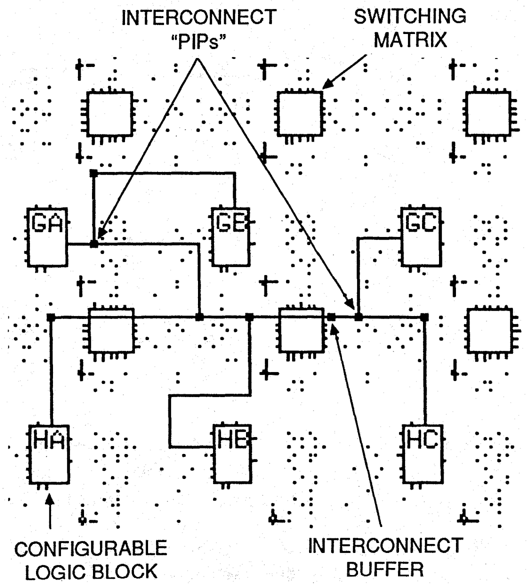


Figure 7. An XACT view of routing resources used to form a typical interconnection network from CLB GA.

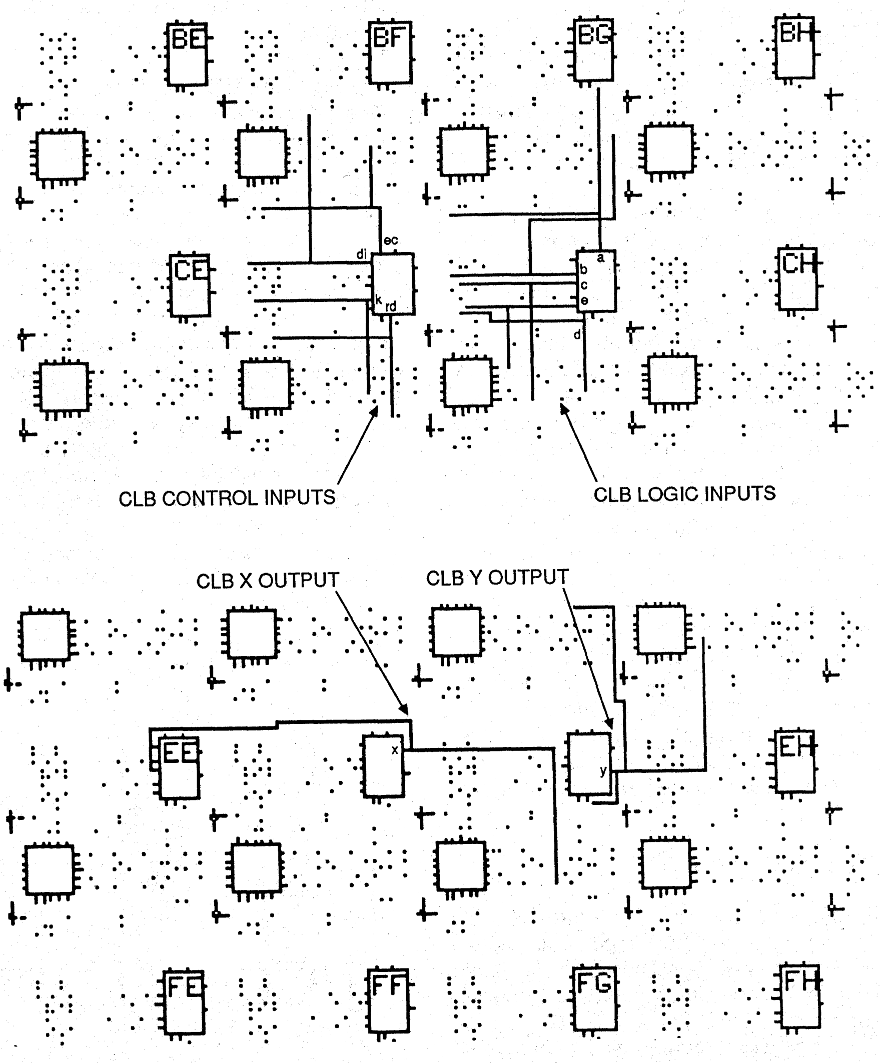


Figure 8. The Xilinx XACT Development System view of the locations of interconnect access, CLB control inputs, logic inputs and outputs. The dot pattern represents the available programmable interconnection points (PIPs).

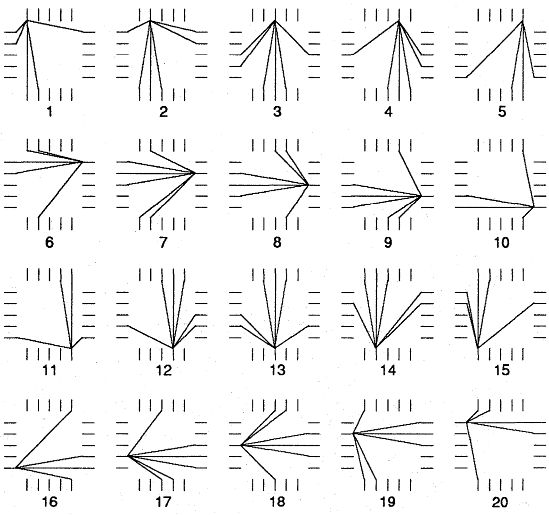
Some of the interconnect "PIPs" are directional. This is indicated on the XACT design editor status line:

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 matricies, 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 Configurable Logic Block, 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



1105 13

Figure 10. Switch matrix interconnection options for each "pin." Switch matrices on the edges are different. Use "Show Matrix" menu option in XACT

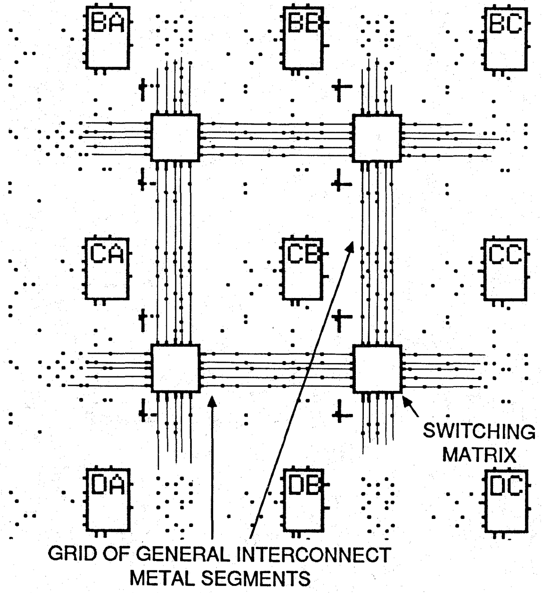


Figure 9. Logic Cell Array general-purpose interconnect is composed of a grid of metal segments which may be interconnected through switch matrices to form networks for CLB and I/O block inputs and outputs.

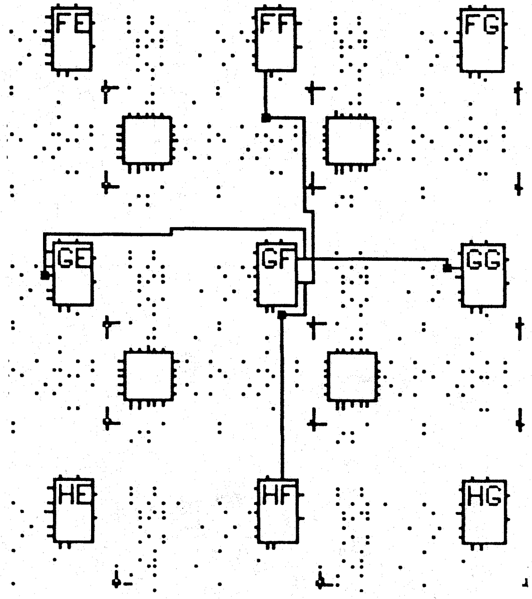


Figure 11. The .x and .y outputs of each CLB have single contact, direct access to inputs of adjacent CLBs.

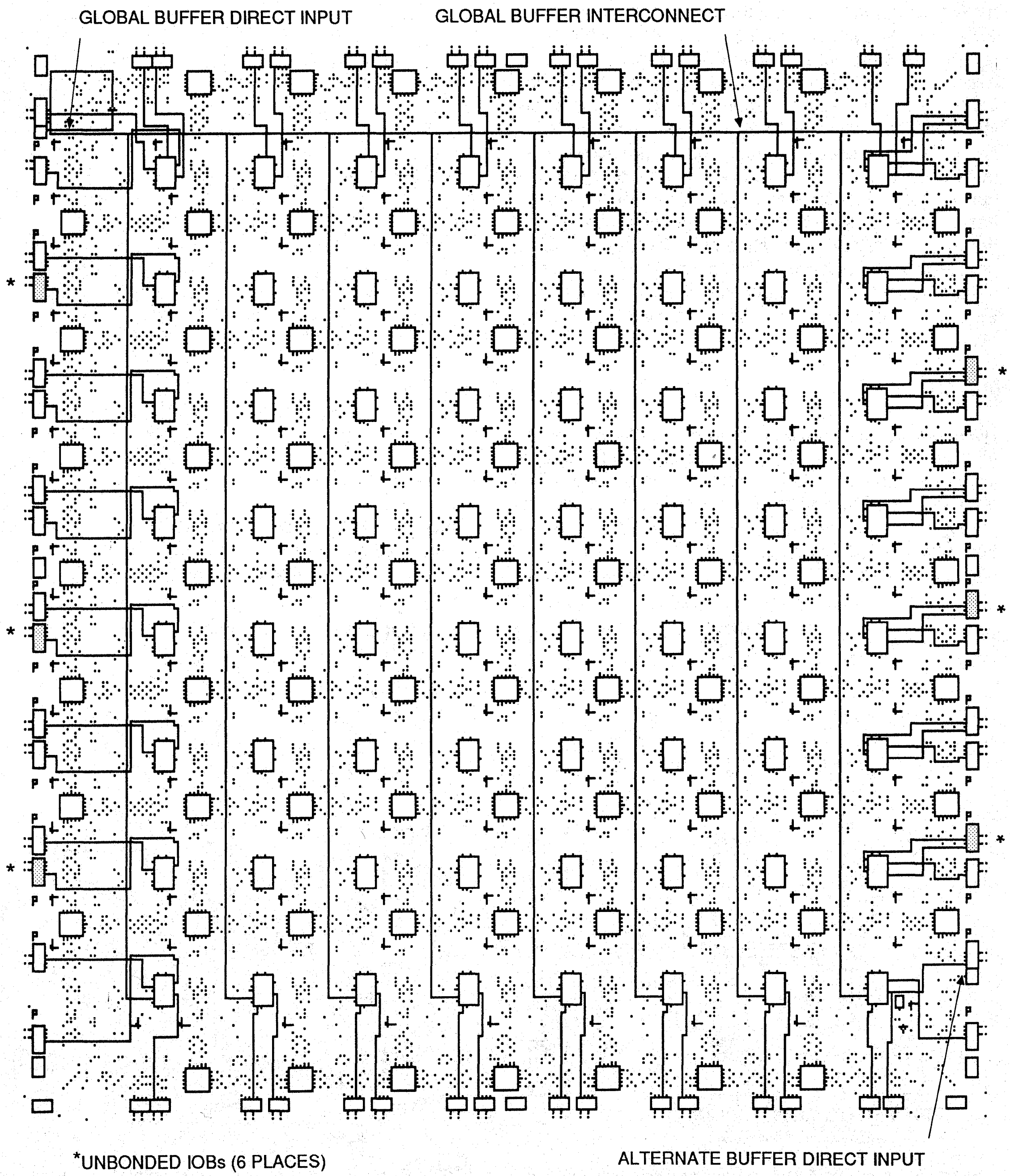

GLOBAL BUFFER DIRECT INPUT
GLOBAL BUFFER INTERCONNECT
*
*
*
*
*
*
*UNBONDED IOBs (6 PLACES)
ALTERNATE BUFFER DIRECT INPUT

Figure 12. X3020 die edge I/O blocks are provided with direct access to adjacent CLBs.

the block immediately above and the *.a* input of the block below. Direct interconnect should be used to maximize the speed of high performance portions of logic. Where logic blocks are adjacent to I/O Blocks, direct connect is provided alternately to the I/O Block 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 I/O Blocks with CLBs are shown in Figure 12.

Long Lines

The long lines bypass the switch matrices and are intended primarily for signals which 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. An additional two 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 connectible half-length lines. On the XC3020 only the outer long lines are.

Long lines can be driven by a logic block or I/O block 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 Logic Cell Array 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 I/O Block 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 three-state buffers is located adjacent to each configurable logic block. These buffers allow logic to drive the horizontal long lines. Logical operation of the three-state buffer controls allows them to implement wide multiplexing functions. Any three-state buffer input can be selected as drive for the horizontal long line bus by applying a low logic level on its three-state control line. See Figure 15a. The user is required to avoid contention which can result from multiple drivers with opposing logic

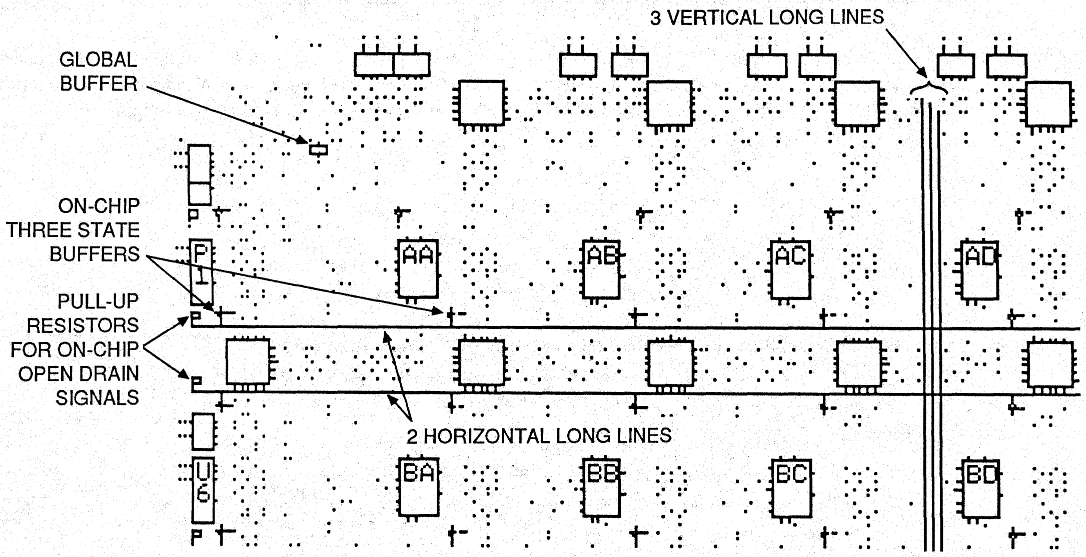


Figure 13. Horizontal and vertical long lines provide high fan-out, low-skew signal distribution in each row and column. The global buffer in the upper left die corner drives a common line throughout the LCA.

levels. Control of the three-state input by the same signal that drives the buffer input, creates an 'open drain' wired-AND function. A logical HIGH on both buffer inputs creates a high impedance which represents no contention. A logical 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 three-state control lines, these buffers form multiplexers (three-state buses). In this case care must be used to prevent contention through multiple active

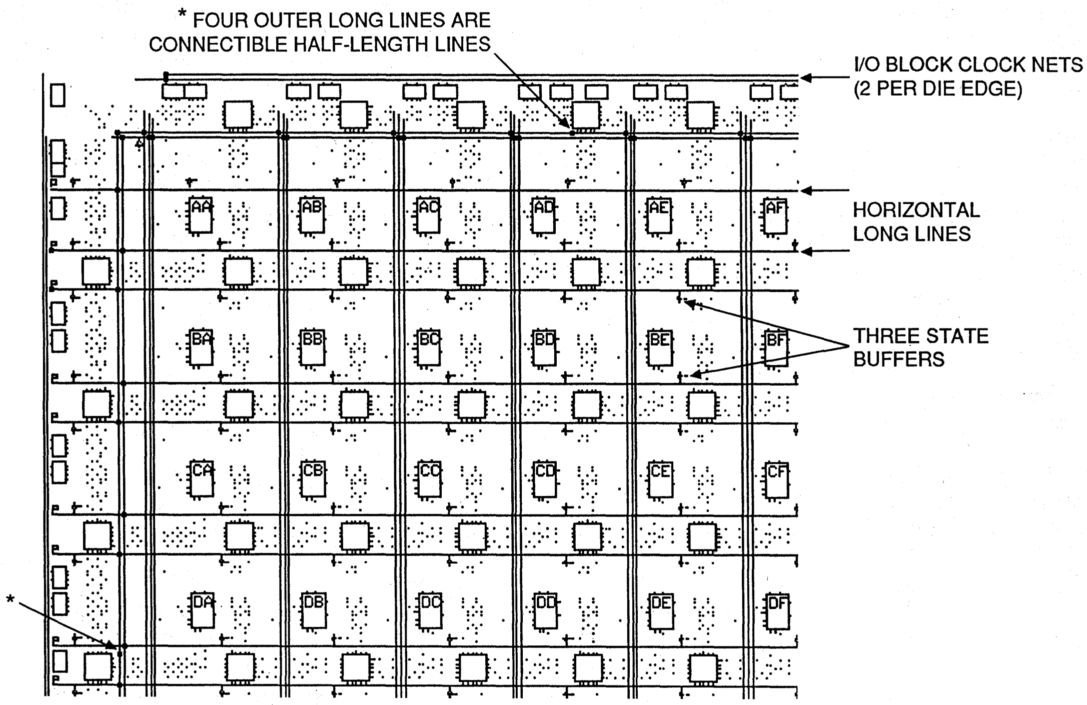

* FOUR OUTER LONG LINES ARE CONNECTIBLE HALF-LENGTH LINES
I/O BLOCK CLOCK NETS (2 PER DIE EDGE)
HORIZONTAL LONG LINES
THREE STATE BUFFERS
AA
AB
AC
AD
AE
AF
BA
BB
BC
BD
BE
BF
CA
CB
CC
CD
CE
CF
DA
DB
DC
DD
DE
DF
*

Figure 14. Programmable interconnection of long lines 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.

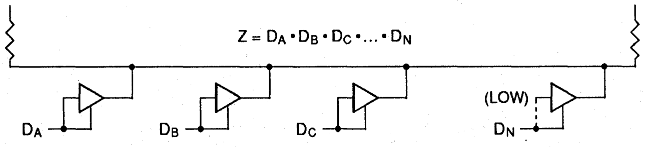
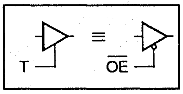
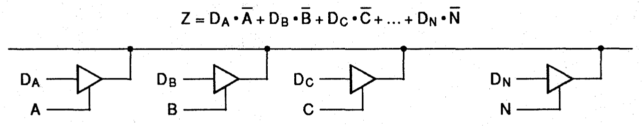


Figure 15a. Three-state buffers implement a Wired-AND function. When all the buffer three state lines are HIGH, (high impedance), the pull-up resistor(s) provide the HIGH output. The buffer inputs are driven by the control signals or a LOW.




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

1105 04

Figure 15b. Three-state buffers implement a Multiplexer where the selection is accomplished by the buffer three-state signal.

buffers of conflicting levels on a common line. Figure 16 shows three 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 I/O Blocks 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 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-half the specified CLB toggle frequency. Use at frequen-

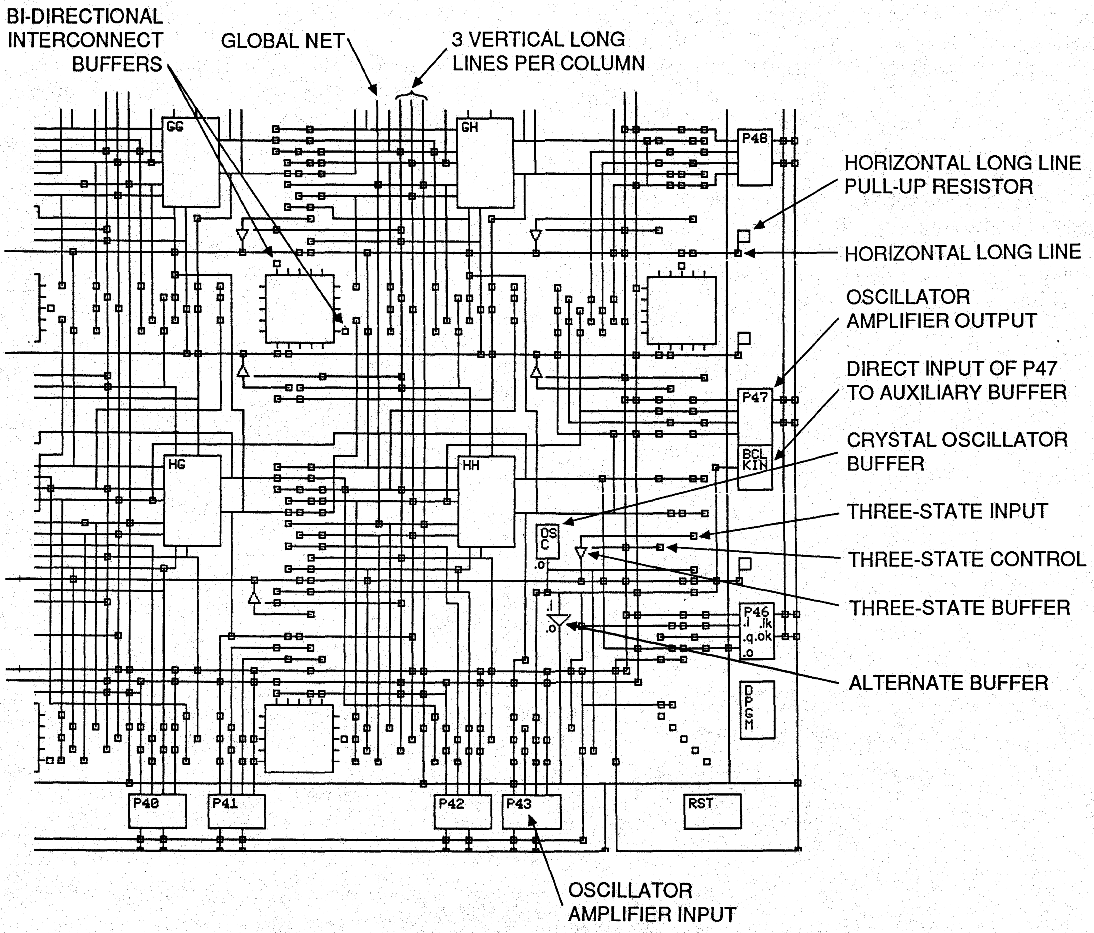

BI-DIRECTIONAL INTERCONNECT BUFFERS
GLOBAL NET
3 VERTICAL LONG LINES PER COLUMN
GG
GH
P48
HORIZONTAL LONG LINE PULL-UP RESISTOR
HORIZONTAL LONG LINE
OSCILLATOR AMPLIFIER OUTPUT
DIRECT INPUT OF P47 TO AUXILIARY BUFFER
P47
BCL KIN
CRYSTAL OSCILLATOR BUFFER
HG
HH
THREE-STATE INPUT
OSC
.O
THREE-STATE CONTROL
THREE-STATE BUFFER
.i
.o
P46
.i .lk
.q.ok
.o
ALTERNATE BUFFER
DPGM
P40
P41
P42
P43
RST
OSCILLATOR AMPLIFIER INPUT

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

cies 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 I/O Blocks and their package pins are available for general user I/O.

PROGRAMMING

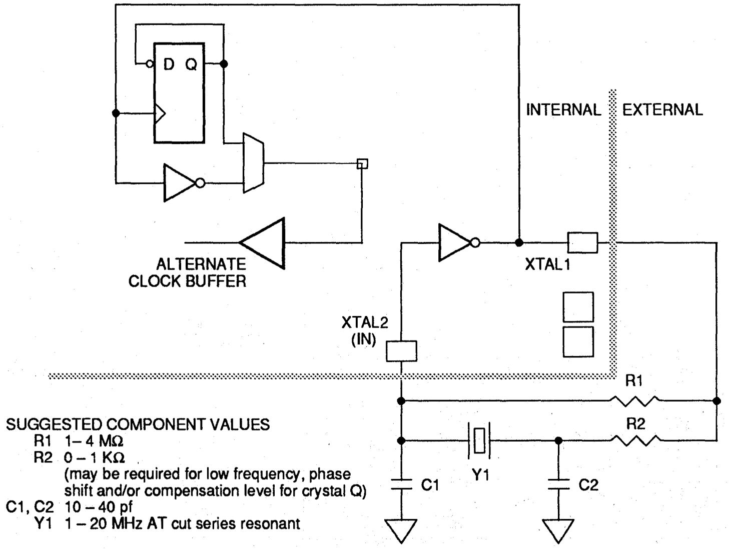
Initialization Phase

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 (2.5 to 3 Volts), 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 over the range of 0.5 to 1.5 MHz. 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 becomes the source of 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) very slow. Figure 18 shows the state sequences. At the end of Initiali-

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


D Q
INTERNAL
EXTERNAL
ALTERNATE
CLOCK BUFFER
XTAL1
XTAL2
(IN)
R1
R2
SUGGESTED COMPONENT VALUES
R1 1– 4 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
C1
Y1
C2

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

1105 14

Figure 17. 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.

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

If a configuration has begun, a re-assertion of RESET for a minimum of three internal timer cycles will be recognized and the LCA will initiate an abort, returning to the Clear state to clear the partially loaded configuration memory words. The LCA will then re-sample 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/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/PROG output signal. DONE/PROG may also be programmed to be an open drain or include a pull-up resistor to accommodate wired ANDing. The High During Configuration (HDC) and Low During Configuration (LDC) are two user I/O pins which are driven active when an LCA is in its Initialization, Clear or Configure states. They and DONE/PROG provide signals for control of external logic signals such as reset, bus enable or PROM enable during configuration. For parallel Master configuration modes these signals provide PROM enable control and allow the data pins to be shared with user logic signals.

User I/O inputs can be programmed to be either TTL or CMOS compatible thresholds. At power-up, all inputs have TTL thresholds and can change to CMOS thresholds

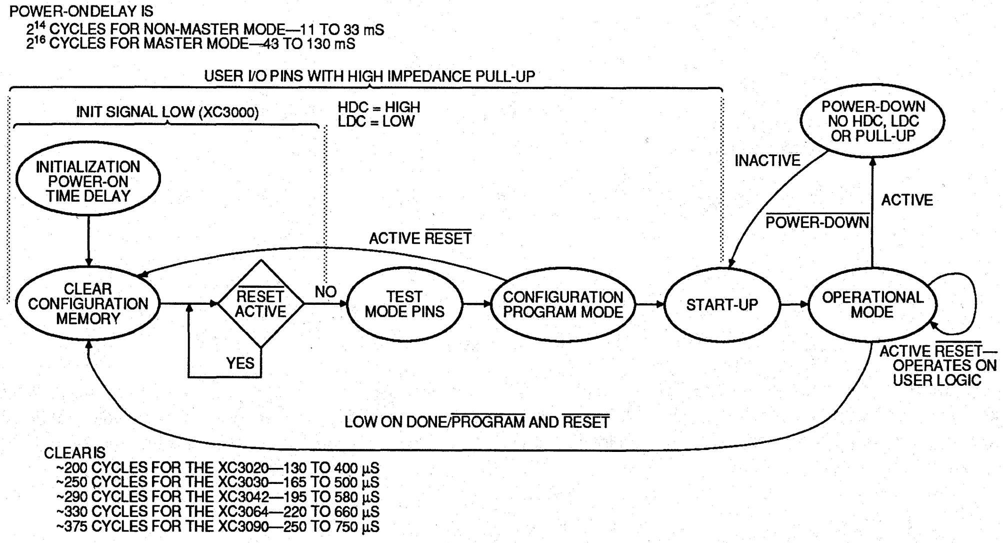

POWER-ONDELAY IS
2^{14} CYCLES FOR NON-MASTER MODE—11 TO 33 mS
2^{16} CYCLES FOR MASTER MODE—43 TO 130 mS
USER I/O PINS WITH HIGH IMPEDANCE PULL-UP
INIT SIGNAL LOW (XC3000)
HDC = HIGH
LDC = LOW
POWER-DOWN NO HDC, LDC OR PULL-UP
INACTIVE
ACTIVE
POWER-DOWN
INITIALIZATION POWER-ON TIME DELAY
ACTIVE RESET
CLEAR CONFIGURATION MEMORY
RESET ACTIVE
NO
YES
TEST MODE PINS
CONFIGURATION PROGRAM MODE
START-UP
OPERATIONAL MODE
ACTIVE RESET— OPERATES ON USER LOGIC
LOW ON DONE/PROGRAM AND RESET
CLEARIS
~200 CYCLES FOR THE XC3020—130 TO 400 µS
~250 CYCLES FOR THE XC3030—165 TO 500 µS
~290 CYCLES FOR THE XC3042—195 TO 580 µS
~330 CYCLES FOR THE XC3064—220 TO 660 µS
~375 CYCLES FOR THE XC3090—250 TO 750 µS
1105 15

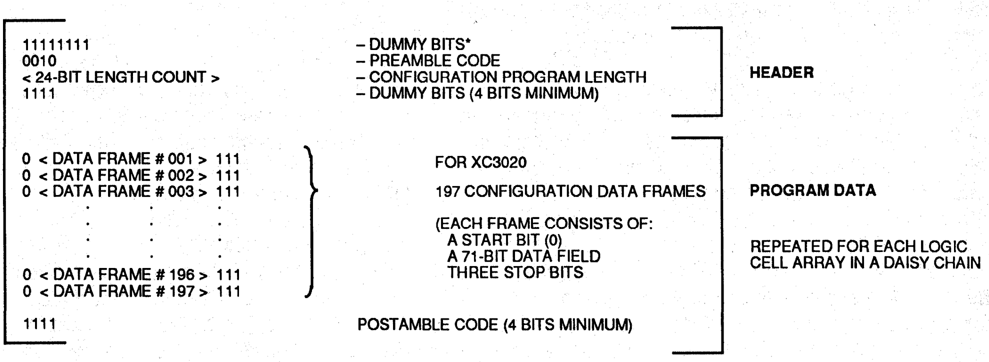
Figure 18. A state diagram of the configuration process for power-up and reprogram.

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 se-



```
11111111                          - DUMMY BITS*
0010                              - PREAMBLE CODE
< 24-BIT LENGTH COUNT >           - CONFIGURATION PROGRAM LENGTH          HEADER
1111                              - DUMMY BITS (4 BITS MINIMUM)

0 < DATA FRAME # 001 > 111           FOR XC3020
0 < DATA FRAME # 002 > 111
0 < DATA FRAME # 003 > 111           197 CONFIGURATION DATA FRAMES        PROGRAM DATA
      .       .       .
      .       .       .              (EACH FRAME CONSISTS OF:
      .       .       .                A START BIT (0)                    REPEATED FOR EACH LOGIC
0 < DATA FRAME # 196 > 111             A 71-BIT DATA FIELD                CELL ARRAY IN A DAISY CHAIN
0 < DATA FRAME # 197 > 111             THREE STOP BITS

1111                              POSTAMBLE CODE (4 BITS MINIMUM)
```

*THE LCA DEVICES REQUIRE 4 DUMMY BITS MIN., XACT 2.10 GENERATES 8 DUMMY BITS

1105 05

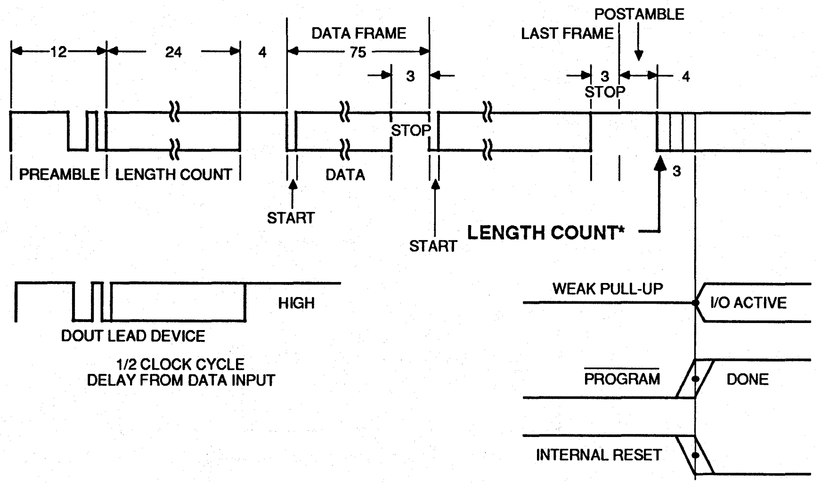
Device	**XC3020**	**XC3030**	**XC3042**	**XC3064**	**XC3090**
Gates	2000	3000	4200	6400	9000
CLBs Row X Col	64 (8 X 8)	100 (10 X 10)	144 (12 X 12)	224 (16 X 14)	320 (20 X 16)
IOBs	64	80	96	120	144
Flip-flops	256	360	480	688	928
Bits per frame (w/ 1 start 3 stop)	75	92	108	140	172
Frames	197	241	285	329	373
Program Data = Bits * Frames + 4 (excludes header)	14779	22176	30784	46064	64160
PROM size (bits) = Program Data + 40 bit Headers	14819	22216	30824	46104	64200

Figure 19. The internal Configuration Data Structure for an LCA 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 ≤ K ≤ 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.

lection 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 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 indeterminant levels which 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 mA of Icc are acceptable.

The configuration bit-stream begins with HIGH preamble bits, a four-bit preamble code and a 24-bit length count. When configuration is initiated, a counter in the LCA is set to 0 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 BOUNDRY. 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 06

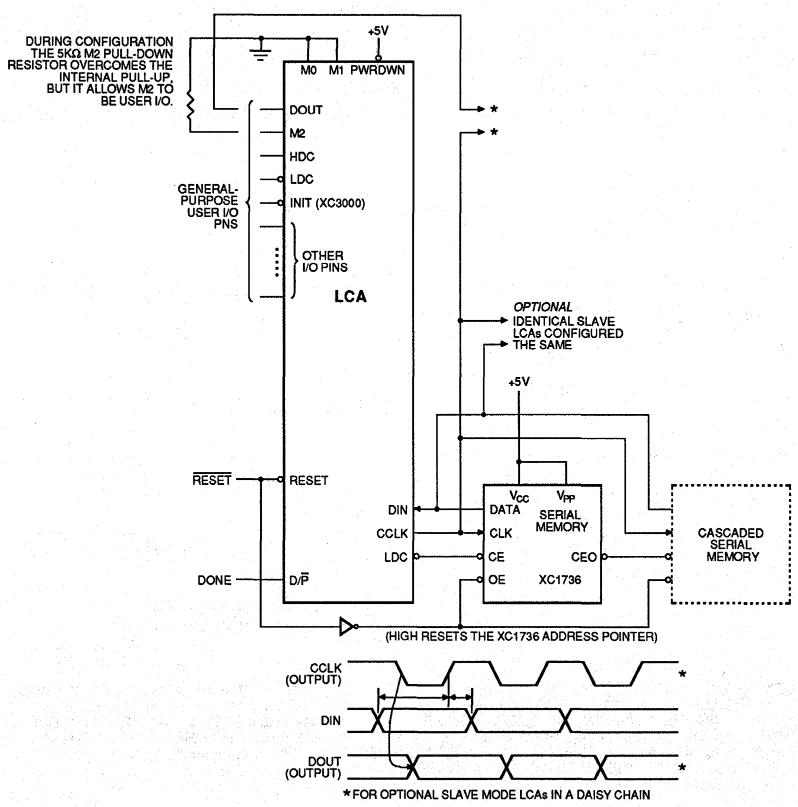
Figure 20. Configuration and start-up of one or more LCAs.

1105 06

Two user programmable pins are defined in the unconfigured Logic Cell array. High During Configuration (HDC) and Low During Configuration (LDC) as well as DONE/$\overline{\text{PROG}}$ may be used as external control signals during configuration. In Master mode configurations it is convenient to use LDC as an active-low EPROM Chip Enable. After the last configuration data-bit is loaded and the length count compares, the user I/O pins become active. Options 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/$\overline{\text{PROG}}$ output can be AND-tied with multiple Logic Cell Arrays 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

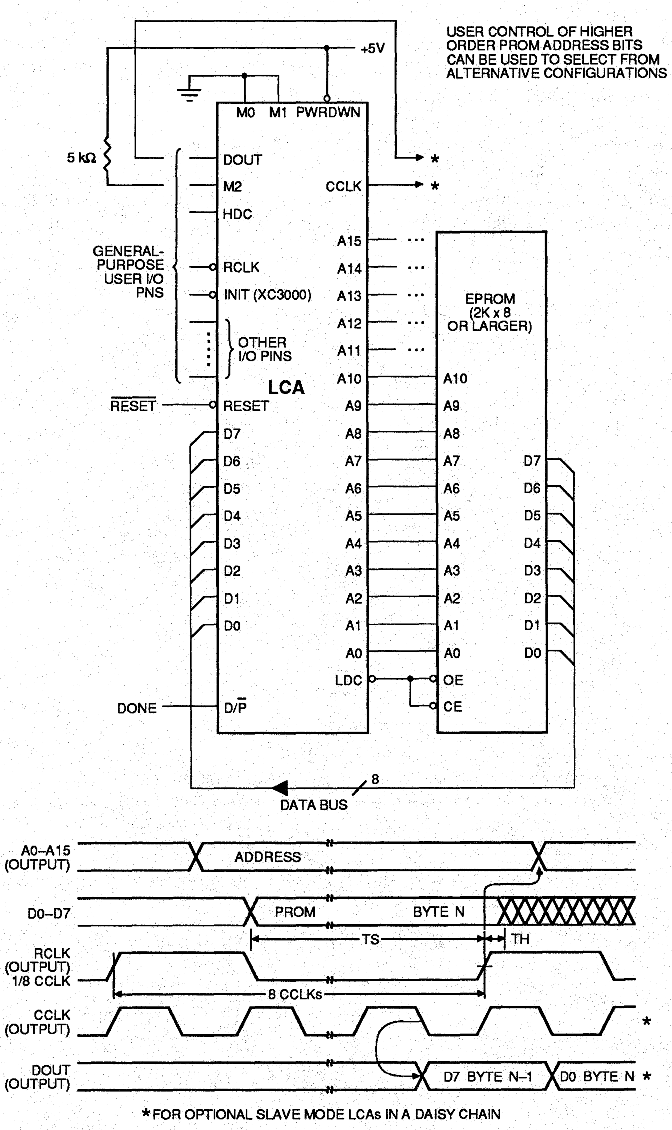
In Master mode, the Logic Cell Array automatically loads configuration data from an external memory device. There are three Master modes which use the internal timing source to supply the configuration clock (CCLK) to time the incoming data. Serial Master mode uses serial configuration data supplied 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-



DURING CONFIGURATION THE 5KΩ M2 PULL-DOWN RESISTOR OVERCOMES THE INTERNAL PULL-UP, BUT IT ALLOWS M2 TO BE USER I/O.
+5V
M0 M1 PWRDWN
DOUT
M2
HDC
LDC
INIT (XC3000)
GENERAL-PURPOSE USER I/O PNS
OTHER I/O PINS
LCA
*
*
OPTIONAL IDENTICAL SLAVE LCAs CONFIGURED THE SAME
+5V
RESET
RESET
DIN
CCLK
LDC
DONE
D/P̄
VCC VPP
DATA SERIAL MEMORY
CLK
CE CEO
OE XC1736
CASCADED SERIAL MEMORY
(HIGH RESETS THE XC1736 ADDRESS POINTER)
CCLK (OUTPUT)
DIN
DOUT (OUTPUT)
*
*
* FOR OPTIONAL SLAVE MODE LCAs IN A DAISY CHAIN
1105 16

Figure 21. Master Serial Mode. The one-time-programmable XC1736 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 XC1736 data output a $\overline{\text{CCLK}}$ cycle before the LCA I/O 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 Master 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.



1105 17

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

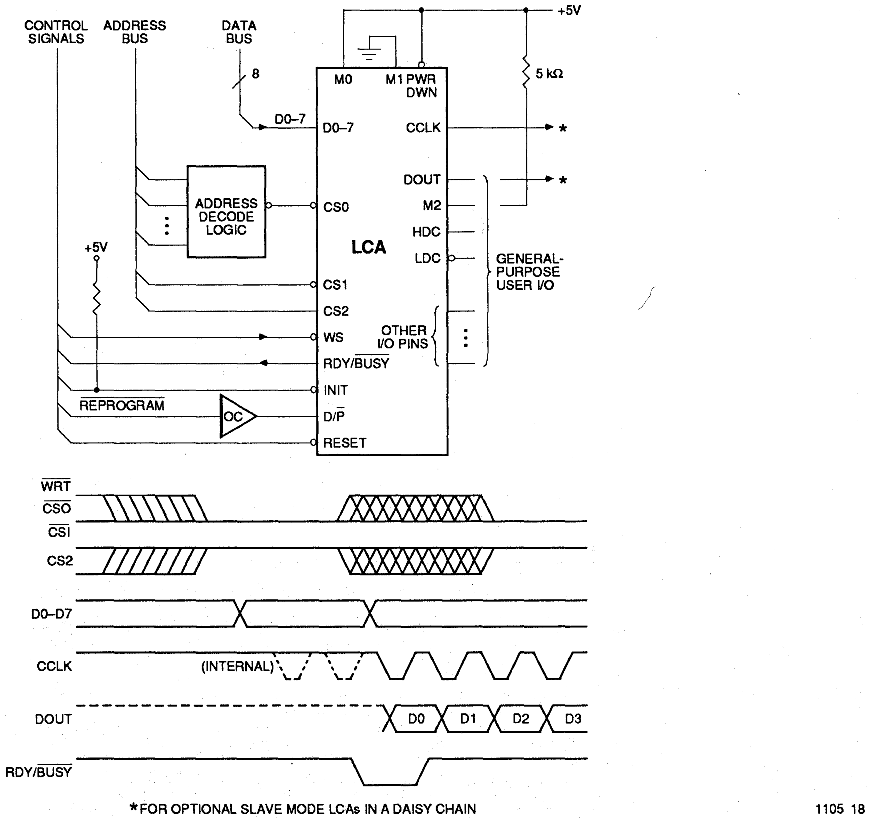
Peripheral mode provides a simplified interface through which the device may be loaded byte-wide, as a processor peripheral. Figure 23 shows the peripheral mode connections. Processor write cycles are decoded from the common assertion of the active low Write Strobe ($\overline{\text{WS}}$), and two active low and one active high Chip Selects ($\overline{\text{CS0}}$, $\overline{\text{CS1}}$, 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/$\overline{\text{BUSY}}$ 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 Xilinx XACT development system is used to create a composite configuration bit stream for selected LCAs



*FOR OPTIONAL SLAVE MODE LCAs IN A DAISY CHAIN

1105 18

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

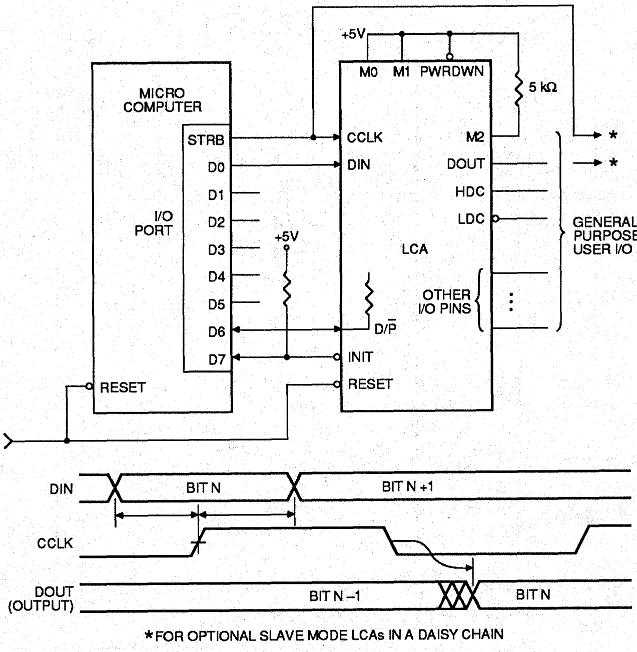
including: a preamble, a length count for the total bit-stream, multiple concatenated data programs and a postamble plus an additional fill bit per device in the serial chain. After loading and passing-on the preamble and length count to a possible daisy-chain, a lead device will load its configuration 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 enable
- 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 bit-stream generation process.



1105 19

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.

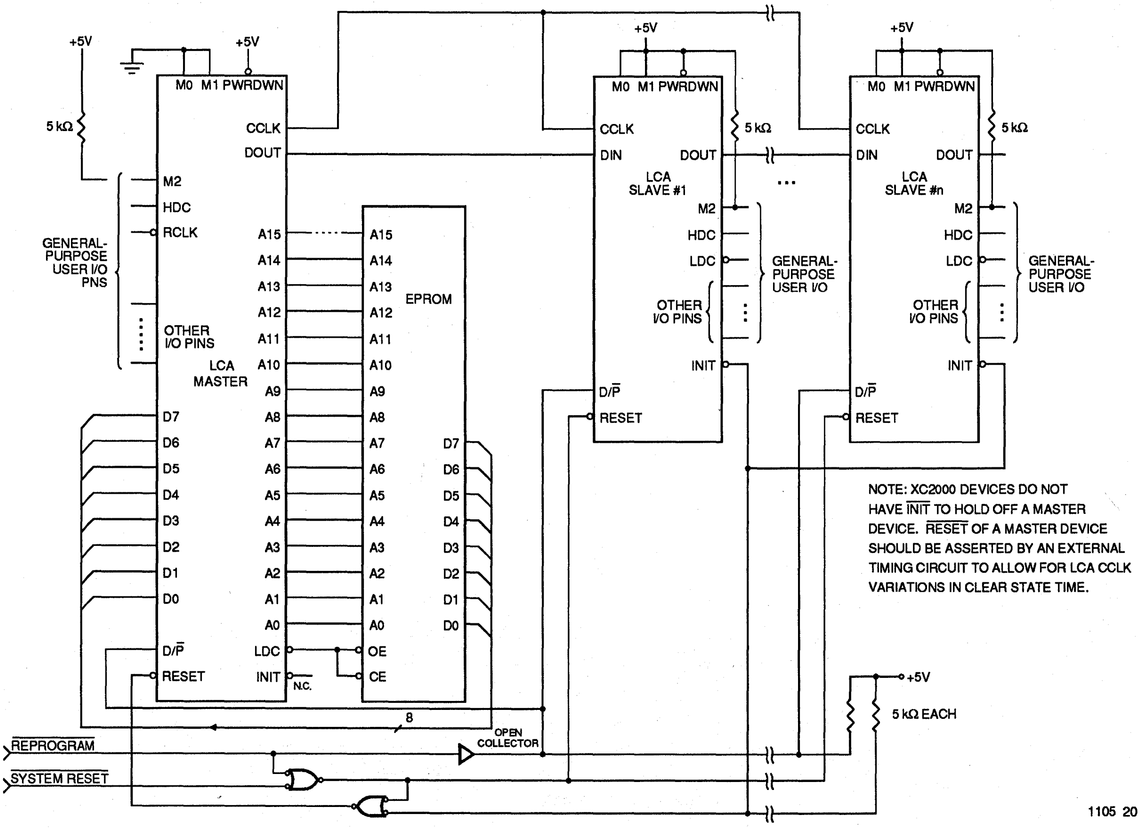
Input Thresholds

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 I/O Block 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 bit-stream 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 bit-stream:

- "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.



NOTE: XC2000 DEVICES DO NOT HAVE $\overline{\text{INIT}}$ TO HOLD OFF A MASTER DEVICE. $\overline{\text{RESET}}$ OF A MASTER DEVICE SHOULD BE ASSERTED BY AN EXTERNAL TIMING CIRCUIT TO ALLOW FOR LCA CCLK VARIATIONS IN CLEAR STATE TIME.

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/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 I/O Block. These data are imbedded into unused configuration bit positions during readback. This state information is used by the Logic Cell Array 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 Logic Cell Array configuration memory can re-written while the device is operating in the user's system. To initiate a re-programming cycle, the dual function package pin DONE/PROG must be given a HIGH to LOW transition. To reduce sensitivity to noise, the input signal is filtered for 2 cycles of the LCA's 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, wire-AND the slave INIT pins and use them to force a RESET on the master (see Figure 25). Reprogram control is often implemented using an external open collector driver which pulls DONE/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/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/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 high performance of the Logic Cell Array 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. The parameter which traditionally describes the overall performance of a gate array is the toggle frequency of a flip-flop. 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 Logic Cell Array 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. Examples of internal worst case timing 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 determing 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 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 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 Figures 28 and 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 bi-directional re-powering 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. Figure 30 illustrates this.

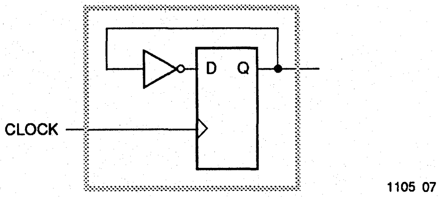
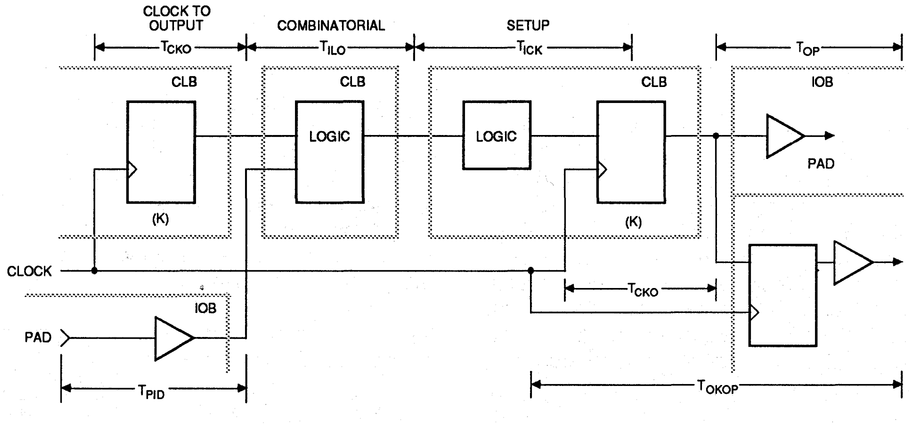


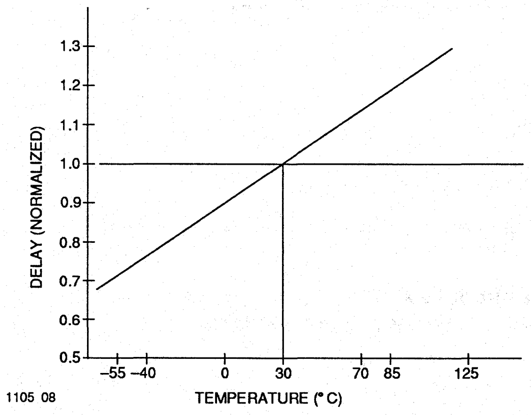
Figure 26. "Toggle" Flip-Flop used to characterize device performance.



1105 21

		Speed Grade	-50		-70		-100		Units
	Description	Symbol	Min	Max	Min	Max	Min	Max	
Logic input to Output	Combinatorial	T_{ILO}		14		9		7	ns
K Clock	To output Logic-input setup Logic-input hold	T_{CKO} T_{ICK} T_{CKI}	 12 0	12	 8 0	8	 7 0	7	ns ns ns
Input/Output	Pad to input (direct)	T_{PID}		9		6		4	ns
	Output to pad (fast)	T_{OP}		15		9		6	ns
	I/O clock to pad (fast)	T_{OKPO}		18		13		10	ns
FF toggle frequency		F_{CLK}		50		70		100	MHz

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.



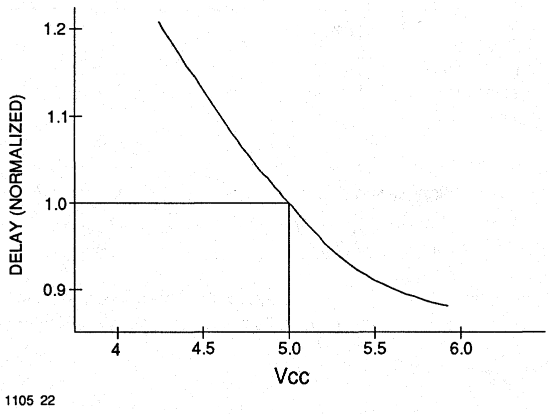


Figure 28. Change in speed performance as a function of temperature, normalized for 30°C.

Figure 29. The speed performance of a CMOS device increases with Vcc within the operating range.

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 Vcc and ground ring surrounding the logic array provides power to the I/O drivers. See Figure 31. 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 all connected and appropriately decoupled. Typically a 0.1 µF capacitor connected near the Vcc and ground pins of 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. 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. More than 32 fast outputs should not be switching in the same direction exactly simultaneously. A few ns of deliberate skew can alleviate this problem of "ground-bounce".

Power Dissipation

The Logic Cell Array exhibits the low power consumption characteristic of CMOS ICs. For any design the user can use Figure 32 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 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 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 Configurable Logic Block output requires about 0.4 mW per MHz of its output frequency.

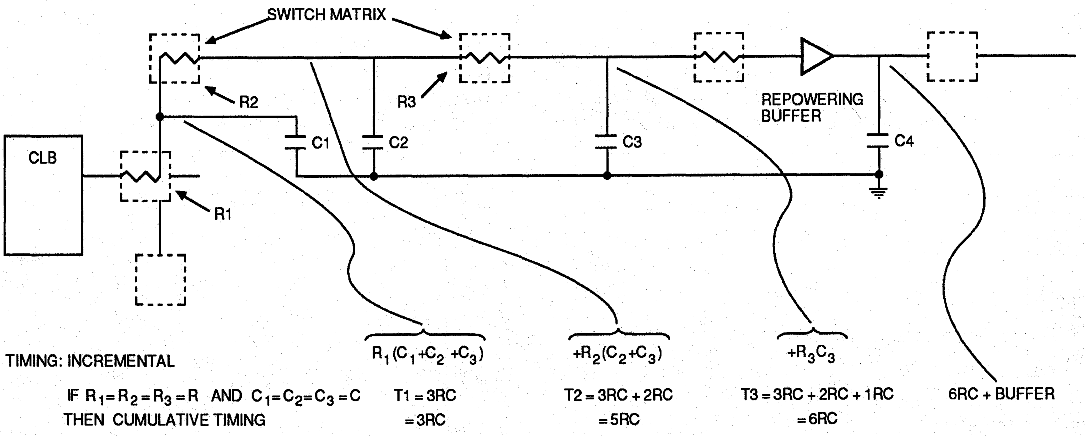


Figure 30. Interconection timing example. Use of the XACT timing calculator or XACT-generated simulation model provide actual worst-case performance information.

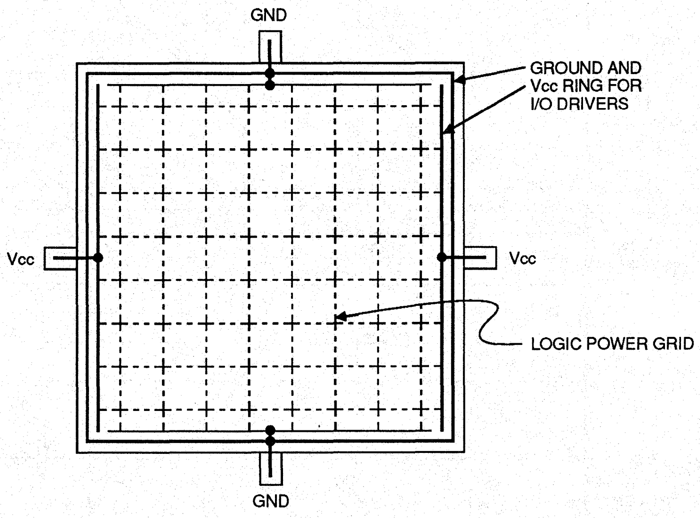
1105 23

Total Power = Vcc • Icco + external (DC + capacitive)
+ 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 volts the required current is typically on the order of 50 nanoamps.

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 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 power pin. This condition can produce invalid power conditions and should be avoided. A large series resistor might be used to limit the current or a bi-polar buffer may be used to isolate the input signal.



1105 24

Figure 31. LCA Power Distribution.

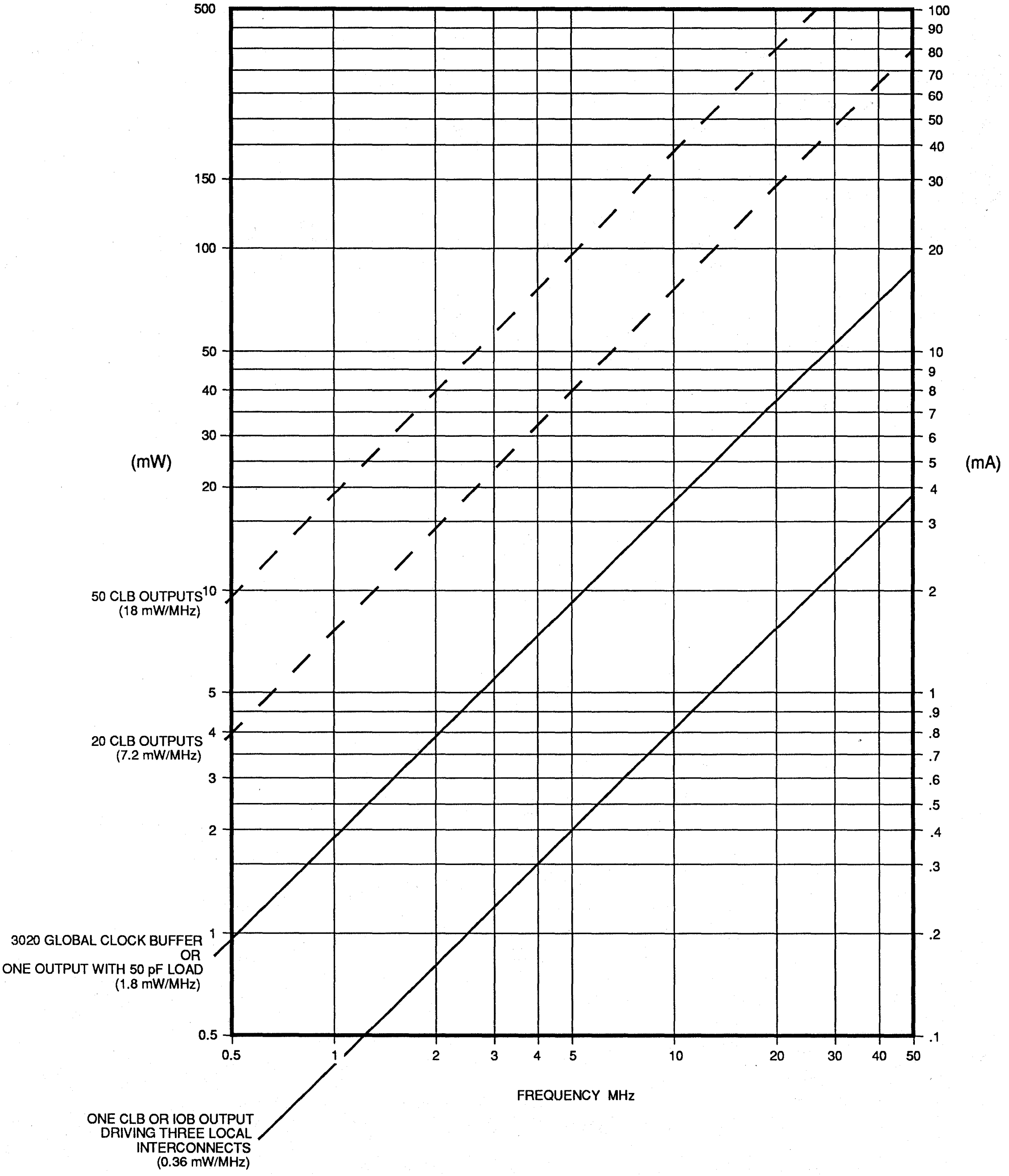


Figure 32. LCA Power Consumption by Element. Total chip power is the sum of Vcc•Icco plus effective internal and external values of frequency dependent capacitive charging currents and duty factor dependent resistive loads.

Component Selection, Ordering Information, Electrical Parameters & Physical Dimensions

		48 PIN		68 PIN		84 PIN		100 PIN		132 PIN		164 PIN	175 PIN	
		PLASTIC DIP	CERAMIC DIP	PLASTIC PLCC	CERAMIC PGA	PLASTIC PLCC	CERAMIC PGA	PLASTIC PQFP	CERAMIC CQFP	PLASTIC PGA	CERAMIC PGA	CERAMIC CQFP	PLASTIC PGA	CERAMIC PGA
		-PD48	-CD48	-PC68	-PG68	-PC84	-PG84	-PQ100	-CQ100	-PP132	-PG132	-CQ164	-PP175	-PG175
XC2064	-33	C	I M	C I	C I M									
	-50	C	I	C I	C I M									
	-70			C I	C I M									
	-100			C	C									
XC2018	-33			C I		C I	C I M B							
	-50			C I		C I	C I M B							
	-70			C I		C I	C I M							
	-100			C		C	C							
XC3020	-50			C I		C I	C I M B	C I	C I M B					
	-70			C I		C I	C I M	C I	C I M					
	-100			C		C	C	C	C					
XC3030	-50			C I		C I	C I M	C I	C I M					
	-70			C I		C I	C I M	C I	C I M					
	-100			C		C	C	C	C					
XC3042	-50					C I	C I M	C I	C I M	C I	C I M B			
	-70					C I	C I M	C I	C I M	C I	C I			
	-100					C	C	C	C	C	C			
XC3064	-50									C I	C I M			
	-70									C I	C I			
	-100									C	C			
XC3090	-50											C I M	C I	C I M B
	-70											C I M	C I	C I M
	-100											C	C	C

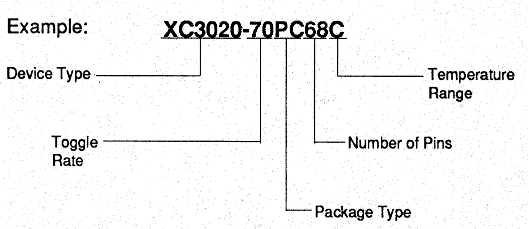
1972 01

XC1736-PD8C	**Plastic 8 Pin, Mini-DIP**	**–40°C to 85°C**
XC1736-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	Military	–55°C to 125°C
B	Military	MIL-STD-883, Class B

ORDERING INFORMATION



COMPATIBLE PACKAGE OPTIONS

A range of Xilinx 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:		
	PLCC 68:	2064-2018-3020-3030
	PLCC 84:	2018-3020-3030-3042
	PGA 84:	2018-3020-3030-3042
	PQFP 100:	3020-3030-3042
	CQFP100:	3020-3030-3042
	PGA 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.

1972

PIN DESCRIPTIONS

1. Permanently Dedicated Pins.

Vcc

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.

$\overline{\text{PWRDWN}}$

A LOW on this CMOS compatible input stops all internal activity to minimize Vcc power, and puts all output buffers in a high impedance state, but configuration is retained. When the $\overline{\text{PWRDWN}}$ pin returns HIGH, the device returns to operation with the same sequence of buffer enable and DONE/$\overline{\text{PROGRAM}}$ as at the completion of configuration. All internal storage elements are reset. If not used, $\overline{\text{PWRDWN}}$ must be tied to Vcc.

$\overline{\text{RESET}}$

This is an active low input which has three functons.

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 $\overline{\text{RESET}}$ are complete, the levels of the "M" lines are sampled and configuration begins.

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

If $\overline{\text{RESET}}$ is asserted after configuration is complete it will provide an asynchronous reset of all IOB and CLB storage elements of the LCA.

CCLK

During configuration, Configuration Clock is an output of an LCA in Master mode or Peripheral mode. LCAs in Slave mode use it as a clock input. During a Readback operation it is a clock input for the configuration data being shifted out.

DONE

The DONE output is configurable as open drain with or without an internal pull-up resistor. At the completion of configuration, the circuitry of the LCA becomes active in a synchronous order, and DONE may be programmed to occur one cycle before or after that.

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

As a Read Trigger, a LOW-to-HIGH input transition, after configuration is complete, will initiate a Readback of configuration and storage element data by CCLK. This operation may be limited to a single request, or be inhibited altogether, by selecting the appropriate readback option when generating the bit stream.

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 to be used, a 5 KΩ resistor should be used to define mode level inputs.

$\overline{\text{RDATA}}$

As an active low Read Data, after configuration is complete, this pin is the output of the readback data.

2. User I/O Pins that can have special functions.

M2

As Mode 2 this input has a passive pullup during configuration. 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 becomes a user programmable I/O pin.

HDC

High During Configuration is held at a HIGH level by the LCA until after configuration. It is available as a control output indicating that configuration is not yet completed. After configuration this pin is a user I/O pin.

$\overline{\text{LDC}}$

Low During Configuration is held at a LOW level by the LCA until after configuration. It is available as a control output indicating that configuration is not yet completed. It is particularly useful in Master mode as a LOW enable for an EPROM. After configuration this pin is a user I/O pin. If used as a LOW EPROM enable, it must 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, which are used in the Peripheral mode to control configuration data entry. The assertion of all 4 generates a write to the internal data buffer. The removal of any assertion, clocks in the D0–D7 data present.

$\overline{\text{RCLK}}$

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 8 pins represent the parallel configuration byte for the parallel Master and Peripheral modes. After configuration is complete they are user programmed I/O pin.

A0–A15

This set of 16 pins present an address output for a configuration EPROM during Master parallel mode. After configuration is complete they are user programmed I/O pin.

DIN

This user I/O pin is used as serial Data input during Slave or Master Serial configuration. This pin is Data 0 input in Master or Peripheral configuration mode.

DOUT

This user I/O pin is used during configuration to output serial configuration data for daisy-chained slaves' Data In.

TCLKIN

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

3. Unrestricted User I/O Pins.

I/O

A pin which may be programmed by the user to be Input and/or Output pin following configuration. Some of these pins present a high impedance pull-up (see next page) or perform other functions before configuration is complete (see above).

XC3000 Family Configuration Pin Assignments

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

AVAILABLE PACKAGES

▨ REPRESENTS A 50KΩ TO 100KΩ PULL-UP

* INIT IS AN OPEN DRAIN OUTPUT DURING CONFIGURATION

(I) REPRESENTS AN INPUT

1105 25

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

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

	XC-3020*		
68 PLCC	XC-3030, XC-3042	84 PLCC	84 PGA
10	$\overline{\text{PWRDN}}$	12	B2
11	TCLKIN-I/O	13	C2
n.c.	**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-$\overline{\text{RDATA}}$	31	J2
26	M0-$\overline{\text{RTRIG}}$	32	L1
27	M2-I/O	33	K2
28	HDC-I/O	34	K3
29	I/O	35	L2
30	$\overline{\text{LDC}}$-I/O	36	L3
31	I/O	37	K4
n.c.	**I/O***	38	L4
32	I/O	39	J5
33	I/O	40	K5
n.c.	**I/O***	41	L5
34	$\overline{\text{INIT}}$-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
n.c.	**I/O***	50	L9
n.c.	**I/O***	51	L10
42	I/O	52	K9
43	XTL2(IN)-I/O	53	L11
44	$\overline{\text{RESET}}$	54	K10
45	DONE-$\overline{\text{PG}}$	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	$\overline{\text{CS0}}$-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	$\overline{\text{CS1}}$-I/O	66	E11
55	D2-I/O	67	E10
—	I/O	68	E9
n.c.	**I/O***	69	D11
56	D1-I/O	70	D10
57	RDY/$\overline{\text{BUSY}}$-$\overline{\text{RCLK}}$-I/O	71	C11
58	D0-DIN-I/O	72	B11
59	DOUT-I/O	73	C10
60	CCLK	74	A11
61	A0-$\overline{\text{WS}}$-I/O	75	B10
62	A1-CS2-I/O	76	B9
63	A2-I/O	77	A10
64	A3-I/O	78	A9
n.c.	**I/O***	79	B8
n.c.	**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
n.c.	**I/O***	6	A4
n.c.	**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-limited.

* Indicates unconnected package pins for the XC3020.

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	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
35	50	I/O*
36	51	I/O*
37	52	M1-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	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	XTAL1-I/O
68	83	D6-I/O
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	$\overline{\text{RCLK-BUSY}}$/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 limited.
* Indicates unconnected package pins for the XC3020.
** Indicates unconnected package pins for the XC3020 and the XC3030.

Table 2d. XC3000 Family 132 Pin PGA Pinouts

PGA Pin Number	XC-3042 XC-3064	PGA Pin Number	XC-3042 XC-3064	PGA Pin Number	XC-3042 XC-3064	PGA Pin Number	XC-3042 XC-3064
C4	**GND**	B13	M1-$\overline{\text{RD}}$	P14	$\overline{\text{RESET}}$	M3	DOUT-I/O
A1	$\overline{\text{PWRDN}}$	C11	**GND**	M11	**VCC**	P1	CCLK
C3	I/O	A14	M0-RT	N13	DONE-$\overline{\text{PG}}$	M4	**VCC**
B2	I/O	D12	**VCC**	M12	D7-I/O	L3	**GND**
B3	I/O	C13	M2-I/O	P13	XTAL1-I/O	M2	A0-$\overline{\text{WS}}$-I/O
A2	**I/O***	B14	HDC-I/O	N12	I/O	N1	A1-CS2-I/O
B4	I/O	C14	I/O	P12	I/O	M1	I/O
C5	I/O	E12	I/O	N11	D6-I/O	K3	I/O
A3	**I/O***	D13	I/O	M10	I/O	L2	A2-I/O
A4	I/O	D14	$\overline{\text{LDC}}$-I/O	P11	**I/O***	L1	A3-I/O
B5	I/O	E13	**I/O***	N10	I/O	K2	I/O
C6	I/O	F12	I/O	P10	I/O	J3	I/O
A5	I/O	E14	I/O	M9	D5-I/O	K1	A15-I/O
B6	I/O	F13	I/O	N9	$\overline{\text{CS0}}$-I/O	J2	A4-I/O
A6	I/O	F14	I/O	P9	**I/O***	J1	**I/O***
B7	I/O	G13	I/O	P8	**I/O***	H1	A14-I/O
C7	**GND**	G14	$\overline{\text{INIT}}$-I/O	N8	D4-I/O	H2	A5-I/O
C8	**VCC**	G12	**VCC**	P7	I/O	H3	**GND**
A7	I/O	H12	**GND**	M8	**VCC**	G3	**VCC**
B8	I/O	H14	I/O	M7	**GND**	G2	A13-I/O
A8	I/O	H13	I/O	N7	D3-I/O	G1	A6-I/O
A9	I/O	J14	I/O	P6	$\overline{\text{CS1}}$-I/O	F1	**I/O***
B9	I/O	J13	I/O	N6	**I/O***	F2	A12-I/O
C9	I/O	K14	I/O	P5	**I/O***	E1	A7-I/O
A10	I/O	J12	I/O	M6	D2-I/O	F3	I/O
B10	I/O	K13	I/O	N5	I/O	E2	I/O
A11	**I/O***	L14	**I/O***	P4	I/O	D1	A11-I/O
C10	I/O	L13	I/O	P3	I/O	D2	A8-I/O
B11	I/O	K12	I/O	M5	D1-I/O	E3	I/O
A12	**I/O***	M14	I/O	N4	$\overline{\text{RCLK-BUSY}}$/RDY-I/O	C1	I/O
B12	I/O	N14	I/O	P2	I/O	B1	A10-I/O
A13	**I/O***	M13	XTAL2-I/O	N3	I/O	C2	A9-I/O
C12	I/O	L12	**GND**	N2	D0-DIN-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-limited.

* Indicates unconnected package pins for the XC3042.

XC3000 Family 164-Pin CQFP Pinouts

CQFP Pin Number	XC-3064 XC-3090
20	$\overline{\text{PWRDN}}$
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-3064 XC-3090
61	**I/O***
62	M1-$\overline{\text{RDATA}}$
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	$\overline{\text{LDC}}$-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	$\overline{\text{INIT}}$-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	$\overline{\text{RESET}}$
102	**Vcc**

CQFP Pin Number	XC-3064 XC-3090
103	DONE-$\overline{\text{PG}}$
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	$\overline{\text{CS0}}$-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	$\overline{\text{CS1}}$-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/$\overline{\text{BUSY}}$-$\overline{\text{RCLK}}$-I/O
139	I/O
140	**I/O***
141	I/O
142	**I/O***

CQFP Pin Number	XC-3064 XC-3090
143	D0-DIN-I/O
144	DOUT-I/O
145	CCLK
146	**Vcc**
147	**GND**
148	A0-$\overline{\text{WS}}$-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-limited.
* Indicates unconnected package pins for the XC-3064.

XC3000 Family 175-Pin PGA Pinouts

PGA Pin Number	XC-3090
B2	$\overline{\text{PWRDN}}$
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-$\overline{\text{RDATA}}$
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	$\overline{\text{LDC}}$-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	$\overline{\text{INIT}}$-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	$\overline{\text{RESET}}$
P14	**VCC**

PGA Pin Number	XC-3090
R14	DONE-$\overline{\text{PG}}$
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	$\overline{\text{CS0}}$-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	$\overline{\text{CS1}}$-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/$\overline{\text{BUSY}}$-$\overline{\text{RCLK}}$-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-$\overline{\text{WS}}$-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-limited.

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

PARAMETRICS

Absolute Maximum Ratings			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	Junction temperature plastic	+ 125	°C
	Junction temperature ceramic	+ 150	°C

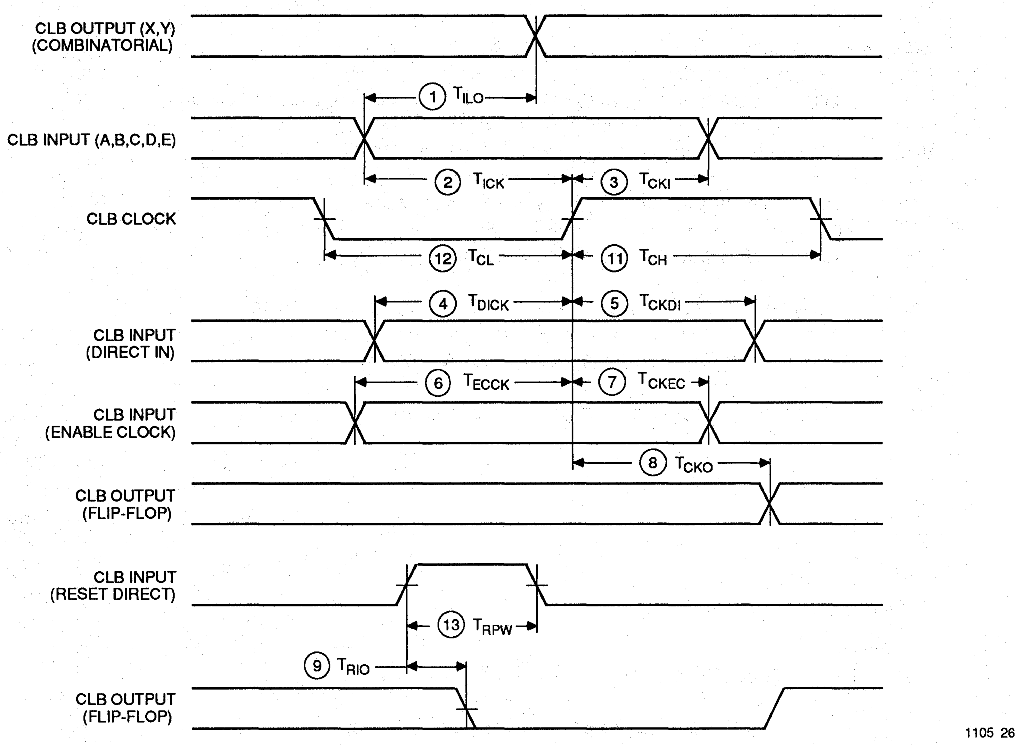
*Note: Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those listed under Recommended Operating Conditions is not implied. Exposure to Absolute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

Recommended Operating Conditions				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	V_{CC}	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

Electrical Characteristics Over Operating Conditions			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} min)			0.32	V
V_{OH}	High-level output voltage (@ I_{OH} = –4.0 mA, V_{CC} min)	Industrial	3.76		V
V_{OL}	Low-level output voltage (@ I_{OL} = 4.0 mA, V_{CC} min)			0.37	V
V_{OH}	High-level output voltage (@I_{OH} = –4.0 mA, V_{CC} min)	Military	3.7		V
V_{OL}	Low-level output voltage (@ I_{OL} = 4.0 mA, V_{CC} min)			0.4	V
I_{CCPD}	Power-down supply current (V_{CC} = 5.0 V @ 70°C)	XC3020		500	µA
		XC3030		800	µA
		XC3042		1150	µA
		XC3064		1650	µA
		XC3090		2500	µA
I_{CCO}	Quiescent LCA supply current in addition to I_{CCPD}[1]				
	Chip thresholds programmed as CMOS levels			500	µA
	Chip thresholds programmed as TTL levels			10	mA
I_{IL}	Leakage Current Commercial/Industrial Temperature		–10	+10	µA
	Leakage Current Military –55°C to 125°C		–20	+20	µA
C_{IN}	Input capacitance, all packages except PGA 175 (sample tested) All Pins except XTL1 and XTL2 XTL1 and XTL2			 10 15	 pF pF
	Input capacitance, PGA 175 (sample tested) All Pins except XTL1 and XTL2 XTL1 and XTL2			 15 20	 pF 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		0.4	3.4	mA

Note: 1. With no output current loads, no active input or long line pull-up resistors, all package pins at V_{CC} 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



BUFFER (Internal) SWITCHING CHARACTERISTIC GUIDELINES

	Speed Grade	-50		-70		-100		Units
Description	Symbol	Min	Max	Min	Max	Min	Max	
Global and Alternate Clock Distribution**								
Either: **Normal** IOB input pad to clock buffer input	TPID		9		6		4	ns
Or: **Fast** (CMOS only) input pad to clock buffer input	TPIDC		5		3		2	ns
Plus: Clock buffer input to any clock k			9		6		5	ns
TBUF driving a Horizontal Longline (L.L.)**								
I to L.L. while T is Low (buffer active)	TID		8		5		4	ns
T↓ to L.L. active and valid	TON		15		9		7	ns
T↑ to L.L. (inactive) with single pull-up resistor	TPUS		34		22		14	ns
with pair of pull-up resistors	TPUF		17		11		7	ns
BIDI								
Bi-directional buffer delay			6		4		3	ns

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

CLB SWITCHING CHARACTERISTIC GUIDELINES (Continued)

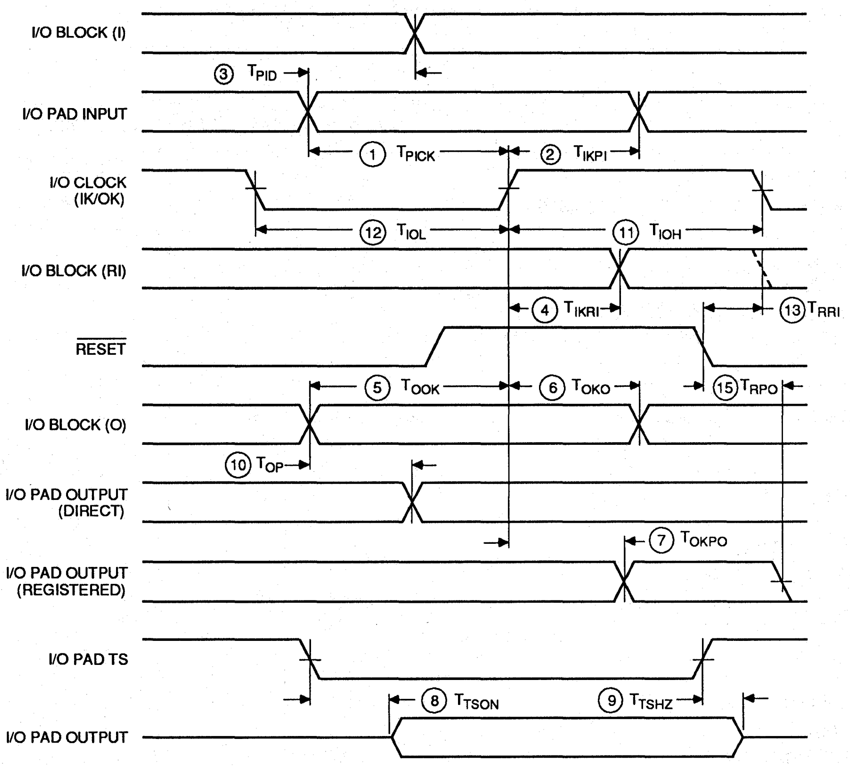
Testing of the switching characteristic guidelines is modeled after testing specified by MIL-M-38510/605. Devices are 100% functionally tested. Benchmark timing patterns are used to provide correlation to the switching characteristic guideline values. Actual worst-case timing is provided by the XACT Timing calculator or Simulation modeling.

	Speed Grade		-50		-70		-100		Units
Description	Symbol		Min	Max	Min	Max	Min	Max	
Combinatorial Delay									
Logic Variables a, b, c, d, e, to outputs x, y	1	TILO		14		9		7	ns
Sequential delay									
Clock k to outputs x, y	8	TCKO		12		8		7	ns
Clock k to outputs x,y when Q is returned through function generators F or G to drive x, y				23		15		12	ns
Set-up time before clock K									
Logic Variables a, b, c, d, e	2	TICK	12		8		7		ns
Data In di	4	TDICK	8		5		4		ns
Enable Clock ec	6	TECCK	10		7		5		ns
Reset Direct inactive rd			2		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	6		4		2		ns
Enable Clock ec	7	TCKEC	0		0		0		ns
Clock									
Clock High time*	11	TCH	9		7		5		ns
Clock Low time*	12	TCL	9		7		5		ns
Max. flip-flop toggle rate*		FCLK	50		70		100		MHz
Reset Direct (rd)									
rd width	13	TRPW	12		8		7		ns
delay from rd to outputs x, y	9	TRIO		12		8		7	ns
Master Reset (MR)									
MR width		TMRW	38		25		21		ns
delay from MR to outputs x, y		TMRQ		30		20		17	ns

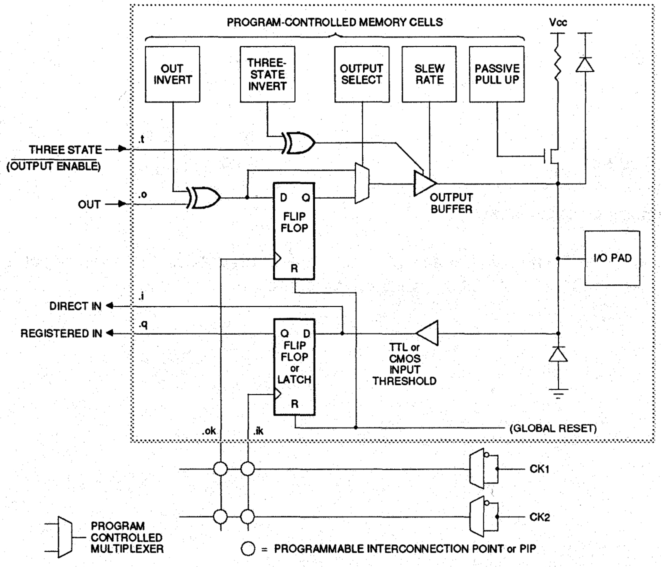
* These timing limits are based on calculations.

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 27



IOB SWITCHING CHARACTERISTIC GUIDELINES (Continued)

Testing of the switching characteristic guidelines is modeled after testing specified by MIL-M-38510/605. Devices are 100% functionally tested. Benchmark timing patterns are used to provide correlation to the switching charcteristic guideline values. Actual worst-case timing is provided by the XACT Timing calculator or Simulation modeling.

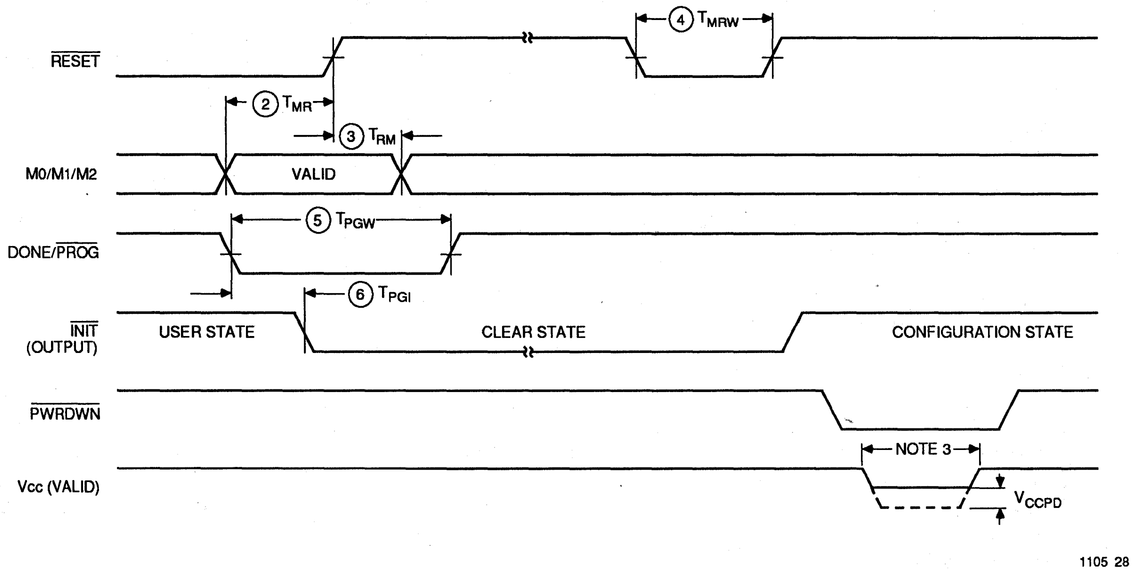
			-50		-70		-100		Units
Description	Symbol		Min	Max	Min	Max	Min	Max	
Propagation Delays (Input)									
Pad to Direct In (i)	3	TPID		9		6		4	ns
Pad to Registered In (q) with latch transparent		TPTG		34		21		17	ns
Clock (ik) to Registered In (q)	4	TIKRI		11		7		6	ns
Set-up Time (Input)									
Pad to Clock (ik) set-up time	1	TPICK	30		20		17		ns
Propagation Delays (Output)									
Clock (ok) to Pad (fast)	7	TOKPO		18		13		10	ns
same (slew rate limited)	7	TOKPO		43		33		27	ns
Output (o) to Pad (fast)	10	TOPF		15		9		6	ns
same (slew-rate limited)	10	TOPS		40		29		23	ns
Three-state to Pad begin hi-Z (fast)	9	TTSHZ		12		8		8	ns
same (slew-rate limited)	9	TTSHZ		37		28		25	ns
Three state to Pad active and valid (fast)	8	TTSON		20		14		12	ns
same (slew -rate limited)	8	TTSON		45		34		29	ns
Set-up and Hold Times (Output)									
Output (o) to clock (ok) set-up time	5	TOOK	15		10		9		ns
Output (o) to clock (ok) hold time	6	TOKO	0		0		0		ns
Clock									
Clock High time	11	TCH	9		7		5		ns
Clock Low time	12	TCL	9		7		5		ns
Max. flip-flop toggle rate		FCLK	50		70		100		MHz
Master Reset Delays									
$\overline{\text{RESET}}$ Pad to Registered In (q)	13	TRRI		35		23		20	ns
$\overline{\text{RESET}}$ Pad to output pad	15	TRPO		50		33		28	ns

Notes: 1. Timing is measured at pin threshold, with 50 pF external capacitive loads (incl. test fixture).
Typical fast mode output rise/fall times are 2 ns and will increase approximately 2%/pF of additional load.
Typical slew rate limited output rise/fall times are approximately 4 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 4 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 chapter (6-14 to 18).

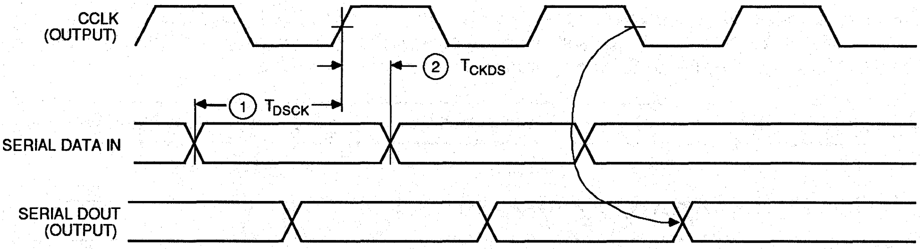
GENERAL LCA SWITCHING CHARACTERISTICS



				-50		-70		-100		Units
	Description	Symbol		Min	Max	Min	Max	Min	Max	
$\overline{\text{RESET}}$ (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				µs
	$\overline{\text{RESET}}$ Width (Low) req. for Abort	4	T_{MRW}	6		6				µs
DONE/$\overline{\text{PROG}}$	Width (Low) required for Re-config.	5	T_{PGW}	6		6		6		µs
	$\overline{\text{INIT}}$ response after D/$\overline{\text{P}}$ is pulled Low	6	T_{PGI}		7		7		7	µs
$\overline{\text{PWRDWN}}$ (3)	Power Down Vcc		V_{CCPD}	2.0		2.0		2.0		V

Notes:
1. Vcc must rise from 2.0 Volts to Vcc minimum in less than 10 ms for master modes.
2. $\overline{\text{RESET}}$ timing relative to valid mode lines (M0, M1, M2) is relevant when $\overline{\text{RESET}}$ is used to delay configuration.
3. $\overline{\text{PWRDWN}}$ transitions must occur during operational Vcc levels.

MASTER SERIAL MODE SWITCHING CHARACTERISTICS GUIDELINES



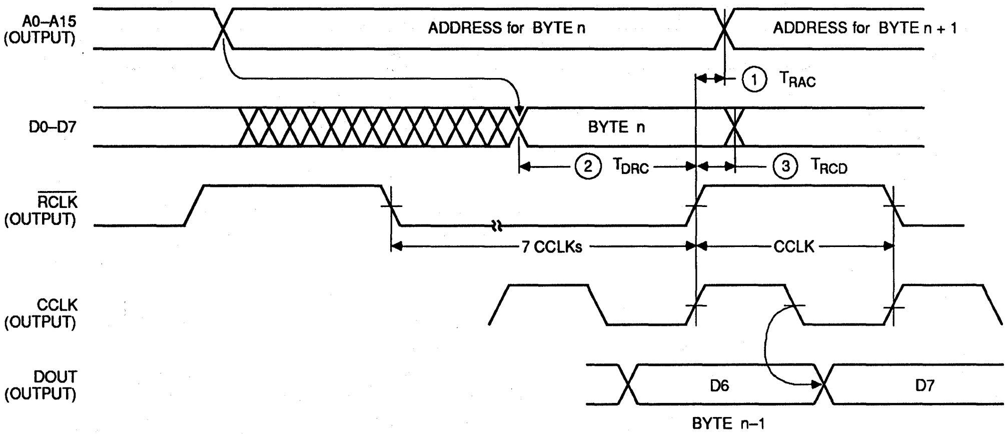
1105 29

			Speed Grade	-50		-70				Units
	Description	Symbol		Min	Max	Min	Max	Min	Max	
CCLK[3]	Data In setup	1	T_{DSCK}	60		60				ns
	Data In hold	2	T_{CKDS}	0		0				ns

Notes: 1. At power-up, Vcc must rise from 2.0 V to Vcc min. in less than 10 ms, otherwise delay configuration using $\overline{\text{RESET}}$ until Vcc is valid.
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.
3. Master serial mode timing is based on slave mode testing.

MASTER PARALLEL MODE PROGRAMMING SWITCHING CHARACTERISTIC GUIDELINES

Testing of the switching characteristic guidelines is modeled after testing specified by MIL-M-38510/605. Devices are 100% functionally tested. Benchmark timing patterns are used to provide correlation to the switching charcteristic guideline values. Actual worst-case timing is provided by the XACT Timing calculator or Simulation modeling.



1105 30

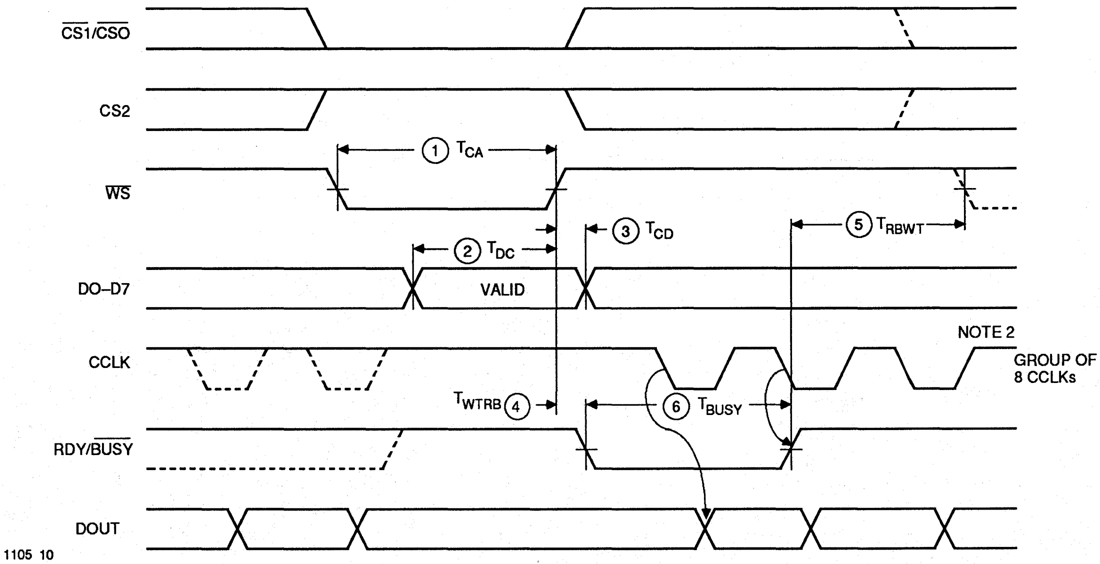
				-50		-70				Units
	Description	Symbol		Min	Max	Min	Max	Min	Max	
RCLK	To address valid	1	TRAC	0	200	0	200			ns
	To data setup	2	TDRC	60		60				ns
	To data hold	3	TRCD	0		0				ns
	RCLK high		TRCH	600		600				ns
	RCLK low		TRCL	4.0		4.0				µs

Notes: 1. At power-up, Vcc must rise from 2.0 V to Vcc min. in less than 10 ms, otherwise delay configuration using $\overline{\text{RESET}}$ until Vcc is valid.

2. Configuration can be contorlled 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

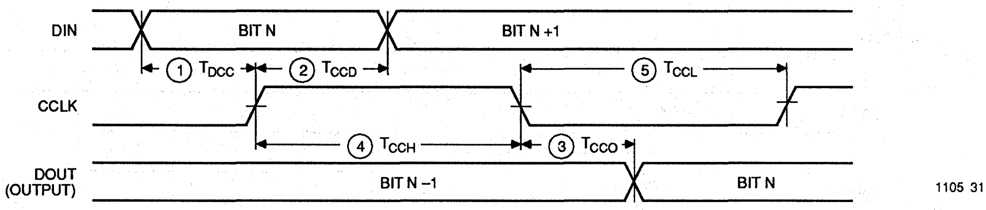


				-50		-70		-100		Units
	Description	Symbol		Min	Max	Min	Max	Min	Max	
Write	WS Low time reqired	1	TCA	0.5		0.5		0.5		µs
	DIN setup time required DIN hold time required	2 3	TDC TCD	60 0		60 0		60 0		ns ns
	RDY/BUSY delay after end of WS	4	TWTRB		60		60		60	ns
RDY	Earliest next WS after end of BUSY	5	TRBWT	0		0		0		ns
	BUSY Low time generated	6	TBUSY	4	9	4	9	4	9	CCLK Periods

Notes:
1. Configuration must be delayed until the INIT of all LCAs is HIGH.
2. Time from end of WS to CCLK cycle for the new byte of data depends on completion of previous byte processing and the phase of the internal timing generator for CCLK.
3. CCLK and DOUT timing is tested in slave mode.

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

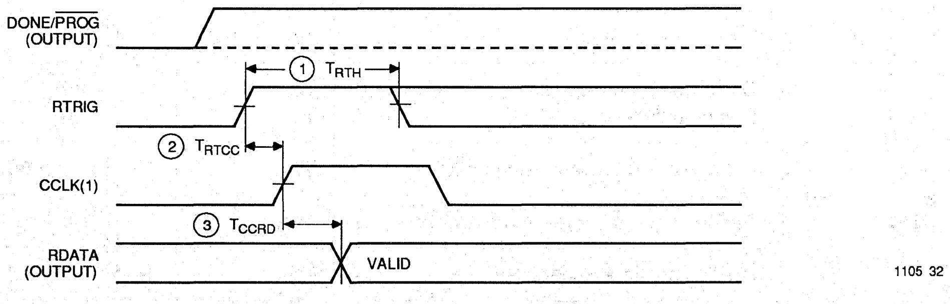
SLAVE MODE PROGRAMMING SWITCHING CHARACTERISTICS



				-50		-70				Units
	Description	Symbol		Min	Max	Min	Max	Min	Max	
CCLK	To DOUT	3	T_{CCO}		100		100			ns
	DIN setup	1	T_{DCC}	60		60				ns
	DIN hold	2	T_{CCD}	0		0				ns
	High time	4	T_{CCH}	0.5		0.5				ns
	Low time	5	T_{CCL}	0.5	5.0	0.5	5.0			µs
	Frequency		F_{CC}		1		1			MHz

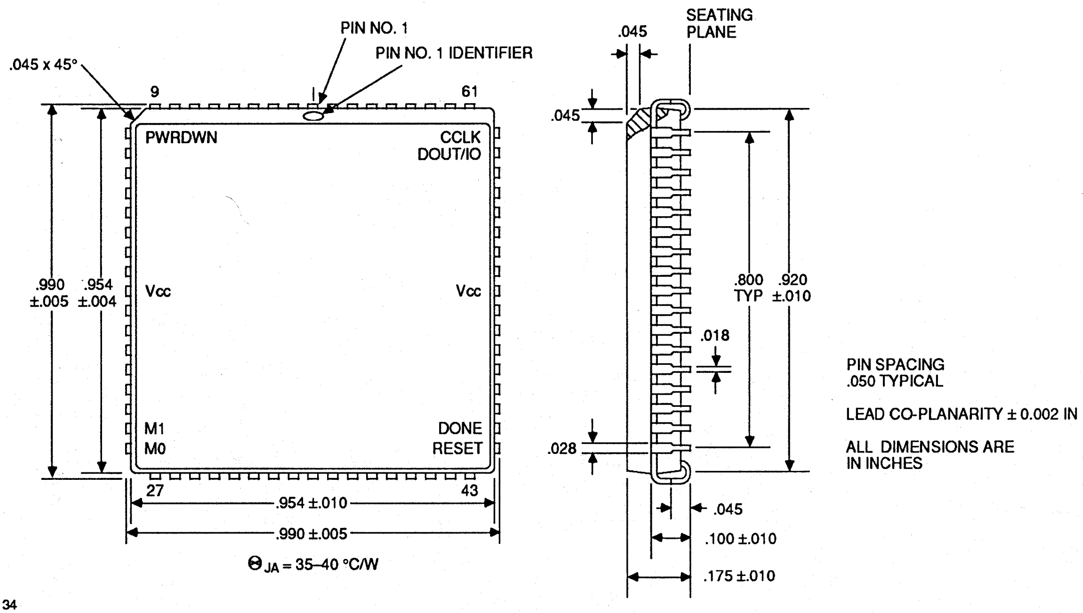
Note: Configuration must be delayed until the INIT of all LCAs is HIGH.

PROGRAM READBACK SWITCHING CHARACTERISTICS



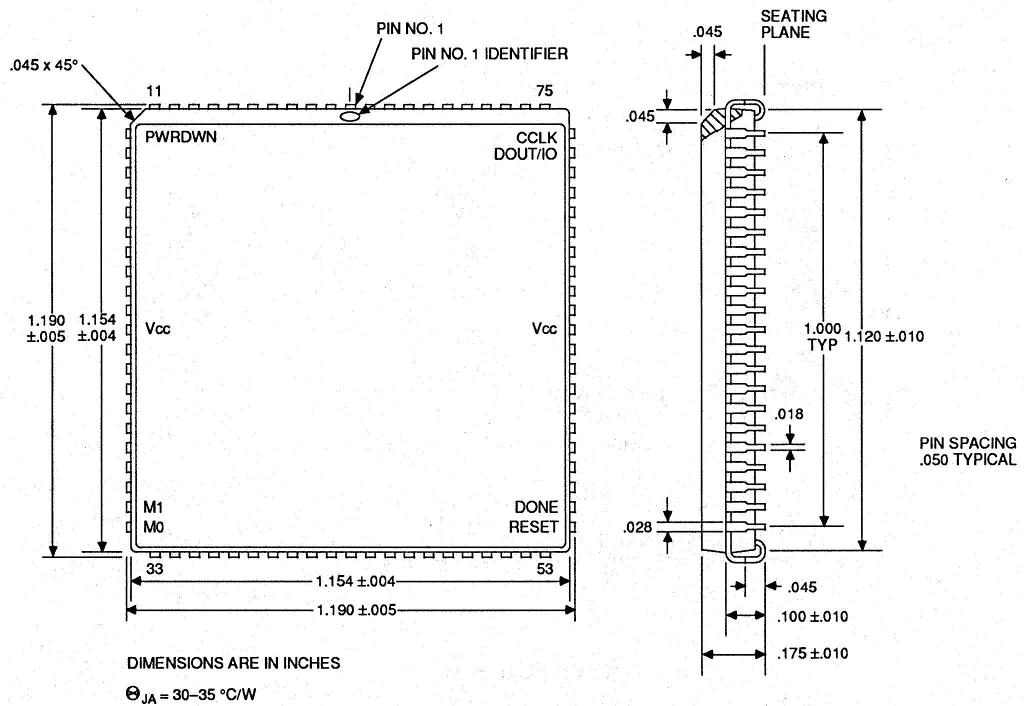
				-50		-70				Units
	Description	Symbol		Min	Max	Min	Max	Min	Max	
RTRIG	RTRIG high	1	T_{RTH}	250		250				ns
CCLK	RTRIG setup	2	T_{RTCC}	10		10				ns
	$\overline{\text{RDATA}}$ delay	3	T_{CCRD}		100		100			ns

Notes: 1. CCLK and DOUT timing are the same as for slave mode.
2. RETRIG (M0 positive transition) shall not be done until after one clock following active I/O pins.
3. Readback should not be initiated until configuration is complete.



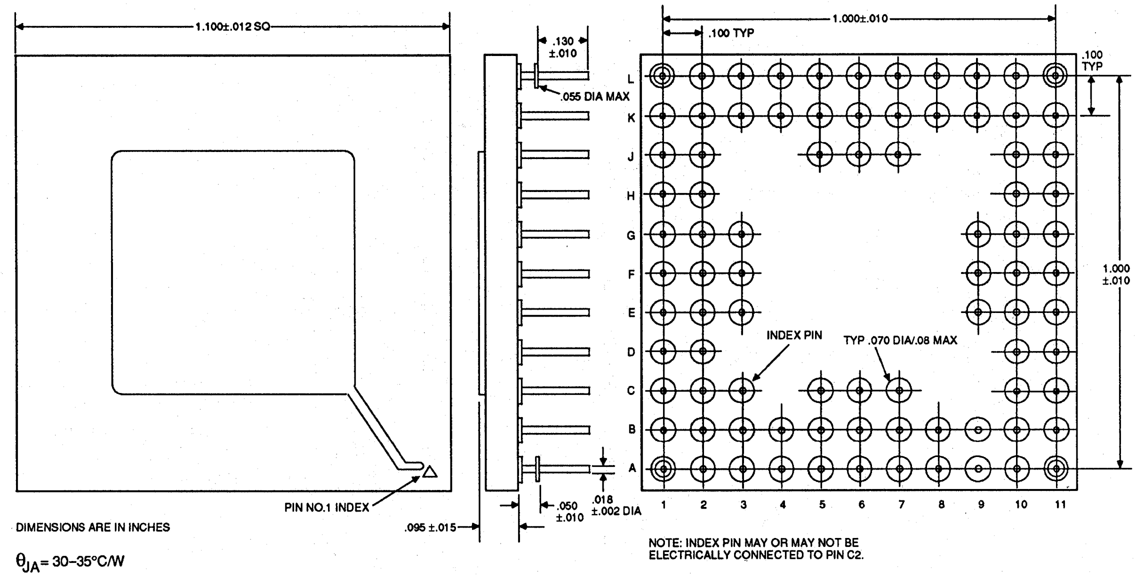
1105 34

68-Pin PLCC Package



1105 36

84-Pin PLCC Package



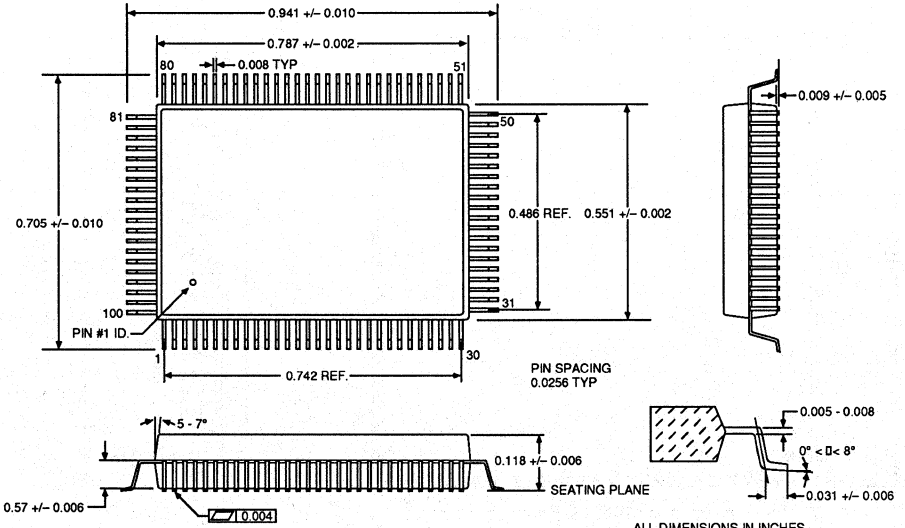
DIMENSIONS ARE IN INCHES

θ_{JA}= 30–35°C/W

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

1105 35

84-Pin PGA Package

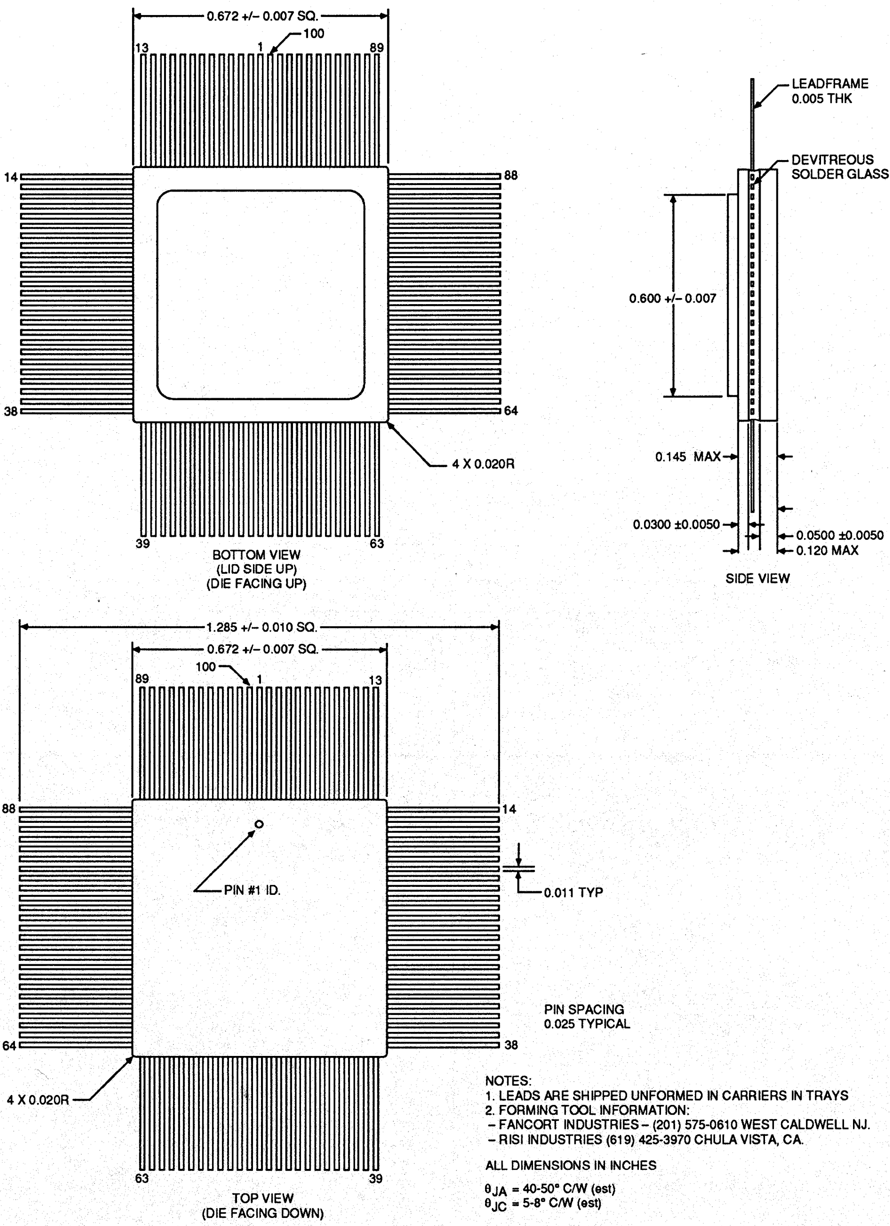


ALL DIMENSIONS IN INCHES

θ_{JA} = 70-80° C/W (est)

1105 39

100-Pin PQFP Package



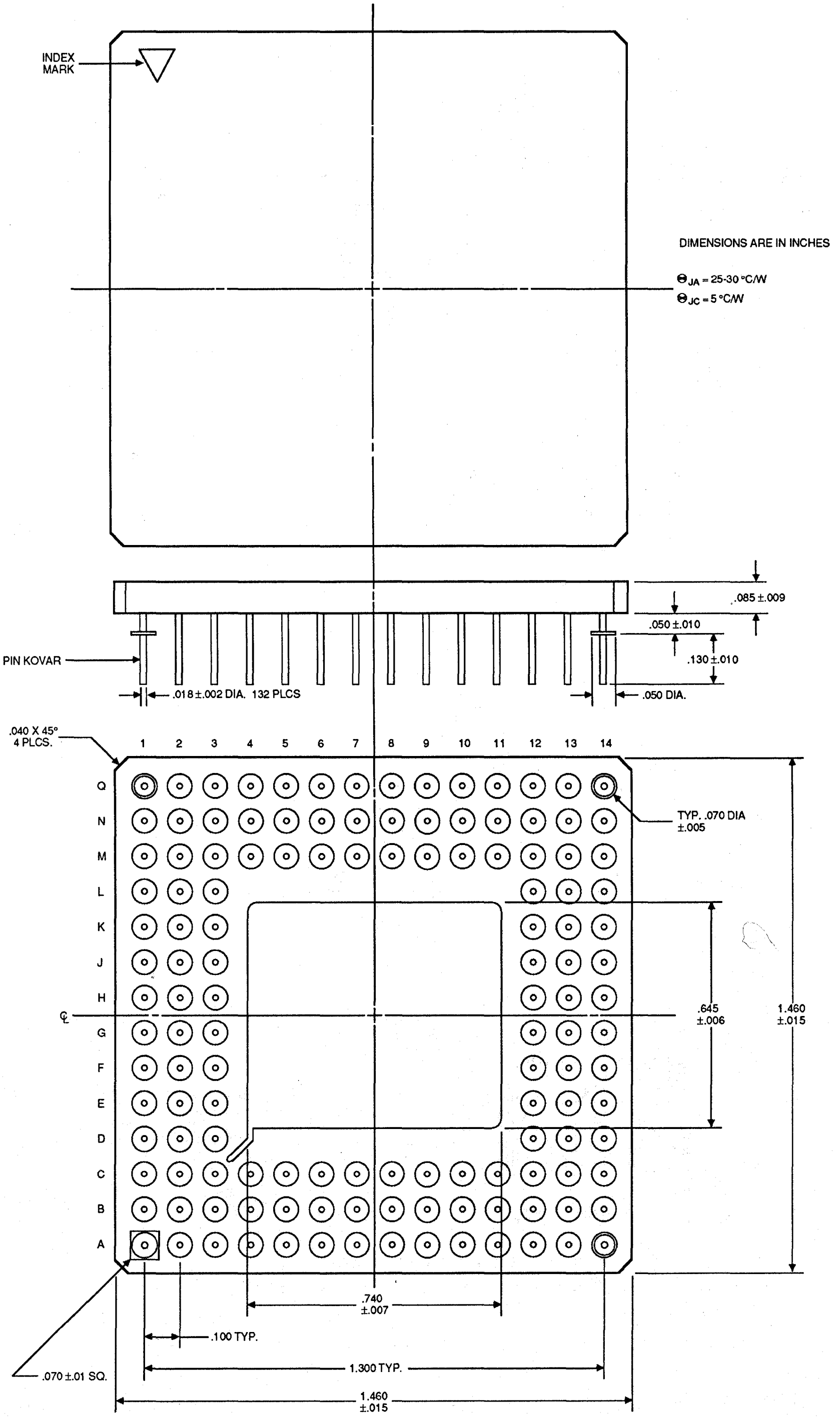
NOTES:
1. LEADS ARE SHIPPED UNFORMED IN CARRIERS IN TRAYS
2. FORMING TOOL INFORMATION:
- FANCORT INDUSTRIES – (201) 575-0610 WEST CALDWELL NJ.
- RISI INDUSTRIES (619) 425-3970 CHULA VISTA, CA.

ALL DIMENSIONS IN INCHES

θ_{JA} = 40-50° C/W (est)
θ_{JC} = 5-8° C/W (est)

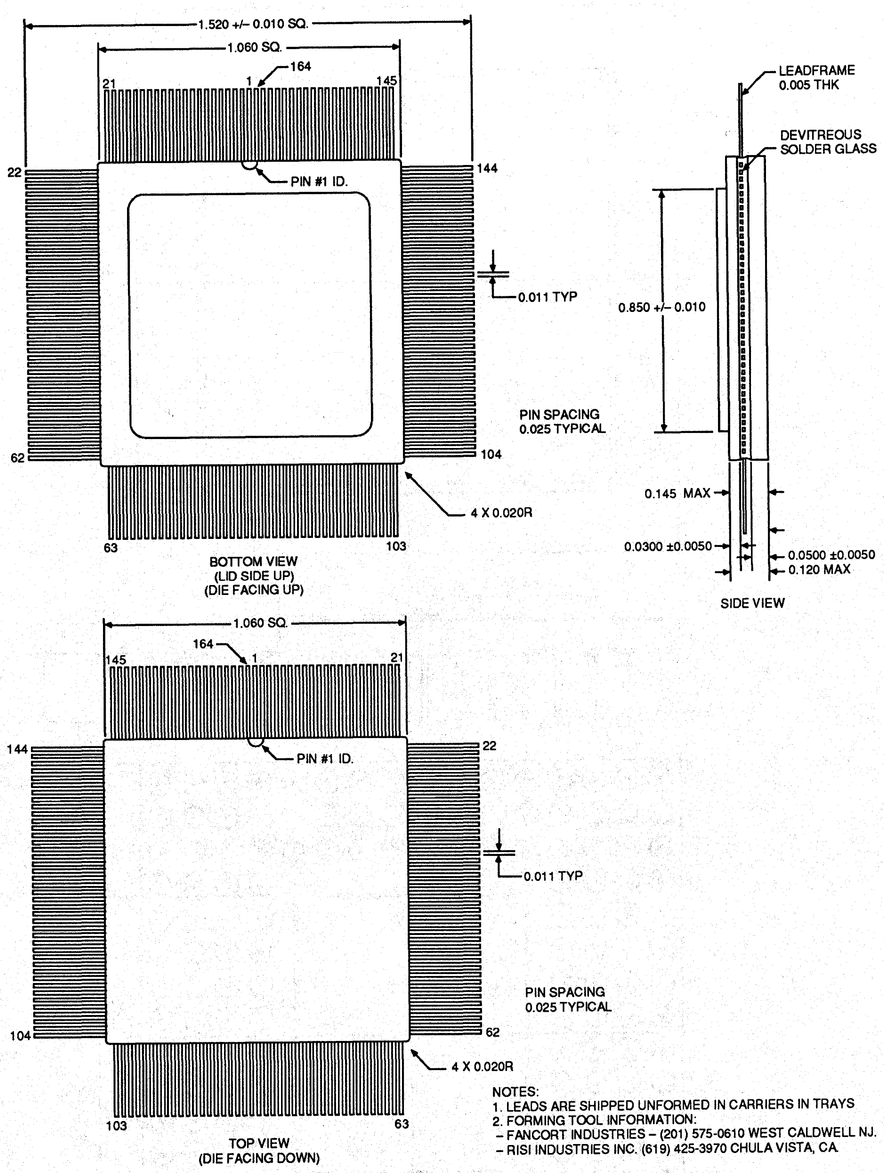
1105 40

100-Pin CQFP Package



1105 38

132-Pin PGA Package



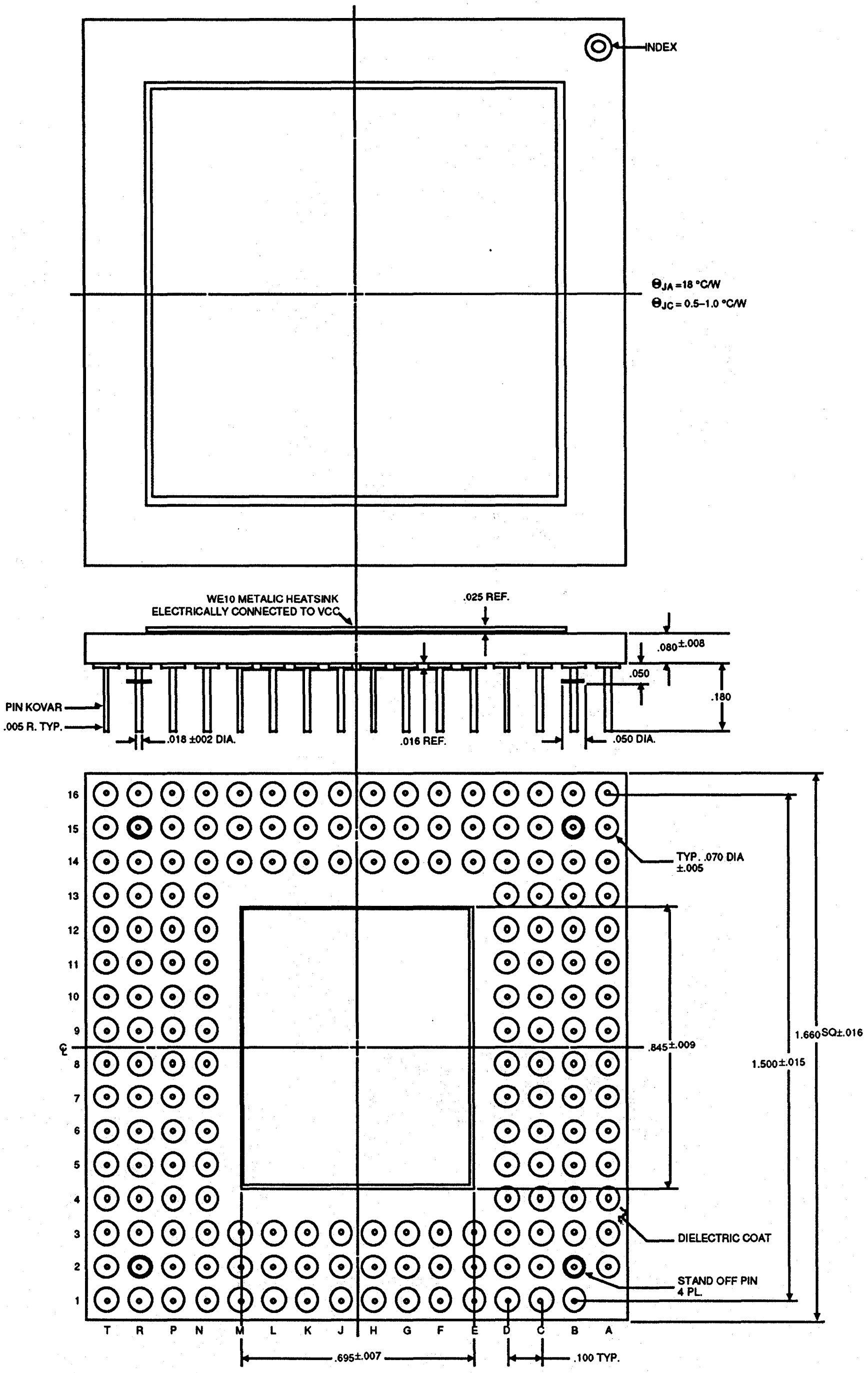
NOTES:
1. LEADS ARE SHIPPED UNFORMED IN CARRIERS IN TRAYS
2. FORMING TOOL INFORMATION:
- FANCORT INDUSTRIES – (201) 575-0610 WEST CALDWELL NJ.
- RISI INDUSTRIES INC. (619) 425-3970 CHULA VISTA, CA

ALL DIMENSIONS IN INCHES

θ_{JA} = 35-45° C/W (est)
θ_{JC} = 3-5° C/W (est)

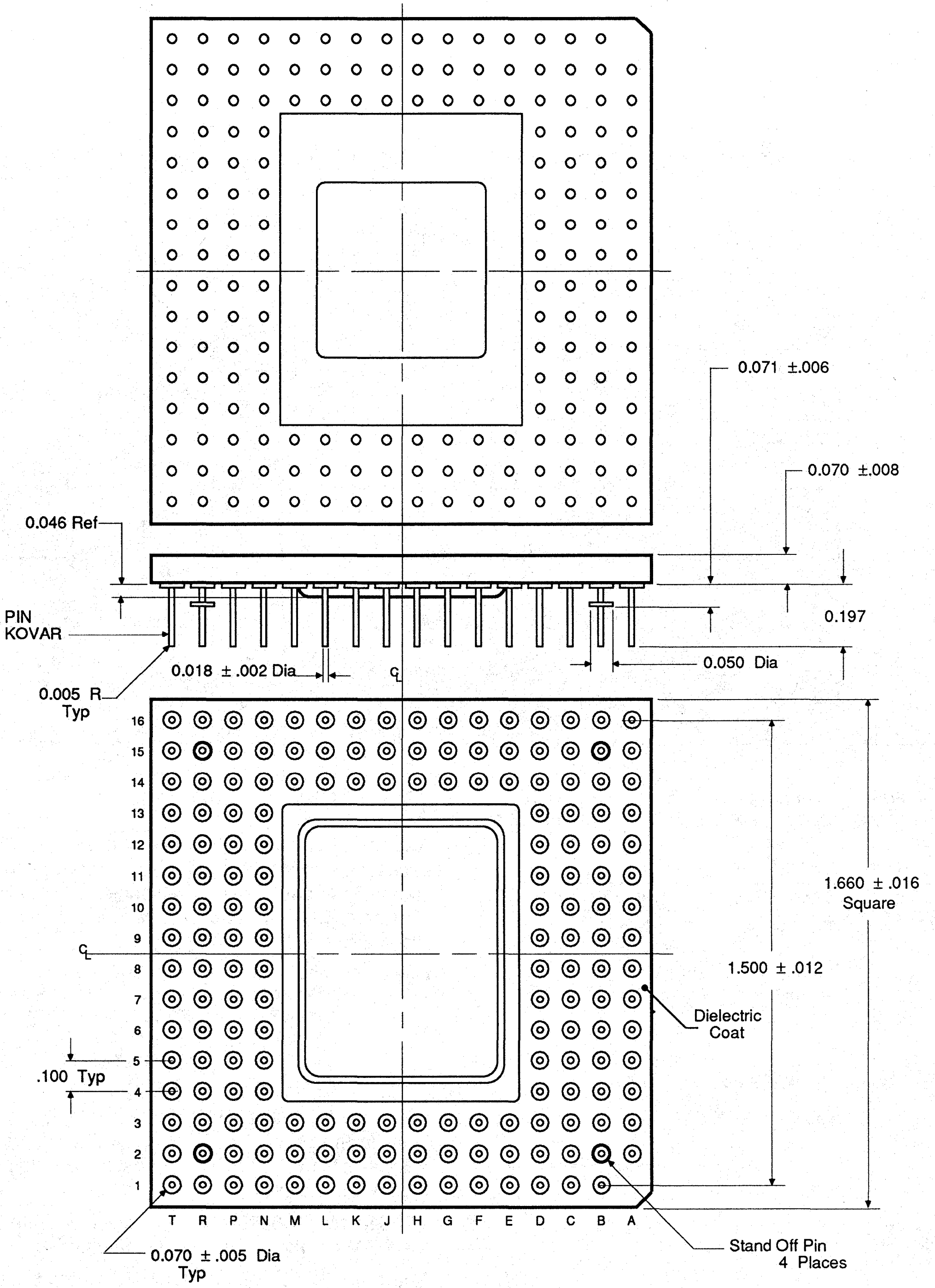
1105 41

164-Pin CQFP Package



1105 37

175-Pin PGA Package



175-Pin PPGA Package

Component Selection & Ordering Information

		48 PIN		68 PIN		84 PIN		100 PIN		132 PIN		164 PIN	175 PIN	
		PLASTIC DIP	CERAMIC DIP	PLASTIC PLCC	CERAMIC PGA	PLASTIC PLCC	CERAMIC PGA	PLASTIC PQFP	CERAMIC CQFP	PLASTIC PGA	CERAMIC PGA	CERAMIC CQFP	PLASTIC PGA	CERAMIC PGA
		-PD48	-CD48	-PC68	-PG68	-PC84	-PG84	-PQ100	-CQ100	-PP132	-PG132	-CQ164	-PP175	-PG175
XC2064	-33	C	I M	C I	C I M									
	-50	C	I	C I	C I M									
	-70			C I	C I M									
	-100			C	C									
XC2018	-33			C I		C I	C I M B							
	-50			C I		C I	C I M B							
	-70			C I		C I	C I M							
	-100			C		C	C							
XC3020	-50			C I		C I	C I M B	C I	C I M B					
	-70			C I		C I	C I M	C I	C I M					
	-100			C		C	C	C	C					
XC3030	-50			C I		C I	C I M	C I	C I M					
	-70			C I		C I	C I M	C I	C I M					
	-100			C		C	C	C	C					
XC3042	-50					C I	C I M	C I	C I M	C I	C I M B			
	-70					C I	C I M	C I	C I M	C I	C I			
	-100					C	C	C	C	C	C			
XC3064	-50									C I	C I M			
	-70									C I	C I			
	-100									C	C			
XC3090	-50											C I M	C I	C I M B
	-70											C I M	C I	C I M
	-100											C	C	C

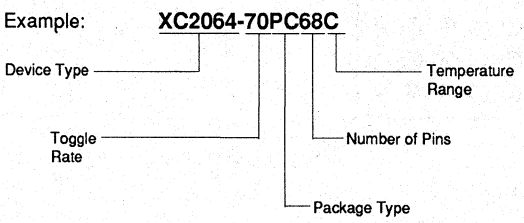
1972 01

XC1736-PD8C	**Plastic 8 Pin, Mini-DIP**	**–40°C to 85°C**
XC1736-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	Military	–55°C to 125°C
B	Military	MIL-STD-883, Class B

ORDERING INFORMATION



COMPATIBLE PACKAGE OPTIONS

A range of Xilinx 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:	PLCC 68:	2064-2018-3020-3030
	PLCC 84:	2018-3020-3030-3042
	PGA 84:	2018-3020-3030-3042
	PQFP 100:	3020-3030-3042
	CQFP100:	3020-3030-3042
	PGA 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.

XILINX

XC2064
XC2018
Logic Cell™ Array

Product Specification

FEATURES

- Fully user-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 usable 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** (usable) gates	**Config-urable Logic Blocks**	**User I/Os**	**Config-uration Program** (bits)
XC2064	1200	64	58	12038
XC2018	1800	100	74	17878

The Logic Cell Array's 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, Configurable Logic Blocks and Programmable Interconnections. The I/O Blocks provide an interface between the logic array and the device package pins. The Configurable Logic Blocks perform user-specified logic functions, and the interconnect resources are programmed to form networks that carry logic signals among blocks.

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

Configuration Memory

The configuration of the Xilinx 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.

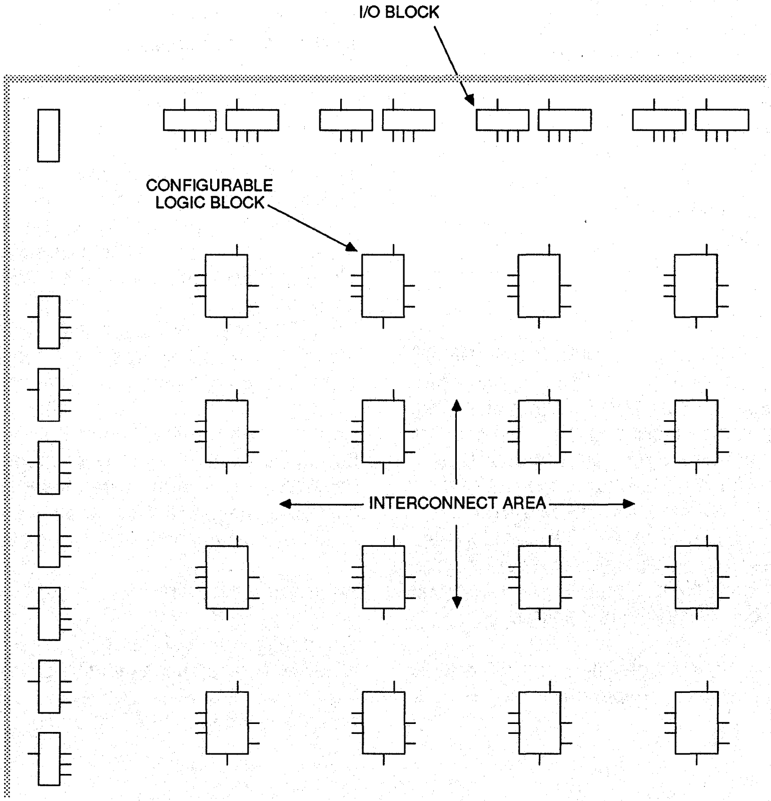
The static memory cell used for the configuration memory in the Logic Cell Array has been designed specifically for high reliability and noise immunity. Based on this design, which is covered by a pending patent application, integrity of the LCA configuration memory is assured even under adverse conditions. 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



1104 01

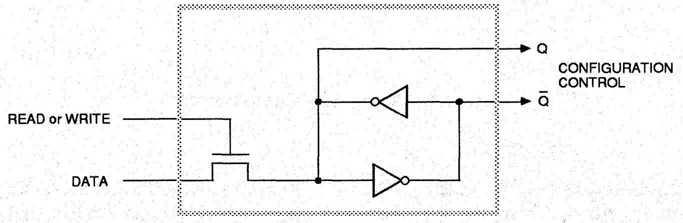
Figure 1. Logic Cell Array Structure

compatible with either TTL (1.4 V) or CMOS (2.2 V) levels. 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 logical three-state buffer control. The user may also select the output buffer three-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.

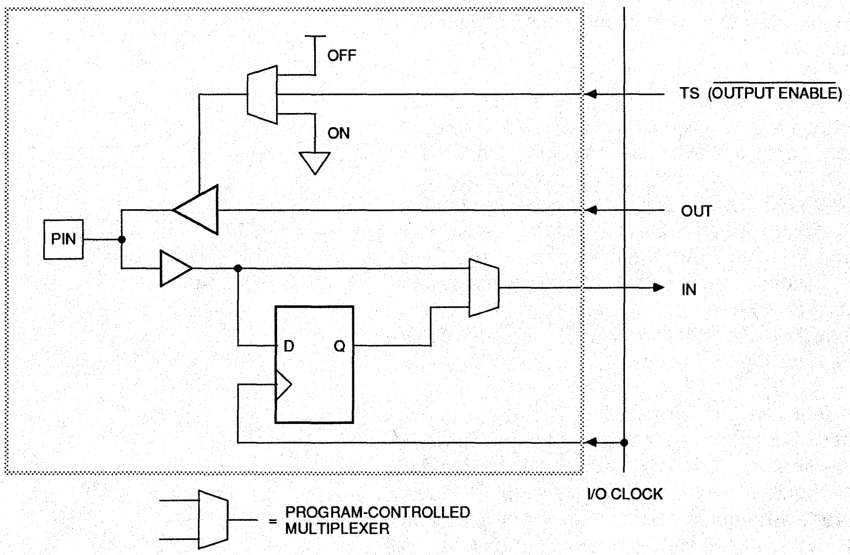
Configurable Logic Block

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 center of the device. The XC2064 has 64 such blocks



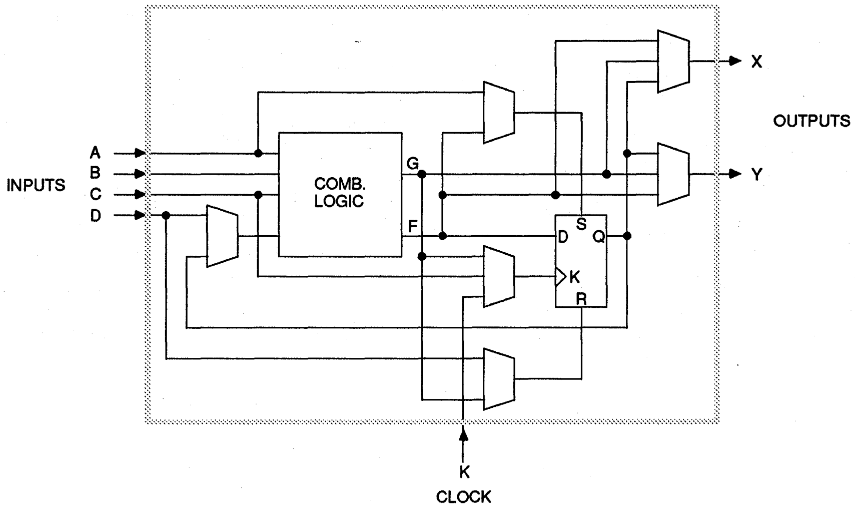
1104 02

Figure 2. Configuration Memory Cell



1104 03

Figure 3. I/O Block



1104 04

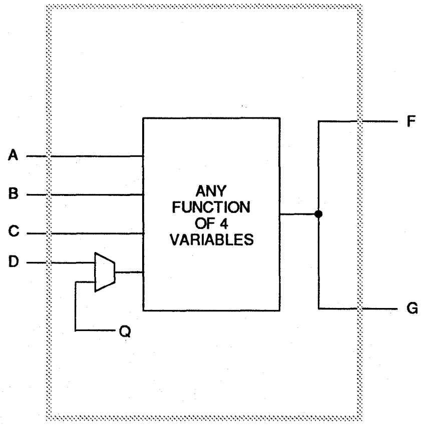
Figure 4. Configurable Logic Block

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 four-variable configuration is selected (Option 1), the F and G outputs are identical. If the two-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 its storage element output "Q". A third form of the combinatorial logic (Option 3) is a special case of the two-function form in which the B input dynamically selects between the two function tables providing a single merged logic function output. This dynamic selec-



OPTION 1

1 FUNCTION OF 4
VARIABLES

tion allows some five-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.

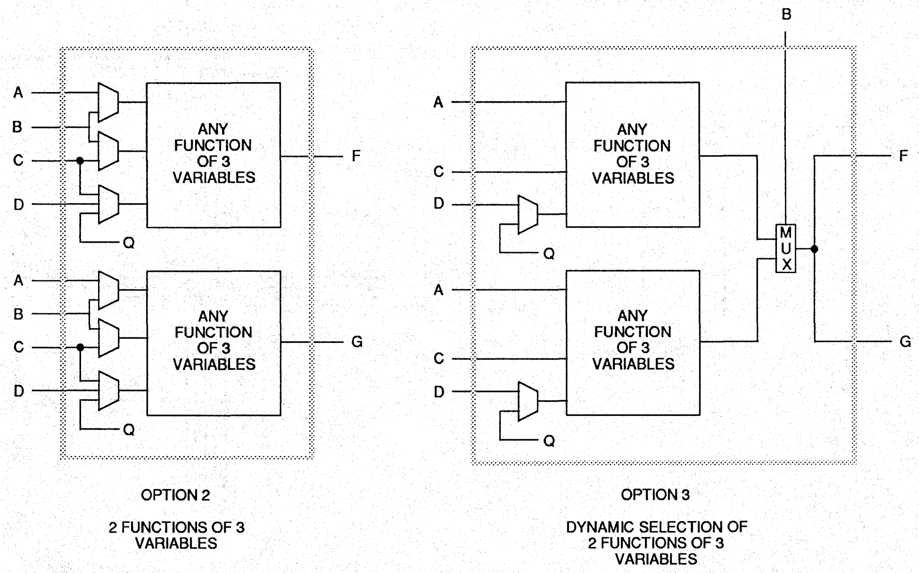
The storage element data input is supplied from the function F output of the combinatorial logic. Asynchronous SET and RESET controls are provided for each storage element. The user may enable these controls independently and select their source. They are active High inputs and the asynchronous reset 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.

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



A
B
C
D
ANY FUNCTION OF 3 VARIABLES
F
Q
A
B
C
D
ANY FUNCTION OF 3 VARIABLES
G
Q
OPTION 2
2 FUNCTIONS OF 3 VARIABLES
B
A
C
D
ANY FUNCTION OF 3 VARIABLES
Q
MUX
F
A
C
D
ANY FUNCTION OF 3 VARIABLES
G
Q
OPTION 3
DYNAMIC SELECTION OF 2 FUNCTIONS OF 3 VARIABLES

1104 05

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

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.

1104 06

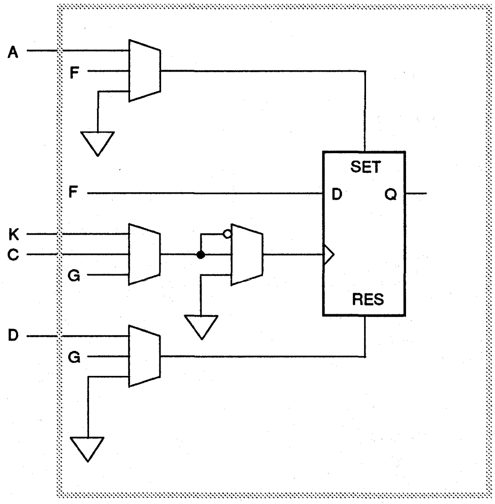
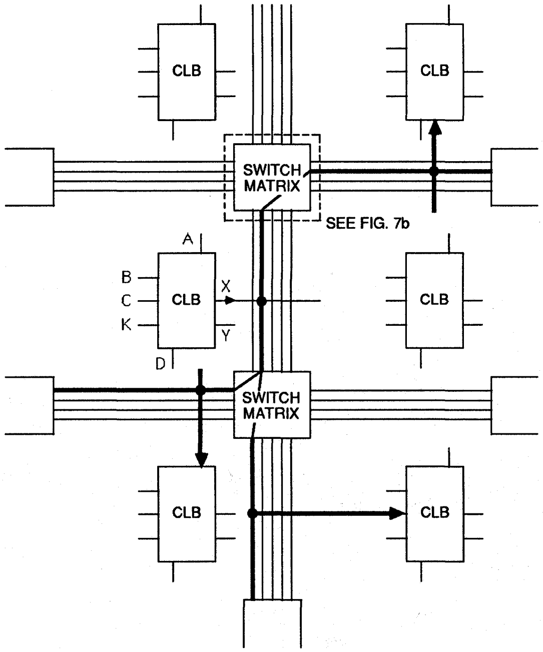


Figure 6. CLB Storage Elememt



1104 07

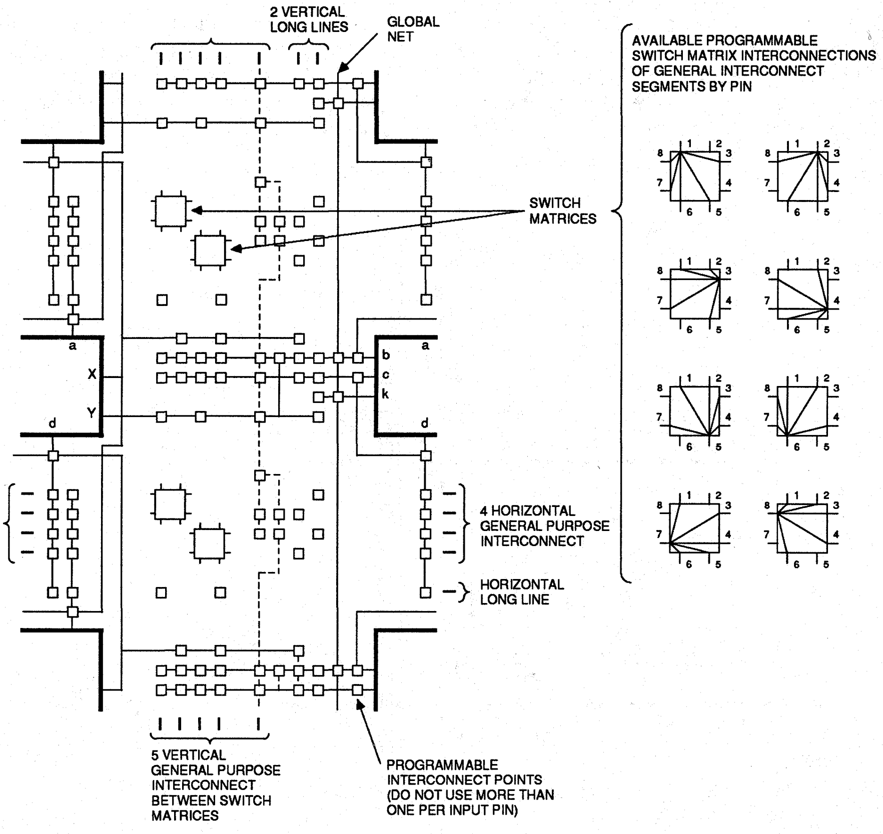
Figure 7a. General-Purpose Interconnect

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.


2 VERTICAL LONG LINES
GLOBAL NET
AVAILABLE PROGRAMMABLE SWITCH MATRIX INTERCONNECTIONS OF GENERAL INTERCONNECT SEGMENTS BY PIN
1
2
3
4
5
6
7
8
SWITCH MATRICES
a
X
Y
d
b
c
k
a
d
4 HORIZONTAL GENERAL PURPOSE INTERCONNECT
HORIZONTAL LONG LINE
5 VERTICAL GENERAL PURPOSE INTERCONNECT BETWEEN SWITCH MATRICES
PROGRAMMABLE INTERCONNECT POINTS (DO NOT USE MORE THAN ONE PER INPUT PIN)

1104 08

Figure 7b. Routing and Switch Matrix Connections

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

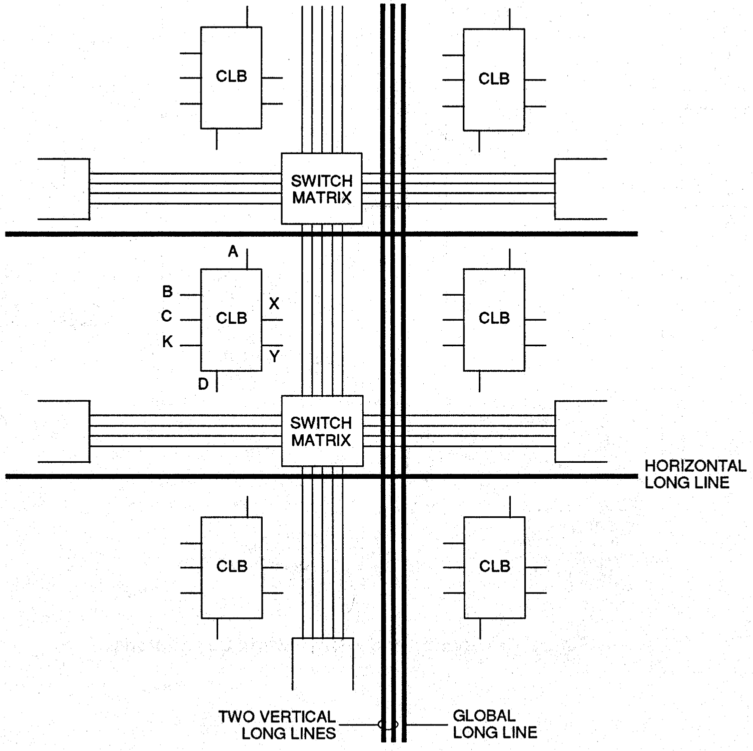
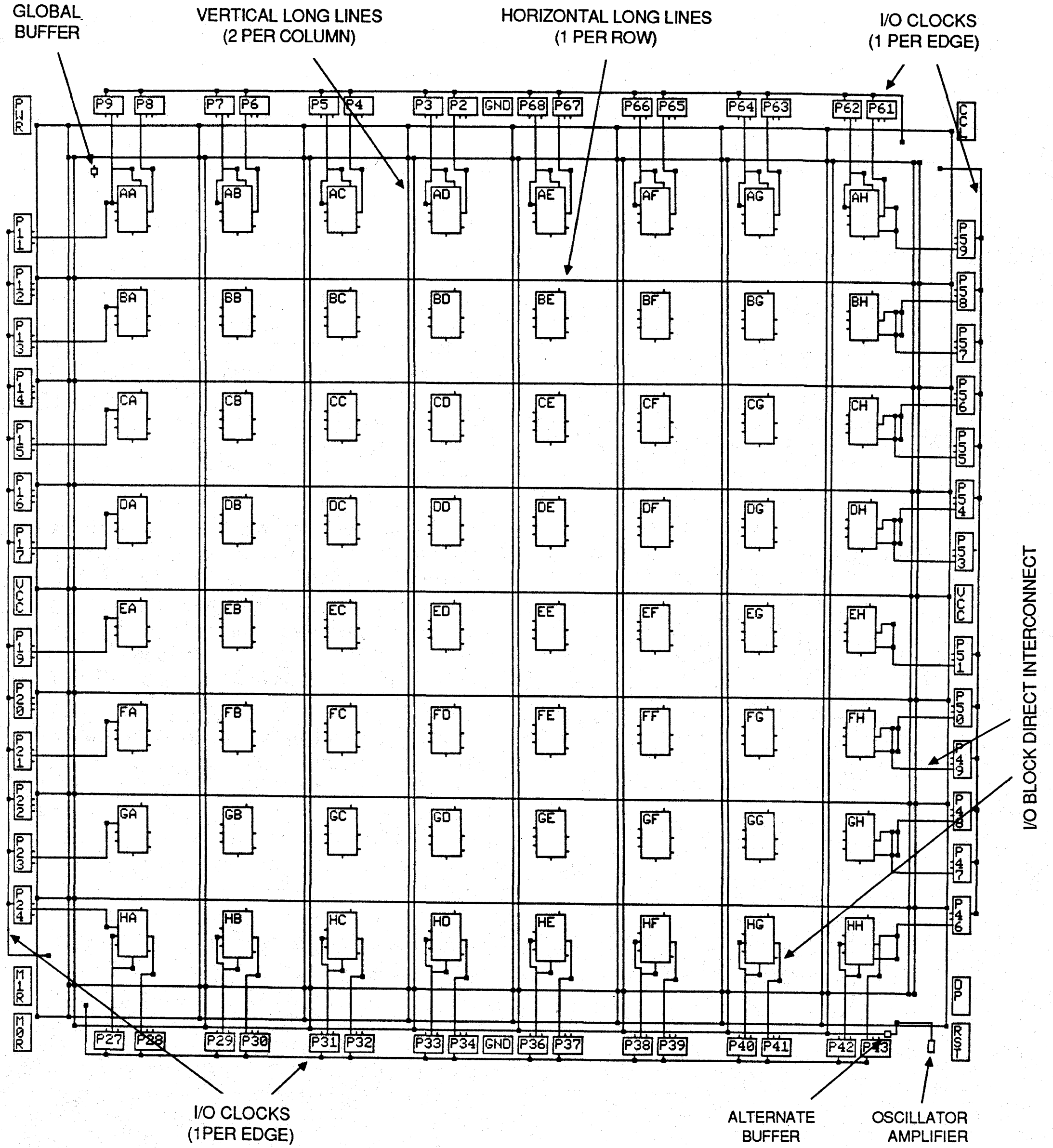


Figure 8a. Long Line Interconnect


GLOBAL BUFFER

VERTICAL LONG LINES
(2 PER COLUMN)

HORIZONTAL LONG LINES
(1 PER ROW)

I/O CLOCKS
(1 PER EDGE)

I/O BLOCK DIRECT INTERCONNECT

I/O CLOCKS
(1PER EDGE)

ALTERNATE
BUFFER

OSCILLATOR
AMPLIFIER

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

the amplifier output impe-dance 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.

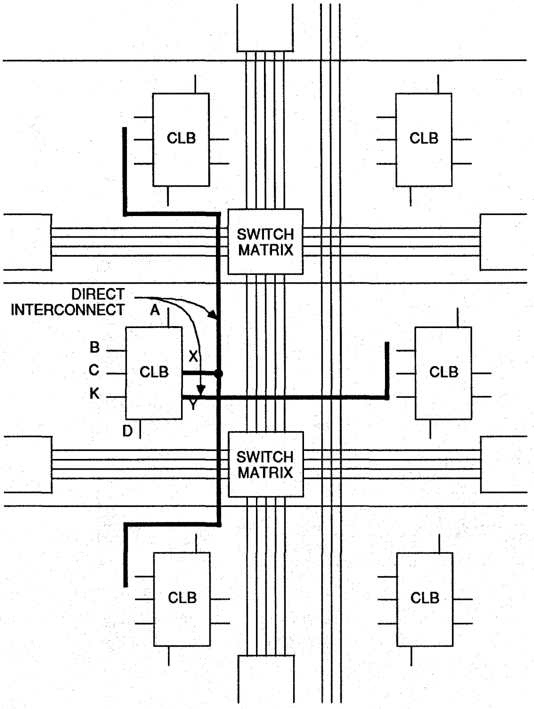


Figure 9. Direct Interconnect

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.

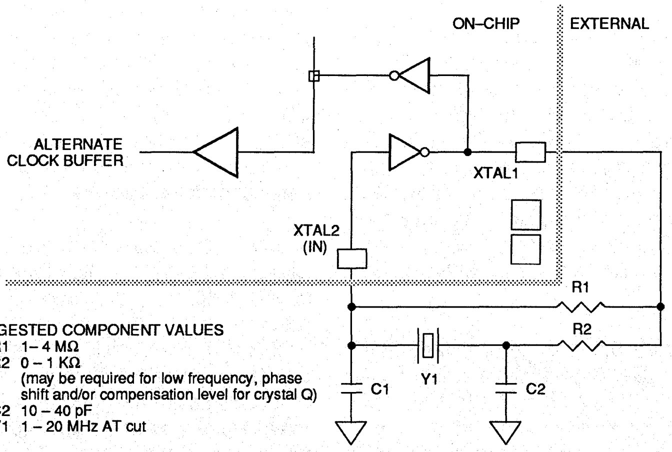
Output buffers capable of driving the specified 4 mA loads under worst-case conditions may be capable of driving 25 to 30 times that current in a best case. Noise can be reduced by minimizing external load capacitance and reducing simultaneous output transitions in the same direction. It may also be 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 Dissipation

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 4mW / 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.

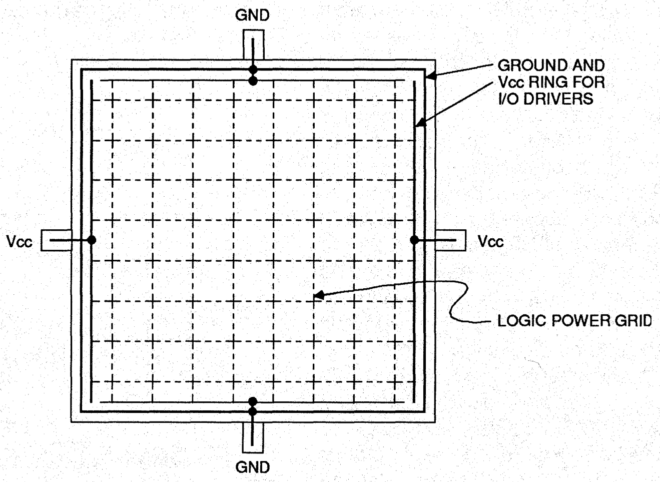


SUGGESTED COMPONENT VALUES
R1 1 – 4 MΩ
R2 0 – 1 KΩ
(may be required for low frequency, phase shift and/or compensation level for crystal Q)
C1, C2 10 – 40 pF
Y1 1 – 20 MHz AT cut

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

1104 11

Figure 10. Crystal Oscillator



1104 12

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, 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 / 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 , RESET, to other portions of the system. High during configuration (HDC) and low during configuration (LDC), are released one CCLK cycle before DONE is asserted. In master mode configurations, it is convenient to use 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.

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 (2.5 to 3 Volts), 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 (15 to 35 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 (60 to 140 ms) to assure that all devices, which it may be driving in a daisy chain, will be ready. Configuration using peripheral or

1104 13

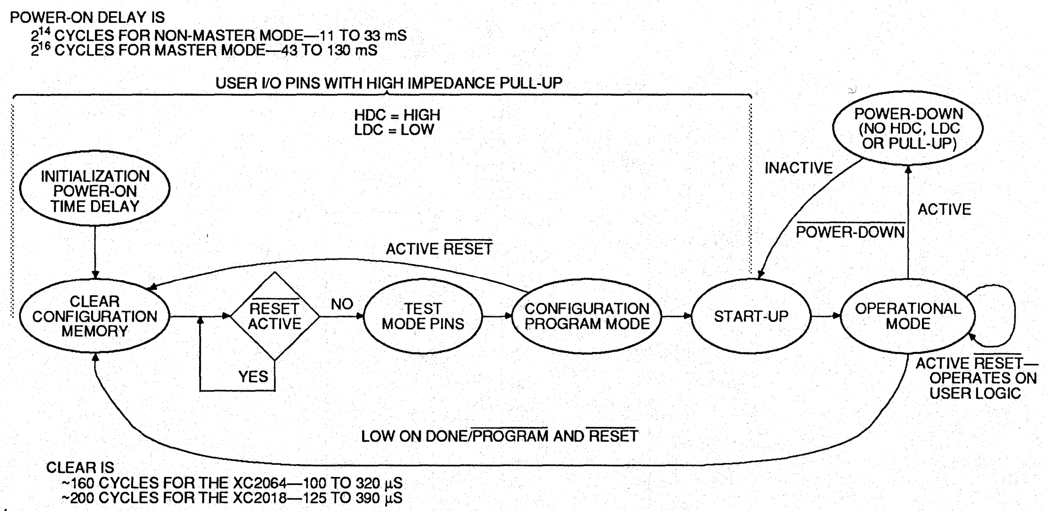
MODE PIN			MODE SELECTED
M0	M1	M2	
0	0	0	MASTER SERIAL
0	0	1	MASTER LOW MODE
0	1	1	MASTER HIGH MODE
1	0	1	PERIPHERAL MODE
1	1	1	SLAVE MODE

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

Figure 14. Configuration Mode Selection

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 3 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 ns) followed by a LOW level (for at least 6 ns) 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.


POWER-ON DELAY IS
2^{14} CYCLES FOR NON-MASTER MODE—11 TO 33 mS
2^{16} CYCLES FOR MASTER MODE—43 TO 130 mS
USER I/O PINS WITH HIGH IMPEDANCE PULL-UP
HDC = HIGH
LDC = LOW
POWER-DOWN (NO HDC, LDC OR PULL-UP)
INACTIVE
ACTIVE
$\overline{\text{POWER-DOWN}}$
INITIALIZATION POWER-ON TIME DELAY
ACTIVE $\overline{\text{RESET}}$
CLEAR CONFIGURATION MEMORY
RESET ACTIVE
NO
YES
TEST MODE PINS
CONFIGURATION PROGRAM MODE
START-UP
OPERATIONAL MODE
ACTIVE $\overline{\text{RESET}}$— OPERATES ON USER LOGIC
LOW ON DONE/$\overline{\text{PROGRAM}}$ AND $\overline{\text{RESET}}$
CLEAR IS
~160 CYCLES FOR THE XC2064—100 TO 320 µS
~200 CYCLES FOR THE XC2018—125 TO 390 µS

1104 14

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

```
11111111                    DUMMY BITS (4 BITS MINIMUM), XACT 2.10 GENERATES 8 BITS
0010                        PREAMBLE CODE                                              HEADER
< 24-BIT LENGTH COUNT >     CONFIGURATION PROGRAM LENGTH
1111                        DUMMY BITS (4 BITS MINIMUM)

0 < DATA FRAME # 001 > 111 }
0 < DATA FRAME # 002 > 111 }
0 < DATA FRAME # 003 > 111 }                          XC2018    XC2064
          .     .     .    }      CONFIGURATION                               PROGRAM DATA
          .     .     .    }        FRAMES              196       160
          .     .     .    }      DATA BITS                                   REPEATED FOR EACH LOGIC
0 < DATA FRAME # 159 > 111 }        PER FRAME            87        71         CELL ARRAY IN A DAISY CHAIN
0 < DATA FRAME # 160 > 111 }

1111                              POSTAMBLE CODE (4 BITS MINIMUM)
```

START-UP REQUIRES THREE CONFIGURATION CLOCKS BEYOND LENGTH COUNT

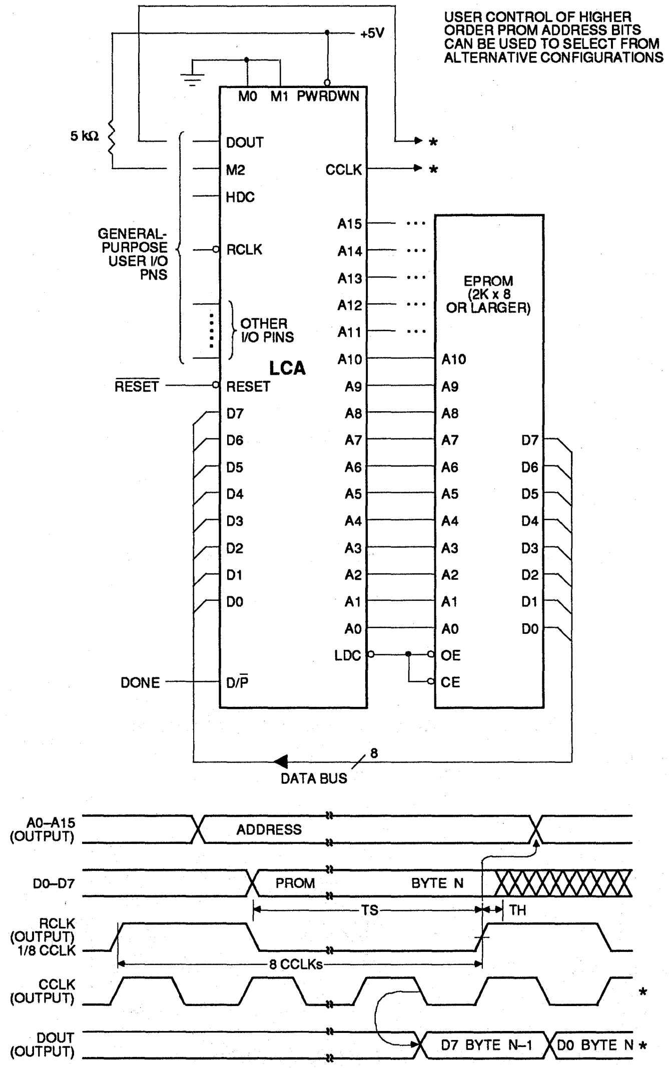
1104 15

Figure 13. XC2064 Internal Configuration Data Arrangement

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 sixteen address outputs and the control signals $\overline{\text{RCLK}}$ (read clock), HDC (high during configuration) and $\overline{\text{LDC}}$ (low during configuration) to execute read cycles from the external memory. Parallel eight-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 read 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 Logic



* FOR OPTIONAL SLAVE MODE LCAs IN A DAISY CHAIN

1104 16

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.

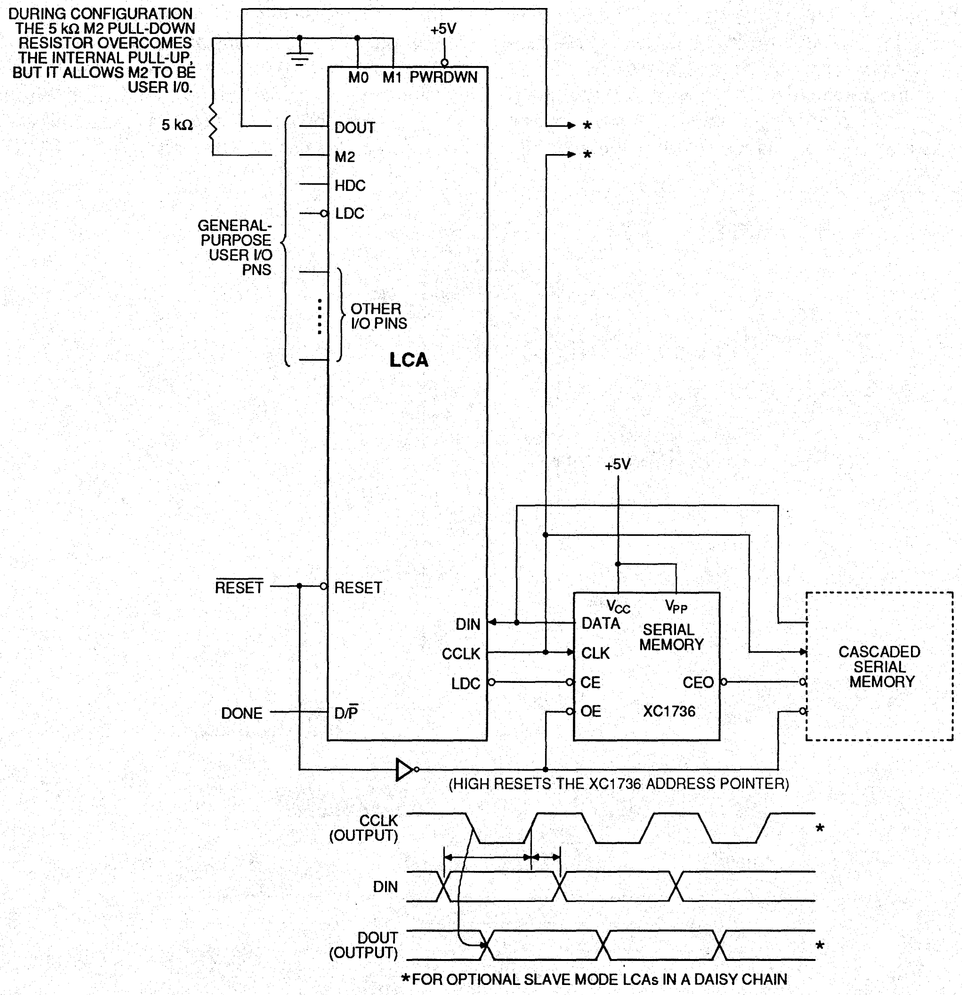
Cell Array can load itself from high memory and enable the processor to begin execution once configuration is completed. The DONE / $\overline{\text{PROG}}$ output pin can be used to hold the processor in a Reset state until the Logic Cell Array has completed the configuration process.

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

Peripheral Mode

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 ($\overline{\text{WS}}$), and two active-low and one active-high chip selects ($\overline{\text{CS0}}$ $\overline{\text{CS1}}$ 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 bit until all bits of the word are written, then continuing in the



1104 17

Figure 15b. Master Serial Mode. The one time programmable XC1736 Serial Configuration PROM supports automatic loading of configuration programs up to 36K bits. Multiple XC1736s can be cascaded to support additional LCAs. An XC2000 LDC signal can provide an XC1736 inhibit as the user I/Os become active.

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 configuration data while providing a HIGH DOUT to down-

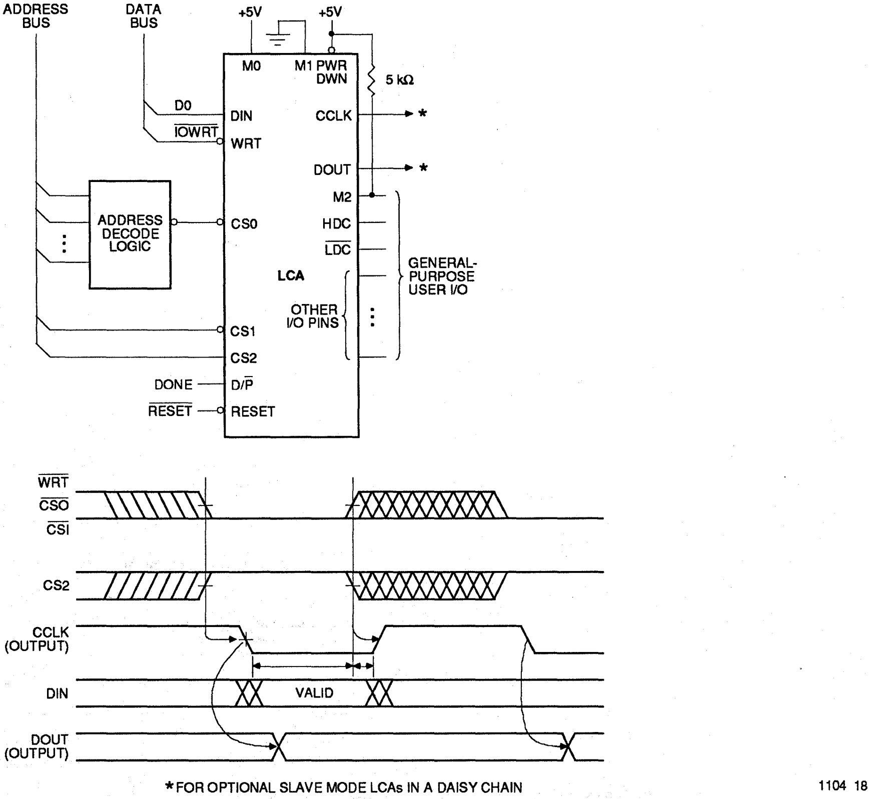


Figure 16. XC2000 Peripheral Mode. Configuration data are loaded using serialized data from a microprocessor.

stream 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

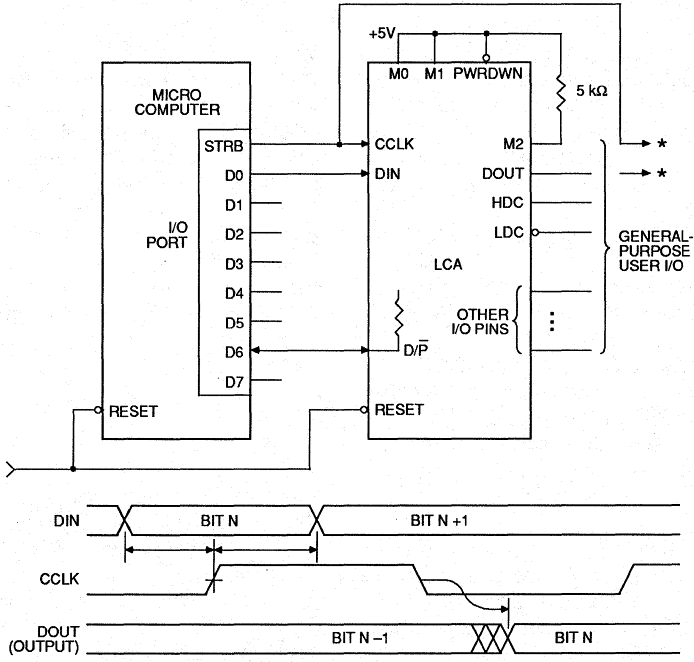
In addition to the normal user logic functions and interconnect, the configuration data include control for several special functions:

- Input thresholds
- Readback enable
- 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.



1104 19

*FOR OPTIONAL SLAVE MODE LCAs IN A DAISY CHAIN

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. Read back 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 rewritten 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 resampled. 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 6μs 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

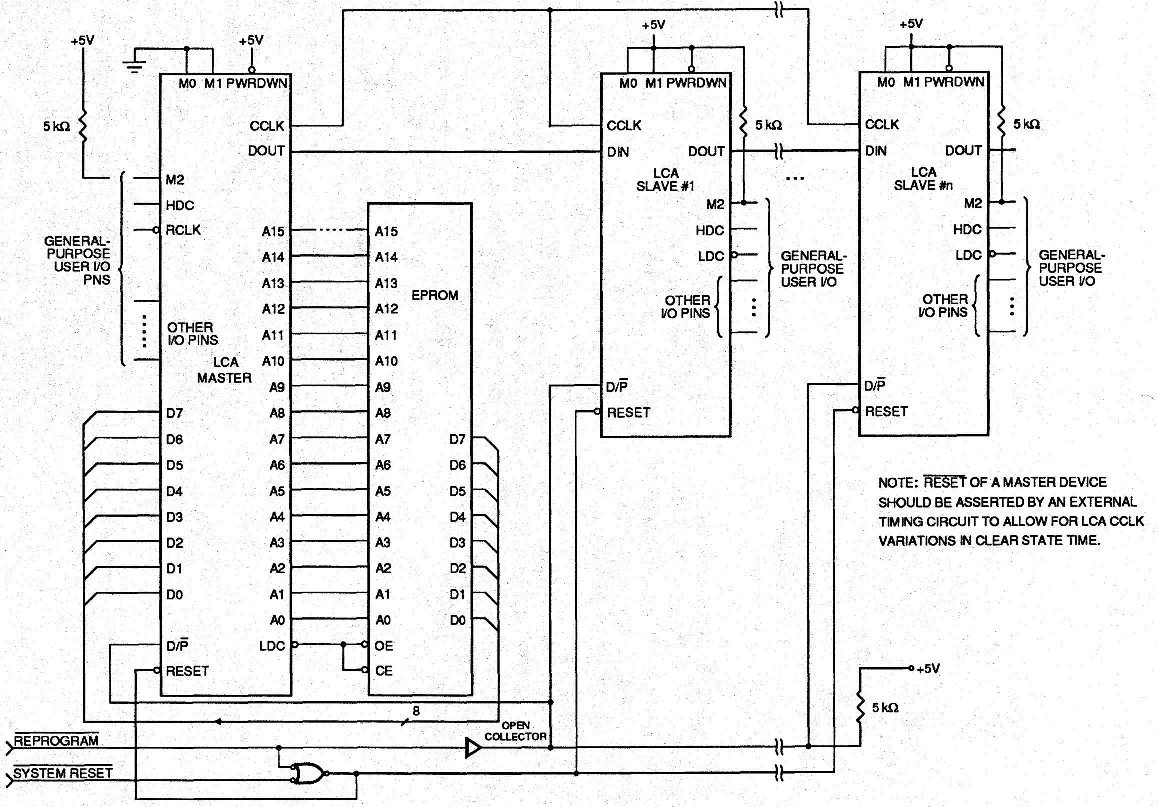
Because the control store of the Logic Cell Array is a CMOS static memory, its cells require only a very low standby current for data retention. In some systems, this low data retention current characteristic facilitates preserving configurations in the event of a primary power loss. The Logic Cell Array 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 volts, the required current is typically on the order of 50 nanoamps. 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 nanoseconds. This short delay provides very good performance under asynchronous clock and data conditions. Short loop delays minimize the probability



NOTE: RESET OF A MASTER DEVICE SHOULD BE ASSERTED BY AN EXTERNAL TIMING CIRCUIT TO ALLOW FOR LCA CCLK VARIATIONS IN CLEAR STATE TIME.

Figure 18. Master Mode with Daisy Chain

1104 20

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.)

of a metastable condition which can result from assertion of the clock during data transitions. Because of the short loop delay characteristic in the Logic Cell Array, 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 thé 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.

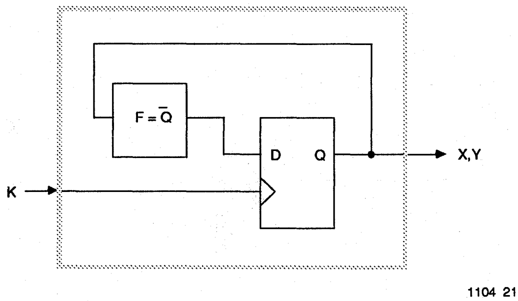


Figure 19. Logic Block Configuration for Toggle Rate Measurement

Logic Block Performance

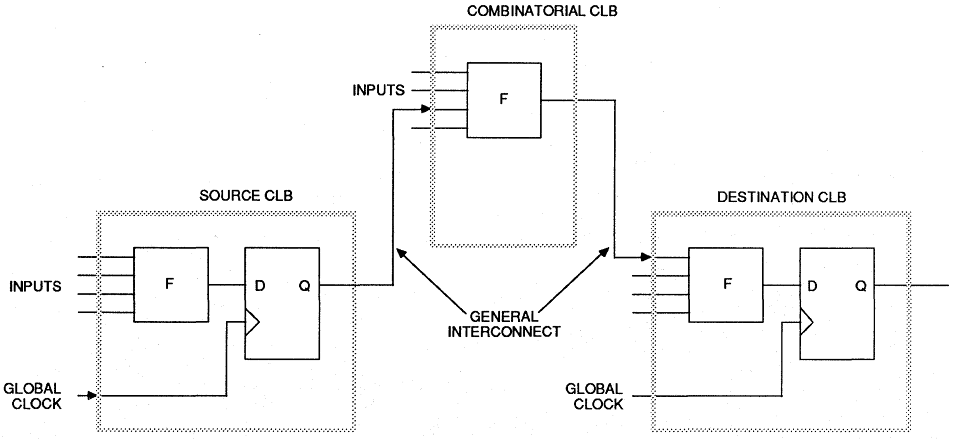
Logic block propagation times are measured from the interconnect point at the input of the combinatorial logic to the output of the block in the interconnect area. Combinatorial performance is independent of logic function because of the table look-up based implementation. Timing is different when the 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 Figures 22 and 23.

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.

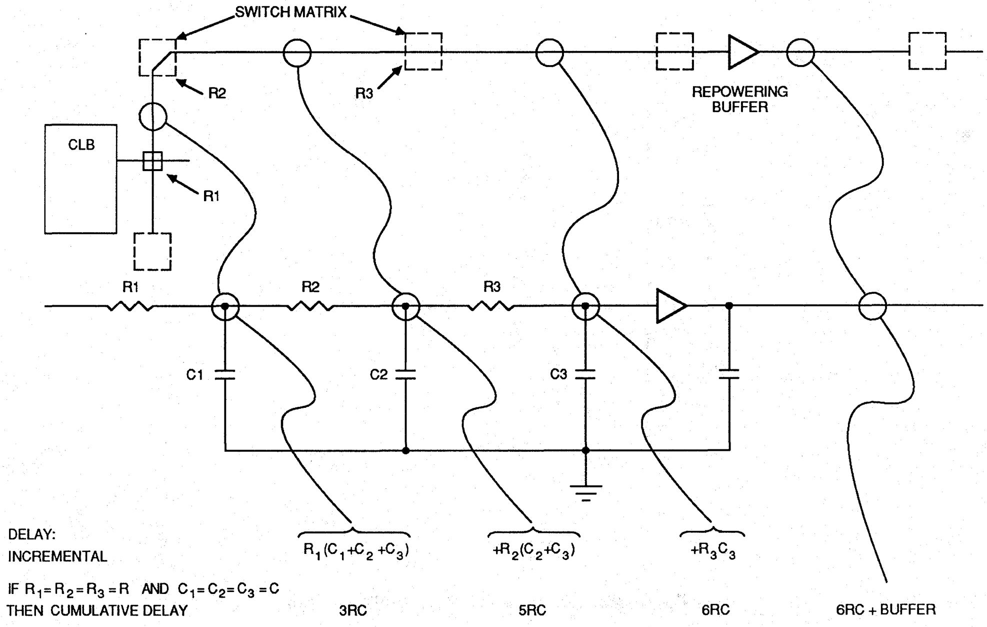
The single metal segment used for long lines exhibits low resistance from end to end, but relatively high capacitance. Signals driven through a programmable switch will have the additional impedance of the switch added to their normal drive impedance.

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.



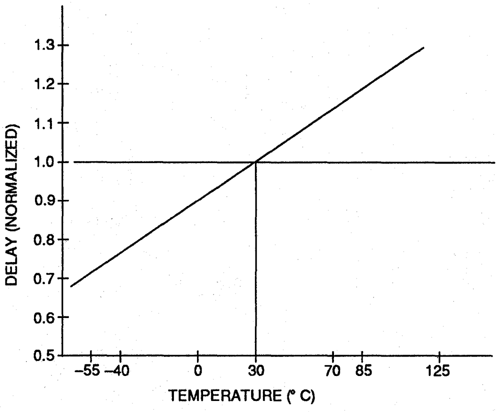
1104 22

Figure 20. Typical Logic Path



1104 23

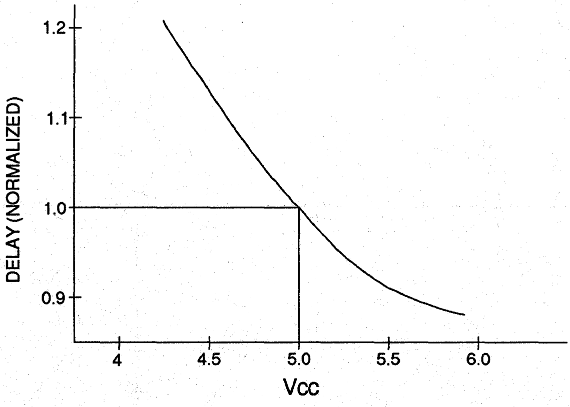
Figure 21. Interconnection Delay Factors Example



NOTE: NORMALIZED FOR 30°C

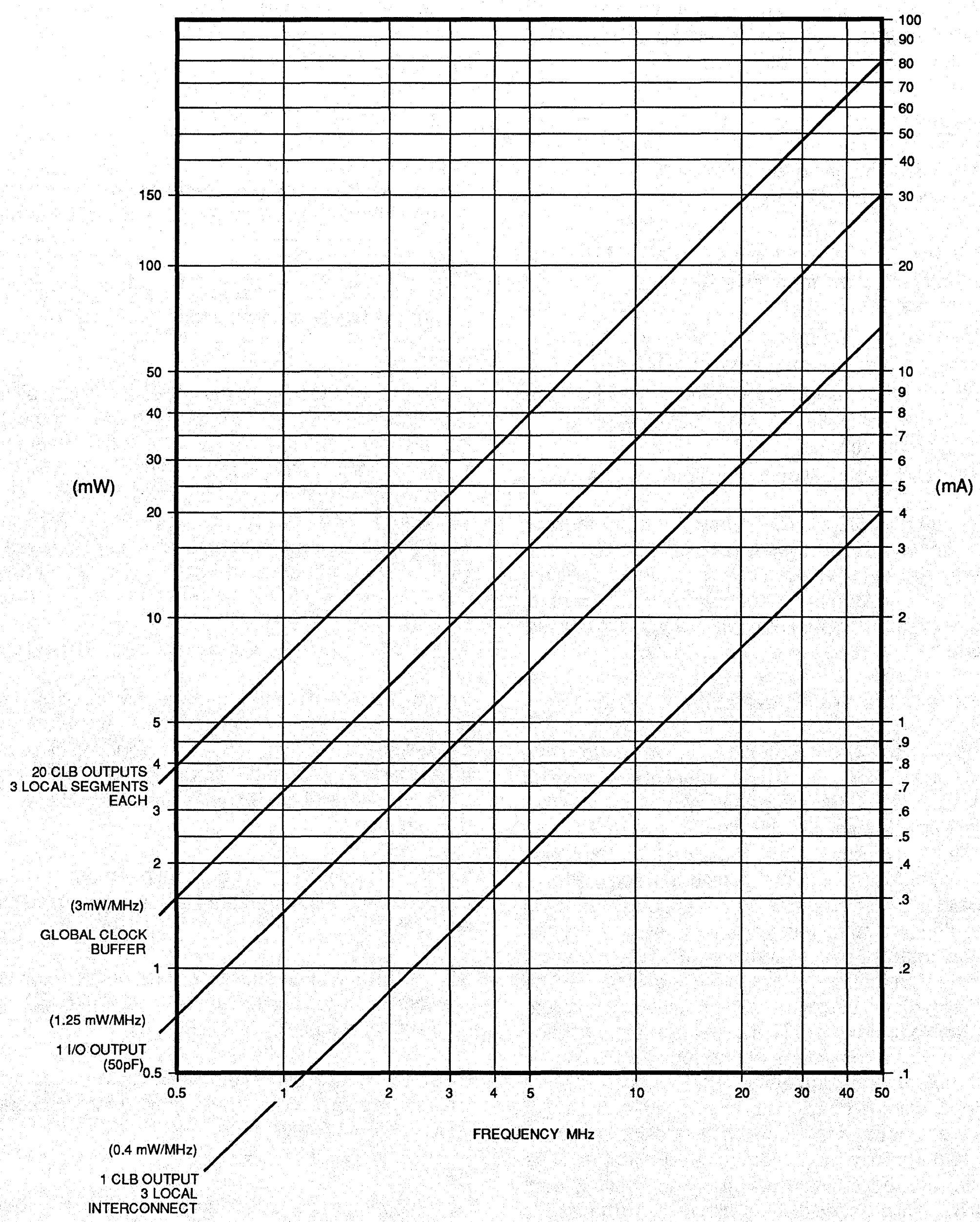
1104 25

Figure 22. Delay *vs.* Temperature



1104 26

Figure 23. Delay *vs.* Power Supply Voltage



1104 27

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, 1.5 to 5.5 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 Schema, 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 XILINX XACTOR In-Circuit Design Verifier directly in the target system and/or the P-SILOS logic simulator.

PIN DESCRIPTIONS

1. Permanently Dedicated Pins.

Vcc

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.

$\overline{\text{PWRDWN}}$

A LOW on this CMOS compatible input stops all internal activity to minimize Vcc power and puts all output buffers in a high impedance state, but Configuration is retained. When the $\overline{\text{PWRDWN}}$ pin returns HIGH the device returns to operation with the same sequence of buffer enable and DONE/$\overline{\text{PROGRAM}}$ as at the completion of configuration. All internal storage elements are reset. If not used, $\overline{\text{PWRDWN}}$ must be tied to Vcc.

$\overline{\text{RESET}}$

This is an active low input which has three functons.

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 $\overline{\text{RESET}}$ are complete, the levels of the "M" lines are sampled and configuration begins.

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

If $\overline{\text{RESET}}$ is asserted after configuration is complete, it will provide an asynchronous reset of all IOB and CLB storage elements of the LCA.

$\overline{\text{RESET}}$ 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. LCAs in Slave mode use it as a clock input. During a Readback operation it is a clock input for the configuration data being shifted out.

DONE

The DONE output is configurable as open drain with or without an internal pull-up resistor. At the completion of configuration, the circuitry of the LCA becomes active in a synchronous order, and DONE may be programmed to occur one cycle before or after that.

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

As a Read Trigger, a LOW-to-HIGH input transition, after configuration is complete, will initiate a Readback of configuration and storage element data by CCLK. This operation may be limited to a single request, or be inhibited altogether, by selecting the appropriate readback option when generating the bit stream.

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 Read back is to be used, a 5 kΩ resistor should be used to define mode level inputs.

$\overline{\text{RDATA}}$

As an active low Read Data, after configuration is complete, this pin is the output of the readback data.

2. User I/O Pins that can have special functions.

M2

As Mode 2, this input has a passive pullup during configuration. 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 becomes a user programmable I/O pin.

HDC

High During Configuration is held at a HIGH level by the LCA until after configuration. It is available as a control output indicating that configuration is not yet completed. After configuration this pin is a user I/O pin.

$\overline{\text{LDC}}$

Low During Configuration is held at a LOW level by the LCA until after configuration. It is available as a control output indicating that configuration is not yet completed. It is particularly useful in Master mode as a LOW enable for an EPROM. After configuration this pin is a user I/O pin.

If used as a LOW EPROM enable, it must 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, which are used in the peripheral mode to control configuration data entry. The assertion of all four generates a LOW CCLK. In master mode, these pins become part of the parallel configuration byte (D4,D3,D2,D1). After configuration is complete, they are user-programmed I/O.

$\overline{\text{RCLK}}$

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 8 pins represent the parallel configuration byte for the parallel Master mode. After configuration is complete they are user programmed I/O pin.

A0–A15

This set of 16 pins present an address output for a configuration EPROM during Master parallel mode. After configuration is complete they are user programmed I/O pin.

DIN

This user I/O pin is used as serial Data input during Slave or Master Serial configuration. This pin is Data 0 input in Master or Peripheral configuration mode.

DOUT

This user I/O pin is used during configuration to output serial configuration data for daisy-chained slaves' Data In.

3. Unrestricted User I/O Pins.

I/O

A pin which may be programmed by the user to be Input and/or Output pin following configuration. Some of these pins present a high impedance pull-up (see next page) or perform other functions before configuration is complete (see above).

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	
$\overline{\text{PWRDWN}}$ (I)					7	10	B2	$\overline{\text{PWR DWN}}$
<<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	
						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	$\overline{\text{RDATA}}$ (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
$\overline{\text{RESET}}$ (I)					31	44	K10	$\overline{\text{RESET}}$
DONE (O)					32	45	K11	$\overline{\text{PROG}}$ (I)
<<HIGH>>					33	46	J10	XTL1 OR I/O
						47	J11	I/O
			D5 (I)		34	48	H10	
						49	H11	
		$\overline{\text{CS0}}$ (I)	D4 (I)		35	50	G10	
		$\overline{\text{CS1}}$ (I)	D3 (I)		36	51	G11	
VCC						52	F10	VCC
<<HIGH>>						53	F11	I/O
		CS2 (I)	D2 (I)		37	54	E10	
						55	E11	
		$\overline{\text{WRT}}$ (I)	D1 (I)		38	56	D10	
$\overline{\text{RCLK}}$			$\overline{\text{RCLK}}$		39	57	D11	
DIN (I)			D0 (I)		40	58	C10	
DOUT (O)					41	59	C11	
CCLK (0)	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 28

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	
$\overline{\text{PWRDWN}}$ (I)					10	12	B2	$\overline{\text{PWR DWN}}$
<<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	
						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	$\overline{\text{RDATA}}$ (O)
M0 (LOW)	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
$\overline{\text{RESET}}$ (I)					44	54	K10	$\overline{\text{RESET}}$
DONE (O)					45	55	J10	$\overline{\text{PROG}}$ (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	
		$\overline{\text{CS0}}$ (I)	D4 (I)		50	62	G11	
		$\overline{\text{CS1}}$ (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	
		$\overline{\text{WRT}}$ (I)	D1 (I)		56	70	D10	
$\overline{\text{RCLK}}$			$\overline{\text{RCLK}}$		57	71	C11	
DIN (I)			D0 (I)		58	72	B11	
DOUT (O)					59	73	C10	
CCLK (0)	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 29

Table 2b. XC2018 Pin Assignments

PARAMETRICS

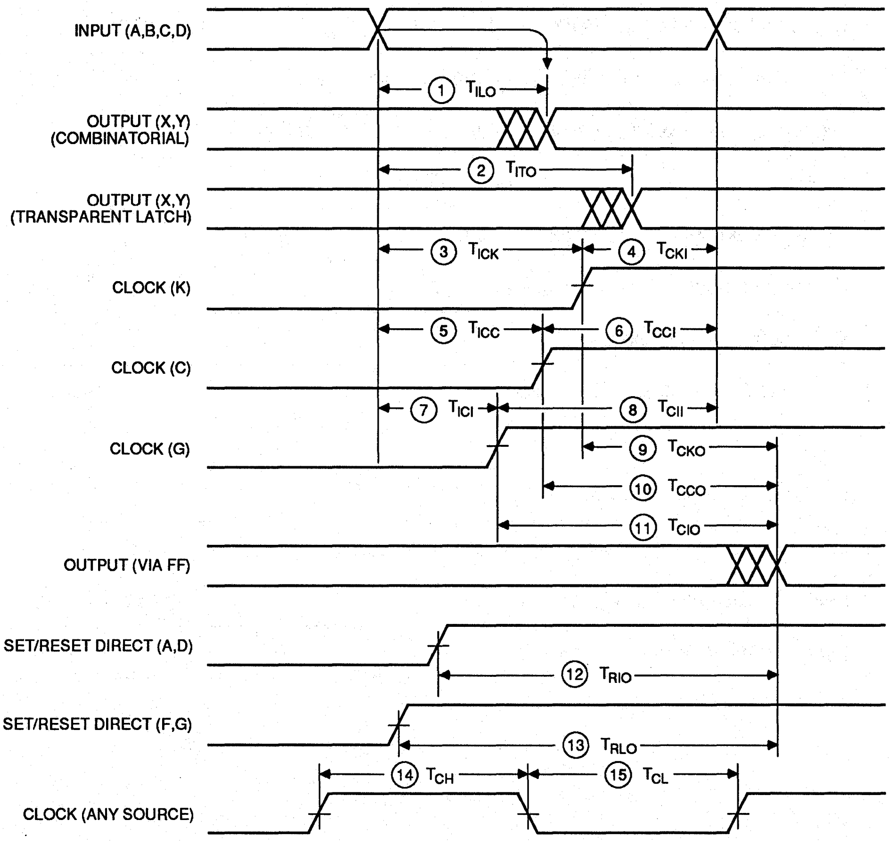
Absolute Maximum Ratings			Units
V_{CC}	Supply voltage relative to GND	–0.5 to 7.0	V
V_{IN}	Input voltage with respect to GND	–0.5 to V_{CC} + 0.5	V
V_{TS}	Voltage applied to three-state output	–0.5 to V_{CC} + 0.5	V
T_{STG}	Storage temperature (ambient)	–65 to +150	°C
T_{SOL}	Maximum soldering temperature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

*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.

Recommended Operating Conditions		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	V_{CC}	V
V_{ILT}	Low-level input voltage — TTL configuration	0	0.8	V
V_{IHC}	High-level input voltage — CMOS configuration	.7 V_{CC}	V_{CC}	V
V_{ILC}	Low-level input voltage — CMOS configuration	0	.2 V_{CC}	V

Electrical Characteristics Over Operating Conditions			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} min)			0.32	V
V_{OH}	High-level output voltage (@ I_{OH} = –4.0 ma V_{CC})	Industrial	3.76		V
V_{OL}	Low-level output voltage (@ I_{OL} = 4.0 ma V_{CC})			0.37	V
V_{OH}	High-level output voltage (@I_{OH} = –4.0 ma V_{CC})	Military	3.7		V
V_{OL}	Low-level output voltage (@ I_{OH} = 4.0 ma V_{CC})			0.4	V
I_{CCO}	Quiescent operating power supply current				
	CMOS thresholds (@ V_{CC} Max)			5	mA
	TTL thresholds (@ V_{CC} Max)			10	mA
I_{CCPD}	Power-down supply current (@ V_{CC} = 5.0 V)			0.5	mA
I_{IL}	Leakage Current Commercial/Industrial Temperature		–10	+10	µA
	Leakage Current Military –55°C to 125°C		–100	+100	µA
C_{IN}	Input capacitance (sample tested)			10	pF

CLB SWITCHING CHARACTERISTICS



1104 30

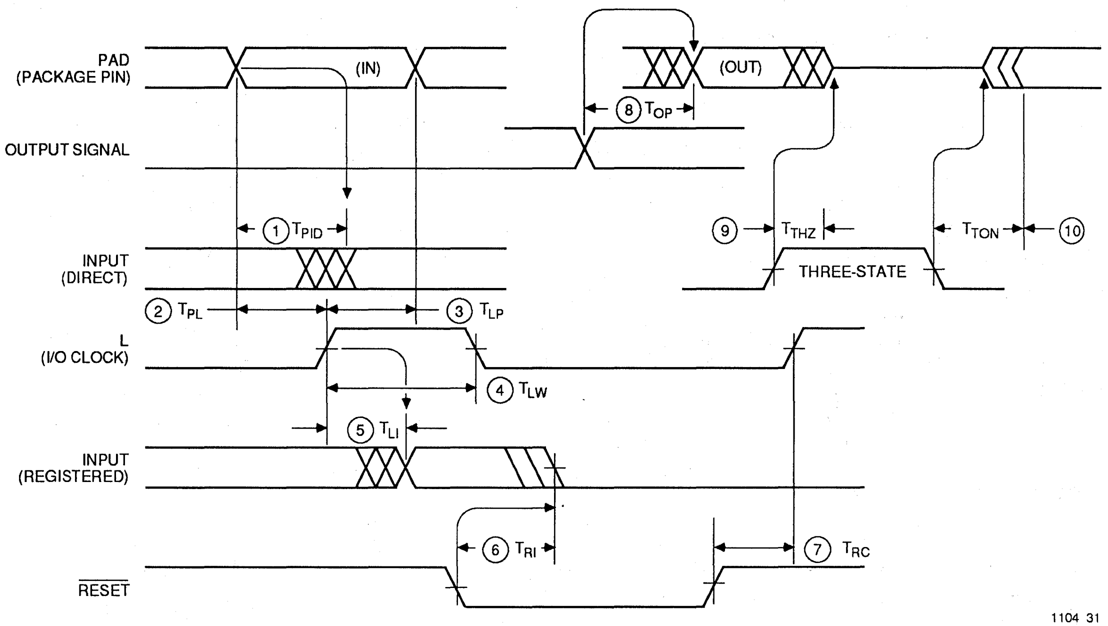
CLB SWITCHING GUIDELINES (Continued)

		Speed Grade		-33		-50		-70		-100		Units
	Description	Symbol		Min	Max	Min	Max	Min	Max	Min	Max	
Logic Input to Output	Combinatorial	1	T_{ILO}		20		15		10		7.5	ns
	Transparent latch	2	T_{ITO}		25		20		14		10	ns
	Additional for Q through F or G to out		T_{QLO}		13		8		6		6	ns
K Clock	To output	9	T_{CKO}		20		15		10		7	ns
	Logic-input setup	3	T_{ICK}	12		8		7		6		ns
	Logic-input hold	4	T_{CKI}	0		0		0		0		ns
C Clock	To output	10	T_{CCO}		25		19		13	9		ns
	Logic-input setup	5	T_{ICC}	12		9		6		5		ns
	Logic-input hold	6	T_{CCI}	6		0		0		0		ns
Logic Input to G Clock	To output	11	T_{CIO}		37		27		20		13	ns
	Logic-input setup	7	T_{ICI}	6		4		3		2		ns
	Logic-input hold	8	T_{CII}	9		5		4		3		ns
Set/Reset direct	Input A or D to out	12	T_{RIO}		25		22		16		10	ns
	Through F or G to out	13	T_{RLO}		37		28		21		14	ns
	Master Reset pin to out		T_{MRQ}		35		25		20		17	ns
	Separation of set/reset		T_{RS}	17		9		7		5*		ns
	Set/Reset pulse-width		T_{RPW}	12		9		7		5*		ns
Flip-flop Toggle rate	Q through F to flip-flop		F_{CLK}	33		50		70		100*		MHz
Clock	Clock High	14	T_{CH}	12		8		7		6		ns
	Clock Low	15	T_{CL}	12		8		7		6		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 paramenters by 20%.

IOB SWITCHING CHARACTERISTIC GUIDELINES

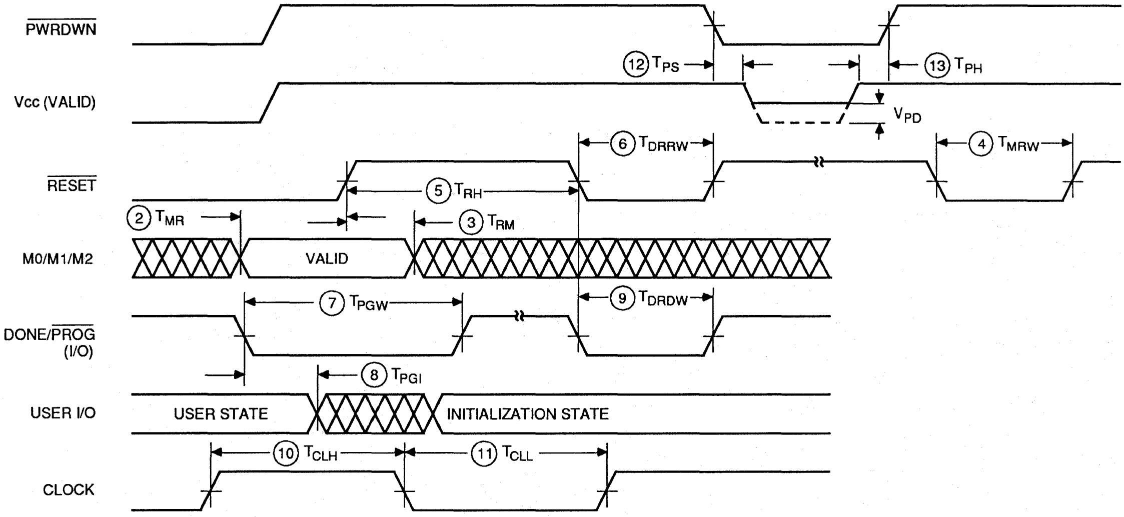


		Speed Grade		-33		-50		-70		-100		Units
	Description	Symbol		Min	Max	Min	Max	Min	Max	Min	Max	
Pad (package pin)	To input (direct)	1	T_{PID}		12		8		6		4	ns
I/O Clock	To input (storage)	5	T_{LI}		20		15		11		8	ns
	To pad-input setup	2	T_{PL}	12		8		6		4		ns
	To pad-input hold	3	T_{LP}	0		0		0		0		ns
	Pulse width	4	T_{LW}		12		9		7		5*	ns
	Frequency			33		50		70		100*		MHz
Output	To pad (output enabled)	8	T_{OP}		15		12		9		7	ns
Three-state	To pad begin hi-Z	9	T_{THZ}		25		20		15		11	ns
	To pad end hi-Z	10	T_{TON}		25		20		15		13	ns
$\overline{\text{RESET}}$	To input (storage)	6	T_{RI}		40		30		25		17	ns
	To input clock	7	T_{RC}	35		25		20		14		ns

Note: Timing is measured at 0.5 VCC levels with 50pF output load.

* These parameters are for clock pulses generated within an LCA. For an externally applied clock, derate these paramenters by 20%.

GENERAL LCA SWITCHING CHARACTERISTICS

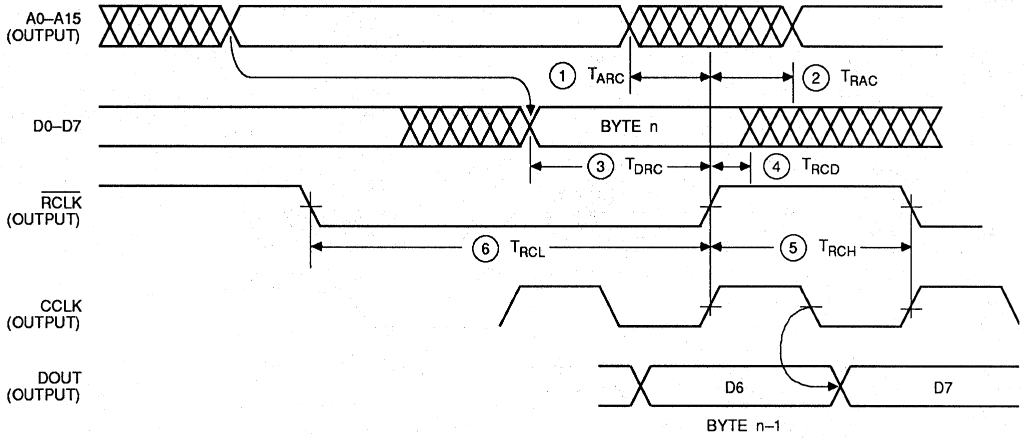


1104 32

		Speed Grade		-33		-50		-70		-100		Units
	Description	Symbol		Min	Max	Min	Max	Min	Max	Min	Max	
RESET[2]	M2, M1, M0 setup	2	TMR	60		60		60		60		ns
	M2, M1, M0 hold	3	TRM	60		60		60		60		ns
	Width—FF Reset	4	TMRW	150		150		150		150		ns
	High before RESET[5]	5	TRH	6		6		6		6		µs
	Device Reset[5]	6	TDRRW	6		6		6		6		µs
DONE/PROG	Progam width (low)	7	TPGW	6		6		6		6		µs
	Initialization	8	TPGI		7		7		7		7	µs
	Device Reset [5]	9	TDRDW	6		6		6		6		µs
CLOCK	Clock (high)	10	TCLH	12		8		7		5		ns
	Clock (low)	11	TCLL	12		8		7		5		ns
PWR DWN[4]	Setup to Vcc	12	TPS	0		0		0		0		ns
	Hold from Vcc	13	TPH	0		0		0		0		ns
	Power Down		VPD	2.0		2.0		2.0		2.0		V

Notes: 1. Vcc must rise from 2.0 Volts to Vcc minimum in less than 10 ms for master mode.
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. PWR DWN must be inactive until after initialization state is complete.
5. After RESET is high, RESET = D/P = LOW for 6µs will abort to CLEAR.

MASTER MODE PROGRAMMING SWITCHING CHARACTERISTICS



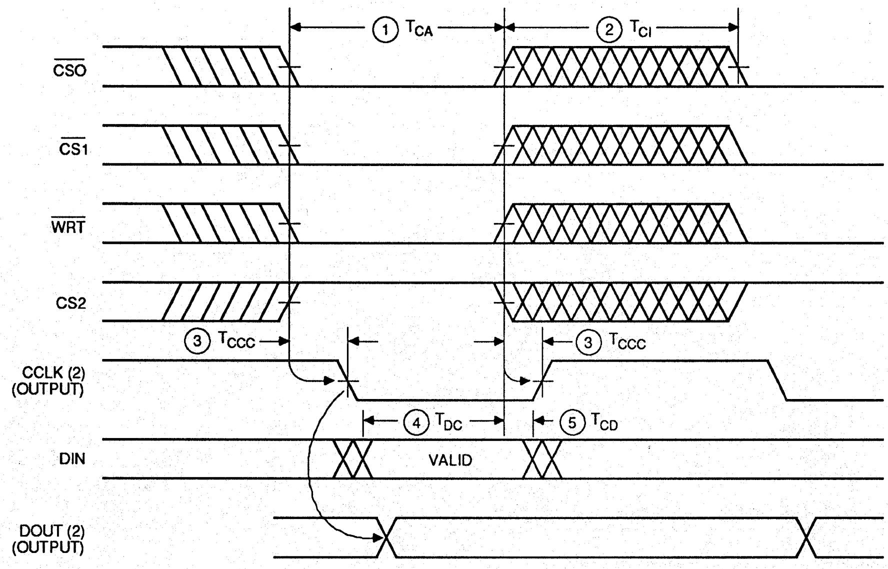
1104 33

		Speed Grade		-33		-50		-70		-100		Units
	Description	Symbol		Min	Max	Min	Max	Min	Max	Min	Max	
RCLK	From address invalid	1	TARC		0		0		0		0	ns
	To address valid	2	TRAC		200		200		200		200	ns
	To data setup	3	TDRC	60		60		60		60		ns
	To data hold	4	TRCD	0		0		0		0		ns
	$\overline{\text{RCLK}}$ high	5	TRCH	600		600		600		600		ns
	$\overline{\text{RCLK}}$ low	6	TRCL	4.0		4.0		4.0		4.0		µs

Notes: 1. CCLK and DOUT timing are the same as for slave mode.
2. At power-up, Vcc must rise from 2.0 V to Vcc min. in less than 10 ms.

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

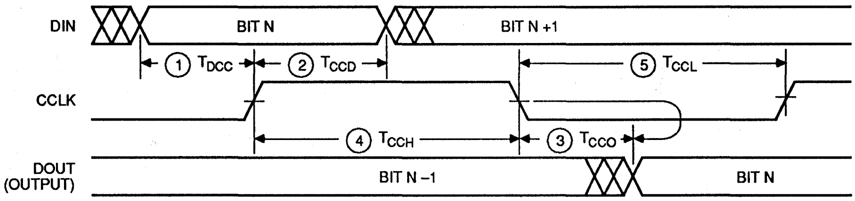


1104 34

	Speed Grade			-33		-50		-70		-100		Units
	Description	Symbol		Min	Max	Min	Max	Min	Max	Min	Max	
Controls[1] ($\overline{\text{CS0}}$, $\overline{\text{CS1}}$, CS2, $\overline{\text{WRT}}$)	Active (last active input to first inactive)	1	T_{CA}	0.25	5.0	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		0.25		µs
	CCLK[2]	3	T_{CCC}		75		75		75		75	ns
	DIN setup	4	T_{DC}	50		50		50		50		ns
	DIN hold	5	T_{CD}	0		0		0		0		ns

Notes:
1. Peripheral mode timing determined from last control signal of the logical AND of ($\overline{\text{CS0}}$, $\overline{\text{CS1}}$, CS2, $\overline{\text{WRT}}$) to transition to active or inactive state.
2. CCLK and DOUT timing are the same as for slave mode.
3. Configuration must be delayed at least 40 ms after Vcc min.

SLAVE MODE PROGRAMMING SWITCHING CHARACTERISTICS

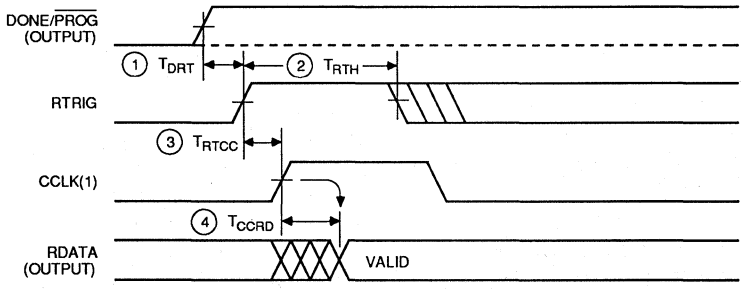


1104 35

		Speed Grade		-33		-50		-70		-100		Units
	Description	Symbol		Min	Max	Min	Max	Min	Max	Min	Max	
CCLK	To DOUT	3	TCCO		65		65		65		65	ns
	DIN setup	1	TDCC	10		10		10		10		ns
	DIN hold	2	TCCD	40		40		40		40		ns
	High time	4	TCCH	0.25		0.25		0.25		0.25		µs
	Low time	5	TCCL	0.25	5.0	0.25	5.0	0.25	5.0	0.25	5.0	µs
	Frequency		FCC		2		2		2		2	MHz

Note: Configuration must be delayed at least 40 ms after Vcc min.

PROGRAM READBACK SWITCHING CHARACTERISTICS



1104 36

		Speed Grade		-33		-50		-70		-100		Units
	Description	Symbol		Min	Max	Min	Max	Min	Max	Min	Max	
RTRIG	PROG setup	1	TDRT	300		300		300		300		ns
	RTRIG high	2	TRTH	250		250		250		250		ns
CCLK	RTRIG setup	3	TRTCC	100		100		100		100		ns
	RDATA delay	4	TCCRD		100		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

		48 PIN		68 PIN		84 PIN		100 PIN		132 PIN		164 PIN	175 PIN	
		PLASTIC DIP	CERAMIC DIP	PLASTIC PLCC	CERAMIC PGA	PLASTIC PLCC	CERAMIC PGA	PLASTIC PQFP	CERAMIC CQFP	PLASTIC PGA	CERAMIC PGA	CERAMIC CQFP	PLASTIC PGA	CERAMIC PGA
		-PD48	-CD48	-PC68	-PG68	-PC84	-PG84	-PQ100	-CQ100	-PP132	-PG132	-CQ164	-PP175	-PG175
XC2064	-33	C	I M	C I	C I M									
	-50	C	I	C I	C I M									
	-70			C I	C I M									
	-100			C	C									
XC2018	-33			C I		C I	C I M B							
	-50			C I		C I	C I M B							
	-70			C I		C I	C I M							
	-100			C		C	C							
XC3020	-50			C I		C I	C I M B	C I	C I M B					
	-70			C I		C I	C I M	C I	C I M					
	-100			C		C	C	C	C					
XC3030	-50			C I		C I	C I M	C I	C I M					
	-70			C I		C I	C I M	C I	C I M					
	-100			C		C	C	C	C					
XC3042	-50					C I	C I M	C I	C I M	C I	C I M B			
	-70					C I	C I M	C I	C I M	C I	C I			
	-100					C	C	C	C	C	C			
XC3064	-50									C I	C I M			
	-70									C I	C I			
	-100									C	C			
XC3090	-50											C I M	C I	C I M B
	-70											C I M	C I	C I M
	-100											C	C	C

1972 01

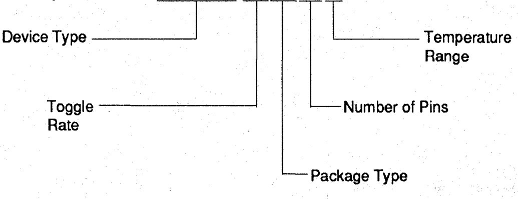
XC1736-PD8C	Plastic 8 Pin, Mini-DIP	–40°C to 85°C
XC1736-CD8M	Ceramic 8 Pin, Mini-DIP	–55°C to 125°C

LCA Temperature Options

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

ORDERING INFORMATION

Example: XC2064-70PC68C



COMPATIBLE PACKAGE OPTIONS

A range of Xilinx 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.

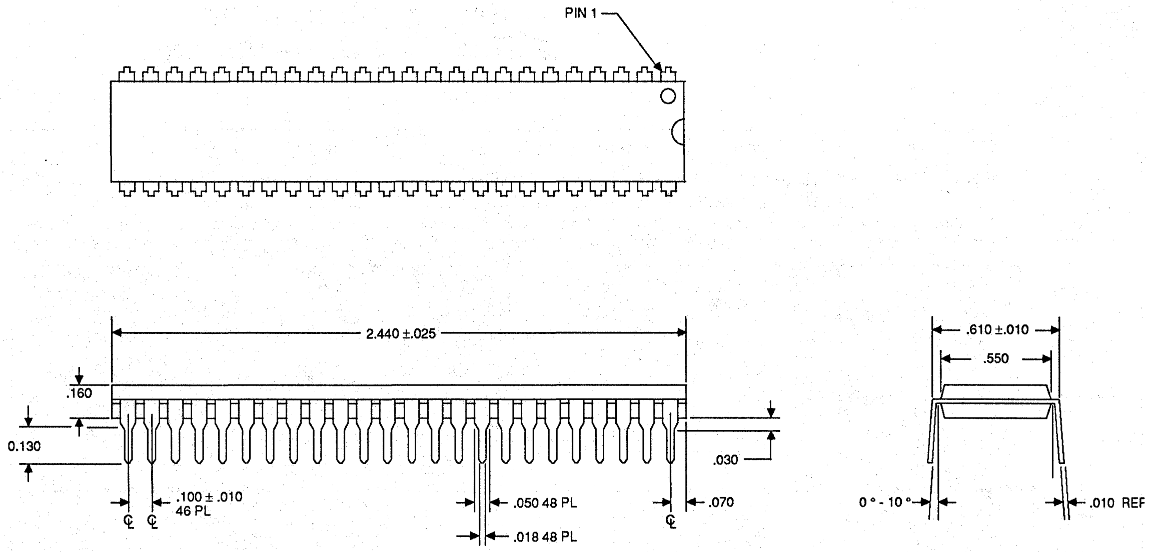
Examples:		
	PLCC 68:	2064-2018-3020-3030
	PLCC 84:	2018-3020-3030-3042
	PGA 84:	2018-3020-3030-3042
	PQFP 100:	3020-3030-3042
	CQFP100:	3020-3030-3042
	PGA 132:	3042-3064

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.

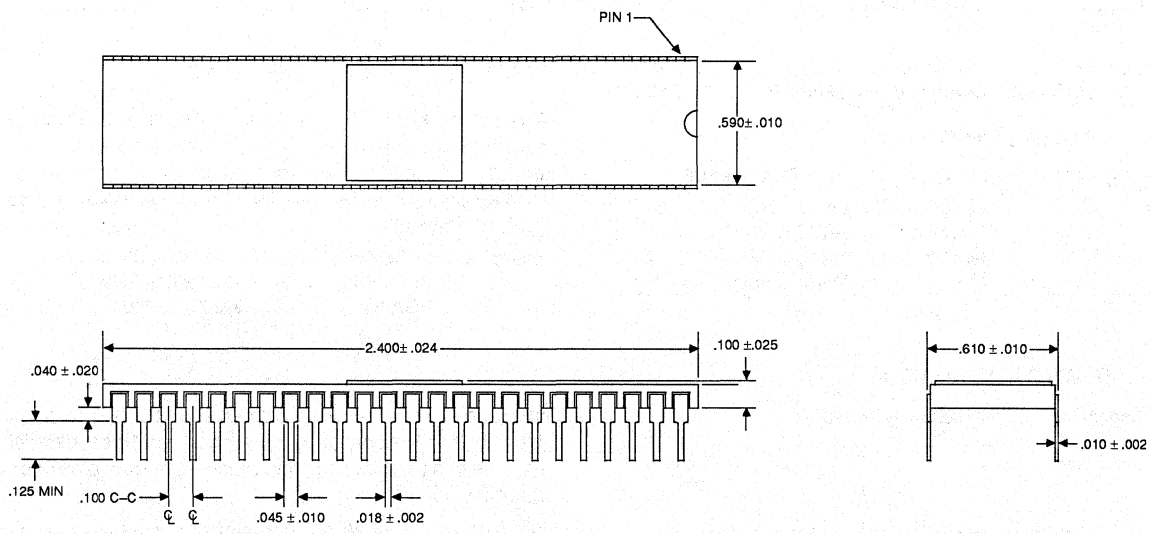
1972



DIMENSIONS IN INCHES
NOT DRAWN TO SCALE

1104 39

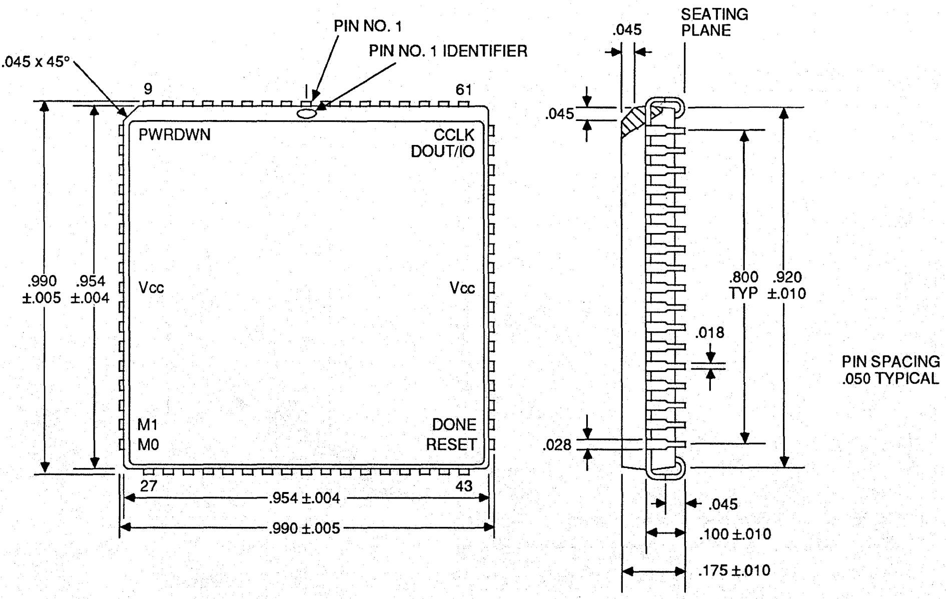
48-Pin Plastic DIP Package



1104 40

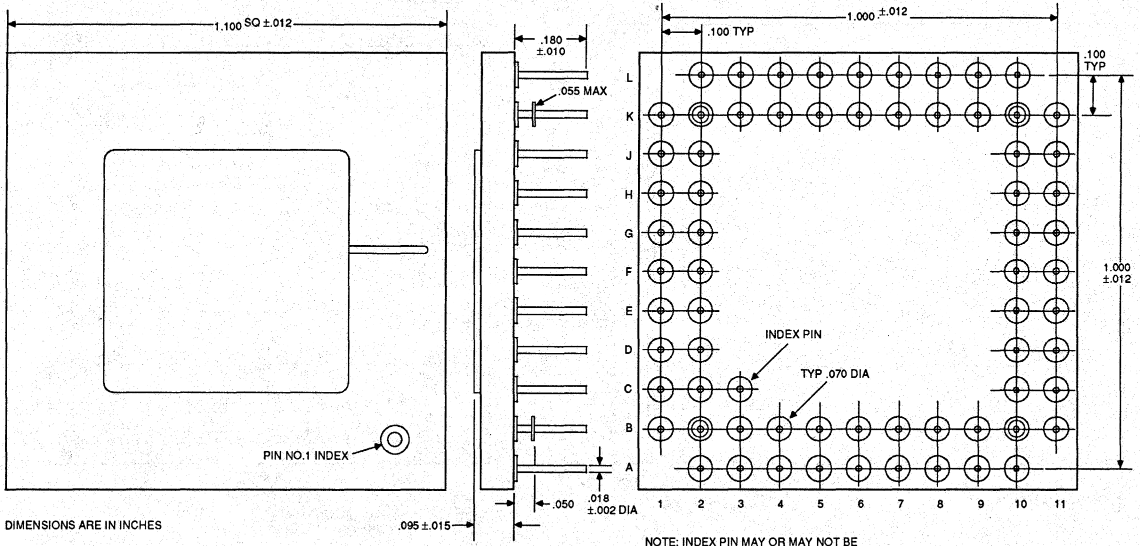
DIMENSIONS IN INCHES
NOT DRAWN TO SCALE

48-Pin Ceramic DIP Package



1104 41

68-Pin PLCC Package

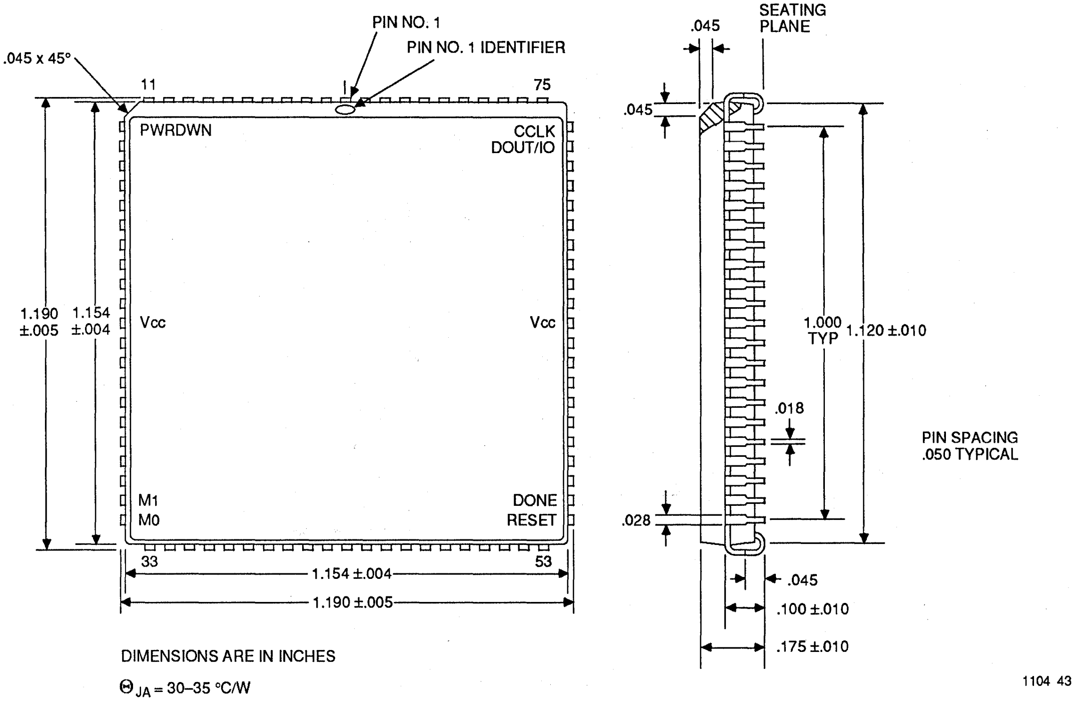


DIMENSIONS ARE IN INCHES

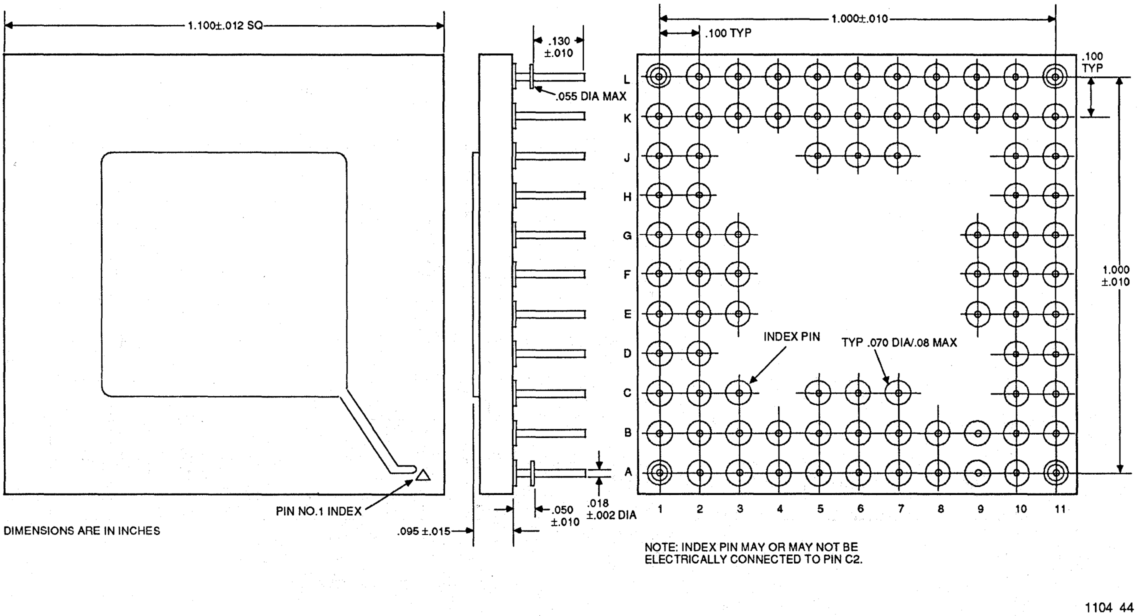
NOTE: INDEX PIN MAY OR MAY NOT BE ELECTRICALLY CONNECTED TO PIN C2.

1104 42

68-Pin PGA Package



84-Pin PLCC Package



84-Pin PGA Package

XILINX

Military Logic Cell™ Arrays

XC2018B, XC3020B, 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. The 2018B and the 3020B were introduced in the first quarter of 1989. The balance of the Xilinx 3000 series family will be available during 1989.

Device	Logic Capacity (usable gates)	Configurable Logic Blocks	User I/Os	Program Data (bits)
XC2018B	1800	100	74	17878
XC3020B	2000	64	64	14779
XC3030B	3000	100	80	22176
XC3042B	4200	144	96	30784
XC3064B	6400	224	120	46064
XC3090B	9000	320	144	64160

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
XC3030	80	CQFP 100	80	CPGA 84	74
XC3042	96	CQFP 100	82	CPGA 132	96
XC3064	120	CQFP 164	120	CPGA 132	110
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 Xilinx XC2018-50PG84B device SMD# 5962-88638. SMD numbers are under development by DESC for other Xilinx devices. Contact your Xilinx representative 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).

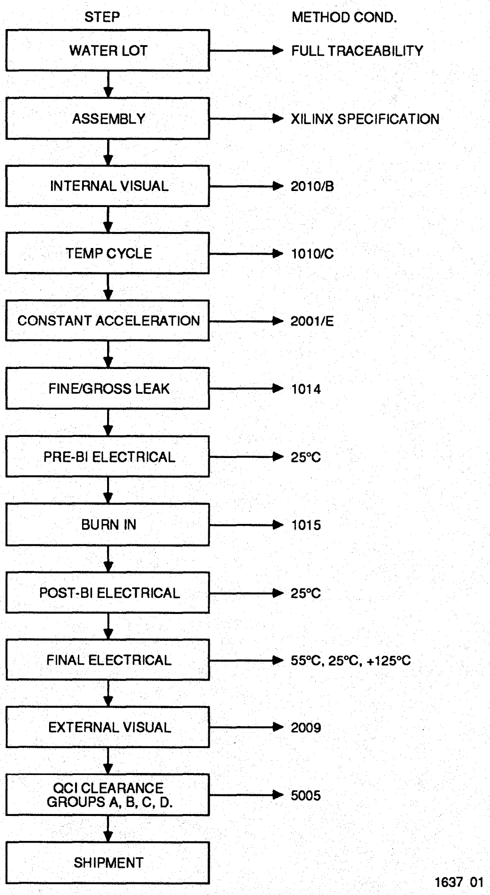
MIL-M-38510 (as invoked by MIL-STD-883)

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



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.

XILINX

XC2018B
Military Logic Cell™ Array

Product Specification. See Note 1.

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

FEATURES

- MIL–STD–883 Class B Processing. Complies with paragraph 1.2.1
- User-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)

DESCRIPTION

The Logic Cell™ Array (LCA) is a high density CMOS programmable gate array. Its patented array architecture consists of three type 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.

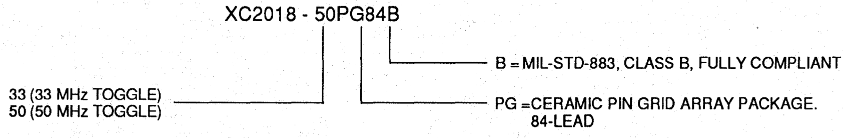
The Logic Cell Array's logic functions and interconnections are determined by the configuration program stored in internal static memory cells. On-chip logic provides for automatic loading of configuration data at power-up or on command. The program data can reside in an EEPROM, EPROM or ROM on the circuit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

ORDERING INFORMATION



XC2018 - 50PG84B

33 (33 MHz TOGGLE)
50 (50 MHz TOGGLE)

B = MIL-STD-883, CLASS B, FULLY COMPLIANT

PG =CERAMIC PIN GRID ARRAY PACKAGE.
84-LEAD

1637 02

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	
$\overline{\text{PWRDWN}}$ (I)					B2	$\overline{\text{PWR DWN}}$
<<HIGH>>					C2	I/O
					B1	
					C1	
					D2	
					D1	
					E3	
					E2	
					E1	
					F2	
VCC					F3	VCC
<<HIGH>>					G3	
					G1	
					G2	
					F1	
					H1	
					H2	
					J1	
					K1	
M1 (LOW)	M1 (HIGH)	M1 (LOW)	M1 (HIGH)	M1 (LOW)	J2	$\overline{\text{RDATA}}$ (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	I/O
					L7	
					K7	
					L6	
					L8	
					K8	
					L9	
			D7 (I)		L10	
			D6 (I)		K9	
					L11	XTL2 OR I/O
$\overline{\text{RESET}}$ (I)					K10	$\overline{\text{RESET}}$
DONE (O)					J10	$\overline{\text{PROG}}$ (I)
<<HIGH>>					K11	XTL1 OR I/O
					J11	I/O
			D5 (I)		H10	
					H11	
					F10	
					G10	
		$\overline{\text{CS0}}$ (I)	D4 (I)		G11	
		$\overline{\text{CS1}}$ (I)	D3 (I)		G9	
VCC					F9	VCC
<<HIGH>>					F11	I/O
		CS2 (I)	D2 (I)		E11	
					E10	
					E9	
					D11	
		$\overline{\text{WRT}}$ (I)	D1 (I)		D10	
RCLK			$\overline{\text{RCLK}}$		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	

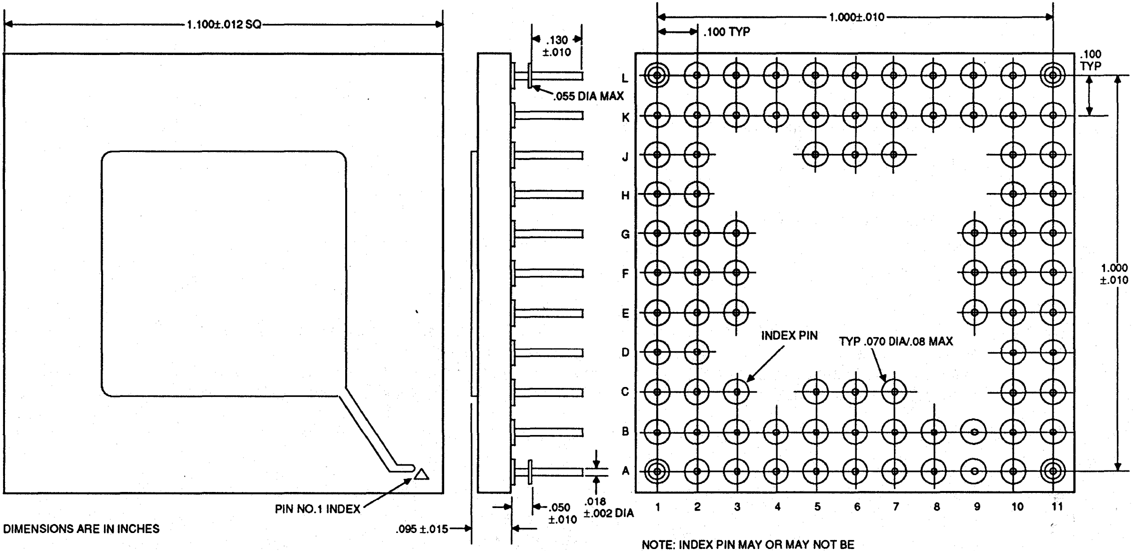
TSC0026 Rev:06

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

1637 03

CASE OUTLINE DRAWING

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

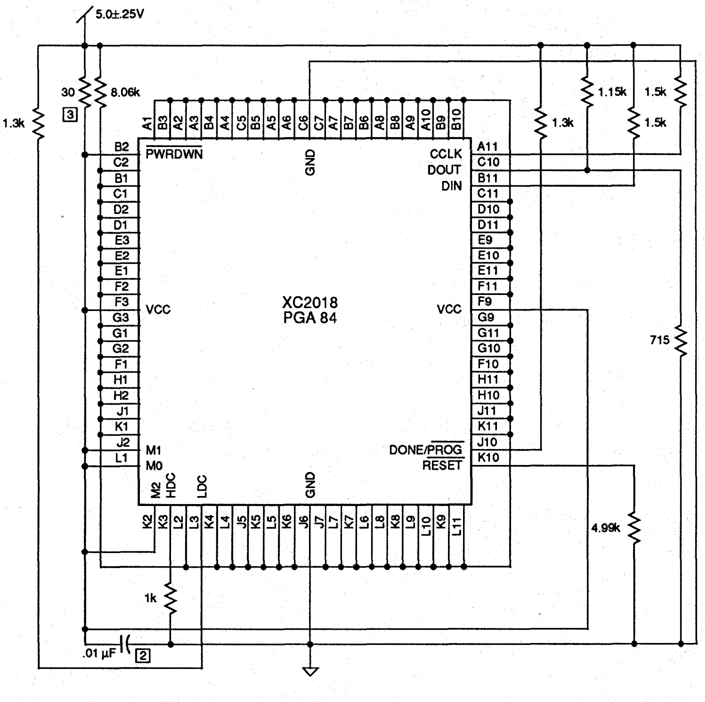


NOTE: INDEX PIN MAY OR MAY NOT BE ELECTRICALLY CONNECTED TO PIN C2.

XC2018: 84-Pin PGA Package

1637 04

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 05

XC2018B Test Specification

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

1637 Tbl 04

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 $-55°C \leq T_C \leq +125°C$ $V_{CC} = 5.0\ V \pm 10\%$	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	I_{CCO}	CMOS Inputs, Vin = V_{CC} = 5.5 V	1,2,3		10	mA
Power Supply Current		TTL Inputs, Vin = V_{CC} = 5.5 V	1,2,3		15	mA
Power-Down Supply Current	I_{CCPD}	Vin = V_{CC} = 5.5 V, $\overline{\text{PWR}}$ $\overline{\text{DWN}}$ = 0 V	1,2,3		0.5	mA
Leakage Current	I_{IL}	V_{CC} = 5.5 V, Vin = 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

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

1637 Tbl 05a

Table 1. Electrical Performance Characteristics

Test	Sym	Conditions −55°C ≤ Tc ≤ +125°C Vcc = 5.0 V ±10%	Group A Subgroups	2018-33 Limits Min	 Max	2018-50 Limits Min	 Max	Units
Switching Characteristics, Peripheral Mode Programming								
Controls ($\overline{CS}$, $\overline{WRT}$)[3,4]								
Last Active Input to First Inactive	T_{CA}	1 See Fig. 4	9,10,11	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		µs
CCLK[5]	T_{CCC}	3	9,10,11		75		75	ns
DIN Setup	T_{DC}	4	9,10,11	50		50		ns
DIN Hold	T_{CD}	5	9,10,11	5		5		ns
Switching Characteristics, Program Readback[6]								
RTRIG Setup	T_{RTH}	1 See Fig. 7	9,10,11	250		250		ns
CCLK,								
RTRIG Setup	T_{RTCC}	2	9,10,11	100		100		ns
RDATA Delay	T_{CCRD}	3	9,10,11		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	ns
T_{PID} + interconnect + 10 (T_{ITO}) + T_{OP}. Measured on 10 cols.	T_{B2}		9,10,11		288		228	ns
T_{PID} + interconnect + 10 (T_{QLO}) + T_{OP}. Measured on 10 cols.	T_{B3}		9,10,11		410		302	ns
Tested on all CLBs with T_{ICK} + interconnect.	T_{B4}		9,10,11		85		62	ns
Tested on all CLBs with T_{ICC} + interconnect.	T_{B5}		9,10,11		66		49	ns
Tested on all CLBs with T_{ICI} + interconnect.	T_{B6}		9,10,11		90		67	ns
T_{PID} + interconnect + 10 (T_{RIO}) + T_{OP}. Measured on 10 rows.	T_{B7}		9,10,11		318		269	ns
T_{PID} + interconnect + T_{PL} + T_{LI} + T_{OP}. Tested on all IOB's.	T_{B8}		9,10,11		274		204	ns

1637 Tbl 05b

Table 1. Electrical Performance Characteristics (Continued)

Test	Sym	Conditions –55°C ≤ Tc ≤ +125°C Vcc = 5.0 V ±10%	Group A Subgroups	2018-33 Limits Min	2018-33 Limits Max	2018-50 Limits Min	2018-50 Limits Max	Units

Application Guidelines, Switching, CLB[7]

Test	Sym	Conditions	Group A Subgroups	2018-33 Min	2018-33 Max	2018-50 Min	2018-50 Max	Units
Logic Input to Output, Combinatorial	TILO	1 See Fig. 1	N/A		20		15	ns
Transparent Latch	TITO	2	N/A		25		20	ns
Additional for Q Through F or G to Out	TQLO		N/A		13		8	ns
K Clock,								
To Output	TCKO	9	N/A		20		15	ns
Logic-Input Setup	TICK	3	N/A	12		8		ns
Logic-Input Hold	TCKI	4	N/A	1		1		ns
C Clock,								
To Output	TCCO	10	N/A		25		19	ns
Logic- Input Setup	TICC	5	N/A	12		9		ns
Logic-Input Hold	TCCI	6	N/A	6		1		ns
Logic Input to G Clock,								
To Output	TCIO	11	N/A		37		27	ns
Logic-Input Setup	TICI	7	N/A	6		4		ns
Logic-Input Hold	TCII	8	N/A	9		5		ns
Set/Reset Direct,								
Input A or D to Out	TRIO	12	N/A		25		22	ns
Through F or G to Out	TRLO	13	N/A		37		28	ns
Master Reset Pin to Out	TMRQ		N/A		55		45	ns
Separation of Set/Reset	TRS		N/A	17		9		ns
Set/Reset Pulse-Width	TRPW		N/A	12		9		ns
Flip-Flop Toggle Rate, Q Through F to Flip-Flop	FCLK		N/A	33		50		MHz
Clock High[8]	TCH	14	N/A	12		8		ns
Clock Low[8]	TCL	15	N/A	12		8		ns

1637 Tbl 05c

Table 1. Electrical Performance Characteristics (Continued)

Test	Sym	Conditions –55°C ≤ T_C ≤ +125°C V_{CC} = 5.0 V ±10%	Group A Subgroups	2018-33 Limits Min	 Max	2018-50 Limits Min	 Max	Units

Application Guidelines, Switching, IOB[7]

Pad (Package Pin) to Input (Direct)	T_{PID}	1 See Fig. 2	N/A		12		8	ns
I/O Clock								
To Input (Storage)	T_{LI}	5	N/A		20		15	ns
To Pad-Input Setup	T_{PL}	2	N/A	12		8		ns
To Pad Input Hold	T_{LP}	3	N/A	0		0		ns
Pulse Width	T_{LW}	4	N/A	12		9		ns
Frequency			N/A		33		50	MHz
Output,								
To Pad (Output Enable)	T_{OP}	8	N/A		15		12	ns
Three-State,								
To Pad Begin hi-Z	T_{THZ}	9	N/A		25		20	ns
To Pad End hi-Z	T_{TON}	10	N/A		25		20	ns
$\overline{\text{RESET}}$,								
To Input (Storage)	T_{RI}	6	N/A		40		30	ns
To Input Clock	T_{RC}	7	N/A		35		25	ns

Application Guidelines, Switching, Slave Mode Programming[7]

CCLK,								
To DOUT	T_{CCO}	3 See Fig. 6	N/A		100		100	ns
DIN Setup	T_{DCC}	1	N/A	10		10		ns
DIN Hold	T_{CCD}	2	N/A	40		40		ns
High Time	T_{CCH}	4	N/A	0.5		0.5		µs
Low Time	T_{CCL}	5	N/A	0.5	1.0	0.5	1.0	µs
Frequency	F_{CC}		N/A		1		1	MHz

Application Guidelines, Switching, Master Mode Programming[7,9]

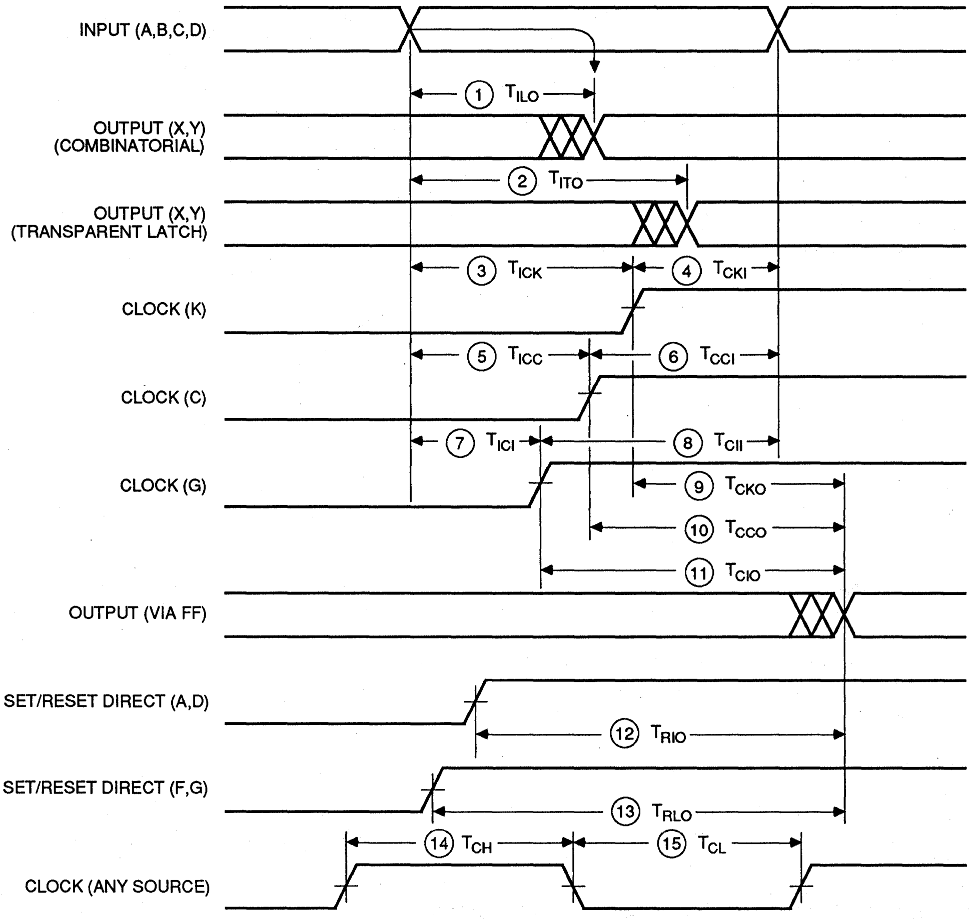
$\overline{\text{RCLK}}$,								
From Address Invalid	T_{ARC}	1 See Fig. 5	N/A		0		0	ns
To Address Valid	T_{RAC}	2	N/A		200		200	ns
To Data Setup	T_{DRC}	3	N/A	60		60		ns
To Data Hold	T_{RCD}	4	N/A	0		0		ns
$\overline{\text{RCLK}}$ High	T_{RCH}	5	N/A	600		600		ns
$\overline{\text{RCLK}}$ Low	T_{RCL}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

Application Guidelines, Switching, General LCA[7]

$\overline{\text{RESET}}$[10]								
M2, M1, M0 Setup	T_{MR}	1 See Fig. 3	N/A	1		1		µs
M2, M1, M0 Hold	T_{RM}	2	N/A	1		1		µs
Width (Low)	T_{MRW}	3	N/A	150		150		ns

1637 Tbl 05d

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

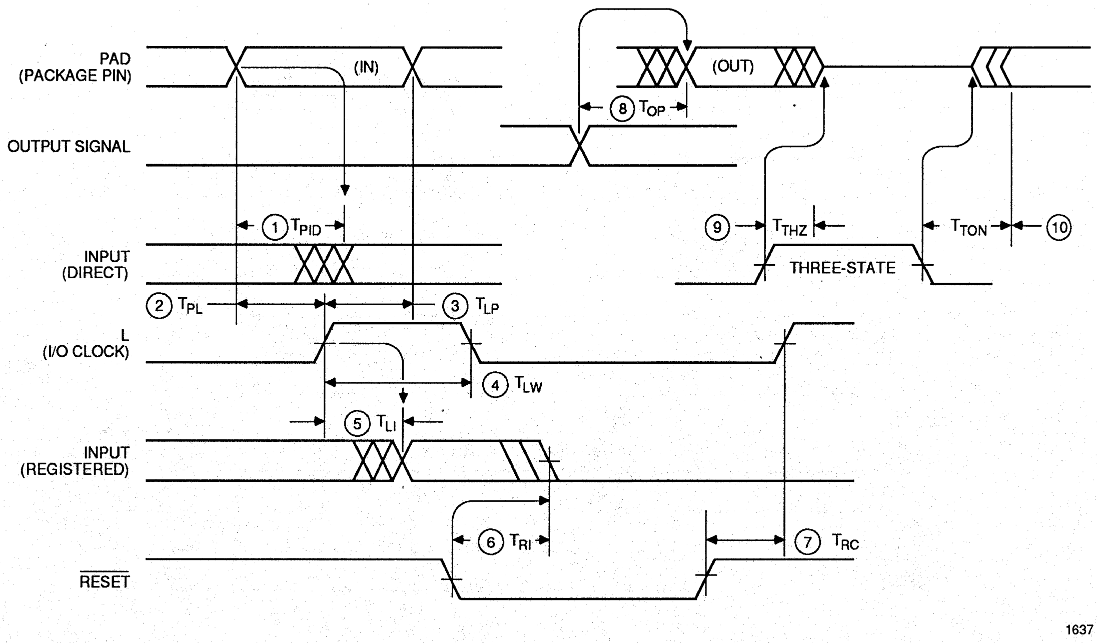


Figure 2. Switching Characteristics, IOB

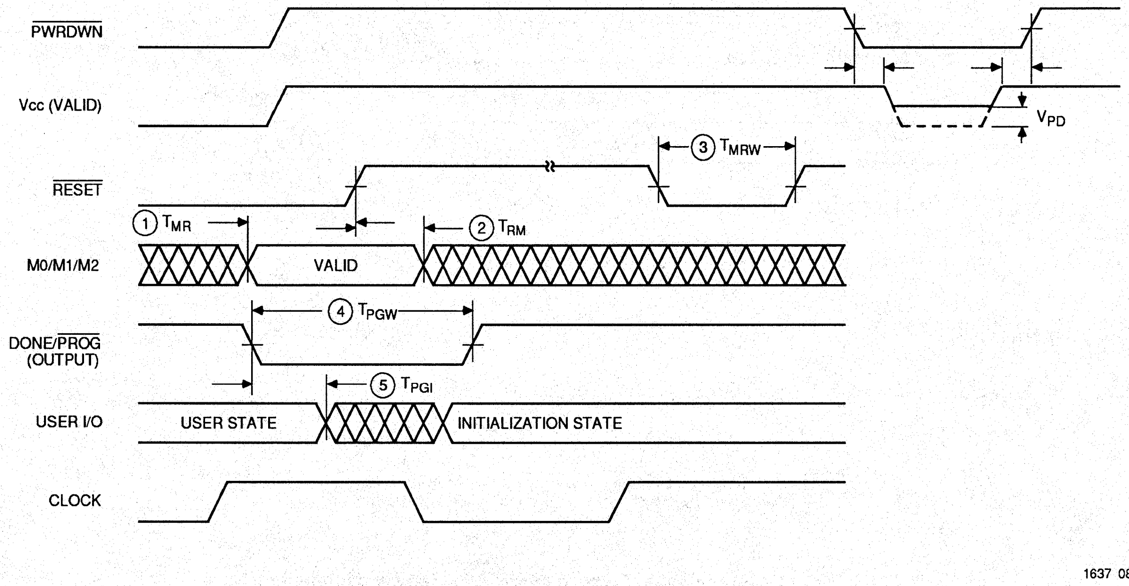
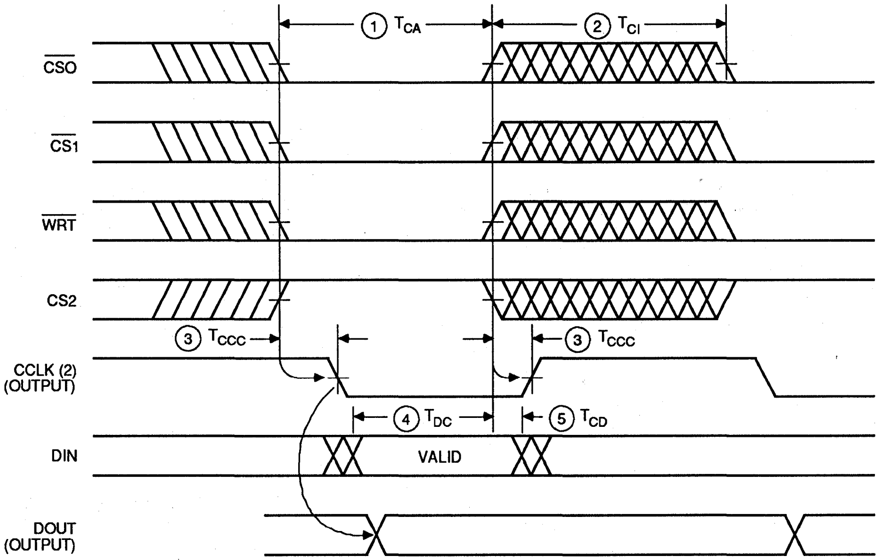
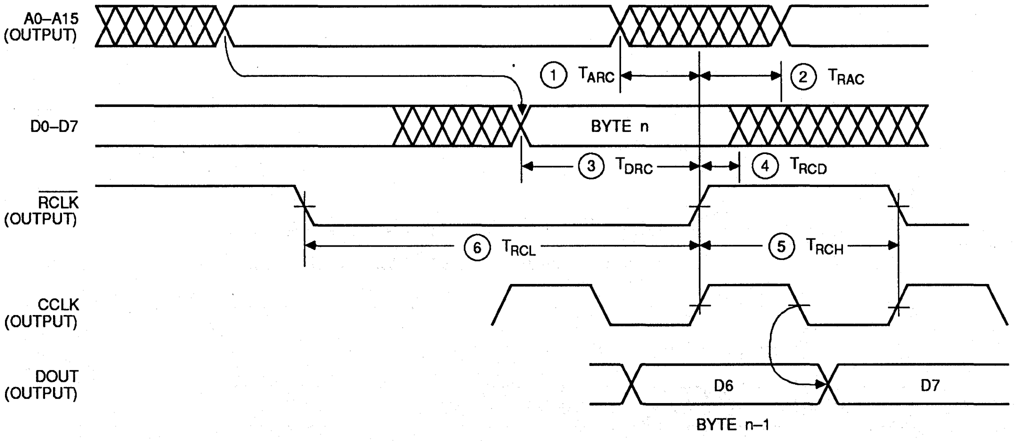


Figure 3. General LCA Switching Characteristics



1637 09

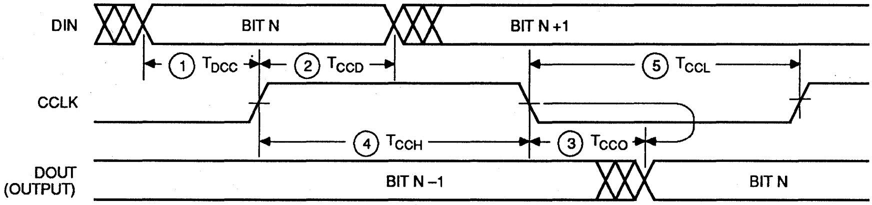
Figure 4. Peripheral Mode Programming Characteristics



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

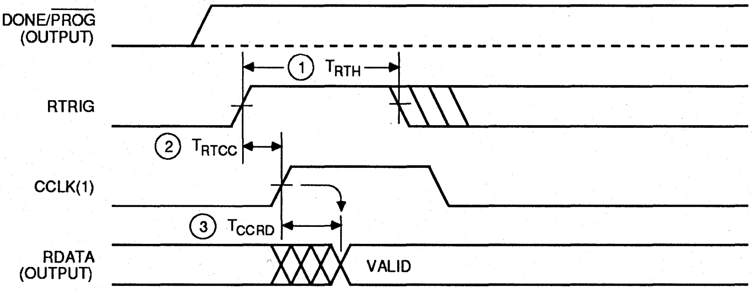
Figure 5. Master Mode Programming Switching Characteristics



Configuration must be delayed at least 40 ms after V_{CC} min.

1637 11

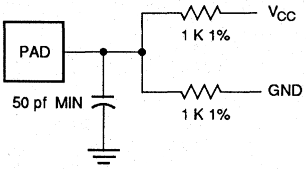
Figure 6. Slave Mode Programming Switching Characteristics



1637 12

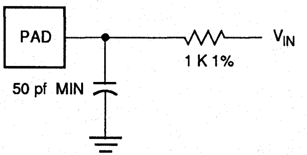
Figure 7. Program Readback Characteristics

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 Table 1 Test Specification from Xilinx.
2. $\overline{\text{PWR DWN}}$ 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 ($\overline{\text{CS0}}$, $\overline{\text{CS1}}$, CS2, $\overline{\text{WRT}}$) 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/$\overline{\text{P}}$ 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 patters, 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. $\overline{\text{RESET}}$ timing relative to power-on and valid mode lines (M0, M1, M2) is relevant only when $\overline{\text{RESET}}$ 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 = 0V, and high = 3V. TTHZ is determined when the output shifts 10% (of the output voltage swing) from VOL level or VOH level. The following circuit is used:



1637 13

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



1637 14

XC3020B
Military Logic Cell™ Array

Product Specification. See Note 1.

Part Number	Logic Capacity (gates)	Config-urable Logic Blocks	User I/O's	Config-uration Program (bits)
XC3020	2000	64	64	14779

FEATURES

- MIL–STD–883 Class B Processing. Complies with paragraph 1.2.1
- User-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)

DESCRIPTION

The Logic Cell™ Array (LCA) is a high density CMOS programmable gate array. Its patented array architecture consists of three type 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 datasheet for a full description.

ORDERING INFORMATION

XC3020 - 50 PG 84 B

B= MIL-STD-883, CLASS B, FULLY COMPLIANT

50 (50 MHz TOGGLE)

PG =CERAMIC PIN GRID ARRAY PACKAGE.
84-LEAD

1637 15

PIN ASSIGNMENTS

CONFIGURATION MODE: <M2:M1:M0>					84 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)	B2	$\overline{\text{PWR DWN}}$ (I)
VCC	VCC	VCC	VCC	VCC	F3	VCC
M1 (HIGH) (I)	M1 (LOW) (I)	M1 (LOW) (I)	M1 (HIGH) (I)	M1 (LOW) (I)	J2	$\overline{\text{RDATA}}$
M0 (HIGH) (I)	M0 (LOW) (I)	M0 (HIGH) (I)	M0 (LOW) (I)	M0 (LOW) (I)	L1	RTRIG (I)
▨ M2 (HIGH) (I)	▨ M2 (LOW) (I)	▨ M2 (HIGH) (I)	▨ M2 (HIGH) (I)	▨ M2 (HIGH) (I)	K2	I/O
HDC (HIGH)	HDC (LOW)	HDC (HIGH)	HDC (HIGH)	HDC (HIGH)	K3	I/O
$\overline{\text{LDC}}$ (LOW)	$\overline{\text{LDC}}$ (LOW)	$\overline{\text{LDC}}$ (LOW)	$\overline{\text{LDC}}$ (LOW)	$\overline{\text{LDC}}$ (LOW)	L3	I/O
* $\overline{\text{INIT}}$	$\overline{\text{INIT}}$	$\overline{\text{INIT}}$	$\overline{\text{INIT}}$	$\overline{\text{INIT}}$	K6	I/O
GND	GND	GND	GND	GND	J6	GND
▨					L11	XTL2 OR I/O
$\overline{\text{RESET}}$ (I)	$\overline{\text{RESET}}$ (I)	$\overline{\text{RESET}}$ (I)	$\overline{\text{RESET}}$ (I)	$\overline{\text{RESET}}$ (I)	K10	$\overline{\text{RESET}}$ (I)
DONE	DONE	DONE	DONE	DONE	J10	$\overline{\text{PROGRAM}}$ (I)
▨		▨ DATA 7 (I)	▨ DATA 7 (I)	▨ DATA 7 (I)	K11	I/O
		▨			J11	XTL1 OR I/O
		▨ DATA 6 (I)	▨ DATA 6 (I)	▨ DATA 6 (I)	H10	I/O
		▨ DATA 5 (I)	▨ DATA 5 (I)	▨ DATA 5 (I)	F10	I/O
		▨ $\overline{\text{CS0}}$ (I)	▨		G10	I/O
		▨ DATA 4 (I)	▨ DATA 4 (I)	▨ DATA 4 (I)	G11	I/O
VCC	VCC	VCC	VCC	VCC	F9	VCC
▨		▨ DATA 3 (I)	▨ DATA 3 (I)	▨ DATA 3 (I)	F11	I/O
		▨ $\overline{\text{CS1}}$ (I)	▨		E11	I/O
		▨ DATA 2 (I)	▨ DATA 2 (I)	▨ DATA 2 (I)	E10	I/O
		▨ DATA 1 (I)	▨ DATA 1 (I)	▨ DATA 1 (I)	D10	I/O
		RDY/$\overline{\text{BUSY}}$	$\overline{\text{RCLK}}$	$\overline{\text{RCLK}}$	C11	I/O
▨ DIN (I)	▨ DIN (I)	▨ DATA 0 (I)	▨ DATA 0 (I)	▨ DATA 0 (I)	B11	I/O
DOUT	DOUT	DOUT	DOUT	DOUT	C10	I/O
▨ CCLK (I)	CCLK	CCLK	CCLK	CCLK	A11	▨ CCLK (I)
▨		▨ $\overline{\text{WS}}$ (I)	A0	A0	B10	I/O
		▨ CS2 (I)	A1	A1	B9	I/O
			A2	A2	A10	I/O
			A3	A3	A9	I/O
			A15	A15	B6	I/O
			A4	A4	B7	I/O
			A14	A14	A7	I/O
			A5	A5	C7	I/O
GND	GND	GND	GND	GND	C6	GND
▨			A13	A13	A6	I/O
			A6	A6	A5	I/O
			A12	A12	B5	I/O
			A7	A7	C5	I/O
			A11	A11	A3	I/O
			A8	A8	A2	I/O
			A10	A10	B3	I/O
			A9	A9	A1	I/O

1637 16

▨ REPRESENTS A 50KΩ TO 100KΩ PULL-UP

* INIT IS AN OPEN DRAIN OUTPUT DURING CONFIGURATION

(I) REPRESENTS AN INPUT

PIN ASSIGNMENTS (Continued)

PGA Pin Number	XC-3020
B2	$\overline{\text{PWRDN}}$
C2	TCLKIN-I/O
B1	NC
C1	I/O
D2	I/O
D1	I/O
E3	I/O
E2	I/O
E1	I/O
F2	I/O
F3	**VCC**
G3	I/O
G1	I/O
G2	I/O
F1	I/O
H1	I/O
H2	I/O
J1	I/O
K1	I/O
J2	M1-$\overline{\text{RDATA}}$
L1	M0-$\overline{\text{RTRIG}}$
K2	M2-I/O
K3	HDC-I/O
L2	I/O
L3	$\overline{\text{LDC}}$-I/O
K4	I/O
L4	NC
J5	I/O
K5	I/O
L5	NC
K6	$\overline{\text{INIT}}$-I/O
J6	**GND**
J7	I/O
L7	I/O
K7	I/O
L6	I/O
L8	I/O
K8	I/O
L9	NC
L10	NC
K9	I/O
L11	XTL2(IN)-I/O

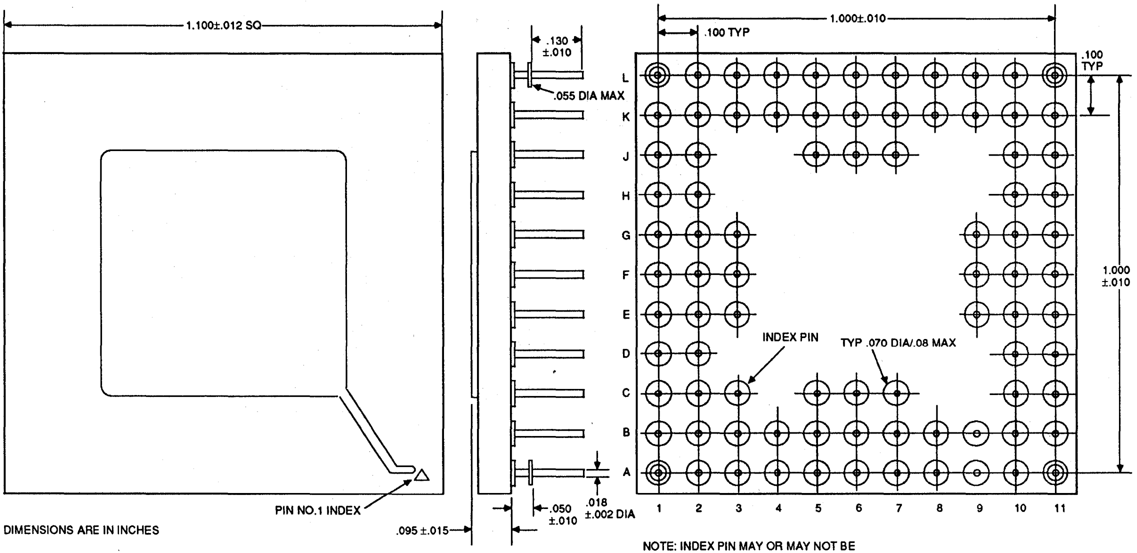
PGA Pin Number	XC-3020
K10	$\overline{\text{RESET}}$
J10	DONE-$\overline{\text{PG}}$
K11	D7-I/O
J11	XTL1(OUT)-BCLKIN-I/O
H10	D6-I/O
H11	I/O
F10	D5-I/O
G10	$\overline{\text{CS0}}$-I/O
G11	D4-I/O
G9	I/O
F9	**VCC**
F11	D3-I/O
E11	$\overline{\text{CS1}}$-I/O
E10	D2-I/O
E9	I/O
D11	NC
D10	D1-I/O
C11	RDY/$\overline{\text{BUSY}}$-$\overline{\text{RCLK}}$-I/O
B11	D0-DIN-I/O
C10	DOUT-I/O
A11	CCLK
B10	A0-$\overline{\text{WS}}$-I/O
B9	A1-CS2-I/O
A10	A2-I/O
A9	A3-I/O
B8	NC
A8	NC
B6	A15-I/O
B7	A4-I/O
A7	A14-I/O
C7	A5-I/O
C6	**GND**
A6	A13-I/O
A5	A6-I/O
B5	A12-I/O
C5	A7-I/O
A4	NC
B4	NC
A3	A11-I/O
A2	A8-I/O
B3	A10-I/O
A1	A9-I/O

1637 Tbl 06

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

CASE OUTLINE DRAWING

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



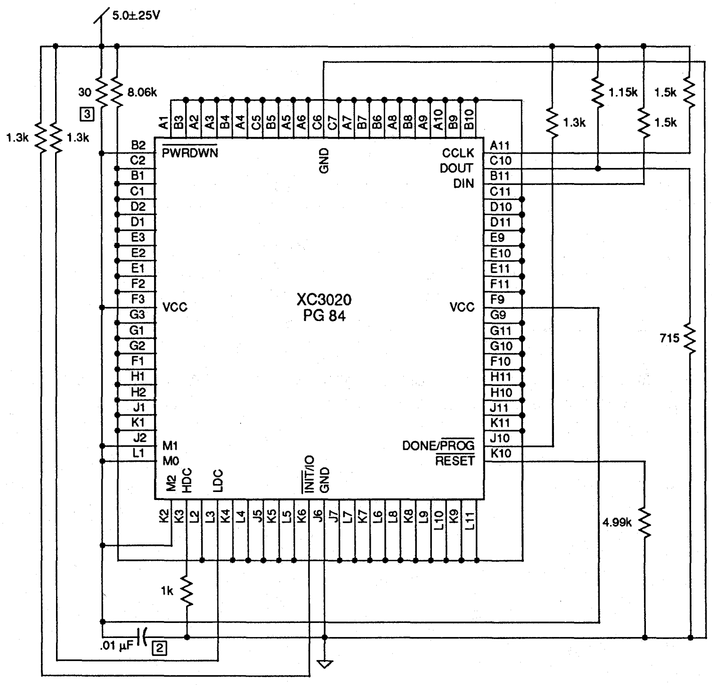
DIMENSIONS ARE IN INCHES

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

XC3020: 84-Pin PGA Package

1637 17

STATIC BURN-IN CIRCUIT



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 Rev:03

XC3020B Test Specification

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

1637 Tbl 07

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 Vcc = 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	mA
		TTL Mode, Vin = V_{CC} = 5.5 V	1,2,3		15	mA
Power-Down Supply Current	I_{CCPD}	Vin = V_{CC} = 5.5 V, $\overline{\text{PWR DWN}}$ = 0 V	1,2,3		0.5	mA
Leakage Current	I_{IL}	V_{CC} = 5.5 V, Vin = 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

1637 Tbl 08a

Table 1. Electrical Performance Characteristics

Test	Sym	Conditions –55°C ≤ Tc ≤ +125°C Vcc = 5.0 V ±10%	Group A Subgroups	Limits Min	 Max	Units
Switching Characteristics, General LCA						
DONE/$\overline{\text{PROG}}$						
Program Width (Low)	TPGW	5 See Fig. 1	9,10,11	6		µs
Initialization	TPGI	6	9,10,11		7	µs
$\overline{\text{PWR}}$ $\overline{\text{DWN}}$[3]						
Power Down Supply	VCCPD		1,2,3	3.5		V
RESET[4]						
M2,M1,M0 Setup	TMR	2	9,10,11	1		µs
M2,M1,M0 Hold	TRM	3	9,10,11	1		µs
Width (low) abort	TMRW	4	9,10,11	6		µs
Switching Characteristics, Peripheral Mode Programming[5]						
$\overline{\text{WRT}}$ LOW	TCA	1 See Fig. 4	9,10,11	0.5		µs
DIN Setup	TDC	2	9,10,11	60		ns
DIN Hold	TCD	3	9,10,11	0		ns
Ready/$\overline{\text{Busy}}$	TWTRB	4	9,10,11		60	ns
Switching Characteristics, Slave Mode Programming[5]						
CCLK,						
To DOUT	TCCO	3 See Fig. 5	9,10,11		100	ns
DIN Setup	TDCC	1	9,10,11	60		ns
DIN Hold	TCCD	2	9,10,11	0		ns
High Time	TCCH	4	9,10,11	0.5		µs
Low Time	TCCL	5	9,10,11	0.5	1.0	µs
Frequency	FCC		9,10,11		1	MHz

1637 Tbl 08b

Table 1. Electrical Performance Characteristics (Continued)

Test	Sym	Conditions −55°C ≤ Tc ≤ +125°C Vcc = 5.0 V ±10%	Group A Subgroups	Limits Min	 Max	Units
Switching Characteristics, Program Readback[6, 7]						
RTRIG Setup	TRTH	1 See Fig. 7	9,10,11	250		ns
CCLK,[7]						
RTRIG Setup	TRTCC	2	9,10,11	200		ns
RDATA Delay	TCCRD	3	9,10,11		100	ns
Benchmark Patterns[8]						
TPID + interconnect + 8 (TILO) + TOP. Measured on 8 cols.	TB1		9,10,11		135	ns
TCKO + TICK + TCKI + interconnect	TB2	Tested on all CLBs	9,10,11		32	ns
TCKO + TQLO + TILO + TDICK + interconnect	TB3	Tested on all CLBs	9,10,11		53	ns
TILO + TECCK + interconnect	TB4	Tested on all CLBs	9,10,11		35	ns
TOKPO + TOPS – TOPF + TPICK	TB5	Tested on all CLBs	9,10,11		73	ns
TCKO + TQLO + TPUS + TICK + interconnect	TB6	One long line pull-up	9,10,11		73	ns
TCKO + TQLO + TPUS + TICK + interconnect	TB7	The other long line pull-up	9,10,11		83	ns
TCKO + TQLO + TIO + TICK + interconnect	TB8	No pull-up, lower long lines	9,10,11		47	ns
TCKO + TQLO + TIO + TICK + interconnect	TB9	No pull-up, upper long lines	9,10,11		57	ns

1637 Tbl 08c

Table 1. Electrical Performance Characteristics (Continued)

Test	Sym	Conditions –55°C ≤ Tc ≤ +125°C Vcc = 5.0 V ±10%	Group A Subgroups	Limits Min	Limits Max	Units
Application Guidelines, Switching, CLB[8]						
Combinatorial	TILO	1 See Fig. 2	N/A		14	ns
Reset to CLB output	TRIO	9	N/A		12	ns
Reset Direct width	TRPW	13	N/A	12		ns
Master Reset pin to CLB out	TMRQ		N/A		30	ns
K Clock[9]						
To CLB output	TCKO	8	N/A		12	ns
Additional for Q returning through F or G to CLB out	TQLO		N/A		11	ns
Logic-input setup	TICK	2	N/A	12		ns
Logic-input hold	TCKI	3	N/A	0		ns
Data In setup	TDICK	4	N/A	8		ns
Data In hold	TCKDI	5	N/A	6		ns
Enable Clock setup	TECCK	6	N/A	10		ns
Enable Clock hold	TCKEC	7	N/A	0		ns
Clock (high)	TCH	11	N/A	9		ns
Clock (low)	TCL	12	N/A	9		ns
Flip-flop Toggle Rate	FCLK		N/A	50		MHz
Application Guidelines, Switching, Internal Buffers[8]						
Clock Buffer	TGCK		N/A		9	ns
TBUF						
Data to Output	TIO		N/A		8	ns
Three-state to Output						
Single Pull-up	TPUS		N/A		34	ns
Pair of Pull-ups	TPUF		N/A		17	ns
Bidirectional	BIDI		N/A		6	ns

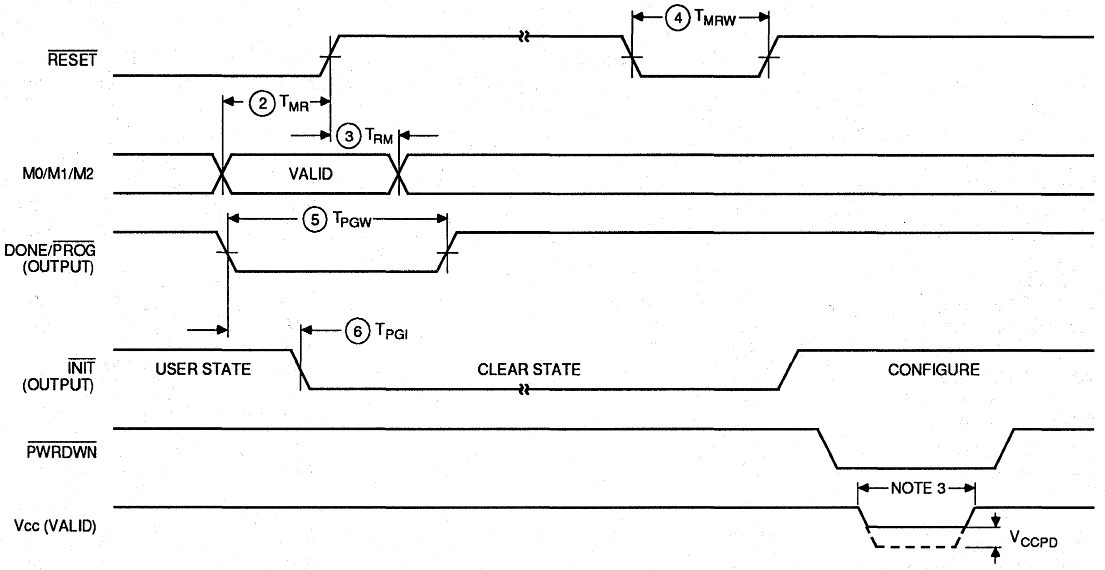
1637 Tbl 08d

Table 1. Electrical Performance Characteristics (Continued)

Test	Sym	Conditions –55°C ≤ Tc ≤ +125°C Vcc = 5.0 V ±10%	Group A Subgroups	Limits Min	Max	Units
Application Guidelines, Switching, IOB[8,10]						
Pad (package pin)		See Fig. 3				
To inputs TCLKIN, BCLKIN	TPIDC		N/A		5	ns
To inputs DIRECT IN	TPID	3	N/A		9	ns
I/O Clock						
To I/O RI input (FF)	TIKRI	4	N/A		11	ns
I/O pad-input setup	TPICK	1	N/A	30		ns
I/O pad-input hold	TIKPI	2	N/A	0		ns
To I/O pad (fast)	TOKPO	7	N/A		18	ns
I/O pad output setup	TOOK	5	N/A	15		ns
I/O pad output hold	TOKO	6	N/A	0		ns
Clock (high)	TIOH	11	N/A	9		ns
Clock (low)	TIOL	12	N/A	9		ns
Output						
To pad (enabled fast)	TOPF	10	N/A		14	ns
To pad (enabled slow)	TOPS	10	N/A		39	ns
Three-State						
To pad begin hi-Z (fast)	TTSHZ	9	N/A		12	ns
To pad valid (fast)	TTSON	8	N/A		20	ns
Master Reset						
To input RI	TRRI	13	N/A		35	ns
To output (FF)	TRPO	14	N/A		50	ns
Application Guidelines, Switching, Master Parallel Mode Programming[8,11]						
$\overline{\text{RCLK}}$,						
To Address Valid	TRAC	1 See Fig.6	N/A	0	200	ns
To Data Setup	TDRC	2	N/A	60		ns
To Data Hold	TRCD	3	N/A	0		ns
$\overline{\text{RCLK}}$ High	TRCH	4	N/A	600		ns
$\overline{\text{RCLK}}$ Low	TRCL	5	N/A	4.0		µs

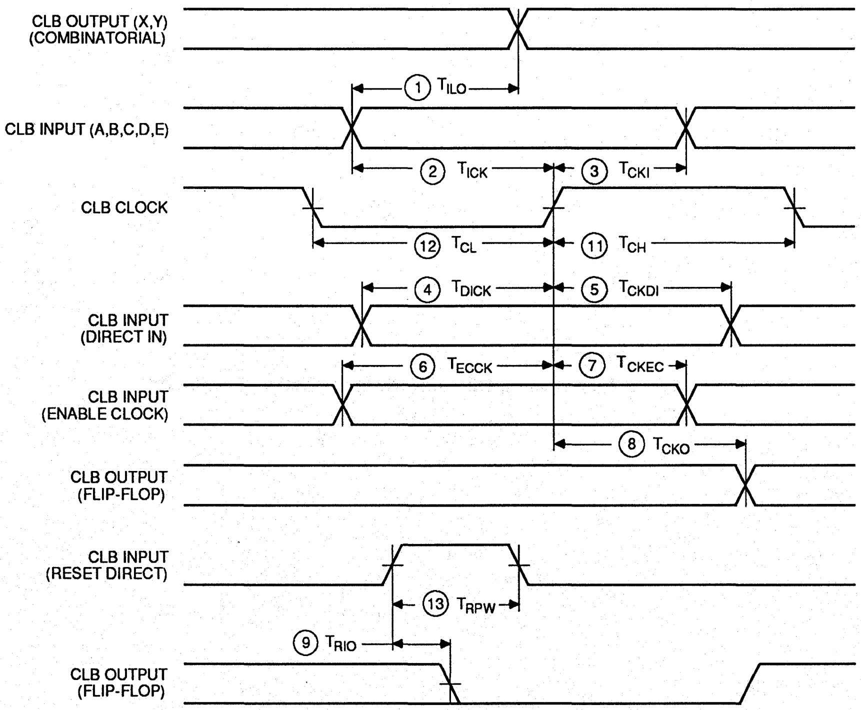
1637 Tbl 08e

Table 1. Electrical Performance Characteristics (Continued)



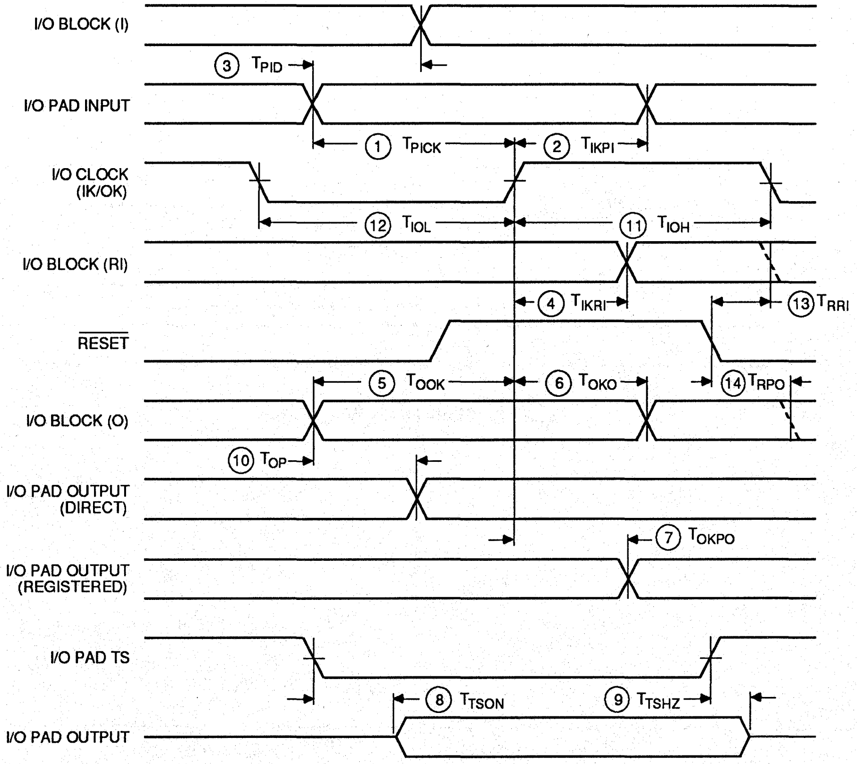
1637 19

Figure 1. General LCA Waveforms



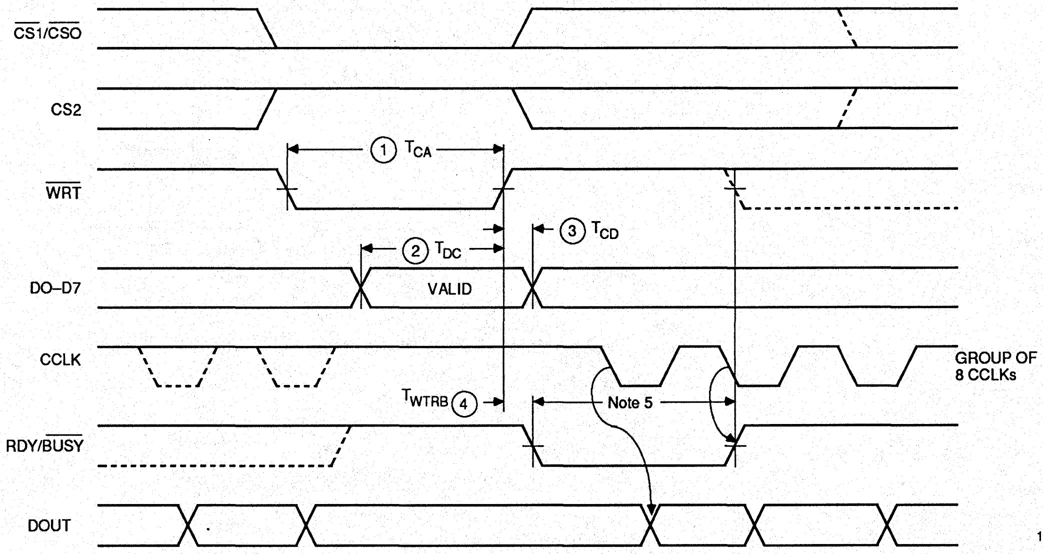
1637 20

Figure 2. CLB Waveforms



1637 21

Figure 3. IOB Waveforms



1637 22

Figure 4. Peripheral Mode Waveforms

TSC0085 Rev:03

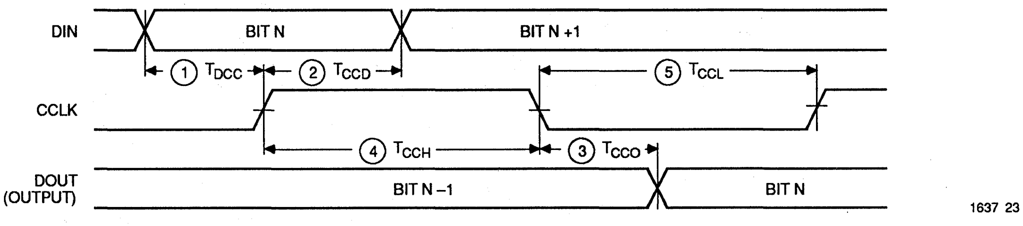


Figure 5. Slave Mode Waveforms

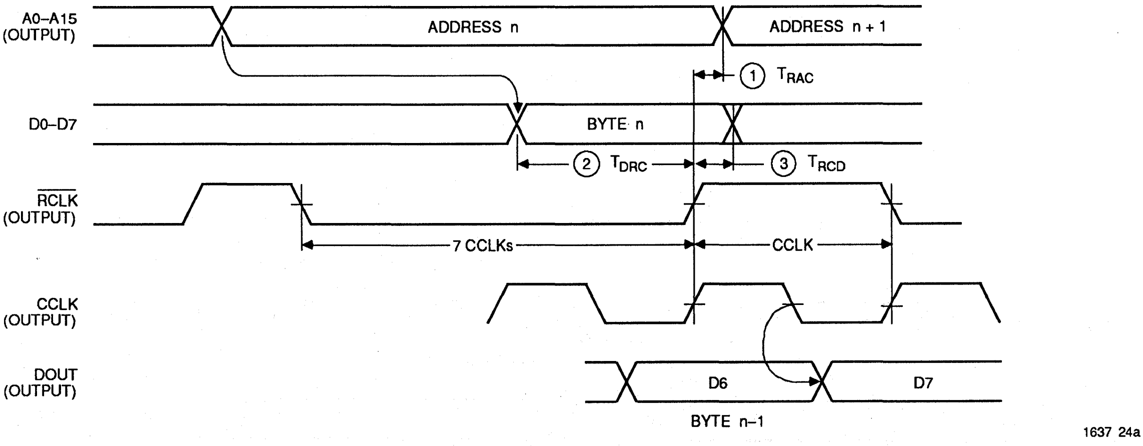
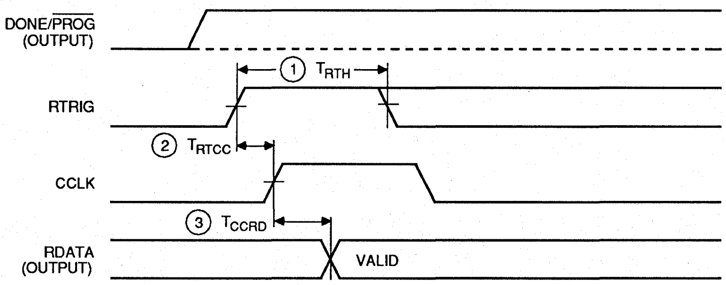


Figure 6. Master Parallel Mode Waveforms

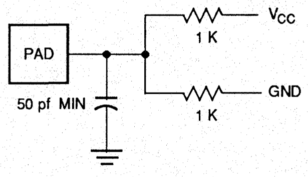


1637 25

Figure 7. Program Readback Waveforms

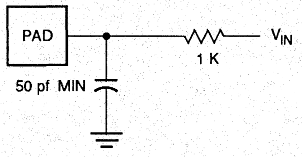
XC3020B Data Sheet Notes

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 Table 1 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 $\overline{\text{INIT}}$ of all LCA's is HIGH. $\overline{\text{WRT}}$ cannot go active until RDY/$\overline{\text{BUSY}}$ goes high.
6. Readback should not be initiated until configuration is complete.
7. CCLK and DOUT timing are 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 (TCKO) plus the shortest possible interconnect delay is always longer than the Data In hold time requirement (TCKDI) 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, Vcc must rise from 2.0V to Vcc minimum in less than 10 ms. Otherwise, delay configuration using $\overline{\text{RESET}}$.
12. All timings except TTSHZ and TTSON 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.
    TTSHZ is determined when the output shifts 10% (of the output voltage swing) from VOL level or VOH level. The following circuit is used:



1637 26

12. (continued)
    TTSON is measured at 0.5 V_{CC} level with VIN = 0 for Tri-State to active high, and VIN = Vcc for Tri-State to active low. The following load circuit is used:



1637 27

TSC0085 Rev:03

XILINX

PRELIMINARY
Subject to Change

XC3090B
Military Logic Cell™ Array

Product Specification. See Note 1.

Part Number	Logic Capacity (gates)	Config-urable Logic Blocks	User I/O's	Config-uration Program (bits)
XC3090	9000	320	144	64160

FEATURES

- MIL–STD–883 Class B Processing. Complies with paragraph 1.2.1
- User-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)

DESCRIPTION

The Logic Cell™ Array (LCA) is a high density CMOS programmable gate array. Its patented array architecture consists of three type 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.

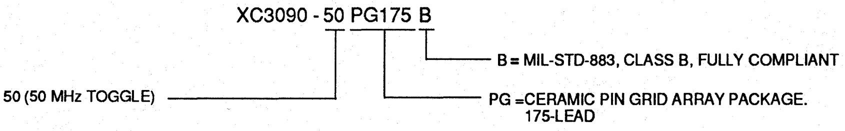
The Logic Cell Array's logic functions and interconnections are determined by the configuration program stored in internal static memory cells. On-chip logic provides for automatic loading of configuration data at power-up or on command. The program data can reside in an EEPROM, EPROM or ROM on the circuit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

ORDERING INFORMATION



XC3090 - 50 PG175 B

B = MIL-STD-883, CLASS B, FULLY COMPLIANT

50 (50 MHz TOGGLE)

PG =CERAMIC PIN GRID ARRAY PACKAGE.
175-LEAD

1637 28

PIN ASSIGNMENTS

CONFIGURATION MODE: <M2:M1:M0>					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)	B2	$\overline{\text{PWR DWN}}$ (I)
VCC	VCC	VCC	VCC	VCC	D9	VCC
M1 (HIGH) (I)	M1 (LOW) (I)	M1 (LOW) (I)	M1 (HIGH) (I)	M1 (LOW) (I)	B14	$\overline{\text{RDATA}}$
M0 (HIGH) (I)	M0 (LOW) (I)	M0 (HIGH) (I)	M0 (LOW) (I)	M0 (LOW) (I)	B15	RTRIG (I)
▨ M2 (HIGH) (I)	▨ M2 (LOW) (I)	▨ M2 (HIGH) (I)	▨ M2 (HIGH) (I)	▨ M2 (HIGH) (I)	C15	I/O
HDC (HIGH)	HDC (LOW)	HDC (HIGH)	HDC (HIGH)	HDC (HIGH)	E14	I/O
$\overline{\text{LDC}}$ (LOW)	$\overline{\text{LDC}}$ (LOW)	$\overline{\text{LDC}}$ (LOW)	$\overline{\text{LDC}}$ (LOW)	$\overline{\text{LDC}}$ (LOW)	D16	I/O
$\overline{\text{INIT}}$	$\overline{\text{INIT}}$	$\overline{\text{INIT}}$	$\overline{\text{INIT}}$	$\overline{\text{INIT}}$	H15	I/O
GND	GND	GND	GND	GND	J14	GND
▨	▨	▨	▨	▨	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)	R15	$\overline{\text{RESET}}$ (I)
DONE	DONE	DONE	DONE	DONE	R14	$\overline{\text{PROGRAM}}$ (I)
▨	▨	▨ DATA 7 (I)	▨ DATA 7 (I)	▨ DATA 7 (I)	N13	I/O
▨	▨	▨	▨	▨	T14	XTL1 OR I/O
▨	▨	▨ DATA 6 (I)	▨ DATA 6 (I)	▨ DATA 6 (I)	P12	I/O
▨	▨	▨ DATA 5 (I)	▨ DATA 5 (I)	▨ DATA 5 (I)	T11	I/O
▨	▨	▨ $\overline{\text{CS0}}$ (I)	▨	▨	R10	I/O
▨	▨	▨ DATA 4 (I)	▨ DATA 4 (I)	▨ DATA 4 (I)	R9	I/O
VCC	VCC	VCC	VCC	VCC	N9	VCC
▨	▨	▨ DATA 3 (I)	▨ DATA 3 (I)	▨ DATA 3 (I)	P8	I/O
▨	▨	▨ $\overline{\text{CS1}}$ (I)	▨	▨	R8	I/O
▨	▨	▨ DATA 2 (I)	▨ DATA 2 (I)	▨ DATA 2 (I)	R7	I/O
▨	▨	▨ DATA 1 (I)	▨ DATA 1 (I)	▨ DATA 1 (I)	R5	I/O
▨	▨	RDY/$\overline{\text{BUSY}}$	$\overline{\text{RCLK}}$	$\overline{\text{RCLK}}$	P5	I/O
▨ DIN (I)	▨ DIN (I)	▨ DATA 0 (I)	▨ DATA 0 (I)	▨ DATA 0 (I)	R3	I/O
DOUT	DOUT	DOUT	DOUT	DOUT	N4	I/O
▨ CCLK (I)	CCLK	CCLK	CCLK	CCLK	R2	▨ CCLK (I)
▨	▨	▨ $\overline{\text{WS}}$ (I)	A0	A0	P2	I/O
▨	▨	▨ CS2 (I)	A1	A1	M3	I/O
▨	▨	▨	A2	A2	P1	I/O
▨	▨	▨	A3	A3	N1	I/O
▨	▨	▨	A15	A15	M1	I/O
▨	▨	▨	A4	A4	L2	I/O
▨	▨	▨	A14	A14	K2	I/O
▨	▨	▨	A5	A5	K1	I/O
GND	GND	GND	GND	GND	J3	GND
▨	▨	▨	A13	A13	H2	I/O
▨	▨	▨	A6	A6	H1	I/O
▨	▨	▨	A12	A12	F2	I/O
▨	▨	▨	A7	A7	E1	I/O
▨	▨	▨	A11	A11	D1	I/O
▨	▨	▨	A8	A8	C1	I/O
▨	▨	▨	A10	A10	E3	I/O
▨	▨	▨	A9	A9	C2	I/O

1637 29

▨ REPRESENTS A 50KΩ TO 100KΩ PULL-UP

* INIT IS AN OPEN DRAIN OUTPUT DURING CONFIGURATION

(I) REPRESENTS AN INPUT

PIN ASSIGNMENTS (Continued)

PGA Pin Number	XC-3090	PGA Pin Number	XC-3090	PGA Pin Number	XC-3090	PGA Pin Number	XC-3090
B2	$\overline{\text{PWRDN}}$	D13	I/O	R14	DONE-$\overline{\text{PG}}$	R3	D0-DIN-I/O
D4	TCLKIN-I/O	B14	M1-$\overline{\text{RDATA}}$	N13	D7-I/O	N4	DOUT-I/O
B3	I/O	C14	**GND**	T14	XTAL1(OUT)-BCLKIN-I/O	R2	CCLK
C4	I/O	B15	M0-$\overline{\text{RTRIG}}$	P13	I/O	P3	**VCC**
B4	I/O	D14	**VCC**	R13	I/O	N3	**GND**
A4	I/O	C15	M2-I/O	T13	I/O	P2	A0-$\overline{\text{WS}}$-I/O
D5	I/O	E14	HDC-I/O	N12	I/O	M3	A1-CS2-I/O
C5	I/O	B16	I/O	P12	D6-I/O	R1	I/O
B5	I/O	D15	I/O	R12	I/O	N2	I/O
A5	I/O	C16	I/O	T12	I/O	P1	A2-I/O
C6	I/O	D16	$\overline{\text{LDC}}$-I/O	P11	I/O	N1	A3-I/O
D6	I/O	F14	I/O	N11	I/O	L3	I/O
B6	I/O	E15	I/O	R11	I/O	M2	I/O
A6	I/O	E16	I/O	T11	D5-I/O	M1	A15-I/O
B7	I/O	F15	I/O	R10	$\overline{\text{CS0}}$-I/O	L2	A4-I/O
C7	I/O	F16	I/O	P10	I/O	L1	I/O
D7	I/O	G14	I/O	N10	I/O	K3	I/O
A7	I/O	G15	I/O	T10	I/O	K2	A14-I/O
A8	I/O	G16	I/O	T9	I/O	K1	A5-I/O
B8	I/O	H16	I/O	R9	D4-I/O	J1	I/O
C8	I/O	H15	$\overline{\text{INIT}}$-I/O	P9	I/O	J2	I/O
D8	**GND**	H14	**VCC**	N9	**VCC**	J3	**GND**
D9	**VCC**	J14	**GND**	N8	**GND**	H3	**VCC**
C9	I/O	J15	I/O	P8	D3-I/O	H2	A13-I/O
B9	I/O	J16	I/O	R8	$\overline{\text{CS1}}$-I/O	H1	A6-I/O
A9	I/O	K16	I/O	T8	I/O	G1	I/O
A10	I/O	K15	I/O	T7	I/O	G2	I/O
D10	I/O	K14	I/O	N7	I/O	G3	I/O
C10	I/O	L16	I/O	P7	I/O	F1	I/O
B10	I/O	L15	I/O	R7	D2-I/O	F2	A12-I/O
A11	I/O	M16	I/O	T6	I/O	E1	A7-I/O
B11	I/O	M15	I/O	R6	I/O	E2	I/O
D11	I/O	L14	I/O	N6	I/O	F3	I/O
C11	I/O	N16	I/O	P6	I/O	D1	A11-I/O
A12	I/O	P16	I/O	T5	I/O	C1	A8-I/O
B12	I/O	N15	I/O	R5	D1-I/O	D2	I/O
C12	I/O	R16	I/O	P5	RDY/$\overline{\text{BUSY}}$-$\overline{\text{RCLK}}$-I/O	B1	I/O
D12	I/O	M14	I/O	N5	I/O	E3	A10-I/O
A13	I/O	P15	XTAL2(IN)-I/O	T4	I/O	C2	A9-I/O
B13	I/O	N14	**GND**	R4	I/O	D3	**VCC**
C13	I/O	R15	$\overline{\text{RESET}}$	P4	I/O	C3	**GND**
A14	I/O	P14	**VCC**				

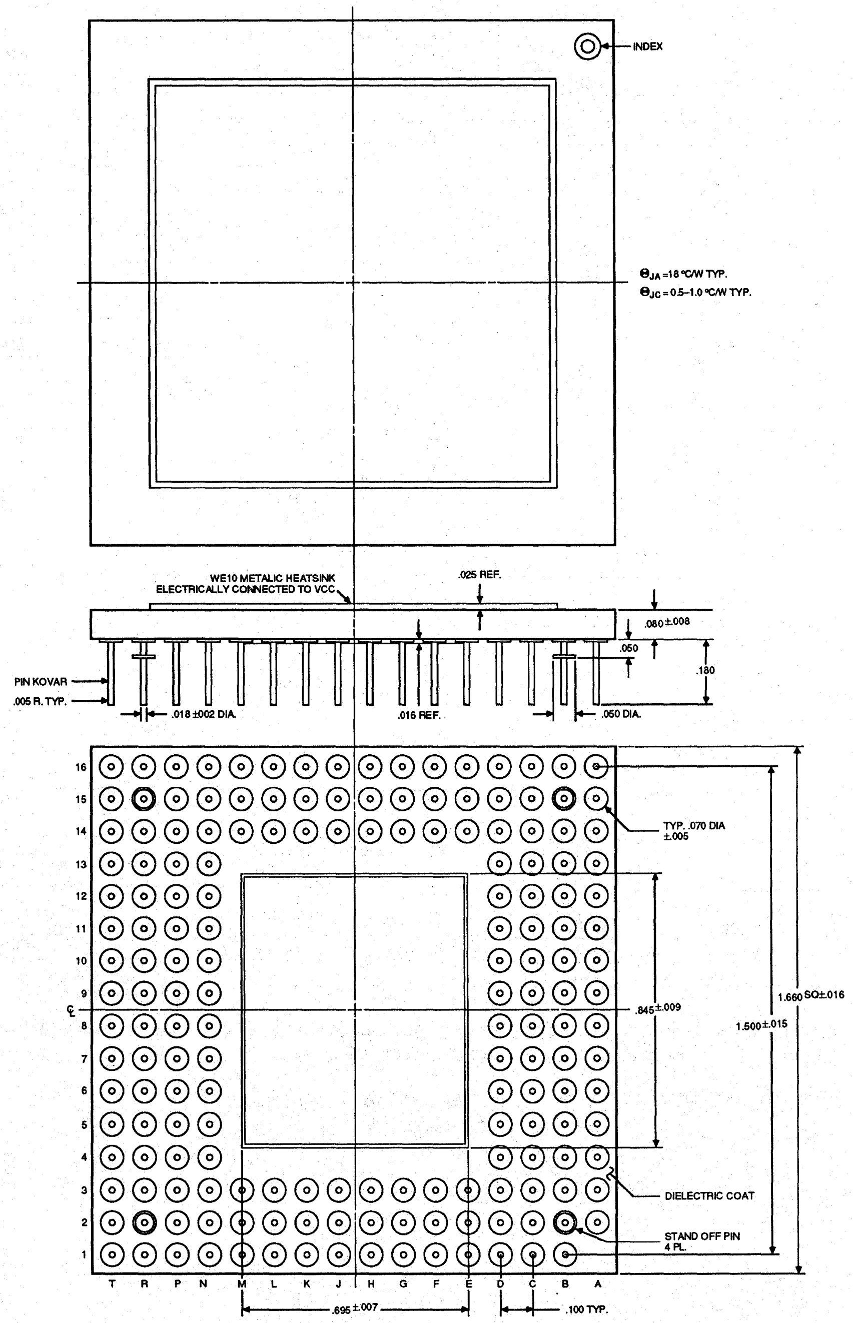
1637 Tbl 09

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

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

CASE OUTLINE DRAWING

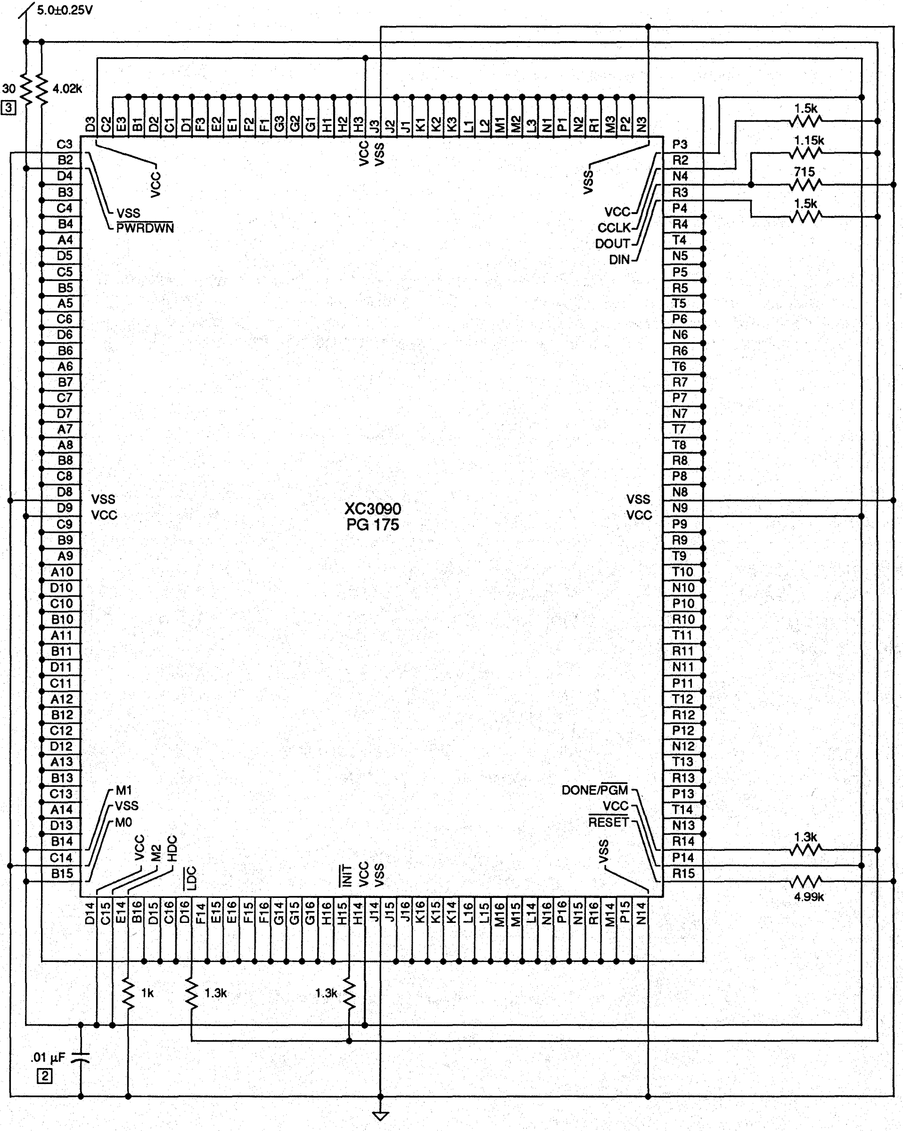
Conforms to MIL-M-38510 Appendix C, Case P-BC.



1637 30

TSC0097 Rev:03

STATIC BURN-IN CIRCUIT



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 31

XC3090B Test Specification

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

1637 Tbl 10

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	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		3	mA
		TTL Mode, Vin = V_{CC} = 5.5 V	1,2,3		15	mA
Power-Down Supply Current	I_{CCPD}	Vin = V_{CC} = 5.5 V, $\overline{\text{PWR DWN}}$ = 0 V	1,2,3		2.5	mA
Leakage Current	I_{IL}	V_{CC} = 5.5 V, Vin = 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

1637 Tbl 11a

Table 1. Electrical Performance Characteristics

Test	Sym	Conditions –55°C ≤ Tc ≤ +125°C Vcc = 5.0 V ±10%		Group A Subgroups	Limits Min	 Max	Units
Switching Characteristics, General LCA							
DONE/$\overline{\text{PROG}}$							
Program Width (Low)	TPGW	5	See Fig. 1	9,10,11	6		µs
Initialization	TPGI	6		9,10,11		7	µs
$\overline{\text{PWR DWN}}$[3]							
Power Down Supply	VCCPD			1,2,3	3.5		V
RESET[4]							
M2,M1,M0 Setup	TMR	2		9,10,11	1		µs
M2,M1,M0 Hold	TRM	3		9,10,11	1		µs
Width (low) abort	TMRW	4		9,10,11	6		µs
Switching Characteristics, Peripheral Mode Programming[5]							
$\overline{\text{WRT}}$ LOW	TCA	1	See Fig. 4	9,10,11	0.5		µs
DIN Setup	TDC	2		9,10,11	60		ns
DIN Hold	TCD	3		9,10,11	0		ns
Ready/$\overline{\text{Busy}}$	TWTRB	4		9,10,11		60	ns
Switching Characteristics, Slave Mode Programming[5]							
CCLK,							
To DOUT	TCCO	3	See Fig. 5	9,10,11		100	ns
DIN Setup	TDCC	1		9,10,11	60		ns
DIN Hold	TCCD	2		9,10,11	0		ns
High Time	TCCH	4		9,10,11	0.5		µs
Low Time	TCCL	5		9,10,11	0.5	1.0	µs
Frequency	FCC			9,10,11		1	MHz

1637 Tbl 11b

Table 1. Electrical Performance Characteristics (Continued)

Test	Sym	Conditions −55°C ≤ Tc ≤ +125°C Vcc = 5.0 V ±10%	Group A Subgroups	Limits Min	 Max	Units
Switching Characteristics, Program Readback[6,7]						
RTRIG Setup	T_{RTH}	1 See Fig. 7	9,10,11	250		ns
CCLK,						
RTRIG Setup	T_{RTCC}	2	9,10,11	200		ns
RDATA Delay	T_{CCRD}	3	9,10,11		100	ns
Clock Low	T_{CCLR}	4	9,10,11	1.2	2.0	µs
Clock High	T_{CCHR}	5	9,10,11	0.5		µs
Benchmark Patterns[8]						
T_{PID} + interconnect + 20 (T_{ILO}) + T_{OP}. Measured on 16 cols.	T_{B1}		9,10,11		303	ns
T_{CKO} + T_{ICK} + T_{CKI} + interconnect	T_{B2}	Tested on all CLBs	9,10,11		32	ns
T_{CKO} + T_{QLO} + T_{ILO} + T_{DICK} + interconnect	T_{B3}	Tested on all CLBs	9,10,11		53	ns
T_{QLO} + T_{ECCK} + interconnect	T_{B4}	Tested on all CLBs	9,10,11		35	ns
T_{OKPO} + T_{OPS} − T_{OPF} + T_{PICK}	T_{B5}	Tested on all CLBs	9,10,11		73	ns
T_{CKO} + T_{QLO} + T_{BUF} + T_{ICK} + interconnect	T_{B6}	One long line pull-up	9,10,11		73	ns
T_{CKO} + T_{QLO} + T_{BUF} + T_{ICK} + interconnect	T_{B7}	The other long line pull-up	9,10,11		83	ns
T_{CKO} + T_{QLO} + T_{BUF} + T_{ICK} + interconnect	T_{B8}	No pull-up, lower long lines	9,10,11		47	ns
T_{CKO} + T_{QLO} + T_{BUF} + T_{ICK} + interconnect	T_{B9}	No pull-up, upper long lines	9,10,11		57	ns

1637 Tbl 11c

Table 1. Electrical Performance Characteristics (Continued)

Test	Sym	Conditions −55°C ≤ Tc ≤ +125°C Vcc = 5.0 V ±10%		Group A Subgroups	Limits Min	 Max	Units

Application Guidelines, Switching, CLB[8]

Test	Sym	Conditions		Group A Subgroups	Min	Max	Units
Combinatorial	TILO	1	See Fig. 2	N/A		14	ns
Reset to CLB output	TRIO	9		N/A		12	ns
Reset Direct width	TRPW	13		N/A	12		ns
Master Reset pin to CLB out	TMRQ			N/A		30	ns
K Clock[9]							
To CLB output	TCKO	8		N/A		12	ns
Additional for Q returning through F or G to CLB out	TQLO			N/A		11	ns
Logic-input setup	TICK	2		N/A	12		ns
Logic-input hold	TCKI	3		N/A	0		ns
Data In setup	TDICK	4		N/A	8		ns
Data In hold (1)	TCKDI	5		N/A	6		ns
Enable Clock setup	TECCK	6		N/A	10		ns
Enable Clock hold	TCKEC	7		N/A	0		ns
Clock (high)	TCH	11		N/A	9		ns
Clock (low)	TCL	12		N/A	9		ns
Flip-flop Toggle Rate	FCLK			N/A	50		MHz

Application Guidelines, Switching, Internal Buffers[8]

Test	Sym	Conditions	Group A Subgroups	Min	Max	Units
Clock Buffer	TGCK		N/A		9	ns
TBUF						
Data to Output	TIO		N/A		8	ns
Three-state to Output						
Single Pull-up	TPUS		N/A		34	ns
Pair of Pull-ups	TPUF		N/A		17	ns
Bidirectional	BIDI		N/A		6	ns

1637 Tbl 11d

Table 1. Electrical Performance Characteristics (Continued)

Test	Sym	Conditions −55°C ≤ Tc ≤ +125°C Vcc = 5.0 V ±10%	Group A Subgroups	Limits		Units
				Min	Max	
Application Guidelines, Switching, IOB[8,10]						
Pad (package pin)		See Fig. 3				
To inputs TCLKIN, BCLKIN	T_{PIDC}		N/A		5	ns
To inputs DIRECT IN	T_{PID}	3	N/A		9	ns
I/O Clock						
To I/O RI input (FF)	T_{IKRI}	4	N/A		11	ns
I/O pad-input setup	T_{PICK}	1	N/A	30		ns
I/O pad-input hold	T_{IKPI}	2	N/A	0		ns
To I/O pad.(fast)	T_{OKPO}	7	N/A		18	ns
I/O pad output setup	T_{OOK}	5	N/A	15		ns
I/O pad output hold	T_{OKO}	6	N/A	0		ns
Clock (high)	T_{IOH}	11	N/A	9		ns
Clock (low)	T_{IOL}	12	N/A	9		ns
Output						
To pad (enabled fast)	T_{OPF}	10	N/A		14	ns
To pad (enabled slow)	T_{OPS}	10	N/A		39	ns
Three-State						
To pad begin hi-Z (fast)	T_{TSHZ}	9	N/A		12	ns
To pad valid (fast)	T_{TSON}	8	N/A		20	ns
Master Reset						
To input RI	T_{RRI}	13	N/A		35	ns
To output (FF)	T_{RPO}	14	N/A		50	ns

1637 Tbl 11e

Table 1. Electrical Performance Characteristics (Continued)

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

1637 Tbl 11f

Table 1. Electrical Performance Characteristics (Continued)

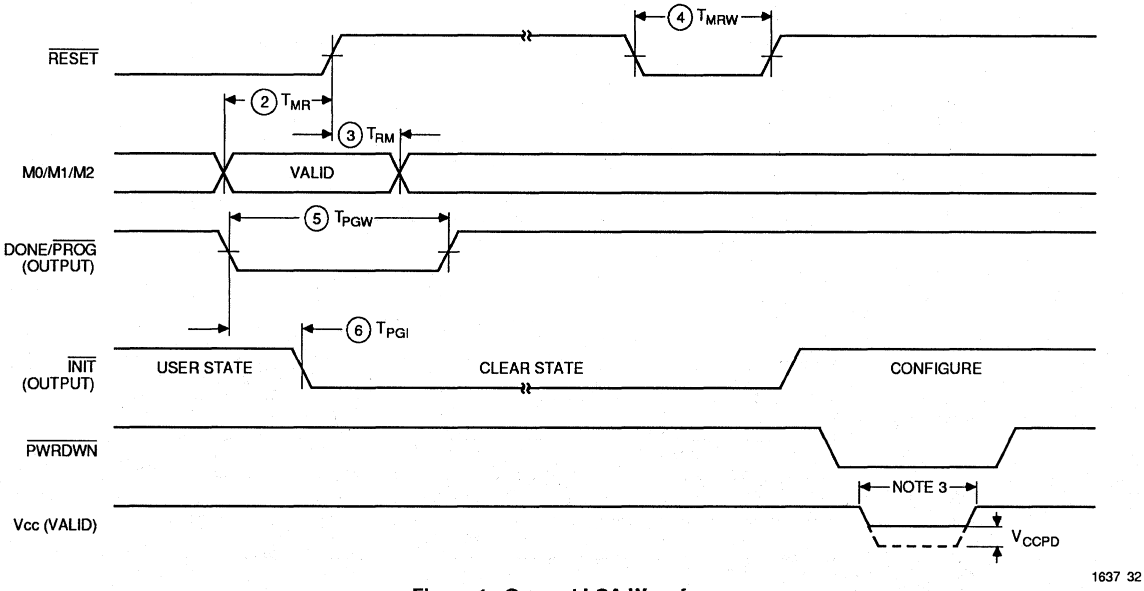


Figure 1. General LCA Waveforms

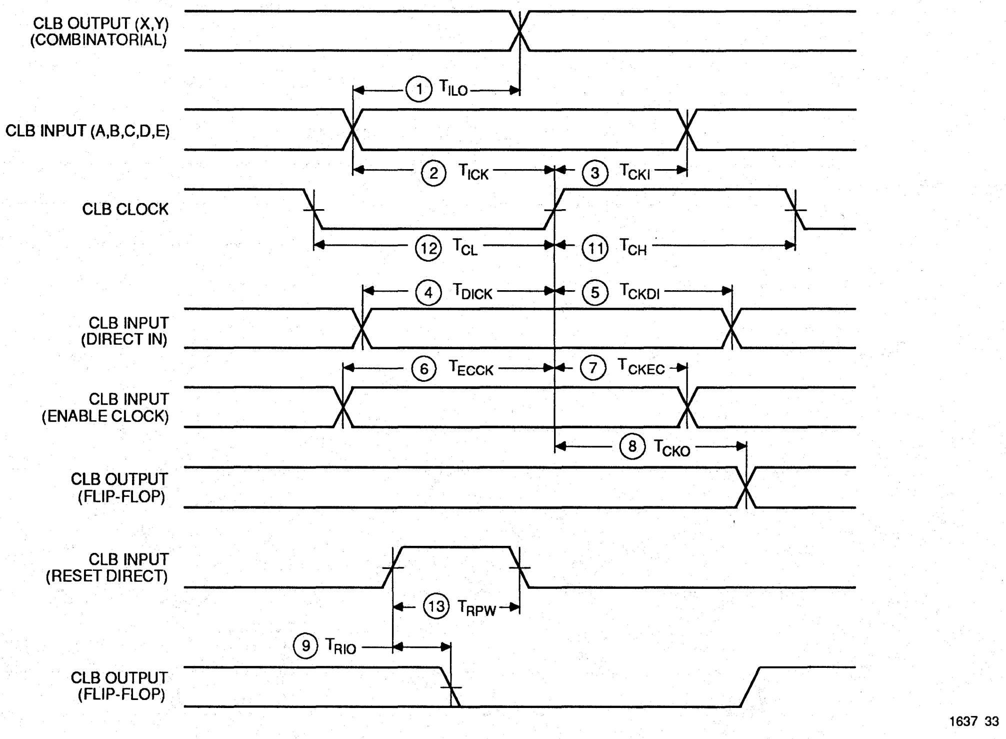


Figure 2. CLB Waveforms

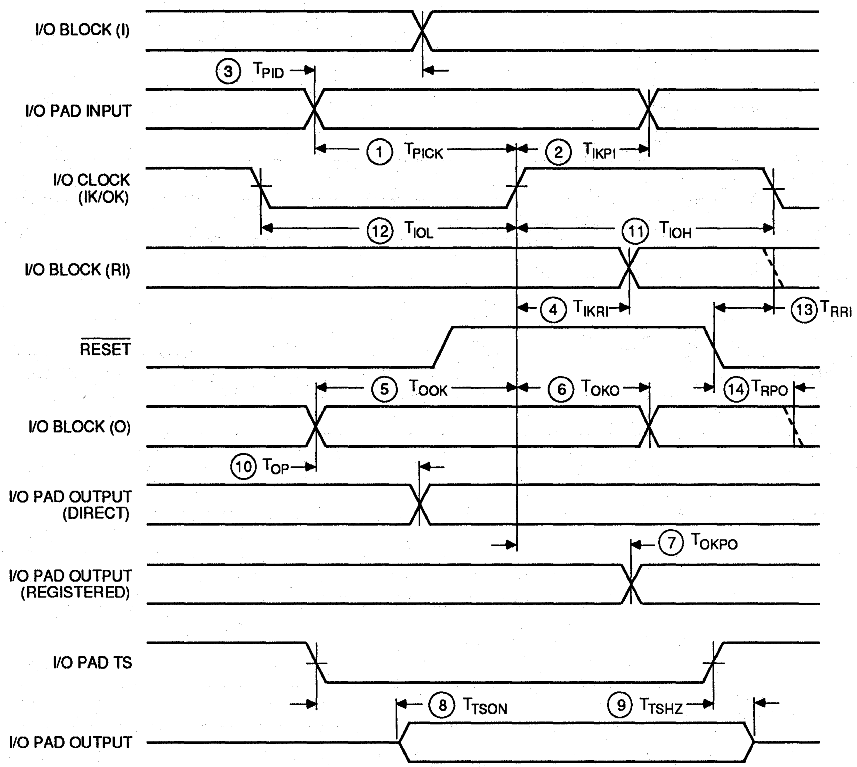


Figure 3. IOB Waveforms

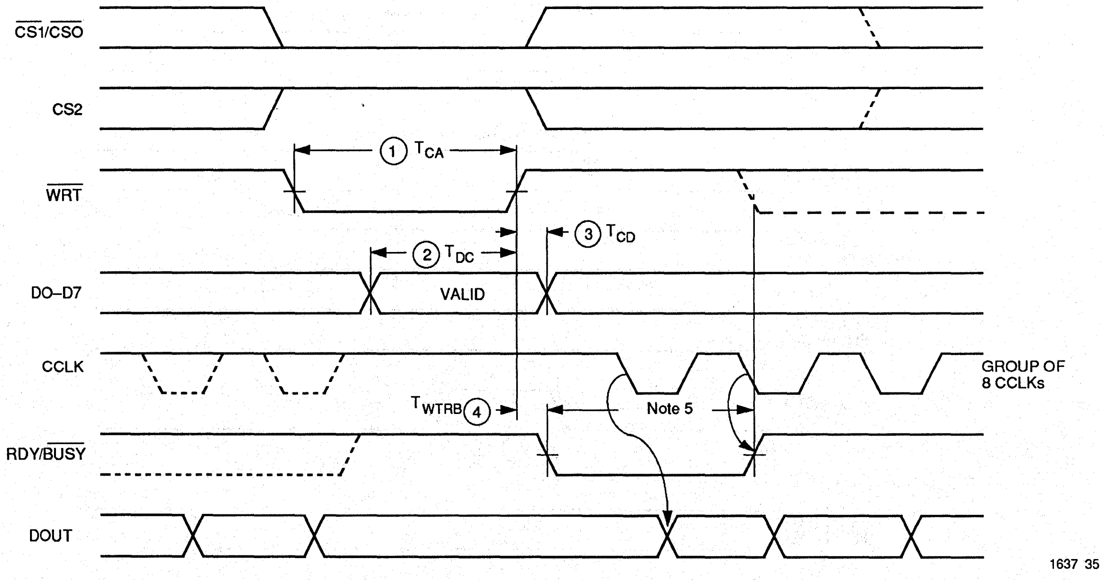


Figure 4. Peripheral Mode Waveforms

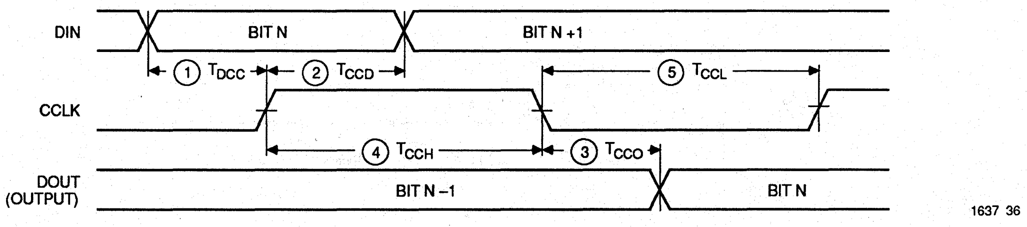


Figure 5. Slave Mode Waveforms

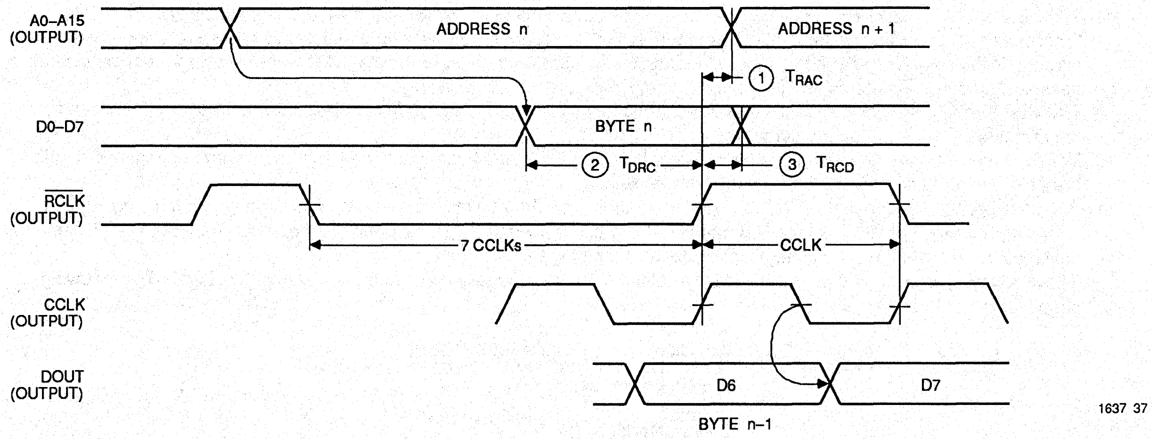
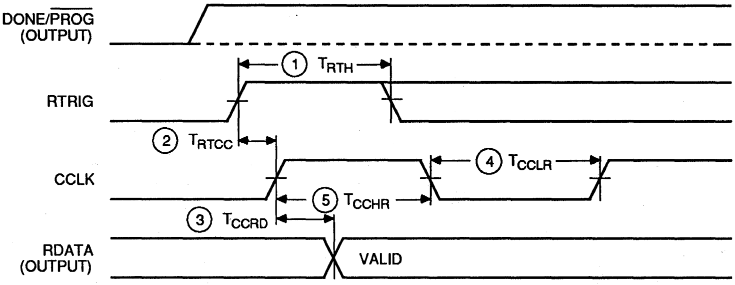


Figure 6. Master Parallel Mode Waveforms



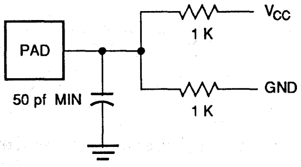
1637 38

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 Table 1 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 LCA's is HIGH. $\overline{\text{WRT}}$ 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 (TCKO) plus the shortest possible interconnect delay is always longer than the Data In hold time requirement (TCKDI) 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, Vcc must rise from 2.0V to Vcc minimum in less than 10 ms. Otherwise, delay configuration using $\overline{\text{RESET}}$.
12. All timings except TTSHZ and TTSON 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.

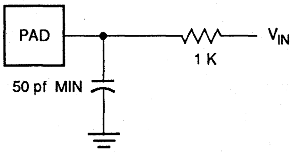
    TTSHZ is determined when the output shifts 10% (of the output voltage swing) from VOL level or VOH level. The following circuit is used:



1637 39

12. (continued)

    TTSON is measured at 0.5 V_{CC} level with VIN = 0 for Tri-State to active high, and VIN = Vcc for Tri-State to active low. The following load circuit is used:



1637 40

XILINX

XC1736 Serial Configuration PROM

Product Specification

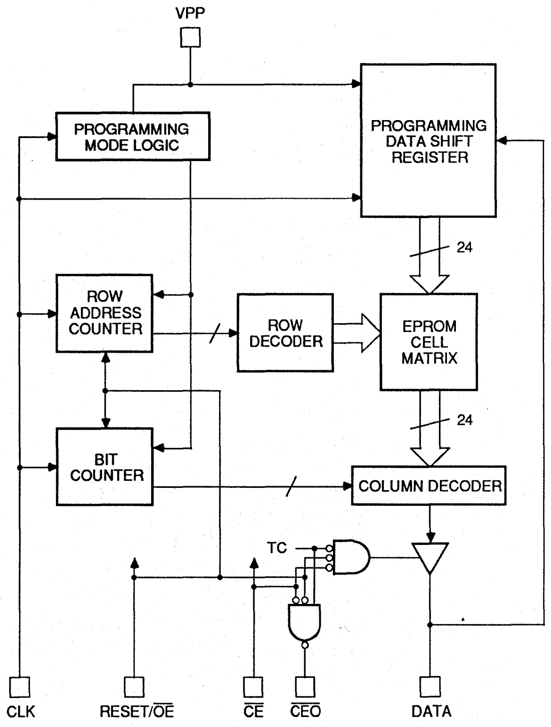
FEATURES

- One-Time Programmable (OTP) 36,288 x 1 bit serial memory designed to store configuration programs for Programmable Gate Arrays
- Simple interface to a XILINX Logic Cell™ Array (LCA) requires only two I/O pins
- Daisy chain configuration support for multiple LCAs
- Cascadable to provide more memory for additional configurations or future higher-density arrays
- Storage for multiple configurations for a single Logic Cell Array
- Low power CMOS EPROM process
- Space-efficient, 8-pin plastic/ceramic DIP package
- PC-based Xilinx programmer for development. Production programming support from leading programmer manufacturers

DESCRIPTION

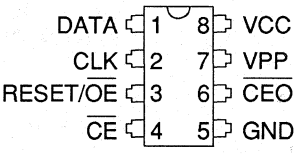
The XC1736 Serial Configuration PROM (SCP) provides an easy-to-use, cost-effective configuration memory for the Xilinx family of programmable gate arrays. Packaged in an economical 8-pin plastic DIP package, the XC1736 uses a simple serial access procedure to configure one or more Logic Cell Arrays (LCAs). The 36,288 x 1 organization of the configuration PROM supplies enough memory to configure three XC2064's, two XC2018's, two XC3020s, one XC3030 or one XC3042. Multiple Serial Configuration PROMs can be cascaded to provide a larger memory for higher density arrays. Multiple configurations for a single LCA can also be loaded from the XC1736.

The XC1736 can be programmed with the PC-based Xilinx XC-DS112 Configuration PROM Programmer or with programmers from other manufacturers. The Logic Cell Array design file is first compiled into a standard HEX format with the XC-DS21 Development System. It can then be transferred to the XC-DS112 programmer connected to the serial port.



1106B 05

Figure 1. XC1736 Block Diagram



1106B 04

XC1736 Pin Assignments

Table 1. XC1736 Pin Assignments

Pin	Name	I/O	Description
1	DATA	O	Three-state DATA output for reading. Input/Output pin for programming.
2	CLK	I	Clock input. Used to increment the internal address and bit counters for reading and programming.
3	RESET/ $\overline{OE}$	I	Output Enable input. A LOW level on both the $\overline{CE}$ and $\overline{OE}$ inputs enables the data output driver. A HIGH level on RESET/$\overline{OE}$ resets both the address and bit counters.
4	$\overline{CE}$	I	Chip Enable input. 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. Used for device selection.
5	GND		Ground pin.
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 follow $\overline{CE}$, but if $\overline{OE}$ goes HIGH, $\overline{CEO}$ will stay HIGH until the entire PROM is read again.
7	VPP		Programming Voltage Supply. Used to enter programming mode (+6V) and to program the memory (+21V) Must be connected directly to Vcc for normal read operation. No overshoot above +22V permitted.
8	Vcc		+5 volt power supply input.

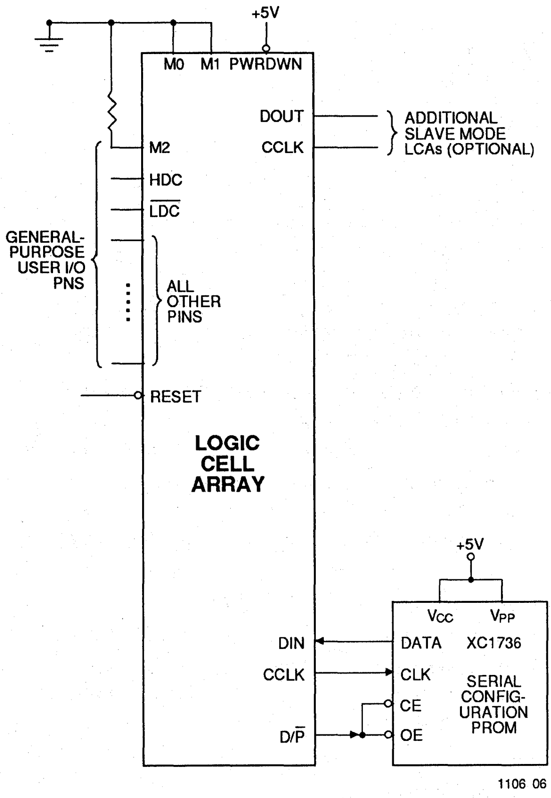


Figure 2. Master Serial Mode Configuration

LCA MASTER SERIAL MODE SUMMARY

The I/O and logic functions of the Xilinx Programmable Gate Array, and their associated interconnections, are established by a configuration program. The program is loaded either automatically upon power up, or on command, 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 will enter Master Serial Mode whenever all three of the LCA's 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/$\overline{\text{P}}$ 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 will clock data out of the SCP on every rising clock edge. At the completion of configuration, the DONE/$\overline{\text{PROG}}$ signal will go high and reset the internal address counters of the SCP.

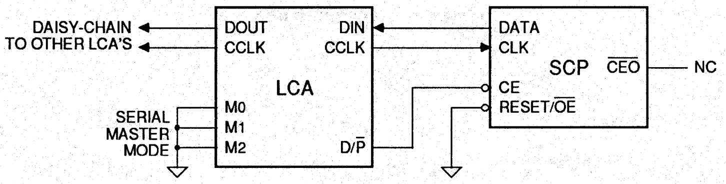
If the user-programmable, dual function DIN and CCLK pins are used only for the configuration process, they should be programmed on the LCA so that no nodes are floating or in contention. For example, both DIN and CCLK can be programmed as output highs during normal operation. An alternate method is to program both DIN and CCLK as inputs, with external pullup resistors attached.

If DIN and CCLK are to be used for another function after configuration, the user must avoid contention. The Low During Configuration ($\overline{\text{LDC}}$) pin can be used to control the SCP's $\overline{\text{CE}}$ and $\overline{\text{OE}}$ inputs to disable the SCP's DATA pin 1 clock cycle before D/$\overline{\text{P}}$ is active.

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

Programming The LCA With Counters Unchanged Upon Completion

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



1106 07

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/$\overline{\text{P}}$ will not go high, so an external signal is required to reset the 1736 counters.

Cascading Serial Configuration PROM's

For multiple LCA's configured as a daisy-chain, or for future LCA's requiring larger configuration memories, cascaded SCP's provide additional memory. A single SCP is large enough for daisy-chains consisting of three XC2064's, two XC2018's, two XC3020's, or a mixture.

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

After configuration is complete, the address counters of all of the cascaded SCP's will be reset when DONE/$\overline{\text{PROG}}$ goes high, forcing the RESET/$\overline{\text{OE}}$ on each SCP to go high.

If the address counters are not to be reset upon completion, then the $\overline{\text{OE}}$ inputs can be tied to ground, as shown in Figure 3. To reprogram the LCA with another program, the DONE/$\overline{\text{PROG}}$ 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 SCP's.

STANDBY MODE

The XC1736 enters a low-power standby Mode whenever $\overline{\text{CE}}$ is asserted high. In this mode the SCP consumes less than 0.5mA of current. The output remains in a high-impedance state regardless of the state of the $\overline{\text{OE}}$ input.

PROGRAMMING MODE

Figure 5 shows the programming algorithm for the XC1736. Note that programming mode is entered by holding VPP high for at least 2 clock edges and is exited by removing power from the device.

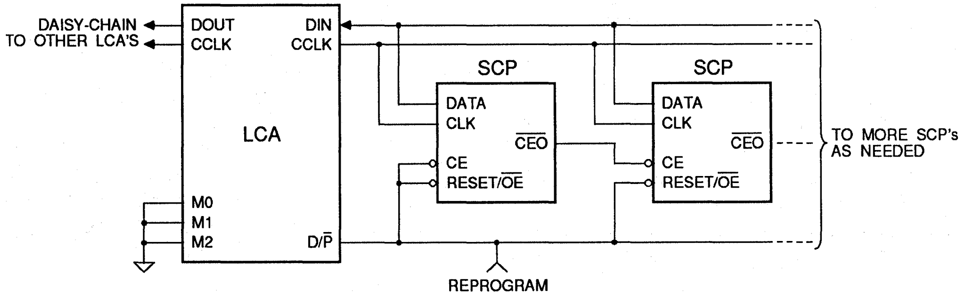
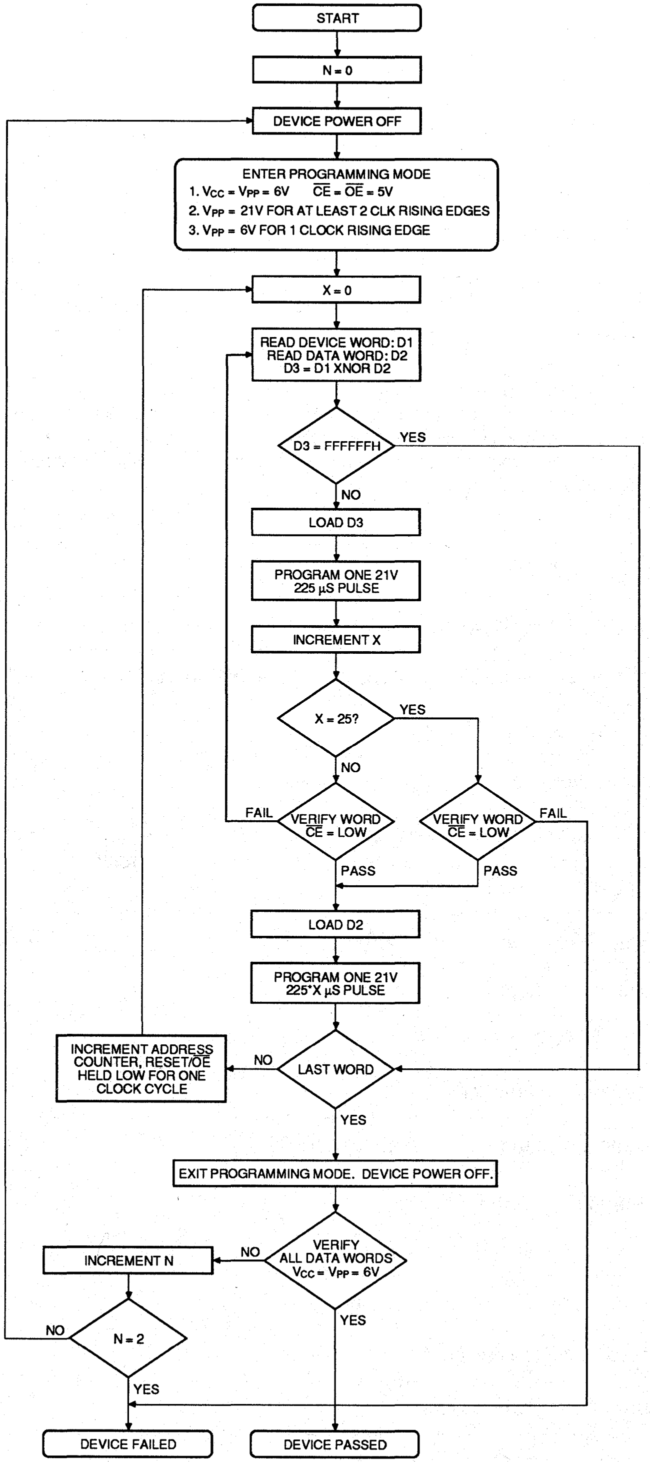


Figure 4. Cascading SCP's



1106 09

Figure 5. Programming Mode

PARAMETRICS

Absolute Maximum Ratings			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 + 125	°C
T_{SOL}	Maximum soldering temperature (10 sec @ 1/16 in.)	+ 260	°C

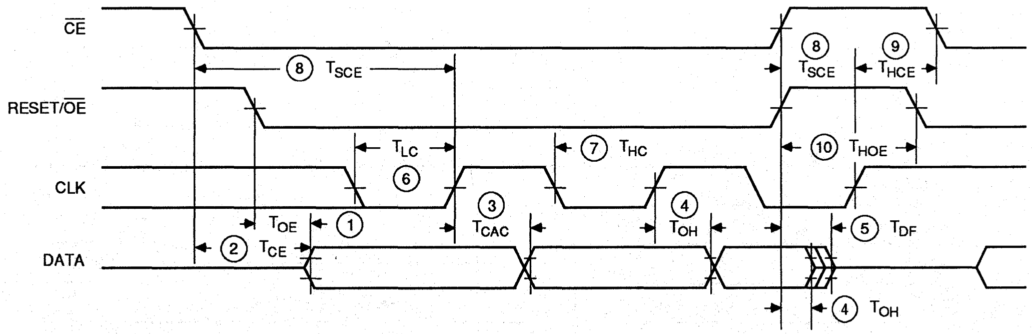
*Note: Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those listed under Recommended Operating Conditions is not implied. Exposure to Absolute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

DC CHARACTERISTICS

Symbol	Description		Min	Max	Units
V_{CC} V_{PP}**	Supply voltage relative to GND Comm/Ind –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_{IH}	High-level input voltage		2.0	V_{CC}	V
V_{IL}	Low-level input voltage		0	0.8	V
V_{OH}	High-level output voltage (I_{OH} = – 4mA)	Commercial	3.86		V
V_{OL}	Low-level output voltage (I_{OL} = 4mA)			0.32	V
V_{OH}	High-level output voltage (I_{OH} = – 4mA)	Industrial	3.76		V
V_{OL}	Low-level output voltage (I_{OL} = 4mA)			0.37	V
V_{OH}	High-level output voltage (I_{OH} = – 4mA)	Military	3.7		V
V_{OL}	Low-level output voltage (I_{OL} = 4mA)			0.4	V
VPP_1	V_{PP} programming voltage (No overshoot permitted)		20	22	V
VPP_2	V_{PP} during programming mode		5.75	6.25	V
ICC_A	Supply current, active mode (I_{CC} + I_{PP})			10	mA
ICC_S	Supply current, standby mode (I_{CC} + I_{PP})			0.5	mA
I_L	Input or output leakage current		–10	10	µA
IPP_P	Supply current, programming mode			50	mA

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

AC CHARACTERISTICS

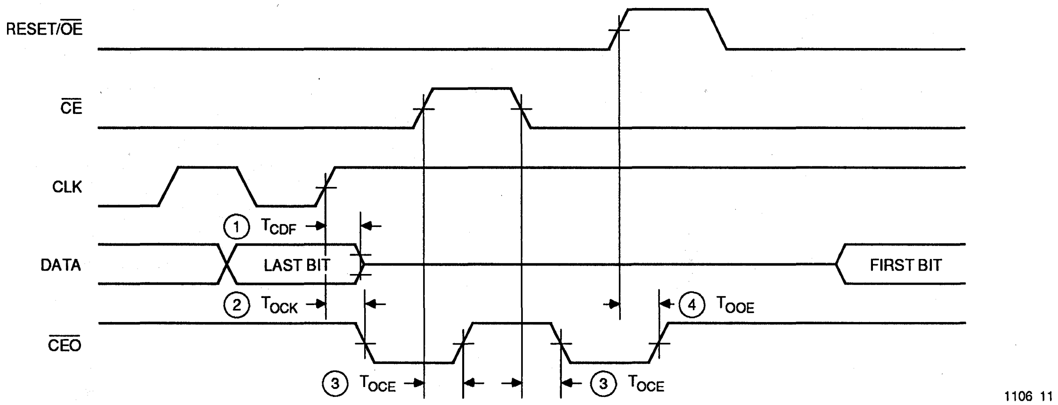


1106 10

Symbol		Description	Limits		Units
			Min	Max	
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	200		ns
7	T_{HC}	CLK High Time	200		ns
8	T_{SCE}	$\overline{CE}$ Setup Time to CLK (Guarantees Counters Will or Will Not Change)	100		ns
9	T_{HCE}	$\overline{CE}$ Hold Time to CLK (Guarantees Counters Will or Will Not Change)	0		ns
10	T_{HOE}	$\overline{OE}$ High Time (Guarantees Counters Are Reset)	100		ns

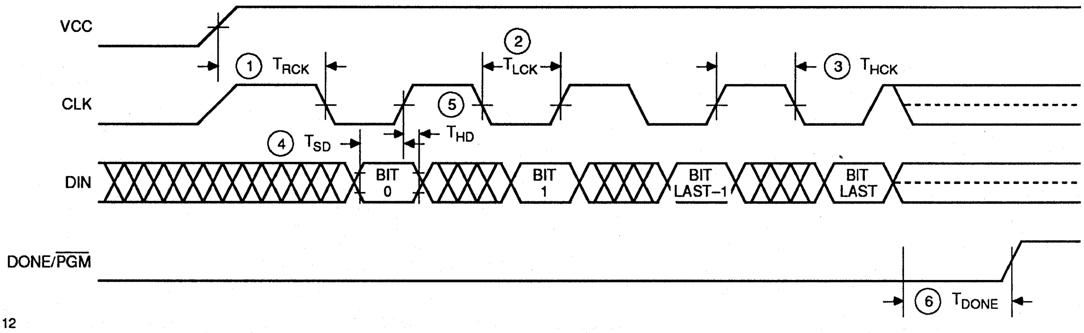
Notes: 1. A.C. test load = 50 pF.
2. Float delays are measured with minimum tester A.C. load and maximum D.C. load.

AC CHARACTERISTICS



Symbol		Description	Limits		Units
			Min	Max	
1	T_{CDF}	CLK to Data Float Delay		40	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

LCA SERIAL MASTER MODE CHARACTERISTICS

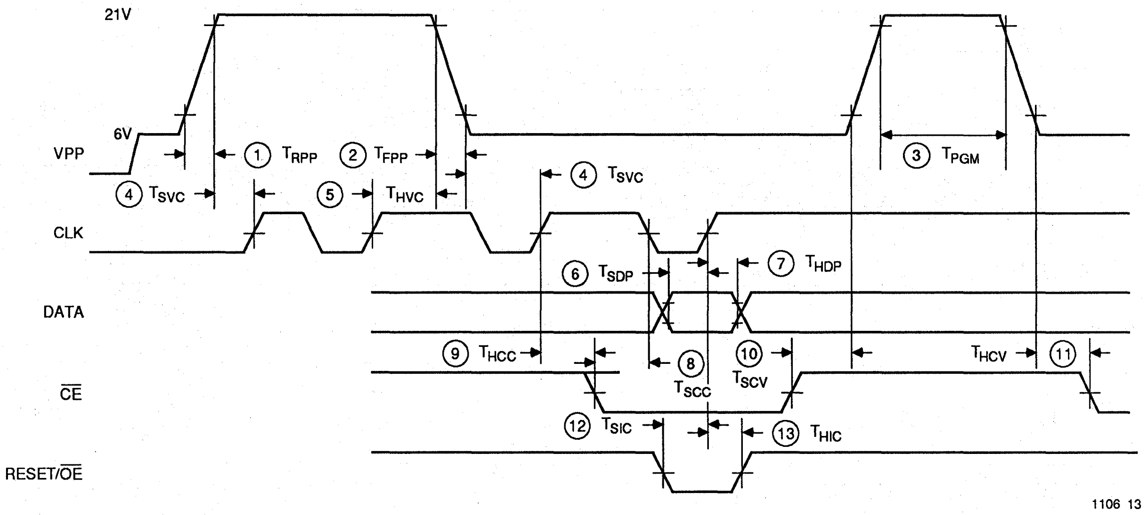


1106 12

Symbol		Description	Limits		Units
			Min	Max	
1	TRCK	Vcc ≈ 3V to RCLK Delay[1]	10	160	ms
2	TLCK	CLK Low Time	250		ns
3	THCK	CLK High Time	250		ns
4	TSD	DIN Setup to CLK	100		ns
5	THD	DIN Hold From CLK	0		ns
6	TDONE	LCA User-Programmed Pins Active to Done Delay		3	µs

[1] This time may be extended by holding the LCA's $\overline{\text{RESET}}$ input low.

PROGRAMMING CHARACTERISTICS



Symbol		Description	Limits		Units
			Min	Max	
1	TRPP	10% to 90% Rise Time of VPP	50	70	µs
2	TFPP	90% to 10% Fall Time of VPP	50	70	µs
3	TPGM	VPP Programming Pulse Width	0.95	1.05	ms
4	TSVC	Setup of VPP to CLK to Enter Programming Mode	100		ns
5	THVC	Hold of VPP to CLK to Enter Programming Mode	300		ns
6	TSDP	Data Setup to CLK for Programming	50		ns
7	THDP	Data Hold to CLK for Programming	0		ns
8	TSCC*	$\overline{\text{CE}}$ Setup to CLK for Programming/Verifying	100		ns
9	THCC*	$\overline{\text{CE}}$ Hold From CLK for Programming/Verifying	200		ns
10	TSCV	$\overline{\text{CE}}$ Setup to VPP for Programming	100		ns
11	THCV	$\overline{\text{CE}}$ Hold From VPP for Programming	50		ns
12	TSIC	$\overline{\text{OE}}$ Setup to CLK to Increment Address Counter	100		ns
13	THIC	$\overline{\text{OE}}$ Hold From CLK to Increment Address Counter	0		ns

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

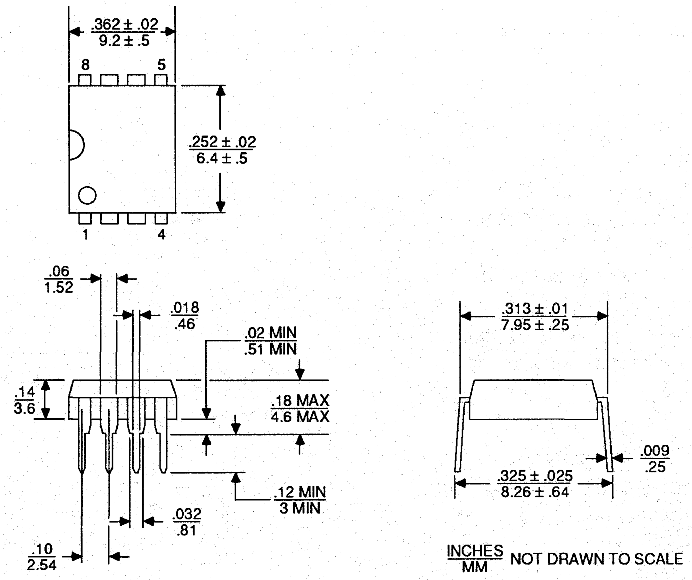
ORDERING INFORMATION

XC1736 - PD8C

PD8 (8 PIN PLASTIC DIP)
CD8 (8 PIN CERAMIC DIP)

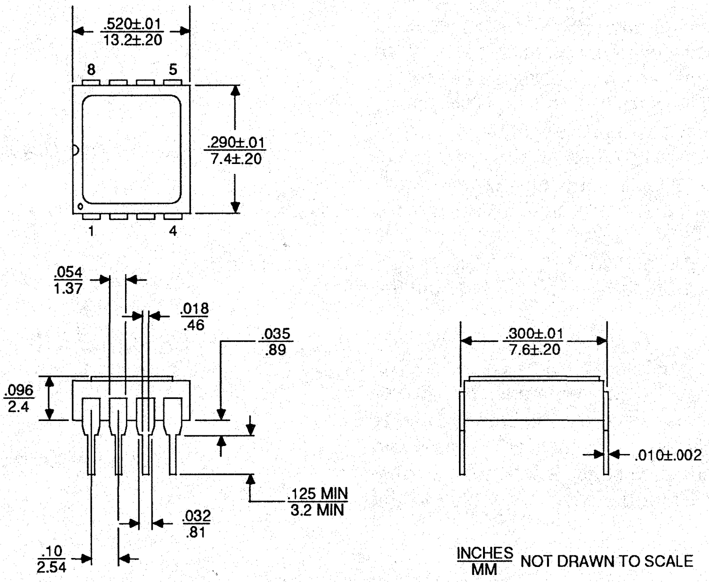
C (COMM/IND –40° TO 85°C)
M (MILITARY TEMP –55° TO 125°)

1106 01



1106 02

8-Pin Plastic DIP Package



1106 14

8-Pin Ceramic Sidebrazed DIP Package

XILINX

XC1764 Serial Configuration PROM

Advance Information

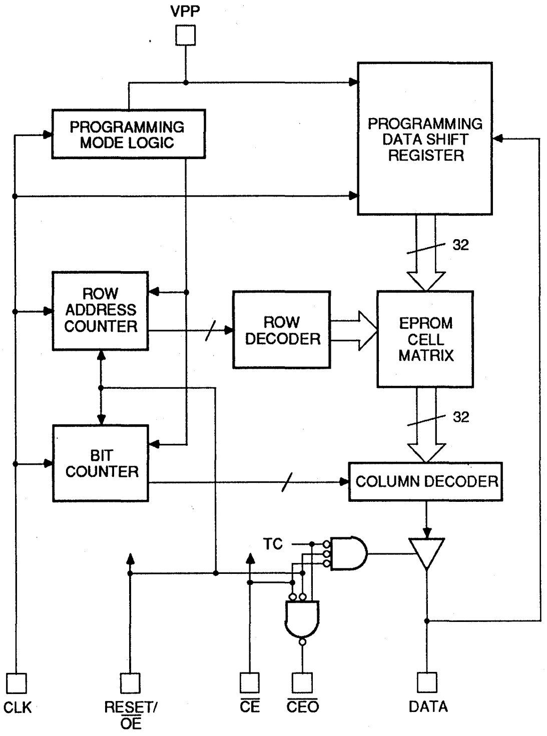
FEATURES

- One-Time Programmable (OTP) 65,536 x 1 bit serial memory designed to store configuration programs for Programmable Gate Arrays
- Simple interface to XILINX Logic Cell™ Arrays (LCA) requires only two I/O pins
- Daisy chain configuration support for multiple XC2000 or XC3000 LCAs
- Cascadable to provide more memory for additional configurations or future higher-density arrays
- Storage for multiple configurations for a single Logic Cell Array
- Low power CMOS EPROM process
- Space-efficient, 8-pin plastic/ceramic DIP package
- PC-based Xilinx programmer for development. Production programming support from leading programmer manufacturers

DESCRIPTION

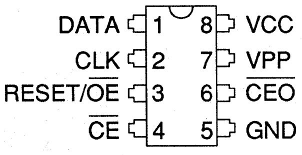
The XC1764 Serial Configuration PROM (SCP) provides an easy-to-use, cost-effective configuration memory for Xilinx programmable gate arrays. Packaged in an economical 8-pin plastic DIP package, the XC1764 uses a simple serial access procedure to configure one or more Logic Cell Arrays (LCAs). The 65,536 x 1 organization of the configuration PROM supplies enough memory to configure one XC3090 or multiple smaller LCA's. Multiple Serial Configuration PROMs can be cascaded to provide a larger memory for higher density arrays. Multiple configurations for a single LCA can also be loaded from the XC1764.

The XC1764 can be programmed with the PC-based Xilinx XC-DS112 Configuration PROM Programmer or with programmers from other manufacturers. The Logic Cell Array design file is first compiled into a standard HEX format with the XC-DS21 Development System. It can then be transferred to the XC-DS112 programmer connected to the PC's serial port.



1951 05

Figure 1. XC1764 Block Diagram



1951 04

XC1764 Pin Assignments

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 devcies, 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
1887 O'Toole Avenue, C103
San Jose, CA 95131
(408) 435-1110

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 - Nepenthe
2471 E. Bayshore Road,
Suite 600
Palo Alto, CA 94303
(415) 856-9332

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 - Nepenthe
2471 E. Bayshore Road,
Suite 600 Palo Alto, CA 94303
(415) 856-9332

XILINX

Component Selection & Ordering Information

		48 PIN		68 PIN		84 PIN		100 PIN		132 PIN		164 PIN	175 PIN	
		PLASTIC DIP	CERAMIC DIP	PLASTIC PLCC	CERAMIC PGA	PLASTIC PLCC	CERAMIC PGA	PLASTIC PQFP	CERAMIC CQFP	PLASTIC PGA	CERAMIC PGA	CERAMIC CQFP	PLASTIC PGA	CERAMIC PGA
		-PD48	-CD48	-PC68	-PG68	-PC84	-PG84	-PQ100	-CQ100	-PP132	-PG132	-CQ164	-PP175	-PG175
XC2064	-33	C	I M	C I	C I M									
	-50	C	I	C I	C I M									
	-70			C I	C I M									
	-100			C	C									
XC2018	-33			C I		C I	C I M B							
	-50			C I		C I	C I M B							
	-70			C I		C I	C I M							
	-100			C		C	C							
XC3020	-50			C I		C I	C I M B	C I	C I M B					
	-70			C I		C I	C I M	C I	C I M					
	-100			C		C	C	C	C					
XC3030	-50			C I		C I	C I M	C I	C I M					
	-70			C I		C I	C I M	C I	C I M					
	-100			C		C	C	C	C					
XC3042	-50					C I	C I M	C I	C I M	C I	C I M B			
	-70					C I	C I M	C I	C I M	C I	C I			
	-100					C	C	C	C	C	C			
XC3064	-50									C I	C I M			
	-70									C I	C I			
	-100									C	C			
XC3090	-50											C I M	C I	C I M B
	-70											C I M	C I	C I M
	-100											C	C	C

1972 01

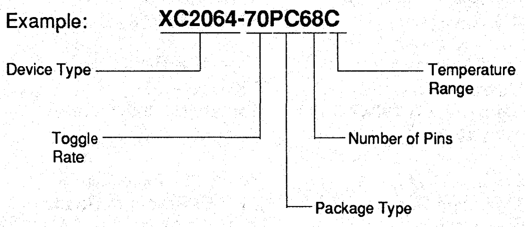
XC1736-PD8C	**Plastic 8 Pin, Mini-DIP**	**–40°C to 85°C**
XC1736-CD8M	**Ceramic 8 Pin, Mini-DIP**	**–55°C to 125°C**

LCA Temperature Options

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

ORDERING INFORMATION

Example: XC2064-70PC68C



COMPATIBLE PACKAGE OPTIONS

A range of Xilinx 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:		
	PLCC 68:	2064-2018-3020-3030
	PLCC 84:	2018-3020-3030-3042
	PGA 84:	2018-3020-3030-3042
	PQFP 100:	3020-3030-3042
	CQFP100:	3020-3030-3042
	PGA 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.

1972

XILINX

SECTION 3

Quality, Testing, Packaging

XILINX

Quality Testing, and Packaging

1962

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 = 7, 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 = 15, 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	N/A to Xilinx packages	N/A to Xilinx packages

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 VDC, 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 Xilinx LCAs]. 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 nonhermetic 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. **Vibration, Variable-Frequency** 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

The Table 2 testing data shows 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 nonhermetic product line. This series of tests is ongoing as a part of the Quality Conformance Program on nonhermetic devices.

XILINX Reliability Testing Summary

Device Types: XC2018, XC2064, XC3020	Process/Technology: 1.2 Micron Double Layer Metal CMOS
Die Attach Method: Silver Epoxy	Package Type: 68 & 84 Lead PLCC
Molding Compound: Sumitomo 6300H	Date: 4Q, 1988

1. High Temperature Life Test 145°C	Combined Sample	Failures	Equivalent Mean Hrs/Device at T = 125°C	Equivalent Device Hrs at T = 125°C	Equivalent Failure Rate in FIT at T_j = 70°C
	765	3	1,041	796,426	56
2. Biased Moisture Life Test T = 85°C; RH = 85%	Combined Sample	Failures	Mean Hrs per Device at T = 85°C	Total Device Hrs at T = 85°C	
	105	1	1000	104,988	
3. Unbiased Pressure Pot Test +121°C, 2 atm sat. steam	Combined Sample	Failures	Mean Hrs per Device	Total Device Hrs	
	504	2	504	105,860	
4. Thermal Shock Test –65°C/+150°C 100 cy. (min)	Combined Sample	Failures	Mean Cycles per Device	Total Device Cycles	
	524	1	469	245,700	
5. Temperature Cycling Test –65°C/+150°C 100 cy. (min)	Combined Sample	Failures	Mean Cycles per Device	Total Device Cycles	
	254	0	334	84,900	
6. Salt Atmosphere Test MIL-STD-883, Method 1009, Cond. A	Combined Sample	Failures	Mean Hrs per Device	Total Device Hrs	
	105	0	24	2,520	
7. Resistance to Solvents Test MIL-STD-883, Method 2105	Combined Sample	Failures			
	28	0			
8. Solderability Test MIL-STD-883, Method 2003	Combined Sample	Failures			
	51	0			

Table 2. Reliability Testing Summary, 1.2 Micron

XILINX XC2018, XC2064, and XC3020 Reliability Testing Summary, Initial Lots

Device Type: XC2018, XC2064, and XC3020	Process/Technology: 1.2 Micron Double Layer Metal CMOS
Die Attach Method: Silver Epoxy	Package Type: 68 & 84 Lead PLCC
Molding Compound: Nitto MP 150 SG	Date: 2Q, 1988

1. High Temperature Life Test 145°C	Combined Sample	Failures	Equivalent Mean Hrs/Device at T = 125°C	Equivalent Device Hrs at T = 125°C	Equivalent Failure Rate in FIT at T_j = 70°C
	1,964	42	2,491	4,892,851	131

Table 2a. Reliability Testing Summary, 1.2 micron

DATA INTEGRITY

Memory Cell Design

An important aspect of the Logic Cell Array's 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 Logic Cell Array user assured of high data integrity in a noisy environment? We must 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

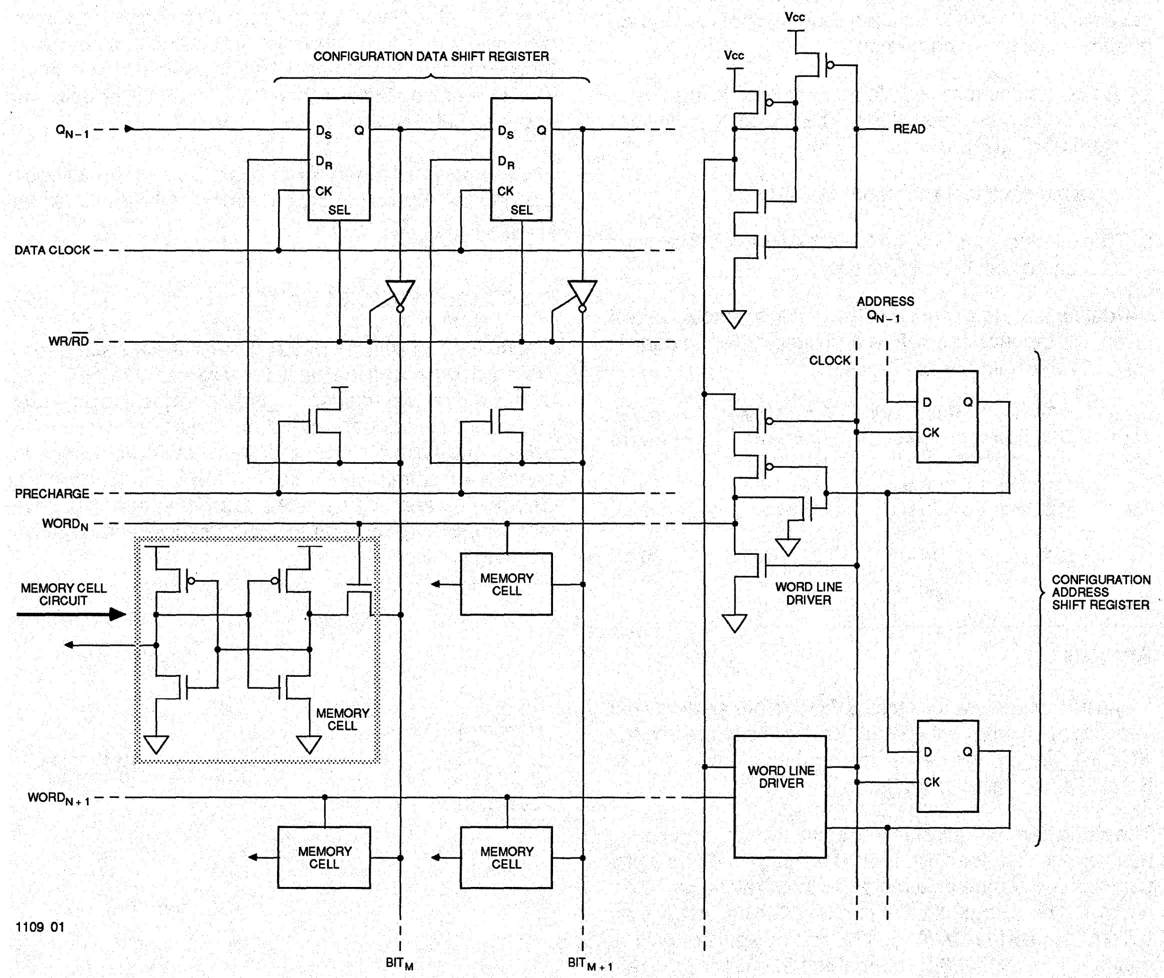


Figure 1. Configuration Memory Cell

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 volts.
2. A pause of variable duration was allowed.
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 sec	70 sec	80 sec	0
2	120 sec	70 sec	190 sec	0
3	300 sec	70 sec	370 sec	0
4	1500 sec	70 sec	1570 sec	0
Total			2210 sec	

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/cm^2/hr). The surface area of the XC-2064 die is less than 0.5 cm^2, 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 test time without error then corresponds 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/cm^2/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/cm^2/hr. Low alpha glasses are specified at 0.8 alpha particles/cm^2/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 cm^2 square). 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 memory cell of the Xilinx Logic Cell Array 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 gives 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.

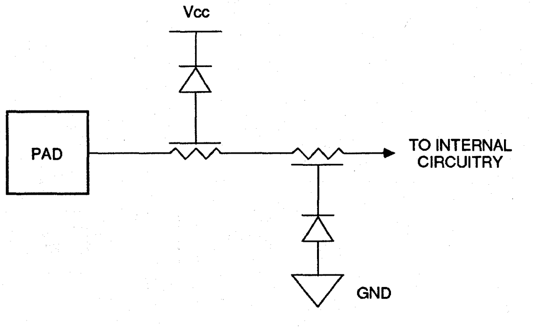


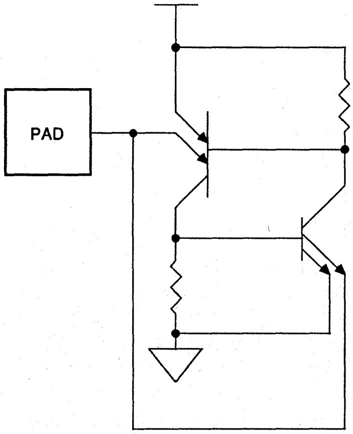
Figure 2. Input Protection Circuitry

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 VBE 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 VCE 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.

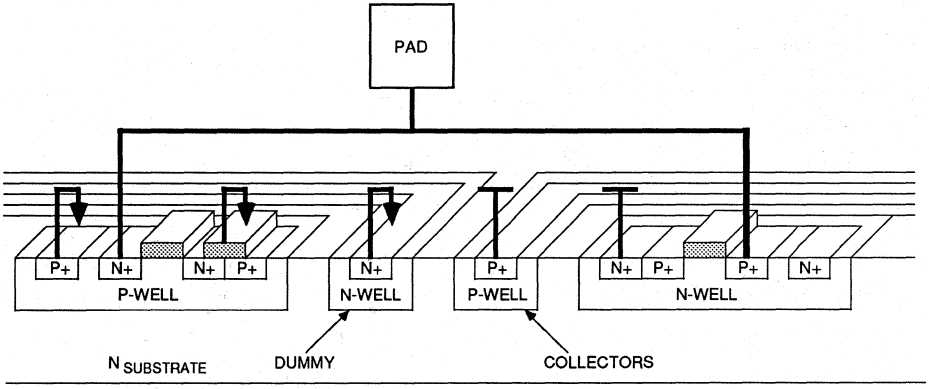
HIGH TEMPERATURE PERFORMANCE

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.



1109 03

Figure 3. SCR Model



1109 04

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 20k, 30k, 50k and 75krads (Si). Each device was exposed to a different radiation level. The radiation was applied at the rate of 1 0rads (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 75krad (Si) level. Test results are summarized in Table 1.

Table 1

Device Number	Gamma Exposure Level	System Functional Test	Tester Results
1	20krad (Si)	Passed	Passed
2	30krad (Si)	Passed	Passed
3	50krad (Si)	Passed	Passed
4	75krad (Si)	Failed*	Failed*

* Would not reconfigure

Total Dose Level: 50krad (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 50krad (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 50krad (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. These factors, coupled with the two hour test time lag and the small sample size, make these tests not fully conclusive. Further testing is planned.

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 Logic Cell Array 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 Logic Cell Array is similar to other standard products, including microprocessors 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 Logic Cell Array test procedure by examining each of the testing requirements:

- All of the 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 config-

ured 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 Logic Cell Array 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 logic cell array is tested as a standard product. Every device is tested for: 1) 100% functionality; 2) D.C. 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 logic cell array is to test the functionality of every element inside the LCA. 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/test pattern consist of: 1) A set of test vectors that configure the LCA 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 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.

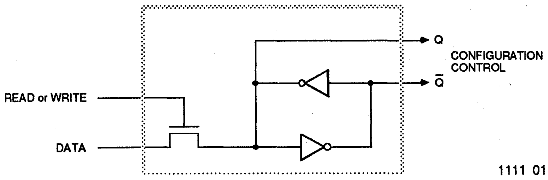


Figure 1. Configuration Memory Cell

The memory cells are directly tested in the logic cell array with three test patterns that are equivalent to those used on a random access memory device. The first test pattern writes 95% of all the RAM cells to a logical 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 logical 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

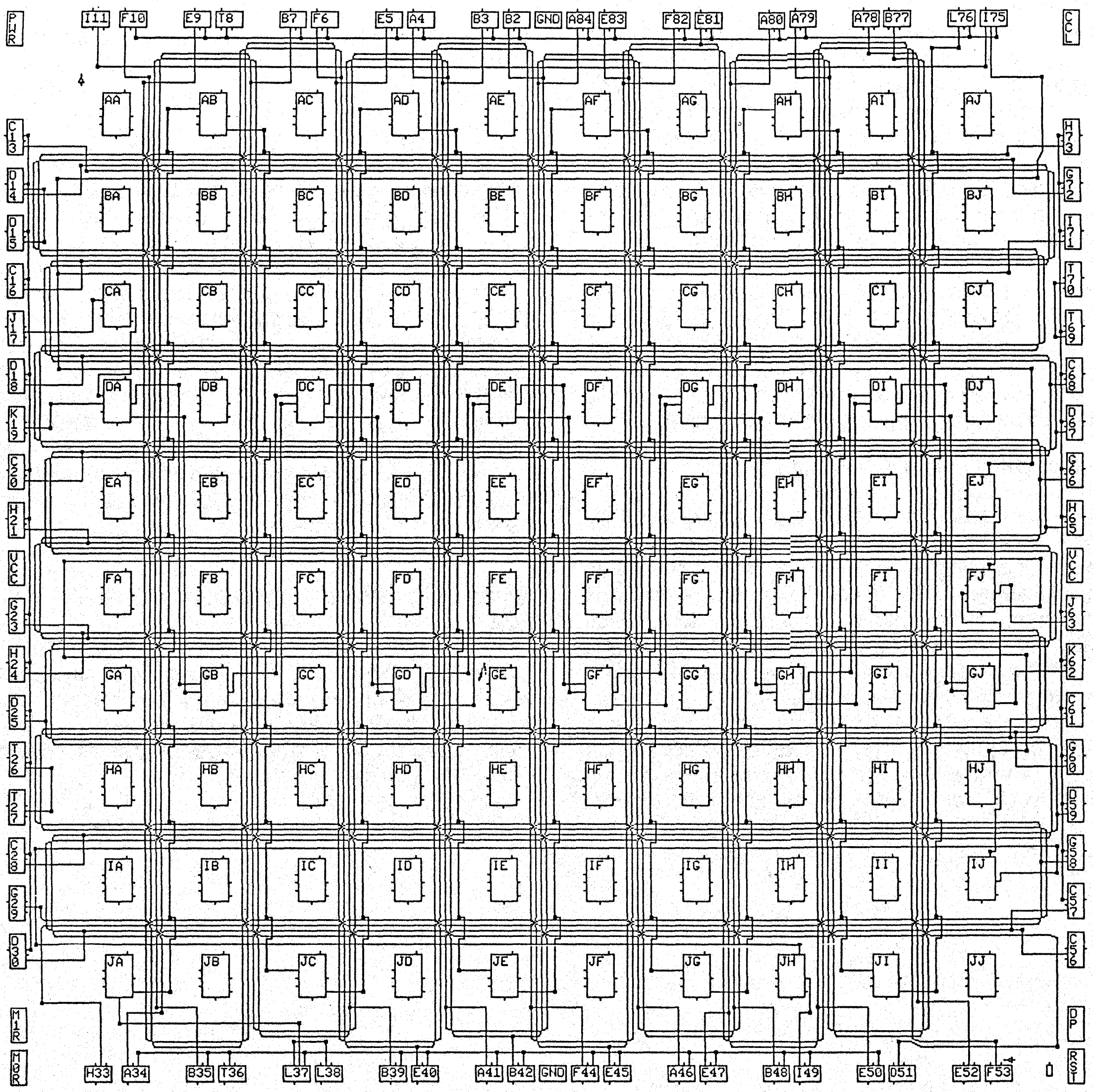

PWR
I11
F10
E9
T8
B7
F6
E5
A4
B3
B2
GND
A84
E83
F82
E81
A80
A79
A78
B77
L76
I75
CCL
C13
D14
D15
C16
J17
D18
K19
C20
H21
VCC
G23
H24
D25
T26
T27
C28
G29
D30
M1R
M0R
H73
G72
I71
T70
T69
C68
D67
G66
H65
VCC
J63
K62
C61
G60
D59
G58
C57
C56
DP
RST
H33
A34
B35
T36
L37
L38
B39
E40
A41
B42
GND
F44
E45
A46
E47
B48
I49
E50
D51
E52
F53
AA AB AC AD AE AF AG AH AI AJ
BA BB BC BD BE BF BG BH BI BJ
CA CB CC CD CE CF CG CH CI CJ
DA DB DC DD DE DF DG DH DI DJ
EA EB EC ED EE EF EG EH EI EJ
FA FB FC FD FE FF FG FH FI FJ
GA GB GC GD GE GF GG GH GI GJ
HA HB HC HD HE HF HG HH HI HJ
IA IB IC ID IE IF IG IH II IJ
JA JB JC JD JE JF JG JH JI JJ

Figure 2. Interconnect Test Pattern

I/O Block Testing

Each I/O Block includes registered and direct input paths and a programmable three-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 Tri-State control (See Figure 3). This allows a test pattern to check the ability of each I/O block to latch and 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 D.C. parameters such as VIH, VIL, VOH, VOL, TTL standby current, CMOS standby current and input/output leakage. The VOH/VOL 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 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

The speed of the device is checked with configuration/test patterns that have been correlated to data sheet A.C. 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 A.C. 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

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 tester with 512K of extended local memory.

Print World: patn01.lca (2018PC84-70), XACT 2.05b Eng, Wed Mar 02 16:02:02 1988 ?rint World: patn01.lca (2018PC84

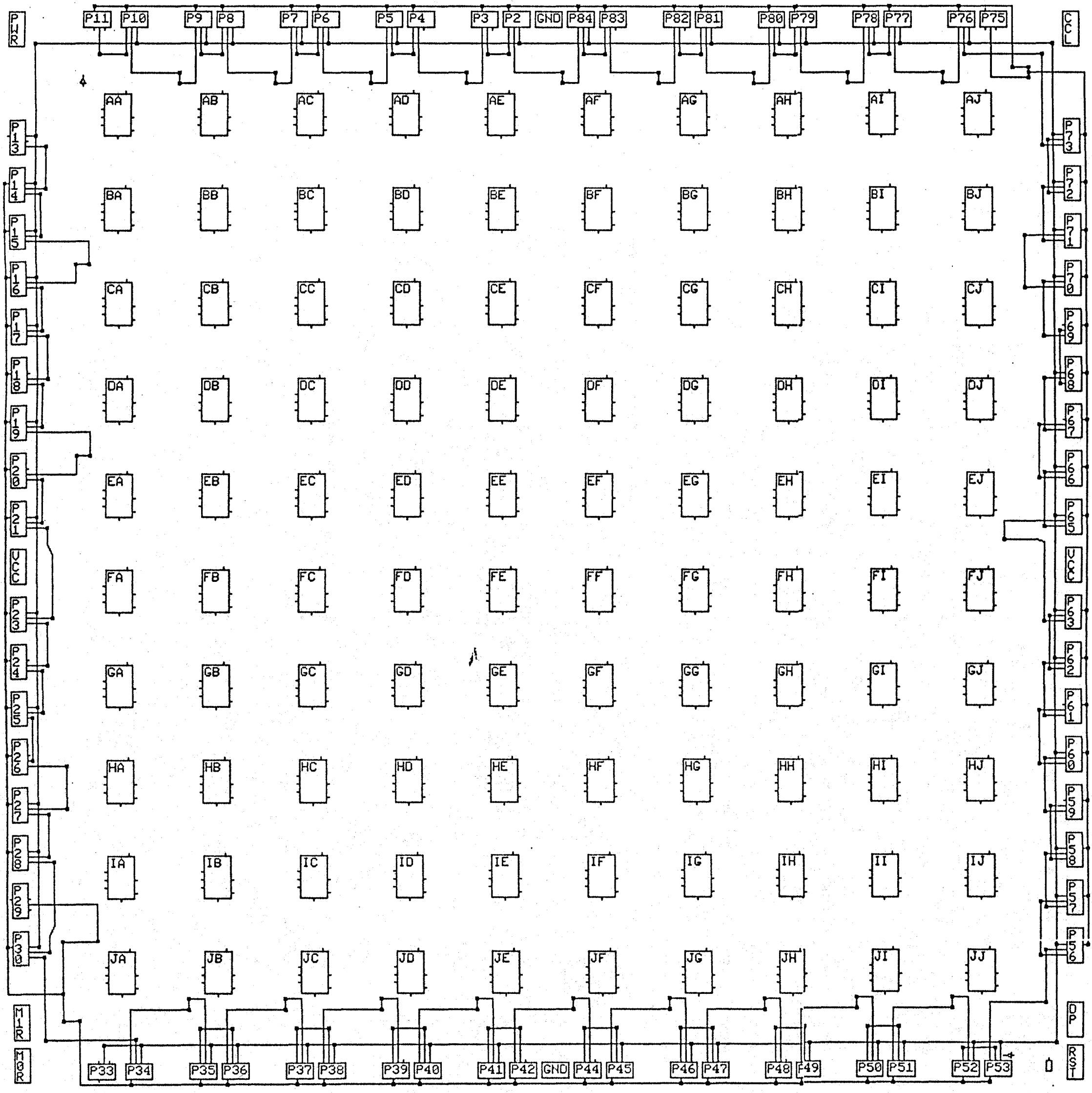

PWR
P11 P10 P9 P8 P7 P6 P5 P4 P3 P2 GND P84 P83 P82 P81 P80 P79 P78 P77 P76 P75
CCL
P13 P14 P15 P16 P17 P18 P19 P20 P21 VCC P23 P24 P25 P26 P27 P28 P29 P30
P73 P72 P71 P70 P69 P68 P67 P66 P65 VCC P63 P62 P61 P60 P59 P58 P57 P56
AA AB AC AD AE AF AG AH AI AJ
BA BB BC BD BE BF BG BH BI BJ
CA CB CC CD CE CF CG CH CI CJ
DA DB DC DD DE DF DG DH DI DJ
EA EB EC ED EE EF EG EH EI EJ
FA FB FC FD FE FF FG FH FI FJ
GA GB GC GD GE GF GG GH GI GJ
HA HB HC HD HE HF HG HH HI HJ
IA IB IC ID IE IF IG IH II IJ
JA JB JC JD JE JF JG JH JI JJ
M1R M0R
P33 P34 P35 P36 P37 P38 P39 P40 P41 P42 GND P44 P45 P46 P47 P48 P49 P50 P51 P52 P53
DP RST

Figure 3. IOB Test Pattern

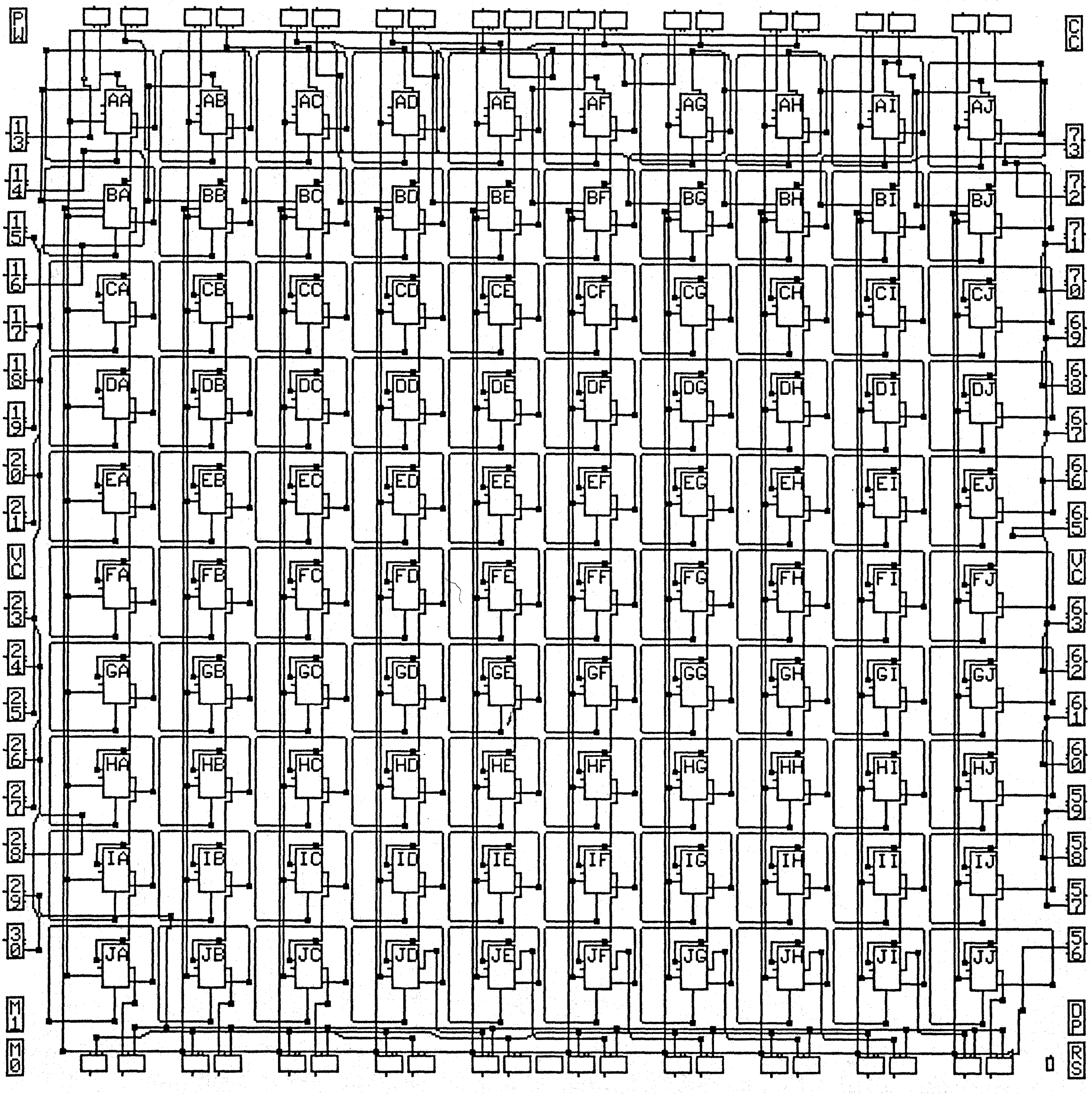

PW
13
14
15
16
17
18
19
20
21
VC
23
24
25
26
27
28
29
30
M1
M0
CC
73
72
71
70
69
68
67
66
65
VC
63
62
61
60
59
58
57
56
DP
RS
AA AB AC AD AE AF AG AH AI AJ
BA BB BC BD BE BF BG BH BI BJ
CA CB CC CD CE CF CG CH CI CJ
DA DB DC DD DE DF DG DH DI DJ
EA EB EC ED EE EF EG EH EI EJ
FA FB FC FD FE FF FG FH FI FJ
GA GB GC GD GE GF GG GH GI GJ
HA HB HC HD HE HF HG HH HI HJ
IA IB IC ID IE IF IG IH II IJ
JA JB JC JD JE JF JG JH JI JJ

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	164 PIN	175 PIN
XC2064	40	58					
XC2018		58	74				
XC3020		58	64	64			
XC3030			74	80			
XC3042			74	82	96		
XC3064					110		
XC3090						142	144

1112B 02

PACKAGE/SPEED/TEMPERATURE SELECTIONS

		48 PIN		68 PIN		84 PIN		100 PIN		132 PIN		164 PIN	175 PIN	
		PLASTIC DIP	CERAMIC DIP	PLASTIC PLCC	CERAMIC PGA	PLASTIC PLCC	CERAMIC PGA	PLASTIC PQFP	CERAMIC CQFP	PLASTIC PGA	CERAMIC PGA	CERAMIC CQFP	PLASTIC PGA	CERAMIC PGA
		-PD48	-CD48	-PC68	-PG68	-PC84	-PG84	-PQ100	-CQ100	-PP132	-PG132	-CQ164	-PP175	-PG175
XC2064	-33	C	I M	C I	C I M									
	-50	C	I	C I	C I M									
	-70			C I	C I M									
	-100			C	C									
XC2018	-33			C I		C I	C I M B							
	-50			C I		C I	C I M B							
	-70			C I		C I	C I M							
	-100			C		C	C							
XC3020	-50			C I		C I	C I M B	C I	C I M B					
	-70			C I		C I	C I M	C I	C I M					
	-100			C		C	C	C	C					
XC3030	-50			C I		C I	C I M	C I	C I M					
	-70			C I		C I	C I M	C I	C I M					
	-100			C		C	C	C	C					
XC3042	-50					C I	C I M	C I	C I M	C I	C I M B			
	-70					C I	C I M	C I	C I M	C I	C I			
	-100					C	C	C	C	C	C			
XC3064	-50									C I	C I M			
	-70									C I	C I			
	-100									C	C			
XC3090	-50											C I M	C I	C I M B
	-70											C I M	C I	C I M
	-100											C	C	C

1972 01

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.

The initial version will be a 164 lead package for the XC3090. First production is scheduled for IQ 1989. Other versions for the rest of the 3000 family will follow with production early in 1989.

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.

The initial versions will be 31 Mil (0.8 mm) pitch, 80 leads and 20 Mil (0.65 mm) pitch, 100 leads. First production is scheduled for 4Q 1988.

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
(312) 677-2850

XILINX

SECTION 4

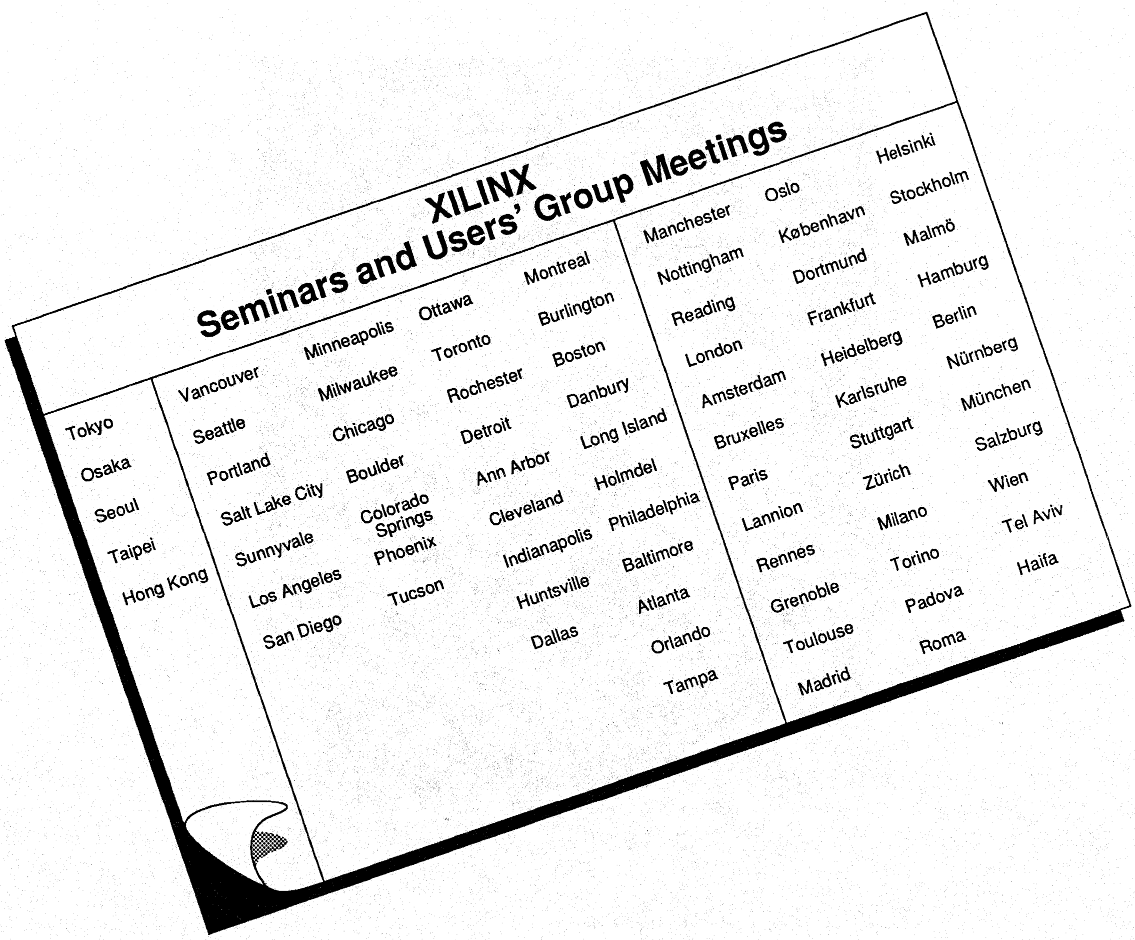
Technical Support

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.

1962

Technical Seminars and Users' Group Meetings



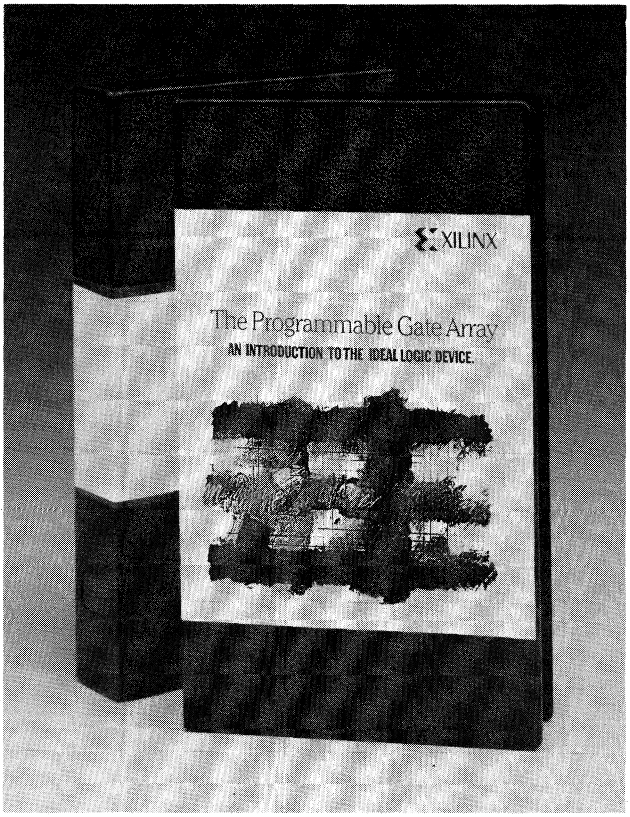
Xilinx sponsors technical seminars at locations throughout North America, Europe, and Asia.

Product-oriented seminars are directed toward new and potential users of Programmable Gate Arrays. These seminars include a basic description of the Logic Cell Array architecture and its 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 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.

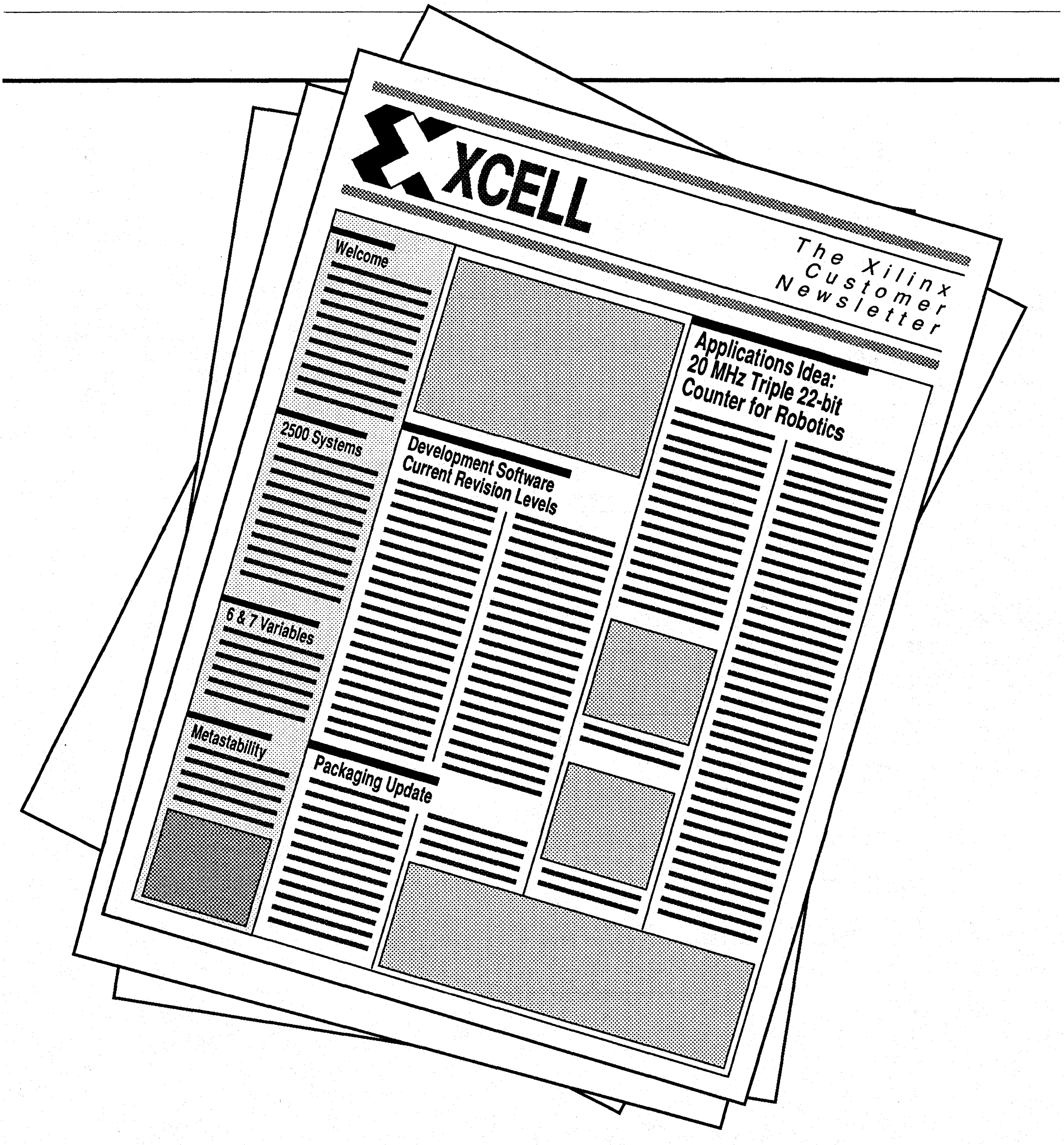
Videotapes


XILINX
The Programmable Gate Array
AN INTRODUCTION TO THE IDEAL LOGIC DEVICE.

A one-hour videotape, 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 videotape 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 videotapes 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 Xilinx Customer Newsletter
Welcome
2500 Systems
6 & 7 Variables
Metastability
Development Software Current Revision Levels
Packaging Update
Applications Idea: 20 MHz Triple 22-bit Counter for Robotics

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 Xilinx LCAs.

Xilinx Technical Bulletin Board

M)SG-SECTION

Enter this section to send and receive messages.

TYPE:

E<CR> [E)NTER]
To send a message.

T<CR> [T)O-YOU]
To read messages.

L<CR> [L)IST]
To list headers of all readable messages in the area. Some messages are private and can be read only by the addressee.

K<CR> [K)ILL]
To delete a mesage you've just read.

R<CR> [R)EPLY]
To reply to a mesage you've just read.

F)ILE-SECTION

The file section is divided into several areas. Enter this section to read, upload and download files.

TYPE:

A<CR> [A)REA]
To list the existing file areas.

After choosing a file area, you can now list all the files in this area.

TYPE:

F<CR> [F)ILES]
To display the available files, the size of each file in bytes and a brief description of the contents of each file.

D<CR> [D)OWNLOAD]
To download one or more files from the bulletin board.

L<CR> [L)IST]
To locate a file in any accessable area

T<CR> [T)YPE]
To display a text file on the screen.

* Files can be uploaded from any file area by typing:

U<CR> [U)PLOAD]

B)ULLETIN

Enter this section to read the "latest and greatest" information on the Xilinx Bulletin Board. A list of bulletins is automatically displayed.

TYPE:

<BULLETIN#><CR>
To display a specific bulletin on the screen.

G)OODBYE

At this time you can leave a message for the system operator. See the M)SG-Section for instructions on how to send messages to other bulletin board users.

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 prompted 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).

1157B

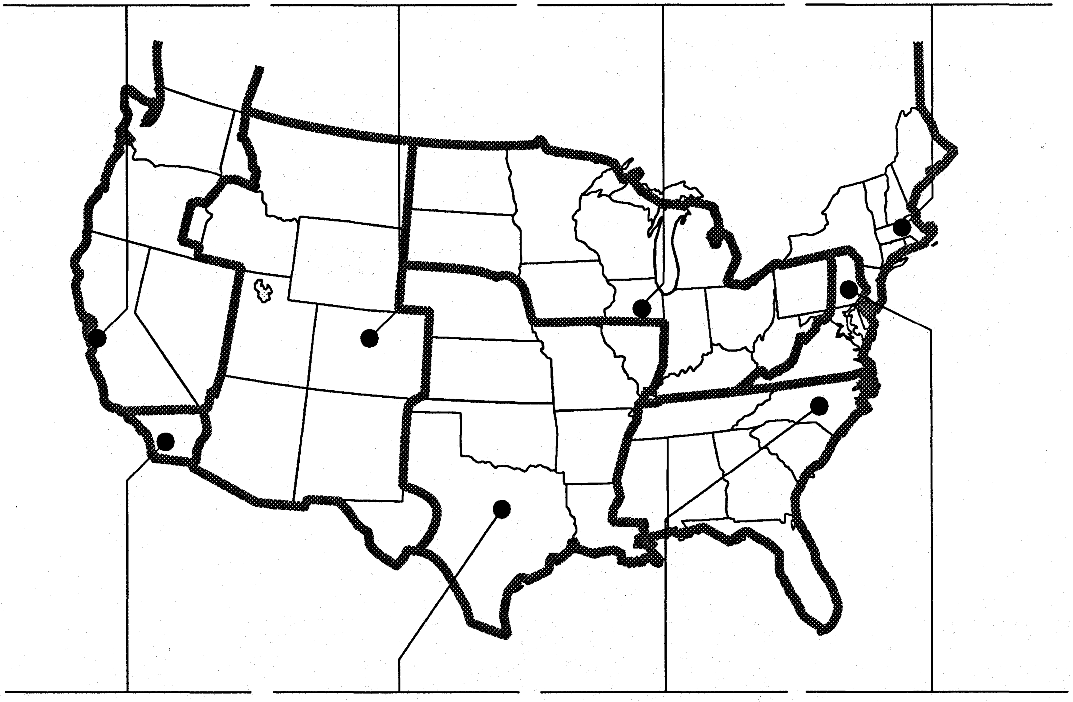
Field Applications Engineers

1270 Oakmead Pkwy.
Suite 201
Sunnyvale, CA 94086
Tel: 408-245-1361

3100 Arapahoe Rd.
Suite 404
Boulder, CO 80303
Tel: 303-443-4780

919 N. Plum Grove Rd.
Suite A
Schaumburg, IL 60173
Tel: 312-490-1972

61 Spit Brook Road
Suite 403
Nashua, NH 03060
Tel: 603-891-1096



2081 Business Center Dr.
Suite 108
Irvine, CA 92715
Tel: 714-955-0831

14506 Sandyside Dr.
Austin, TX 78728
Tel: 512-251-7148

10704 Spiralwood Ct.
Raleigh, NC 27612
Tel: 919-846-3922

65 Valley Stream Parkway
Suite 140
Malvern, PA 19355
Tel: 215-296-8302

North America
There are 9 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
The Xilinx European sales office in England has a resident Field Applications Engineer (tel 44-73081-6725).

Japan
Xilinx Japan is located in Tokyo and has a resident Field Applications Engineer (tel 3-297-9191).

1157B

XILINX

Programmable Gate Array Training Course

The Xilinx Programmable Gate Array Training Course is a comprehensive class covering the Logic Cell Array component architecture and Xilinx development systems, with emphasis on the XC3000 family. This course is intended for design engineers using Programmable Gate Arrays in their applications who want to get "up-to-speed" as quickly as possible.

Courses run for a full four days, Monday through Thursday, 9 AM to 5 PM. Course notes, binders and lunches will be provided by Xilinx. A substantial amount of the class time will be spent performing lab exercises on the Xilinx development system (2 students per development system). These development systems will be available to the students on the Friday following the class for optional individual work and consultations with the instructor(s).

TUITION: The tuition fee is $850 per student.

ENROLLMENT: To enroll, call the Training Administrator at Xilinx headquarters (408) 559-7778 or contact your local Xilinx sales offices.

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:

Training Administrator
Xilinx
2100 Logic Drive
San Jose, CA 95124

Enrollments will be acknowledged with a confirmation letter.

PREREQUISITES: Students are assumed to have a background in digital logic design and familiarity with PC's and MS/DOS.

LOCATION: All courses are held at Xilinx corporate headquarters in San Jose, CA. Xilinx is located approximately 10 miles from San Jose International Airport and 45 miles from San Francisco International Airport. A detailed map, including a listing of nearby hotels, will be included with the enrollment confirmation letter.

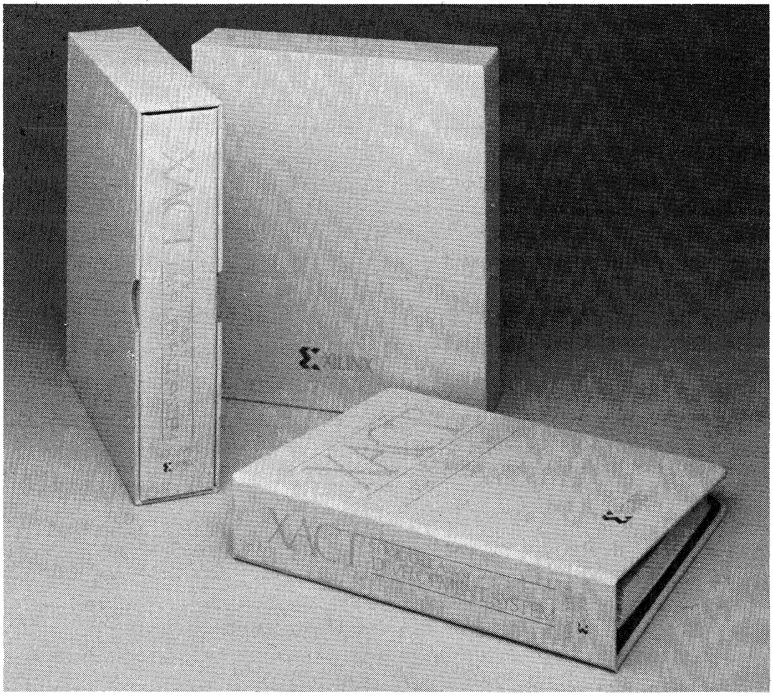
XILINX PROGRAMMABLE GATE ARRAY TRAINING COURSE OUTLINE

DAY 1:	**I.**	**BASIC ARCHITECTURE** LOGIC CELL ARRAY ARCHITECTURAL OVERVIEW XC3000 FAMILY ARCHITECTURE CLB STRUCTURE IOB STRUCTURE BASICS OF INTERCONNECT DEDICATED PINS
	II.	**DESIGN METHODOLOGY OVERVIEW**
	III.	**DESIGN ENTRY** SCHEMATIC CAPTURE PALASM TO XNF TRANSLATOR LOGIC OPTIMIZATION Lab Exercise — Design Entry
DAY 2:	**IV.**	**DESIGN IMPLEMENTATION** LOGIC PARTITIONING AUTOMATIC PLACE AND ROUTE CONFIGURATION BIT STREAM GENERATOR BIT STREAM FORMAT PROM FORMATTER
	V.	**LCA ARCHITECTURE** LCA CONFIGURATION LOADING MODES
	VI.	**DESIGN IMPLEMENTATION** ADVANCED TOPICS Lab Exercise — Design Implementation
DAY 3:	**VII.**	**DESIGN VERIFICATION** SIMULATION XACT DOWNLOAD CABLE XACTOR2 IN-CIRCUIT DEBUGGER READBACK
	VIII.	**XACT LCA DESIGN EDITOR** Lab Exercise — Simulation
DAY 4:	**IX.**	**LCA ARCHITECTURE** XC3000 INTERCONNECT DETAILS
	X.	**SUMMARY** ESTIMATING SIZE AND PERFORMANCE BENEFITS OF PGA TECHNOLOGY Lab Exercise — Using the XACT LCA Editor

TECHDOC 1950

XACT Manuals

LCA Development System Manual, Volume 1 and 2
LCA Macro Library



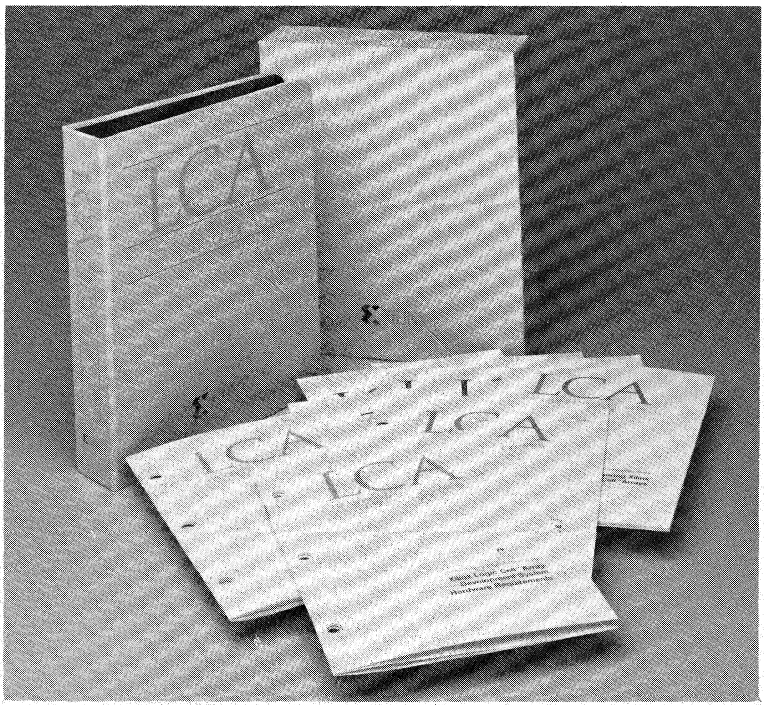
The first two binders of this 3-volume set are the LCA Development System Manuals, providing exhaustive reference information on:

- 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.

XILINX

User's Guide



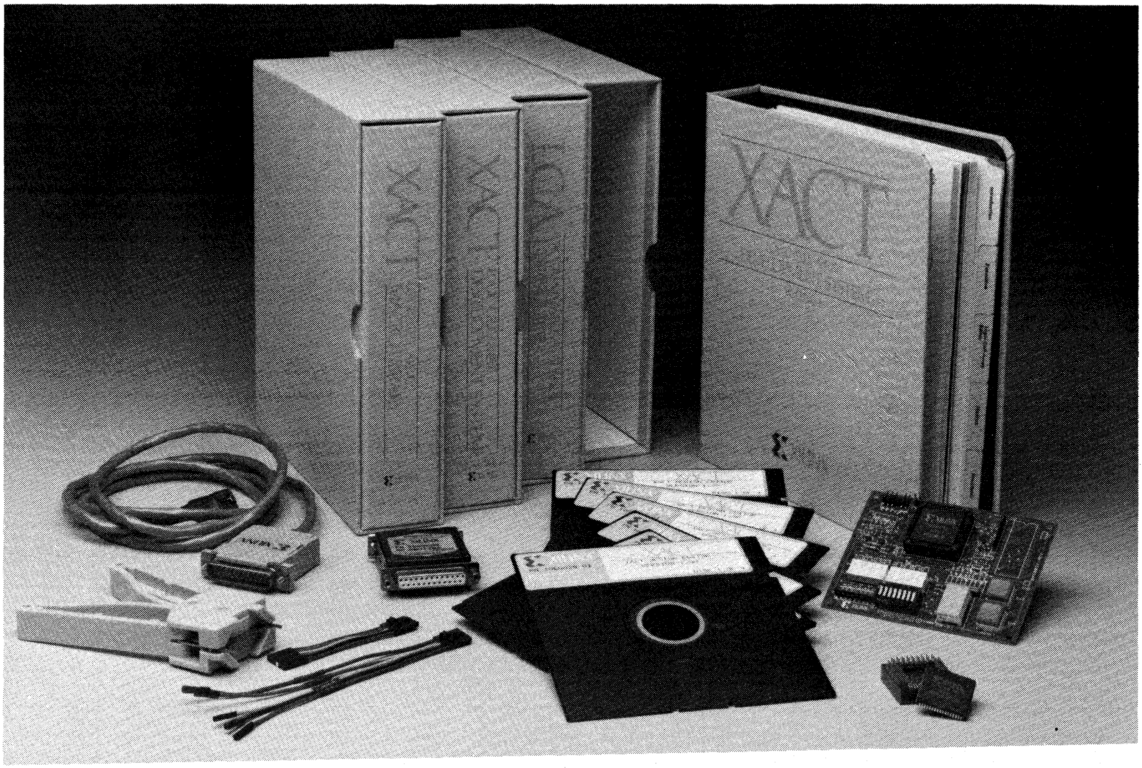
The Xilinx User's Guide, included with every system, is a binder with several self-contained application notes giving practical and tutorial information.

The Users Guide currently has the following manuals:

- Introduction
  Development System Hardware Requirements
  Basic Design Flow
  Introduction to Hierarchical Design and Merging
- Getting Started
  Xilinx Programmable Gate Array Design Flow
- Design Entry
  Designing LCAs With Boolean Equations
- Design Implementation
  Configuring Xilinx Logic Cell Arrays
  Advanced Design Methodology
  Fundamentals of Placement and Routing
- Design Verification
  Simulating Bidirectional I/O's Using Silos
  Readback and Signature Analysis
- Design Examples

This series of notes is being expanded.

1157B



1157B

SECTION 5

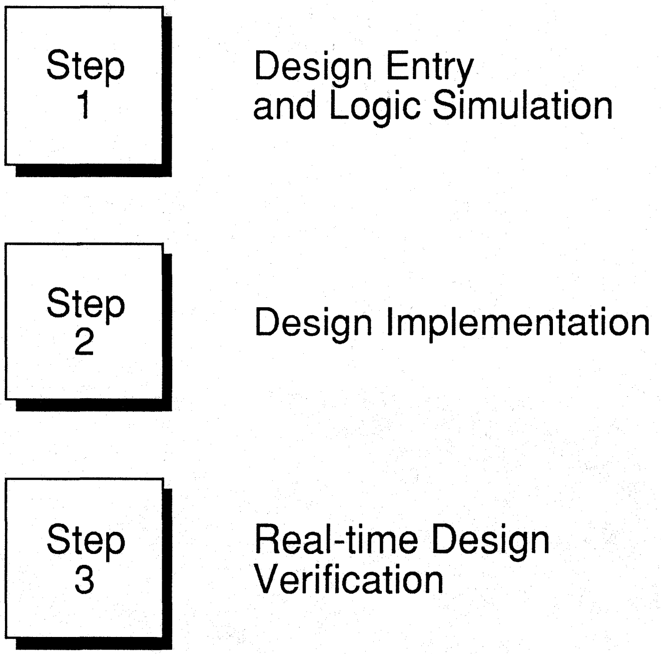
Development Systems

XILINX

Development Systems

1962

Xilinx Design Flow Overview (PC or Workstation)



Step 1

Design Entry
and Logic Simulation

Step 2

Design Implementation

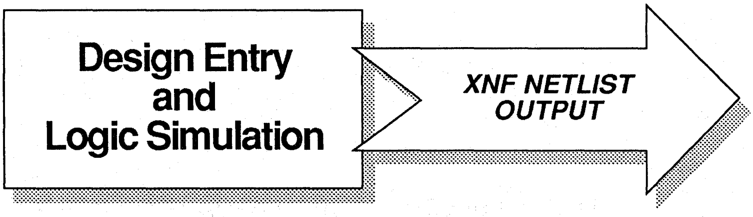
Step 3

Real-time Design
Verification

1955 01

Logic Cell Array Design Flow

STEP 1



Design Entry and Logic Simulation

XNF NETLIST OUTPUT

❑ Open development system supports design entry and simulation on popular CAE systems

❑ Interfaces available for PC and workstation-based environments:

  □ FutureNet, Schema, CASE, VIEWlogic, PCAD, OrCAD, etc.

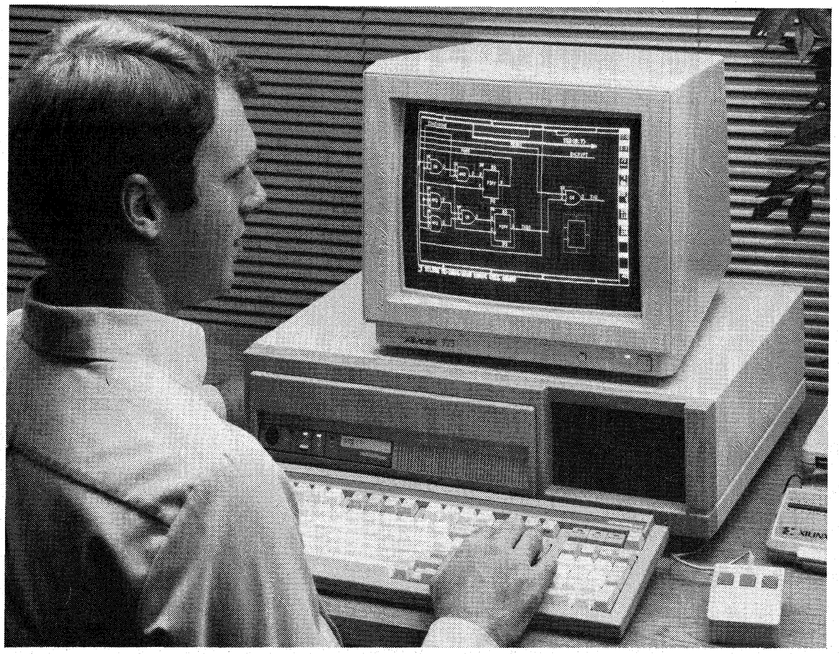
  □ Mentor, Daisy, Valid

  □ SILOS, CADAT, OrCAD simulators

❑ Standard macro library includes over 100 elements

❑ Logic simulation can be used to verify design function

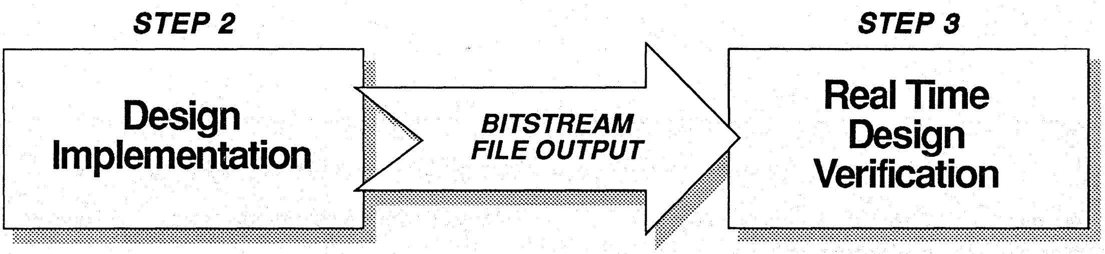
❑ ABEL, CUPL, Log/IC, PLDesigner state machine designs may be merged with schematic input



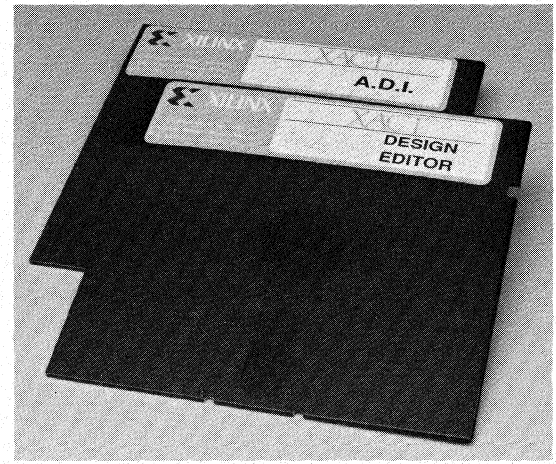
Schematic entry on your PC/workstation

1955

Logic Cell Array Design Flow

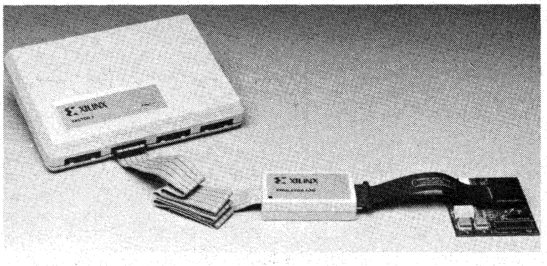


- ❑ Complete system translates design into programmable gate arrays
- ❑ Partitions gate level design logic into Logic Cell Array architecture (CLB/IOB)
- ❑ Automatic logic reduction and partitioning removes unused logic (e.g. unused counter outputs)
- ❑ Logic synthesis software optimizes design for Logic Cell Array architecture
- ❑ All programs run on IBM™ PC/AT™ or compatible personal computer
- ❑ Automated programs run on SUN3 and Apollo engineering workstation computers

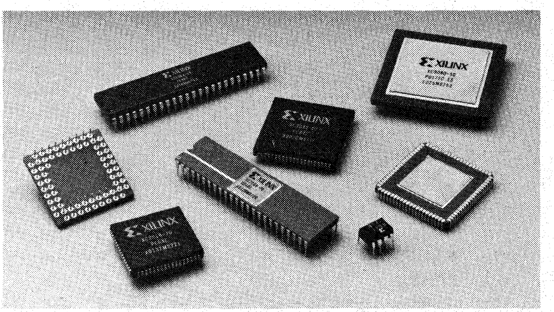


Automated Design Implementation software

- ❑ LCA user-programmability permits real time, in-circuit debugging
- ❑ Download cable allows LCA to be programmed in-circuit from designer's PC during debugging
- ❑ XC-DS28 XACTOR In-Circuit Design Verifier reads and displays the internal LCA storage element states



DS28 XACTOR In-Circuit Design Verifier facilitates real-time, in-circuit analysis

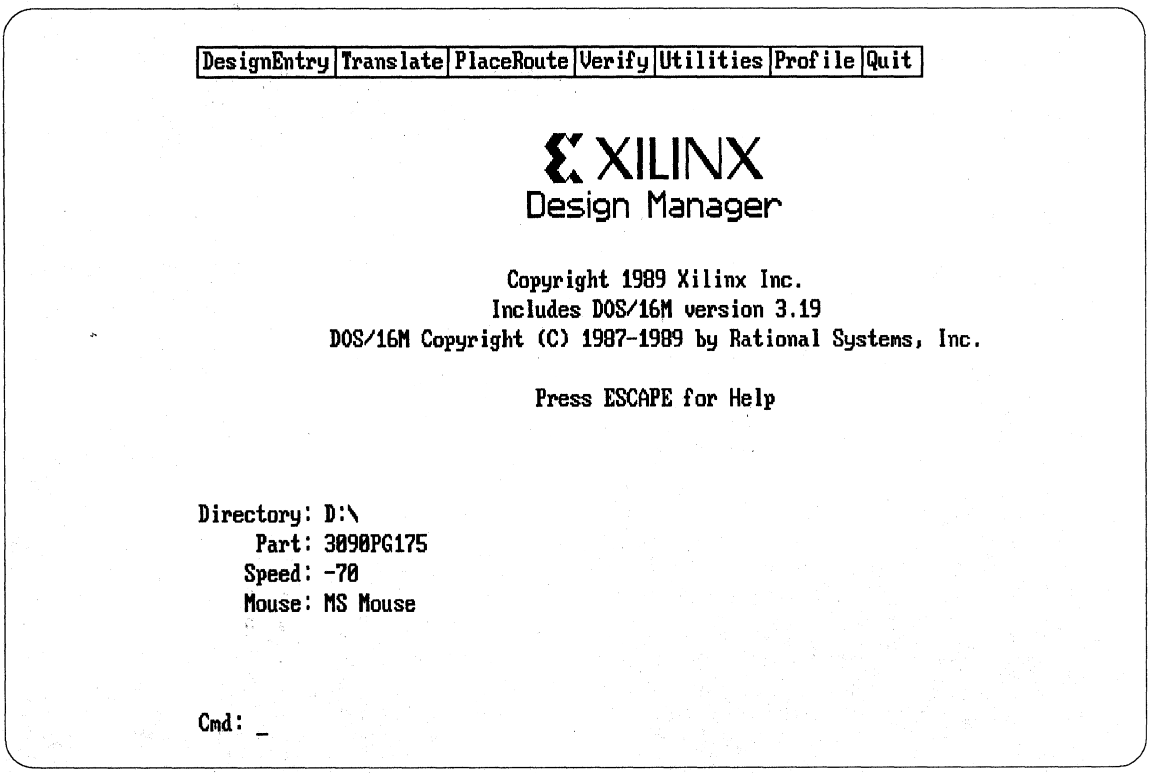


Immediate production

1955

The Xilinx Design Manager—Simplifies the Design Flow

- Lets you run all Xilinx software from menus
- Automates the *entire* translation, optimization, mapping, and merging process
- Provides on-line help for all programs and options



```
Copyright 1989 Xilinx Inc.
Includes DOS/16M version 3.19
DOS/16M Copyright (C) 1987-1989 by Rational Systems, Inc.

Press ESCAPE for Help

Directory: D:\
     Part: 3090PG175
    Speed: -70
    Mouse: MS Mouse

Cmd: _
```

1954 01

Design Documentation

The Design Manager maintains an up-to-date text file listing:

- All sub-design files (e.g., "PAL" modules, schematics, etc.)
- Which program options are used to translate the design

Extensive On-line Help

The Design Manager contains on-line Help for

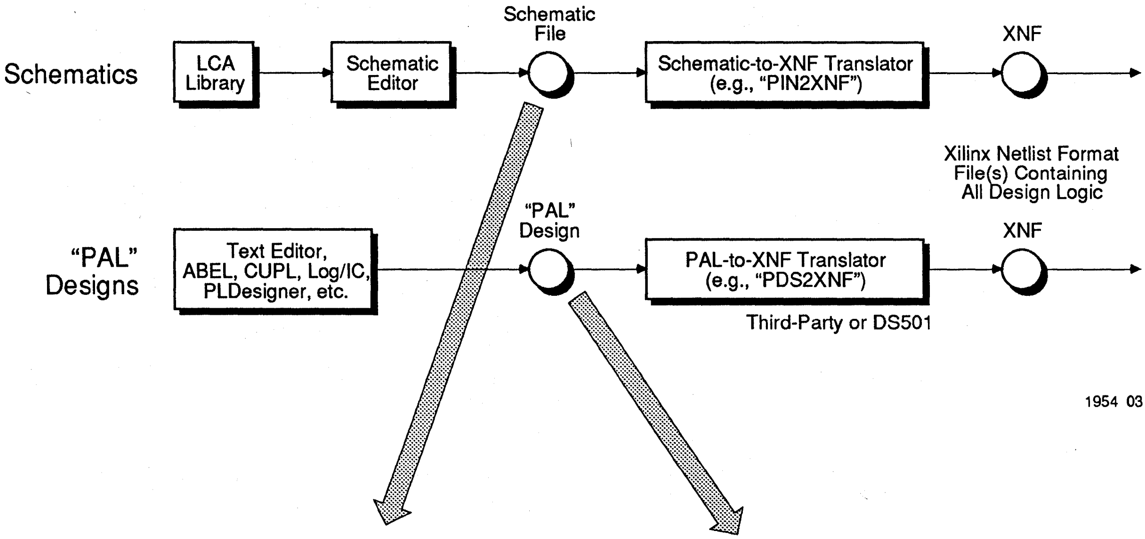
- Every menu
- Every program
- Every program option
- Design flow suggestions

Design Processing Sequence	Design Manager Menu	Description
①	Design Entry	Create your design by running your schematic editor while inside the Design Manager
②	Translate	With one mouse click, *automatically* • Translate all schematic pages and "PAL" designs to Xilinx format • Optimize "PAL" designs and state machines with Logic Synthesis • Map all logic into CLBs and IOBs • Merge PAL designs and state machines into schematic
③	Place Route	Run XACT Design Editor and/or APR to place/route design
④	Place Route	Run MAKEBITS program to compile design into a configuration bitstream for an LCA
⑤	Verify	Run simulator (e.g., SILOS) or in-circuit verifier (e.g., XACTOR) to debug your design

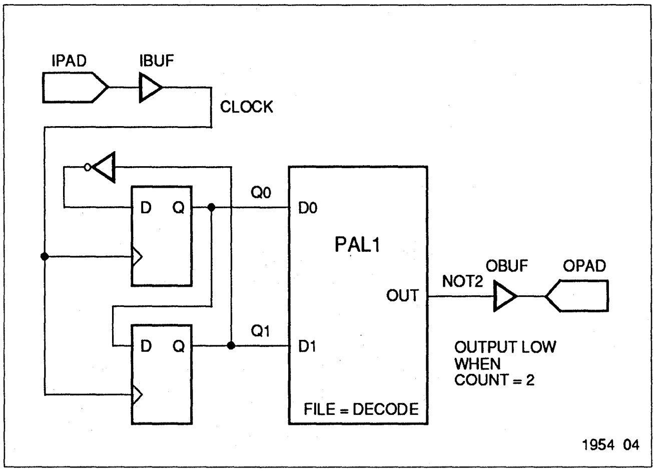
The Xilinx Design Manager provides a highly automated environment for converting your designs into working programmable gate array designs.

This sequence is illustrated — for a very simple design — on the following pages...

Design Entry and XNF Translation


Schematics
LCA Library
Schematic Editor
Schematic File
Schematic-to-XNF Translator (e.g., "PIN2XNF")
XNF
Xilinx Netlist Format File(s) Containing All Design Logic
"PAL" Designs
Text Editor, ABEL, CUPL, Log/IC, PLDesigner, etc.
"PAL" Design
PAL-to-XNF Translator (e.g., "PDS2XNF")
XNF
Third-Party or DS501

1954 03



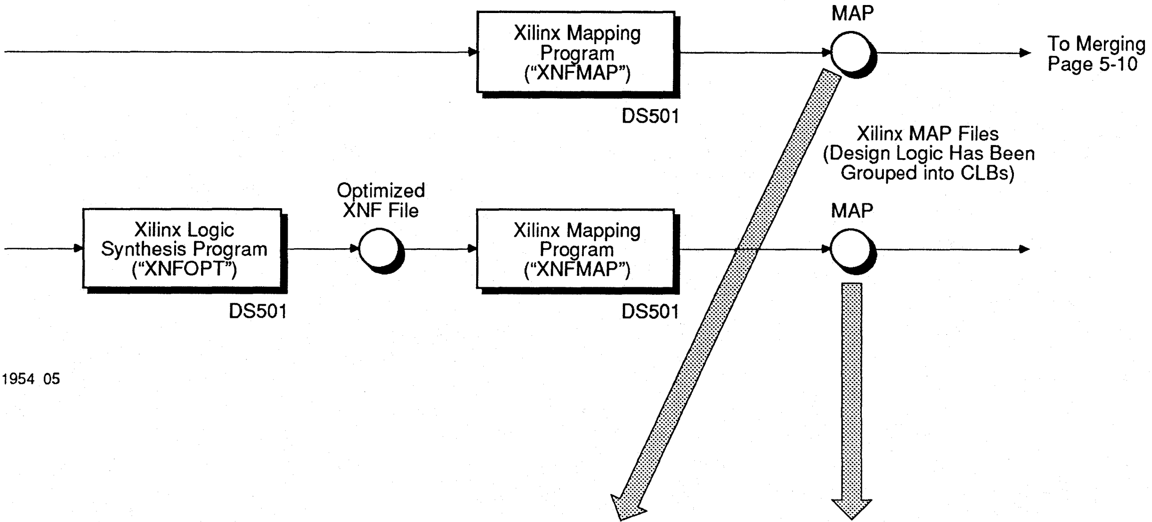
1954 04

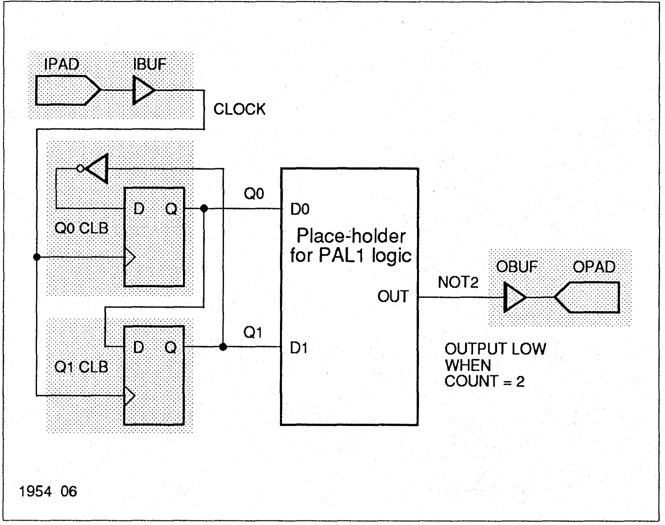
```
TITLE      DECODE.PDS
AUTHOR
COMPANY    XILINX
DATE
CHIP       DECODE   PAL10H8
;Input Pins
               d0 d1
;Output Pins
               out
;
;Define counter states
;
STRING ZERO       '/d1 * /d0'
STRING ONE        '/d1 *  d0'
STRING TWO        ' d1 * /d0'
STRING THREE      ' d1 *  d0'

EQUATIONS
out = ZERO + ONE + THREE
```

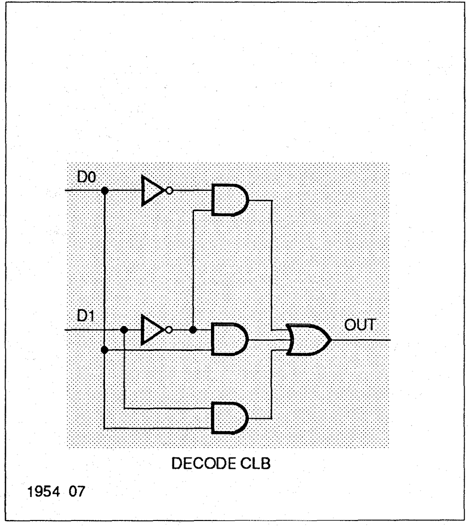
Schematics can include any number of "PAL" devices created with Boolean equations and/or state machines

Very simple "PAL" design included in schematic at left with PAL1 symbol

Optimization and Mapping


Xilinx Mapping Program ("XNFMAP")
DS501
MAP
To Merging Page 5-10
Xilinx MAP Files (Design Logic Has Been Grouped into CLBs)
Xilinx Logic Synthesis Program ("XNFOPT")
DS501
Optimized XNF File
Xilinx Mapping Program ("XNFMAP")
DS501
MAP
1954 05

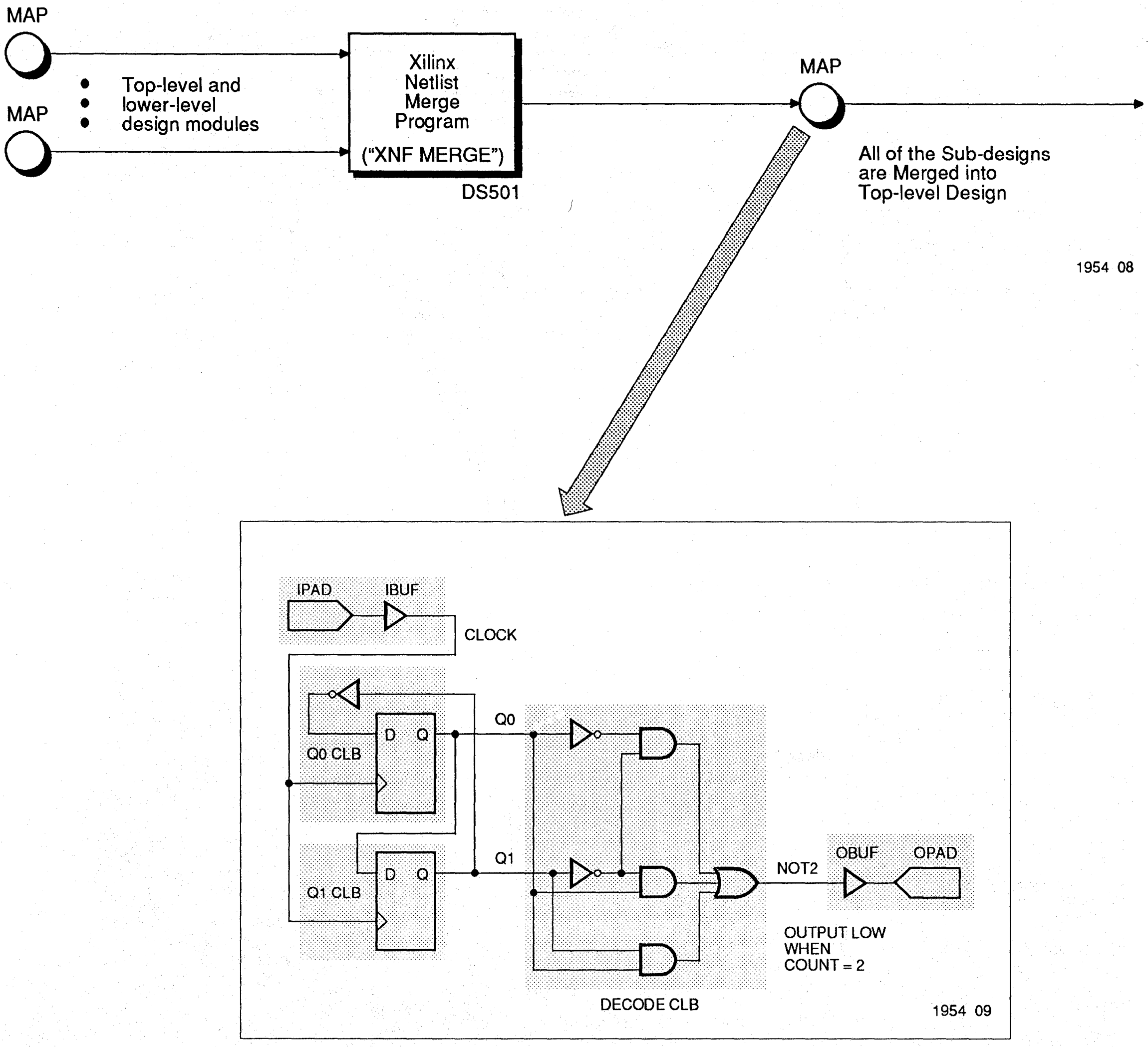




A graphical representation of the top-level MAP file. Unused logic (if any) has been deleted and the remaining logic has been grouped ("mapped") into Configurable Logic Blocks and I/O Blocks.

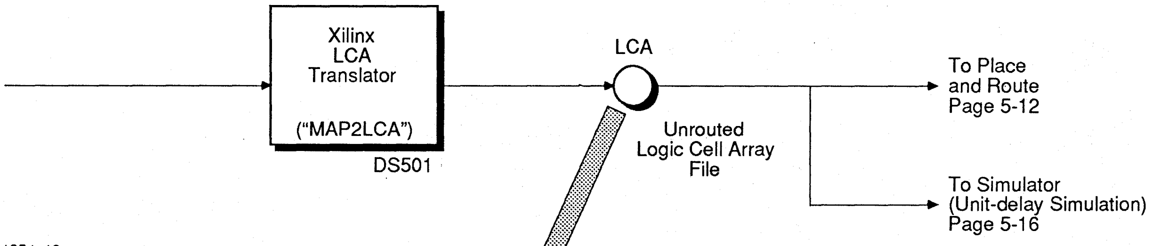
A Graphical representation of the CLB containing the "PAL" logic. (In a typical PAL design, of course, several CLBs would be used.)

Merging

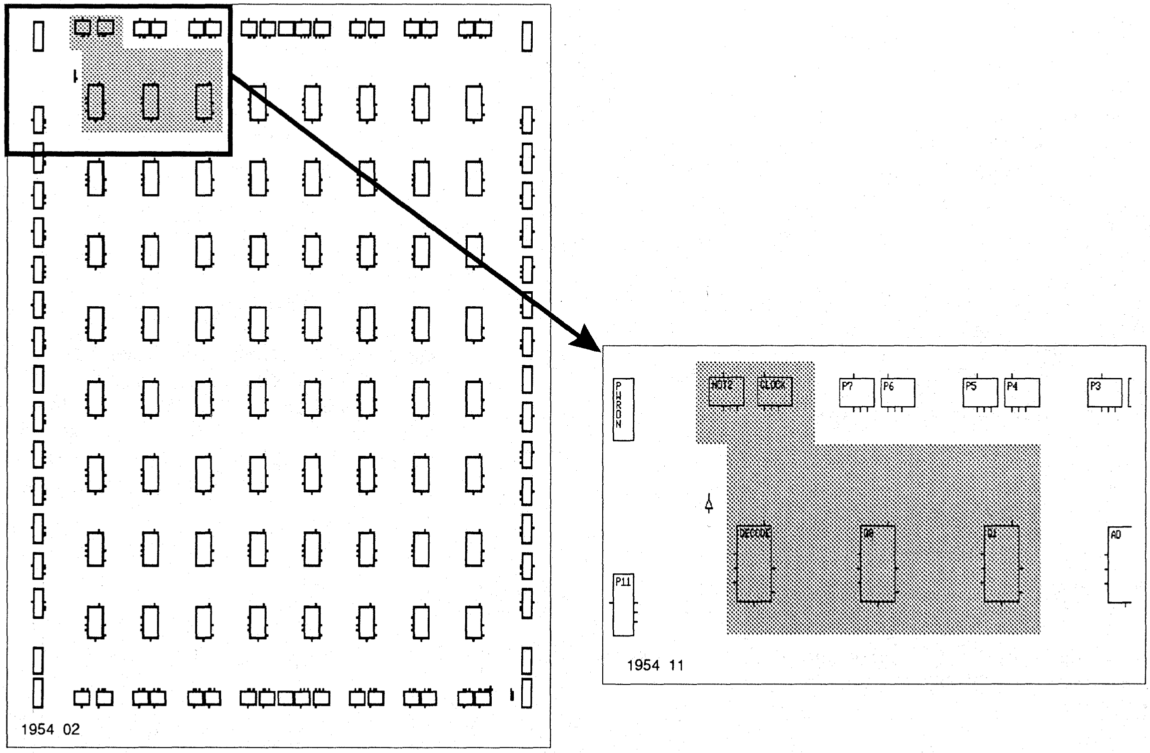


The merged design contains the CLBs and IOBs for the entire design.

Translating to a LCA File



1954 10

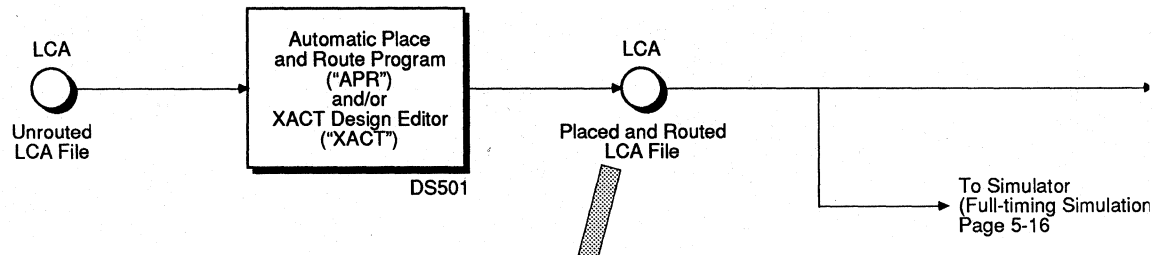


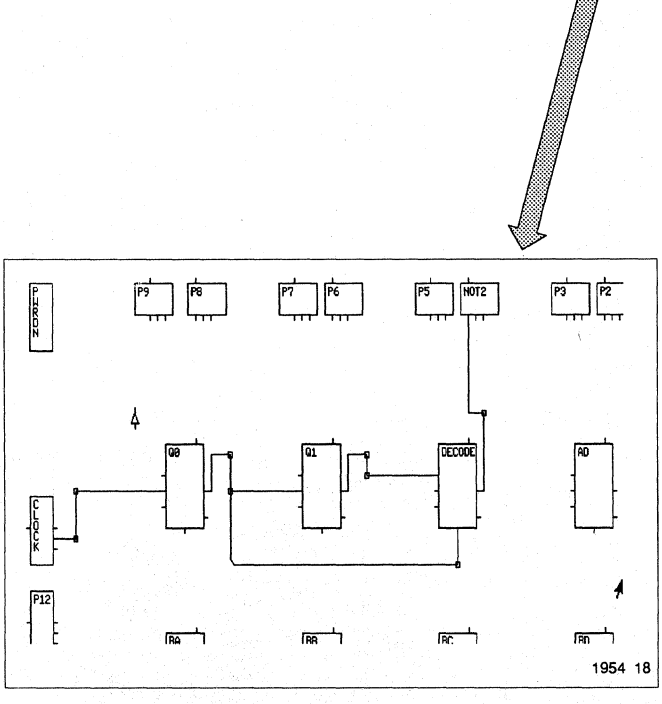
1954 02

1954 11

Initially (before Place and Route) the LCA design is unrouted, and the Configurable Logic and I/O blocks are put in random locations.

Placing and Routing the LCA file

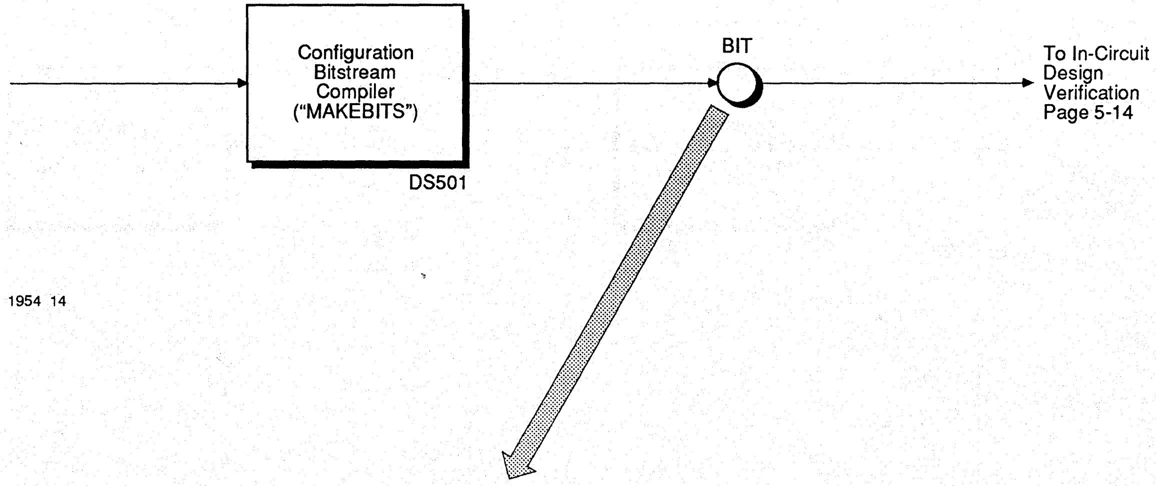




A simple placed and routed design (closeup of upper-left corner of 2064 PC68).

For complex designs, interactive placement and routing of critical logic with the XACT Design Editor is followed by APR to automatically place/route the remaining design. If necessary, additional interactive XACT edit on post-APR design is possible.

Bitstream Generation



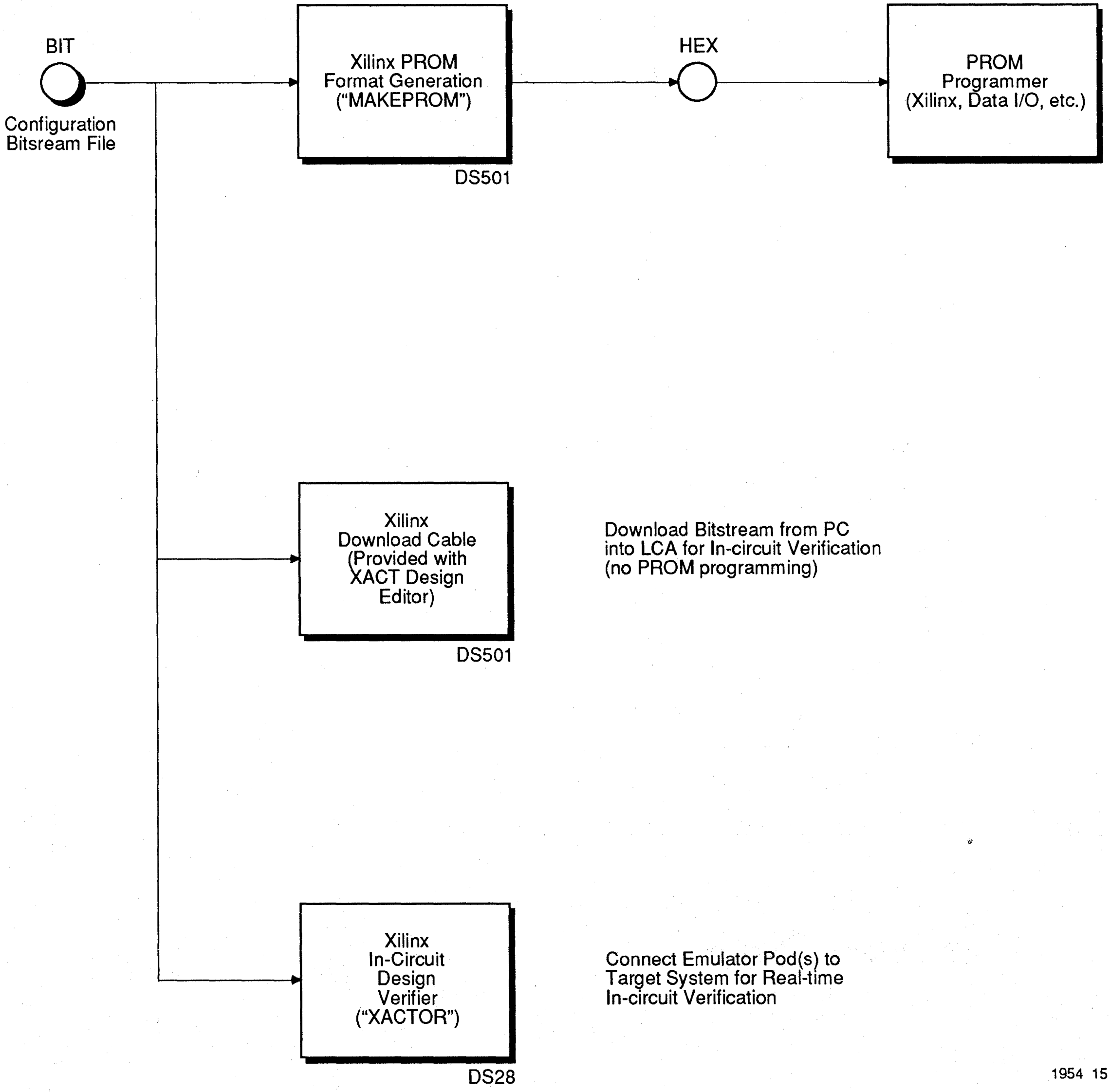
1954 14

```
1111111100100000000000111001111001001111
0011111111011111111111000101111011111101
0111111010110111011101110111111110111111
0111011101101111011111110111111101111111
0011111111111111111111111111101111111111
0111001101110111011101110111111110111111
1111111100100000000000111001110001001111
0011100011110111110111111111111111111111
0011111110111111111101111110111111111111
0011111111111101111111111111101111101111
0111111011101110111111101111111011111110
0111111010110111011...
```

LCA Configuration Bitstream

The BIT file contains the binary configuration data which programs an LCA to perform the design function.

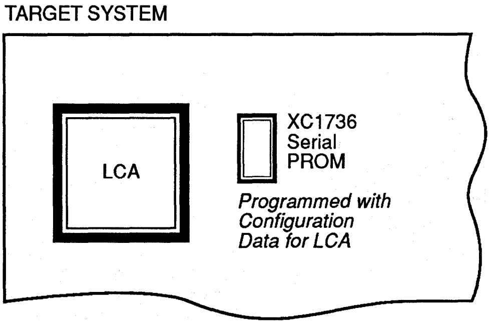
Real-time, In-circuit Verification


BIT
Configuration Bitsream File
Xilinx PROM Format Generation ("MAKEPROM")
DS501
HEX
PROM Programmer (Xilinx, Data I/O, etc.)
Xilinx Download Cable (Provided with XACT Design Editor)
DS501
Download Bitstream from PC into LCA for In-circuit Verification (no PROM programming)
Xilinx In-Circuit Design Verifier ("XACTOR")
DS28
Connect Emulator Pod(s) to Target System for Real-time In-circuit Verification

1954 15

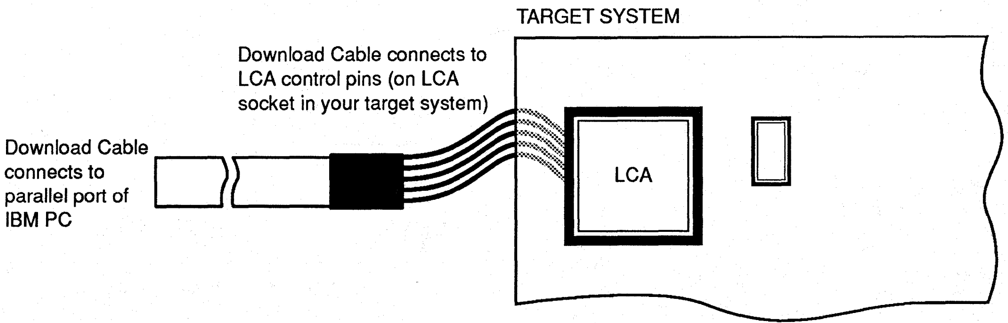
In-circuit verification lets you immediately see how your LCA designs function...

Program a PROM...



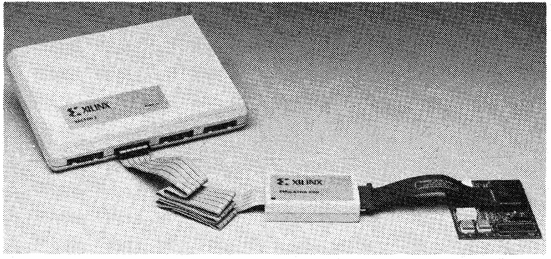
1954 16

Use the Download Cable...



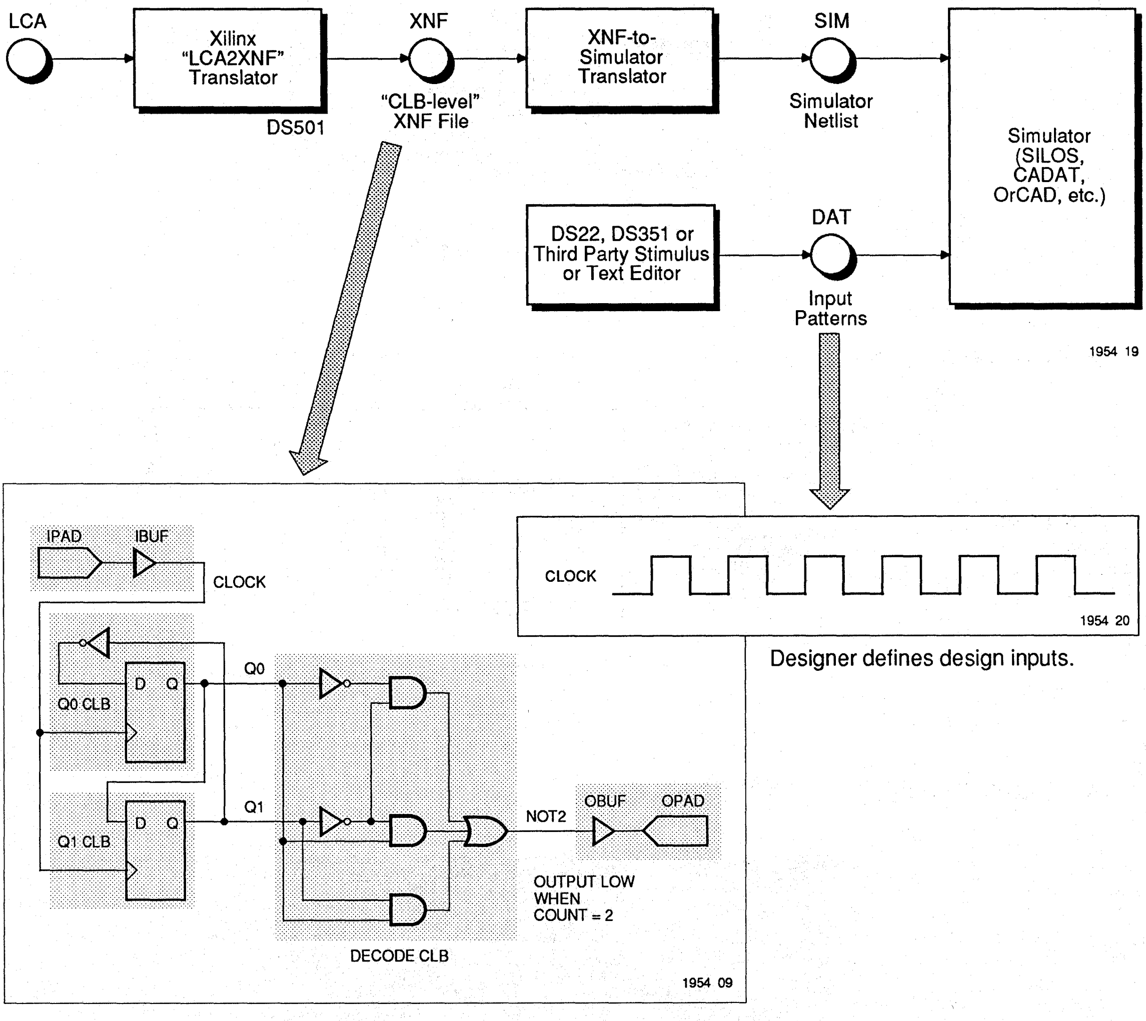
1954 17

...Or use the XACTOR Design Verifier



The DS28 controller (left) can control up to four DS26 emulation pods (center). A DS27 emulation header (right) connects each pod to an LCA socket in the target system.

Simulation

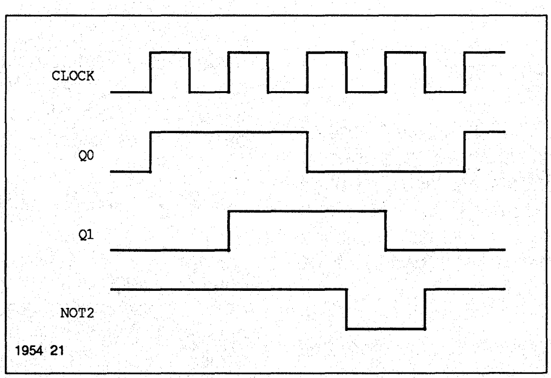

LCA
Xilinx "LCA2XNF" Translator
DS501
XNF
"CLB-level" XNF File
XNF-to-Simulator Translator
SIM
Simulator Netlist
Simulator (SILOS, CADAT, OrCAD, etc.)
DS22, DS351 or Third Party Stimulus or Text Editor
DAT
Input Patterns
1954 19
IPAD
IBUF
CLOCK
Q0
D
Q
Q0 CLB
Q1
D
Q
Q1 CLB
NOT2
OBUF
OPAD
OUTPUT LOW WHEN COUNT = 2
DECODE CLB
1954 09
CLOCK
1954 20

Designer defines design inputs.

LCA designs are simulated at the physical CLB level with worst-case timing. (Nets "inside" CLBs are not generally accessed.)

Simulation

Simulation ***allows design analysis*** under worst-case temperature, voltage, and process conditions.



Each I/O pin and CLB output can be observed with the simulator. Graphic or text display of signals is generated in response to input stimulus for verification of AC performance and logic behavior of design.

System Architecture for Personal Computers & Engineering Workstations

DESIGN FLOW

The Xilinx Development System has three major components

- DESIGN ENTRY
- DESIGN IMPLEMENTATION
- HARDWARE CIRCUIT VERIFICATION

The Xilinx Logic Libraries and XNF Interface Products support design entry with popular schematic logic drawing systems supplied by multiple vendors, allowing easy entry to the Xilinx Development System. Logic entry from Boolean equations or a variety of state machine language systems, through PALASM format, 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 array design.

While completely automated implementation is desirable for low complexity designs, the designer often prefers an interactive process, especially in high performance high complexity 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 new release of Automated Design Implementation software gives the designer an option to have direct control over specific logic mapped into CLBs (partitioning) for better distribution of logic signal routing through the LCA array. The XACT™ Design Editor is extremely versatile, ranging from design entry to CLB and signal routing manipulations. This combination of automated and interactive design editing capability is a unique feature provided by Xilinx.

Logic simulation or actual in-circuit emulation allows functional verification, while timing analysis allows verification of critical timing paths under worst case conditions. The system contains a compiler to generate bit-stream patterns which uniquely configures the LCA array according to the designer's specification.

An important feature of the Xilinx Development System is the capability to incorporate design changes which are frequently encountered during verification. Small changes can be made to the schematics and then automatically processed into the existing design with minimal impact using a unique "incremental design" capability. Thus the designer has the complete capability to develop "production quality" Programmable Gate Arrays resident on a PC or Engineering Workstation.

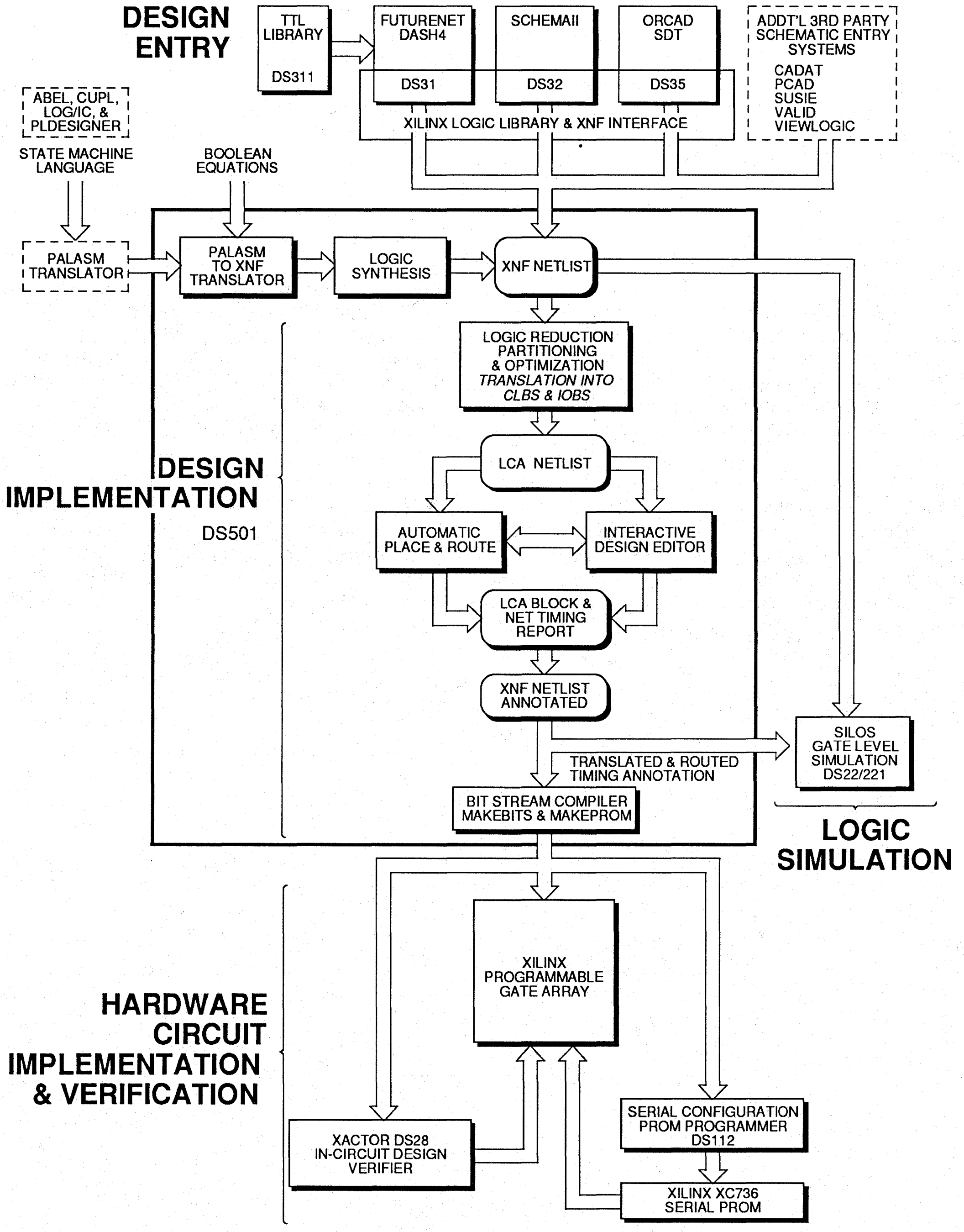
PERSONAL COMPUTER SUPPORT

Design Entry is supported on a variety of schematic capture design systems resident on IBM or compatible personal computers ranging from PC AT to PS/2 as defined in the hardware system requirements section. These design systems support the Xilinx Logic Library and interface to the Xilinx Design Implementation System through several netlist transfer programs.

The logic library and interface programs are supported by Xilinx and a range of third party vendors. Xilinx has provided a standard interface file specification (XNF format) to simplify file transfers into the Xilinx Development System. Xilinx directly supports the FutureNet DASH or DASH-LCA (DS31), Schema II (DS32), and OrCAD SDT (DS35) schematic capture products. An additional collection of TTL logic macrofunctions (DS311) is available with the FutureNet DASH & DASH-LCA systems.

All design implementation (including logic synthesis, partitioning, optimization, APR, interactive XACT design edits, timing analysis, and bit-stream compilation) is executed on the PC system. The new Design Manager enhancement to the Design Implementation 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 Design Manager environment, while the sequence of program commands is generated and stored for documentation prior to execution.

1956


DESIGN ENTRY
TTL LIBRARY DS311
FUTURENET DASH4 DS31
SCHEMAII DS32
ORCAD SDT DS35
ADDT'L 3RD PARTY SCHEMATIC ENTRY SYSTEMS
CADAT
PCAD
SUSIE
VALID
VIEWLOGIC
XILINX LOGIC LIBRARY & XNF INTERFACE
ABEL, CUPL, LOG/IC, & PLDESIGNER
STATE MACHINE LANGUAGE
BOOLEAN EQUATIONS
PALASM TRANSLATOR
PALASM TO XNF TRANSLATOR
LOGIC SYNTHESIS
XNF NETLIST
LOGIC REDUCTION PARTITIONING & OPTIMIZATION TRANSLATION INTO CLBS & IOBS
LCA NETLIST
DESIGN IMPLEMENTATION
DS501
AUTOMATIC PLACE & ROUTE
INTERACTIVE DESIGN EDITOR
LCA BLOCK & NET TIMING REPORT
XNF NETLIST ANNOTATED
TRANSLATED & ROUTED TIMING ANNOTATION
SILOS GATE LEVEL SIMULATION DS22/221
BIT STREAM COMPILER MAKEBITS & MAKEPROM
LOGIC SIMULATION
XILINX PROGRAMMABLE GATE ARRAY
HARDWARE CIRCUIT IMPLEMENTATION & VERIFICATION
XACTOR DS28 IN-CIRCUIT DESIGN VERIFIER
SERIAL CONFIGURATION PROM PROGRAMMER DS112
XILINX XC736 SERIAL PROM

1956 01

WORKSTATION SUPPORT

A variety of design systems has been developed for engineering workstations. Xilinx has developed Logic Library and Interface Products which allow designs entered schematically or in state machine language to be implemented in Xilinx LCAs. The design flow sequences from a variety of popular design systems on three workstations (Apollo, SUN, & DAISY Logician^{TM}) are illustrated in the accompanying figure.

APOLLO SYSTEM

The Mentor IdeaStation^{TM} for schematic entry and simulation is supported on the Apollo DN3000, DN3500, DN4000, and DN4500 computers. Xilinx offers a Logic Library and Interface product (DS343) for XNF netlist file transfer into the Xilinx Design Implementation System. This interface also generates post placement and route timing information associated with logic mapped into CLBs and the signal routing delays between CLBs.

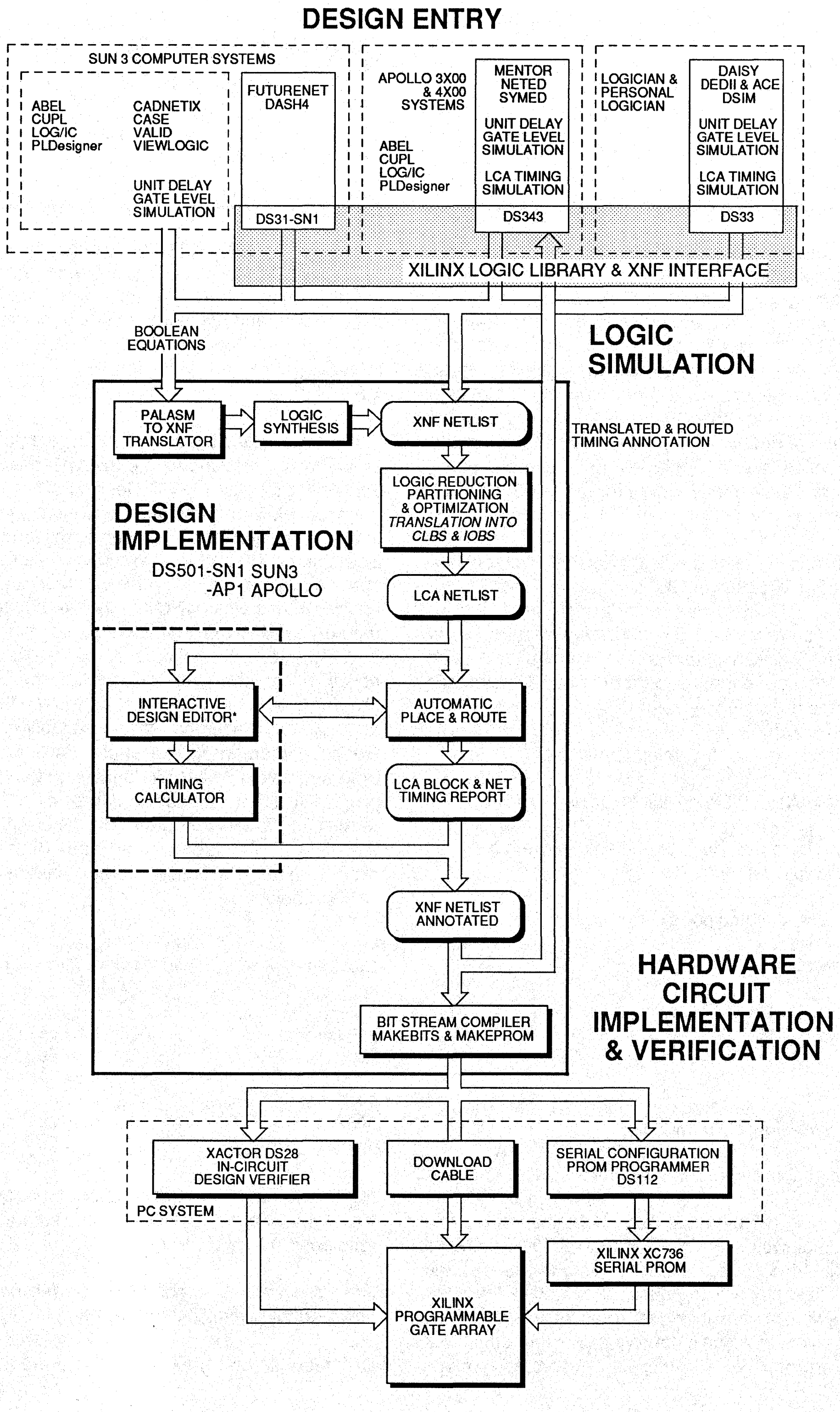
The Xilinx Development System for the Apollo computer (DS501-AP1) offers features identical to the PC version. The Enhanced Automated Design Implementation capability (ADI and Bit-stream Compilation) is completely supported on the Apollo computer. For designers requiring the interactive XACT edit capability, the product currently contains PC resident software. While the interactive XACT editor on the Apollo is being developed, the PC version is included in the DS501-AP1.

The bit-stream compiler (MAKEBITS and MAKEPROM) is included on the Apollo for completely automated design flow from Mentor schematic drawings to bit patterns for LCA configuration. Bit-stream configuration files can be transferred from the Apollo system to a Xilinx PROM programmer (DS112) or XACTOR In-Circuit Verifier (DS28) or a PC system download cable connected to the target system.

SUN3 SYSTEM

The SUN3 workstation supports logic entry on FutureNet DASH, logic entry/unit delay simulation on several systems (including Valid, Cadnetix, ViewLogic, & Case), or state machine language & Boolean equations (including ABEL, CUPL, LOG/IC, & PLDesigner). These third party vendors provide interface products for XNF netlist file transfer into the Xilinx Design Implementation System.

The Xilinx development system for the Sun3 (DS501-SN1) offers features identical to the PC version. The automated design sequence is also completely supported on the SUN3 computer from design capture to generation of bit-stream patterns for LCA configuraton. While the interactive XACT editor on the SUN3 is being developed, the PC version is included in the DS501-SN1.


DESIGN ENTRY
SUN 3 COMPUTER SYSTEMS
ABEL
CUPL
LOG/IC
PLDesigner
CADNETIX
CASE
VALID
VIEWLOGIC
UNIT DELAY
GATE LEVEL
SIMULATION
FUTURENET
DASH4
DS31-SN1
APOLLO 3X00
& 4X00
SYSTEMS
ABEL
CUPL
LOG/IC
PLDesigner
MENTOR
NETED
SYMED
UNIT DELAY
GATE LEVEL
SIMULATION
LCA TIMING
SIMULATION
DS343
LOGICIAN &
PERSONAL
LOGICIAN
DAISY
DEDII & ACE
DSIM
UNIT DELAY
GATE LEVEL
SIMULATION
LCA TIMING
SIMULATION
DS33
XILINX LOGIC LIBRARY & XNF INTERFACE
BOOLEAN
EQUATIONS
LOGIC
SIMULATION
PALASM
TO XNF
TRANSLATOR
LOGIC
SYNTHESIS
XNF NETLIST
TRANSLATED & ROUTED
TIMING ANNOTATION
LOGIC REDUCTION
PARTITIONING
& OPTIMIZATION
TRANSLATION INTO
CLBS & IOBS
DESIGN
IMPLEMENTATION
DS501-SN1 SUN3
-AP1 APOLLO
LCA NETLIST
INTERACTIVE
DESIGN EDITOR*
AUTOMATIC
PLACE & ROUTE
TIMING
CALCULATOR
LCA BLOCK & NET
TIMING REPORT
XNF NETLIST
ANNOTATED
HARDWARE
CIRCUIT
IMPLEMENTATION
& VERIFICATION
BIT STREAM COMPILER
MAKEBITS & MAKEPROM
XACTOR DS28
IN-CIRCUIT
DESIGN VERIFIER
DOWNLOAD
CABLE
SERIAL CONFIGURATION
PROM PROGRAMMER
DS112
PC SYSTEM
XILINX XC736
SERIAL PROM
XILINX
PROGRAMMABLE
GATE ARRAY

1956 02

Development System PC Hardware Requirements

Xilinx provides an integrated Development System for design and implementation of Logic Cell Arrays. The LCA development system operates on an IBM® PC/AT™ or IBM PS/2 model 60 or 80 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. Several popular workstation suppliers have developed compatible software which allows logic diagram design entry, simulation and LCA partitioning programs to be run on their systems.

The system configuration needed to run the Xilinx software consists of a 80286 or 80386 based IBM PC or "compatible". The software runs under the MS-DOS operating system version 3.0 and later, requires a 20 Megabyte hard disk, a mouse and a color monitor. A math co-processor can enhance performance of Automatic Placement and Routing by 10 or 20%.

The recommended configuration consists of:

- A "386" PC/AT or PS/2 model 60 or higher
- 40 M byte hard disk drive plus a 1.2 M Byte high density floppy disk drive
- Two RS-232-C serial ports
- One parallel port
- Enhanced Graphics Adapter and Display (EGA and EGD)
- Mouse
- MS-DOS version 3.0 or higher
- IBM compatible BIOS and keyboard

Highest system performance can be obtained by using a 80386 based system or a PS/2 model 70. To assure integrity, all Xilinx software is tested on IBM systems and several of the more "compatible" systems. Xilinx 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 BIOS or Keyboard Controller. Xilinx software includes system exercises called PMTEST and PMINFO to help test IBM compatibility and measure relative performance.

MEMORY

In protected mode the processor uses extended address space, which is found above the 1 Meg address. Earlier versions of the Xilinx XACT Design Editor used expanded memory (EMS or LIM standard) which is not suitable for the higher performance required by the larger array chips. Some of the Xilinx programs run in conventional memory space while the larger programs have a loader program of about 100 K bytes, which resides in conventional base memory and installs the Xilinx program in extended address space. See Figure 1. If the extended memory space is limited and becomes filled, the balance of the program will be "backfilled" above the loader in conventional memory. The availability of sufficient extended memory for an entire Xilinx protected mode program makes it possible to invoke nonprotected mode programs such as text editors or utilities without terminating a Xilinx program. For different LCA designs, the XACT Design Editor has different minimal requirements of available memory, as displayed by the BIOS at power-up.

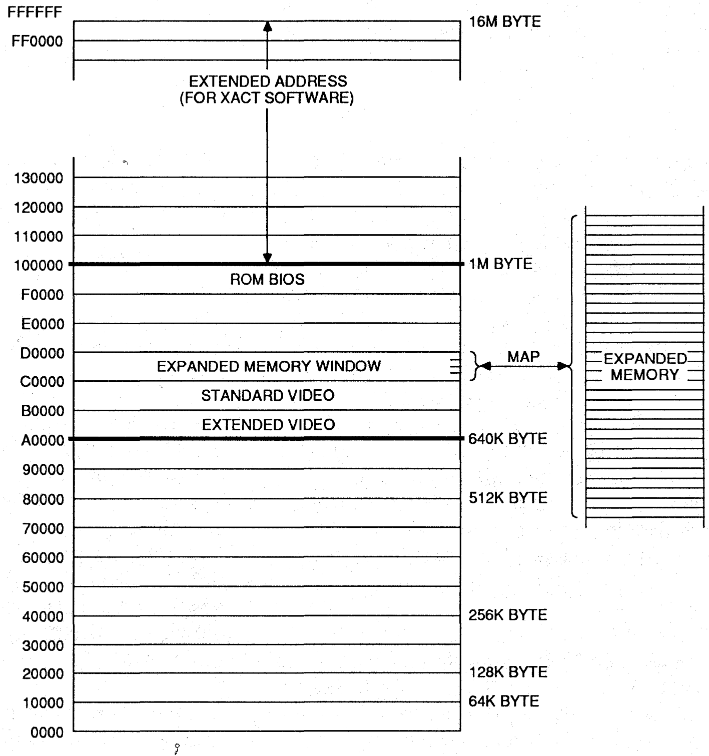
LCA Gates	Total Memory Required for XACT 2.1
2000 or less	2.50 MBytes
3000	3.25 MBytes
4200	4.00 MBytes
6400	5.25 MBytes
9000	6.50 MBytes

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 640K bytes of its first Megabyte available to any user.

Disk requirements for Xilinx software are a function of the program options which the user chooses, the number and type of active LCA designs and the number of other user software packages. A high density 1.2 M byte 5.25" floppy drive is needed for installation of the Xilinx software on the hard disk. The Xilinx software is also available on PS/2 3.5" diskettes. Some of the larger Xilinx programs are:

PROGRAM	APROX. SIZE
XACT	3.5 M byte
PIN2XNF	1.5 M byte
APR	1.0 M byte



1164 01

Figure 1. IBM Memory Map

The Compaq 386 reserves the 384K bytes of memory between 640K and 1M for internal use, making them unavailable to XACT.

I/O Ports

The Xilinx 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 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.

Display

The original IBM PC provided for two display options. Monochrome Display for text, and four color, 320x200 pixel, 8x8 dot character, Color Graphics Display for text and graphics. The interface to the displays was done by the Monochrome Display Adapter (MDA) and Color Display Adapter (CGA). IBM has added the higher resolution 16 color, 640x350 pixel, 8x14 dot character, Enhanced Graphic Display (EGD) and the corresponding Enhanced Graphic Adaptor (EGA). The Xilinx LCA Development System operates with a CGA or EGA display, but, future software enhancements might not support CGA and the higher resolution EGA is therefore recommended. The use of the EGA display is necessary for FutureNet schematic capture. The PS/2 compatible VGA is preferred for the higher density 3000 family LCAs. The Hercules monochrome display is not compatible.

Mouse

The Xilinx Development System programs are compatible with several varieties of mice. These include Mouse Systems PC Mouse (no device driver required), Microsoft (serial or parallel), LogiTech C7 and the FutureNet mouse. The Xilinx software will directly support any mouse which emulates the PC Mouse or has a device driver which provides Microsoft compatibility and defines it's COM port.

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. Examples 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 as:

```
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.

LCA Macro Library Listings

		XC3000				XC200		
		#CLBs	XACT PRIMITIVES	SCHEMATIC LOGIC MACROS	TTL MACROS DS40	#CLBs	XACT PRIMITIVES	SCHEMATIC LOGIC MACROS
PADS								
PIN	Input Pad	0	✓	✓	-	0	✓	✓
PINQ	Input Pad with Registered Input	0	✓	✓	-	0	✓	✓
PINR	Input Pad with Registered and Direct Inputs	0	✓	-	-	-	-	-
PIO	Input/Output Pad	0	✓	✓	-	0	✓	✓
PIOQ	Input/Output Pad with Registered Input	0	✓	✓	-	0	✓	✓
POUT	Output Pad	0	✓	✓	-	0	✓	✓
POUTZ	Output Pad with 3-State Control	0	✓	✓	-	0	✓	✓
POUTQ	Output Pad with Registered Output	0	✓	-	-	-	-	-
PREG	Pad with Data Register	0	✓	✓	-	0	✓	✓
PREG2	Pad with 2-Bit Shift Register	0	✓	-	-	-	-	-
IOB SCHEMATIC ELEMENTS								
TBUF	Internal Three-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	-	✓	-	-	-	-
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	-	✓
BUF	Internal non-inverting Buffer	0	-	✓	-	0	-	✓
INV	Inverter	1	-	✓	-	1	-	✓
PULLUP	Input pull-up Resistor	0	-	✓	-	-	-	-
OINV	Inverting Output Buffer	0	-	✓	-	-	-	✓
IOFF	I/O Pad as Flip-flop	-	-	-	-	0	-	✓

		XC3000				XC200		
		#CLBs	XACT PRIMITIVES	SCHEMATIC LOGIC MACROS	TTL MACROS DS40	#CLBs	XACT PRIMITIVES	SCHEMATIC LOGIC MACROS
GENERAL								
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	✓	✓
GXOR2	Dual Exclusive-OR	-	-	-	-	1	✓	-
GPAR	Parity Test (Even = Low)	-	-	-	-	1	✓	✓
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 Adder	6	-	-	✓	-	-	-
HX518	8-Bit Identity Comparator	5	-	-	✓	-	-	-
HX521	8-Bit Identity Comparator	5	-	-	✓	-	-	-
HX125	Three-State Bus Buffer	0	-	-	✓	-	-	-
HX240	Octal Inverting Buffer, Three-State Outputs	4	-	-	✓	-	-	-
HX241	Octal Non-inverting Buffer, Three-State Outputs	1	-	-	✓	-	-	-
HX244	Octal Non-inverting Buffer, Three-State Outputs	0	-	-	✓	-	-	-
HX245	Octal Bidirectional Transceiver	1	-	-	✓	-	-	-
HX540	Octal Inverting, Three-State Outputs	0	-	-	✓	-	-	-
HX541	Octal Non-inverting, Three-State Outputs	0	-	-	✓	-	-	-
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 Three-State Outputs	4	-	-	✓	-	-	-

FLIP-FLOPS

		XC3000				XC200		
		#CLBs	XACT PRIMITIVES	SCHEMATIC LOGIC MACROS	TTL MACROS DS40	#CLBs	XACT PRIMITIVES	SCHEMATIC LOGIC MACROS
DFF	D Flip-Flop	1	-	-	-	1	-	-
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	-	-	-	-	1	✓	✓
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	✓	✓
FDMR	D Flip-Flop with 2-Input Data Mux with Reset	1	✓	✓	-	1	✓	✓
FDMS	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	-	-	-	1	✓	✓
FT2R	2-Input Toggle Flip-Flop with Reset	1	-	-	-	1	✓	✓
NDFF	Negative Edge Flip-Flop Primitive	1	-	✓	-	1	-	✓
PDFF	Positive Edge Flip-Flop Primitive	1	-	✓	-	1	-	✓

		XC3000				XC200		
		#CLBs	XACT PRIMITIVES	SCHEMATIC LOGIC MACROS	TTL MACROS DS40	#CLBs	XACT PRIMITIVES	SCHEMATIC LOGIC MACROS
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	✓	✓
M4-1E	4-to-1 Mux with Enable	2	✓	✓	-	3	✓	✓
74-352	4-to-1 Mux with Enable, Low Output	2	✓	✓	-	3	✓	✓
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	✓	✓
MZ8-1	8-to-1 Mux Using Three-State Buffers	0	✓	-	-	-	-	-
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

		XC3000				XC200		
		#CLBs	XACT PRIMITIVES	SCHEMATIC LOGIC MACROS	TTL MACROS DS40	#CLBs	XACT PRIMITIVES	SCHEMATIC LOGIC MACROS
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 Three-State Outputs	4	-	-	✓	-	-	-
HX377	Octal D Flip-flops with Clock Enable	4	-	-	✓	-	-	-
HX577	Octal D Flip-flops with Reset and Three-State Outputs	4	-	-	✓	-	-	-

Serial to Parallel

		XC3000				XC200		
		#CLBs	XACT PRIMITIVES	SCHEMATIC LOGIC MACROS	TTL MACROS DS40	#CLBs	XACT PRIMITIVES	SCHEMATIC LOGIC MACROS
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	-	-	✓	-	-	-
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 Three-State Register Output	9	-	-	✓	-	-	-

COUNTERS

		XC3000				XC200		
		#CLBs	XACT PRIMITIVES	SCHEMATIC LOGIC MACROS	TTL MACROS DS40	#CLBs	XACT PRIMITIVES	SCHEMATIC LOGIC MACROS
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	✓	✓
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	-	-	✓	-	-	-

COUNTERS (Continued)

		XC3000				XC200		
		#CLBs	XACT PRIMITIVES	SCHEMATIC LOGIC MACROS	TTL MACROS DS40	#CLBs	XACT PRIMITIVES	SCHEMATIC LOGIC MACROS
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/ ClkEn, 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, ParEnaL, 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 Three-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	✓	✓

XILINX

XC-DS53 XACT Design Implementation System with FutureNet DASH-LCA

Product Brief

FEATURES

- **Complete bundled development system for programmable gate array design entry and implementation using schematics and/or Boolean equations**
- FutureNet DASH™ LCA schematic editor provides easy-to-use hierarchical LCA design capability
- PALASM™ format text input permits combining PLD equations and schematics in the same design
- Extensive macro library of over 100 logic elements (PAL™ devices, TTL counters, registers, etc.)
- Logic optimization software uses logic synthesis algorithms to optimize programmable gate array designs for density or performance
- Automatic logic reduction and partitioning software removes unused, disabled logic
- Automatic placement and routing
- XACT Design Editor provides interactive physical editing, timing calculator, design consistency checking, and configuration bitstream compiling
- Download cable transfers configuration bitstreams directly from PC to LCA during design debugging
- Runs on IBM® PC/AT™ or compatible computer

GENERAL

The XC-DS53 PGA Development System provides designers with a complete system for entering, implementing, compiling, and verifying designs for Xilinx programmable gate arrays.

The FutureNet DASH-LCA schematic editor combined with a Xilinx PALASM-to-LCA translator program provides powerful design entry and documentation capability. Designs can be entered as schematics, Boolean equations, or both.

DASH-LCA supports unlimited levels of hierarchy and an extensive library of standard logic macros for LCA design. In addition, designers can flag critical timing paths to ensure critical signals are routed with minimum delay. The optional XC-DS40 TTL library adds over 50 additional 7400 MSI TTL functions for LCA design using FutureNet DASH-LCA or DASH.

Xilinx PALASM-to-LCA software permits text files containing Boolean equations for *any type* of PLD to be translated into an LCA, or automatically linked to PAL symbols in a schematic.

The XC-DS53 also includes the Automated Design Implementation software, which permits designers to efficiently implement programmable gate array designs without detailed knowledge of LCAs. The logic optimization software uses logic synthesis algorithms to optimize for smallest area or highest performance. Automatic logic reduction and partitioning software eliminates unused logic, then partitions the design into Logic Cell Array physical resources (logic and I/O blocks). Finally, the Automatic Placement and Routing program is invoked to arrange logic blocks within the LCA to minimize signal delays on user-specified critical timing paths.

Designers can interactively place and route portions of the physical design, if necessary, then compile a configuration bitstream using the graphics-based XACT Design Editor (included in the XC-DS53). XACT's timing calculator permits point-to-point timing calculations for critical path analysis, or a simulator (such as the optional XC-DS22 SILOS Simulator) can be used to simulate design function and timing.

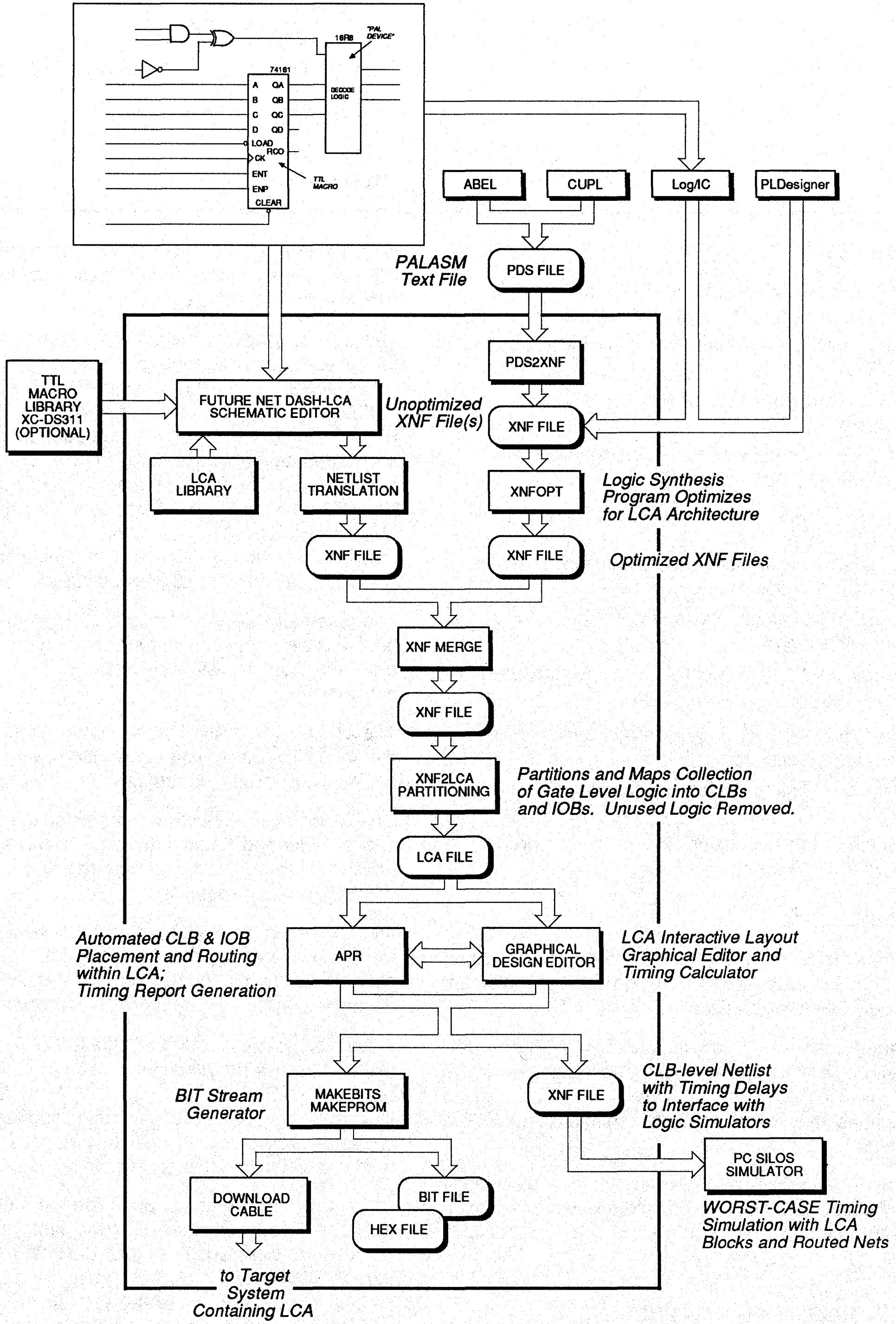
During in-circuit design analysis, designers can save time by using a Download Cable provided with XACT to transfer the configuration program from the PC directly into an LCA under development in a system.

Xilinx provides ongoing support for XACT users. Software updates (for new devices) are included for the first year. Continued support updates are available for purchase with the XC-SC53 Annual Support Agreement. In addition, Xilinx operates a technical support telephone hotline and an electronic bulletin board to provide software enhancements and factory support.

XC-DS53 Includes:

- FutureNet DASH-LCA Schematic Editor and XC-DS31 LCA Library
- Complete XC-DS23 Automated Design Implementation software (PALASM input, logic optimizer, automated placement and routing)
- Complete XC-DS21 XACT Design Editor (Interactive placement and routing, configuration bitstream compiler, download cable and demo board)

1963


16R8
"PAL DEVICE"
DECODE LOGIC
74161
A
B
C
D
LOAD
CK
ENT
ENP
CLEAR
QA
QB
QC
QD
RCO
TTL MACRO
ABEL
CUPL
Log/IC
PLDesigner
PALASM Text File
PDS FILE
PDS2XNF
TTL MACRO LIBRARY XC-DS311 (OPTIONAL)
FUTURE NET DASH-LCA SCHEMATIC EDITOR
Unoptimized XNF File(s)
XNF FILE
LCA LIBRARY
NETLIST TRANSLATION
XNFOPT
Logic Synthesis Program Optimizes for LCA Architecture
XNF FILE
XNF FILE
Optimized XNF Files
XNF MERGE
XNF FILE
XNF2LCA PARTITIONING
Partitions and Maps Collection of Gate Level Logic into CLBs and IOBs. Unused Logic Removed.
LCA FILE
Automated CLB & IOB Placement and Routing within LCA; Timing Report Generation
APR
GRAPHICAL DESIGN EDITOR
LCA Interactive Layout Graphical Editor and Timing Calculator
BIT Stream Generator
MAKEBITS MAKEPROM
XNF FILE
CLB-level Netlist with Timing Delays to Interface with Logic Simulators
PC SILOS SIMULATOR
WORST-CASE Timing Simulation with LCA Blocks and Routed Nets
DOWNLOAD CABLE
BIT FILE
HEX FILE
to Target System Containing LCA

1963 01

Software Included in XC-DS53

XILINX

XC-DS501 XACT Design Implementation System

Product Brief

XC-DS501 FEATURES

- Complete system for implementing programmable gate array designs into Logic Cell™ Array
- Accepts any combination of schematic netlists and PALASM2™ text files
- Supports input from ABEL, CUPL, Log/IC, and PLDesigner
- PLD conversion software allows design entry with PALASM2™ format equations
- Logic synthesis software optimizes designs for Logic Cell Array architecture
- Automated logic reduction and partitioning removes unused, disabled logic
- Automated placement and routing of logic minimizes design cycle time
- Available on PC/AT™, Apollo, and Sun3 computers
- Interactive logic and routing editor
- Boolean equation or Karnaugh map alternatives to specify logic functions
- Point to point timing calculations for critical path analysis
- Download cable to transfer configuration programs from PC to LCA in target system

GENERAL

Designers require both Advanced Design Implementation (ADI) and the Interactive XACT Design Editor to implement their Logic Cell Array designs.

The Automated Design Implementation (ADI) software converts design descriptions for programmable gate arrays into Logic Cell Array (LCA) implementations. Design descriptions may combine both schematics and PLD equations.

The XACT Design Editor provides an interactive placement/routing editor to modify or complete design routing, a bitstream compiler, and a download cable for in-circuit verification.

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. ADI software allows designers to merge multiple modes of design entry into a single design.

The following figure illustrates the design flow from a design schematic with some glue logic, a TTL macro, and a "PLD" symbol. The "PLD" design is entered using ABEL, CUPL, Log/IC, PLDesigner, or directly as PALASM2™ Boolean Equations. The multiple mode design is combined into a programmable gate array in three steps. First, the schematic is converted by the netlist translator to a Xilinx Netlist Format (XNF) file. Second, the PALASM2 text file is translated (PDS2XNF) and then optimized (XNFOPT), resulting in a second XNF file. The design files are then merged, partitioned into CLBs and IOBs.

The automatic placement and routing program (APR) is very flexible. Routing resources can be specified to eliminate clock skews and minimize routing delays for critical paths.

The XACT™ Interactive Design Editor can then be used to modify design placement and routing, when required to meet critical timing requirements.

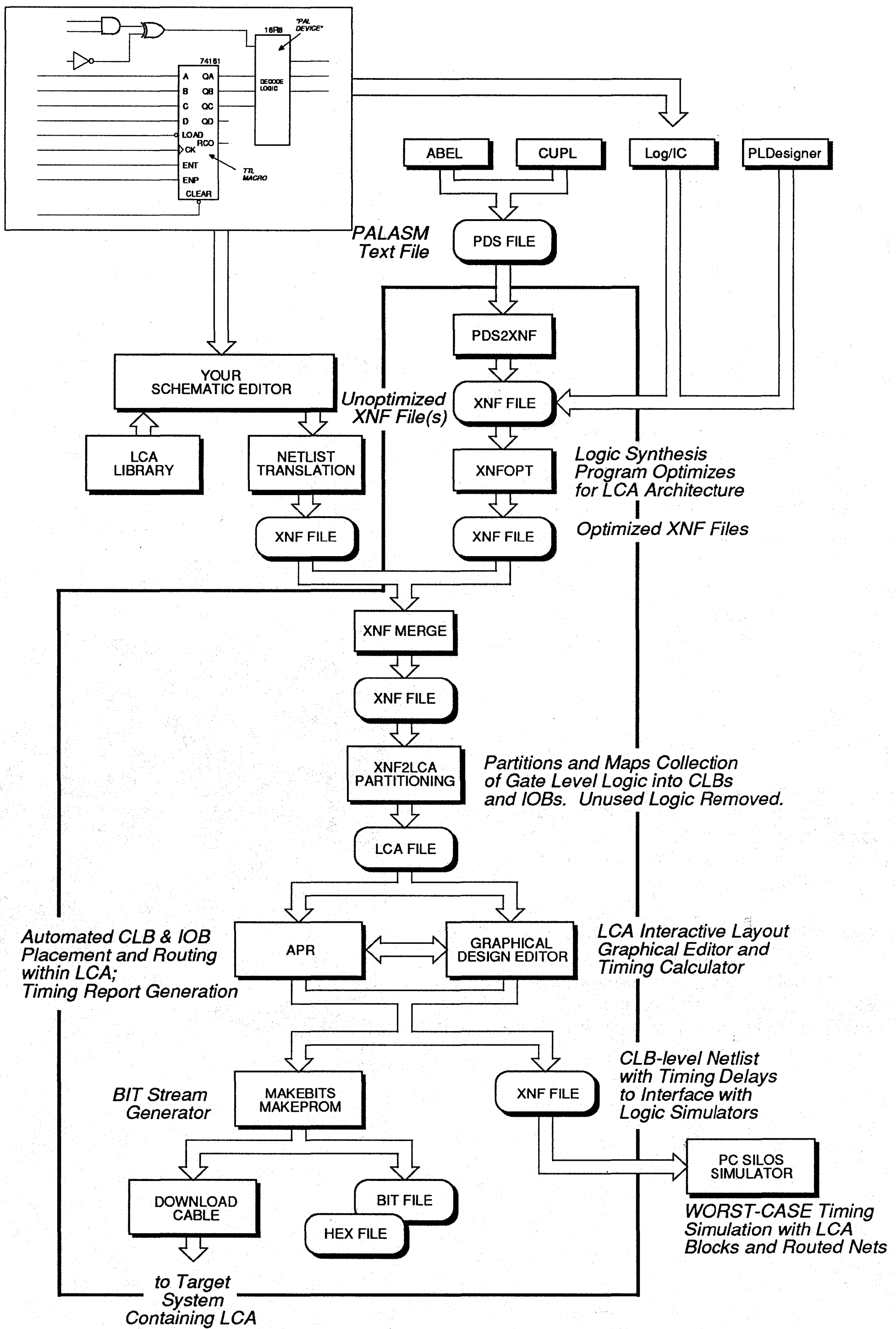
Checks for logic connectivity and design rule violation are easily performed using the XACT Design Editor. All unused internal nodes are automatically configured to minimize power dissipation.

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 included with the XACT Design Editor is useful for transferring configuration programs serially from the PC workstation to a Logic Cell Array installed in a system. During product development and verification this capability can be used to save the time required to write a modified configuration program into an EPROM.

Xilinx provides ongoing support for users of the Design Implementation software. For the first year, software updates are included. After that, the user may purchase the XC-SC501 Annual Support Agreement to continue to receive the latest software releases. Xilinx also maintains a Technical Hotline and a user electronic bulletin board to provide timely product support.

1952


16R8
"PAL DEVICE"
74161
A QA
B QB
C QC
D QD
LOAD RCO
CK
ENT
ENP
CLEAR
DECODE LOGIC
TTL MACRO
ABEL
CUPL
Log/IC
PLDesigner
PALASM Text File
PDS FILE
PDS2XNF
YOUR SCHEMATIC EDITOR
Unoptimized XNF File(s)
XNF FILE
LCA LIBRARY
NETLIST TRANSLATION
XNFOPT
Logic Synthesis Program Optimizes for LCA Architecture
XNF FILE
XNF FILE
Optimized XNF Files
XNF MERGE
XNF FILE
XNF2LCA PARTITIONING
Partitions and Maps Collection of Gate Level Logic into CLBs and IOBs. Unused Logic Removed.
LCA FILE
Automated CLB & IOB Placement and Routing within LCA; Timing Report Generation
APR
GRAPHICAL DESIGN EDITOR
LCA Interactive Layout Graphical Editor and Timing Calculator
BIT Stream Generator
MAKEBITS MAKEPROM
XNF FILE
CLB-level Netlist with Timing Delays to Interface with Logic Simulators
PC SILOS SIMULATOR
WORST-CASE Timing Simulation with LCA Blocks and Routed Nets
DOWNLOAD CABLE
BIT FILE
HEX FILE
to Target System Containing LCA

1952 01

Software Included in XC-DS501

1952 01

XILINX

XC-DS31 FutureNet DASH™ Schematic Entry Interface and Design Library

Product Brief

FEATURES

- Library and translator for users with the FutureNet 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
- User control for flagging critical paths for the Automated Placement and Routing keep
- Converts schematic drawings to a Xilinx Netlist Format (XNF) output file
- Output compatibility with XC-DS501 XACT Design Implementation System
- Runs on an IBM® PC/AT™ or compatible personal computer

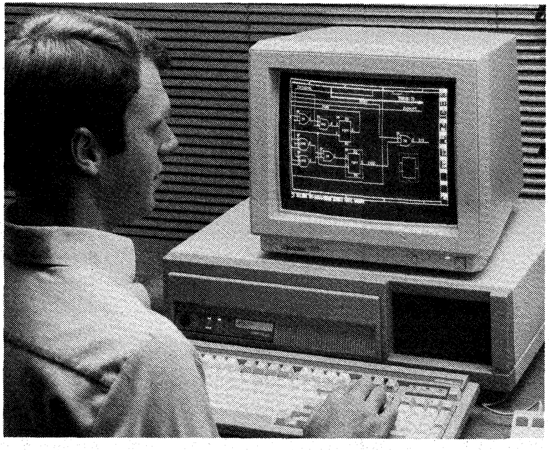
GENERAL

Schematic entry and automatic partitioning of Logic Cell Array designs shortens logic reduction and product development times. Complex designs can be specified schematically and quickly implemented for in-circuit design verification.

Xilinx's FutureNet DASH Schematic Designer Library provides the symbol library and conversion utility to permit designers to enter Logic Cell Array designs with the FutureNet 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 PC-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.

Xilinx provides ongoing support for users of the FutureNet 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

XILINX

XC-DS311 7400 TTL Library for FutureNet DASH and DASH-LCA

Product Brief

FEATURES

- Over 50 7400 TTL library elements for use with FutureNet DASH™ Schematic Designer
- Permits XC3000 family Logic Cell™ Array design using familiar 7400 MSI functions
- Supports XC-DS531 PGA Development Systems with FutureNet DASH-LCA

GENERAL

The XC-DS311 7400 TTL Library is a set of over 50 FutureNet DASH logic symbols for schematic entry of Logic Cell Array (LCA™) designs. The library can be used with the XC-DS531 Development System, or with the FutureNet DASH Schematic Designer and XC-DS31 Future Net DASH Schematic Interface and Library. Use of the XC-DS-311 Library allows the designer to enter complex programmable gate array designs in terms of familiar TTL functions.

The XC-DS311 TTL Library supplements the library of more than 100 logic elements that are included with XC-DS531.

Xilinx provides ongoing support for XACT users. For the first year, software updates (such as support for new devices) are included at no charge. After that, users may purchase the XC-SC311 to continue receiving software updates. In addition, Xilinx operates a technical support telephone hotline and an electronic bulletin board to provide software enhancements and factory support.

LIST OF TTL MACROS

7442	4-to-10 Line Decoder
7448	BCD-to-Seven-Segment Decoder
7477	4-Bit Latch
7483	4-Bit Binary Adder With Fast Carry
7485	4-Bit Magnitude Comparator
74125	Quad Three-State Bus Buffer
74138	1-of-8 Decoder/Demultiplexer
74139	Dual 1-of-4 Decoder
74147	10-to-4 Line Priority Encoder
74148	8-to-3 Line Priority Encoder
74151	8-Input Multiplexer
74152	8-Input Multiplexer
74153	Dual 4-Input Multiplexer
74154	1-of-16 Decoder/Demultiplexer
74157	Quad 2-Input Multiplexer
74158	Quad 2-Input Multiplexer
74160	Presettable Decade Counter
74161	Presettable Binary Counter
74162	Presettable Decade Counter w/ Sync Clear
74163	Synchronous Binary Counter w/ Sync Clear
74164	Serial-In, Parallel-Out Shift Register
74166	Parallel Load 8-Bit Shift Register
74168	4-Bit BCD Synchronous Up/Down Counter
74169	4-Bit Binary Synchronous Up/Down Counter
74174	Hex D Flip-Flop with Master Reset
74179	4-Bit Parallel Access Shift Register
74194	4-Bit Bidirectional Universal Shift Register
74195	4-Bit Parallel Access Shift Register
74198	8-Bit Bidirectional Shift Register
74199	8-Bit Shift Register with Clock Inhibit
74240	Octal Inv Buffer, Three-State Outputs
74241	Octal Non-Inverting Three-State Buffers
74244	Octal Non-Inv Buffer, Three-State Outputs
74245	Octal Bidirectional Transceiver
74257	Quad 2-Input Multiplexer
74258	Quad 2-Input Multiplexer
74259	8-Bit Addressable Latch
74273	Octal D Flip-Flop
74278	4-Bit Cascadable Priority Register
74280	9-Bit Parity Checker/Generator
74283	4-Bit Binary Full Adder
74298	Quad 2-Input Multiplexer with Storage
74352	Dual 4-to-1 Data Selector/Multiplexer
74373	Octal Latch with Three-State Outputs
74374	Octal D Flip-Flops with Three-State Outputs
74377	Octal D Flip-Flop with Clock Enable
74390	Dual 4-Bit Decade Counters with Clear
74393	Dual 4-Bit Binary Counters with Clear
74518	8-Bit Indentity Comparator
74521	8-Bit Indentity Comparator
74540	Octal Inv Buffer, Three-State Outputs
74541	Octal Non-Inv Buffer, Three-State Outputs
74577	Octal D Flip-Flops with Reset and Three-State Outputs
74590	8-Bit Binary Counter with Output Register
74595	8-Bit Shift Register with Output Register

1957

XC-DS32 Schema II Schematic Entry Interface and Design Library

Product Brief

FEATURES

- Library and translator for users with the Schema II Schematic Editor
- 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
- 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 an IBM® PC/AT™ or compatible personal computer

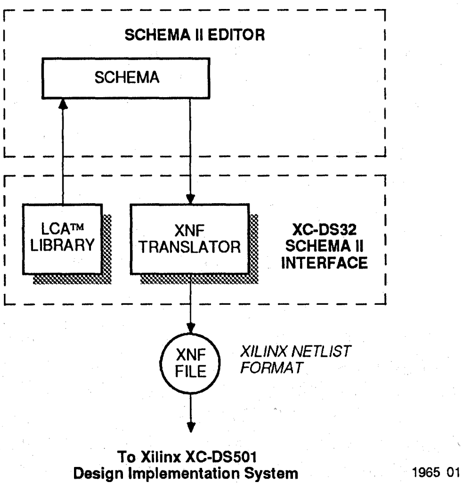
GENERAL

Schematic entry and automatic partitioning of Logic Cell Array designs shortens logic reduction and product development times. Complex designs can be specified schematically and quickly implemented for in-circuit design verification.

Xilinx's Schema II Schematic Entry Interface provides the symbol library and conversion utility to permit designers to enter Logic Cell Array designs with the Schema II 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 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 Schema II Schematic Entry Interface. For the first year, software updates are included. After that, the user may purchase the XC-SC32 Annual Support Agreement to continue to receive the latest software releases.



XC-DS33 Daisy Schematic Entry Interface, Library, Unit-Delay Simulation

Product Brief

FEATURES

- Schematic entry via the Daisy DED II and ACE Schematic editors
- Unit-delay simulation with Daisy simulator
- 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
- 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

GENERAL

Schematic entry and automatic partitioning of Logic Cell Array designs shortens logic reduction and product development times. Complex designs can be specified schematically and quickly implemented for in-circuit design verification.

Xilinx's Daisy schematic interface provides the symbol library and conversion utility to permit designers to enter Logic Cell Array designs with either DED II or ACE on Daisy workstations. The Xilinx library provides the logic, I/O, and macro symbols to be used in the schematic. A Xilinx utility converts the Daisy format to an XNF output file. Once partitioned, the design may be placed and routed with the PC-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.

Xilinx provides ongoing support for users of the Daisy Schematic Interface. For the first year, software updates are included. After that, the user may purchase the XC-SC33 Annual Support Agreement to continue to receive the latest software releases.

DAISY SYSTEM REQUIREMENTS

Any Daisy workstation listed on the chart on the next page can be used to produce a design netlist in the Xilinx Netlist Format using a utility provided with the library.

An IBM PC/AT equipped with the XC-DS501 Design Implementation System is then used to physically implement the design in an LCA.

PC/AT-based Daisy workstations (such as the Daisy ENTRY system) can have the XC-DS501 software installed in the PC's DOS partition. Designers using non-PC/AT workstations can transfer the schematic netlist to an MSDOS floppy and complete the LCA physical design on a PC/AT computer.

Personal LOGICIAN® and LOGICIAN® are registered trademarks of Daisy Systems Corporation.

1967

XILINX

XC-DS343 Mentor Interface Schematic Entry, Timing Simulation Interface, and Library

Product Brief

FEATURES

- **Mentor certified**
- Mentor IDEA® Station can be used for schematic entry and simulation of programmable gate array designs
- Simulation with post placement/routing LCA timing
- Macro library includes over 100 standard logic elements (counters, multiplexers, registers, etc.)
- Primitive library includes flip-flops, latches, AND, OR, XOR, NAND, NOR gates
- Xilinx Netlist Format (XNF) output is compatible with XC-DS501 Design Implementation System

GENERAL

The Xilinx XC-DS343 Mentor Interface allows designers to use their Mentor IDEA Station to enter and simulate programmable gate array schematics. The Logic Cell™ Array library includes over 100 macros for commonly used counters, registers, multiplexers, etc.

The resulting Mentor design files are then processed using Xilinx's XC-DS501 Design Implementation System, which permits rapid design development and real-time in-circuit design verification.

Xilinx provides ongoing support for users of the Mentor Interface. For the first year, software updates are included. After that, the user may purchase the XC-SC343 Annual Support Agreement to continue to receive the latest software releases. Xilinx also maintains a Technical Hotline and an electronic bulletin board to provide technical product support to LCA users.

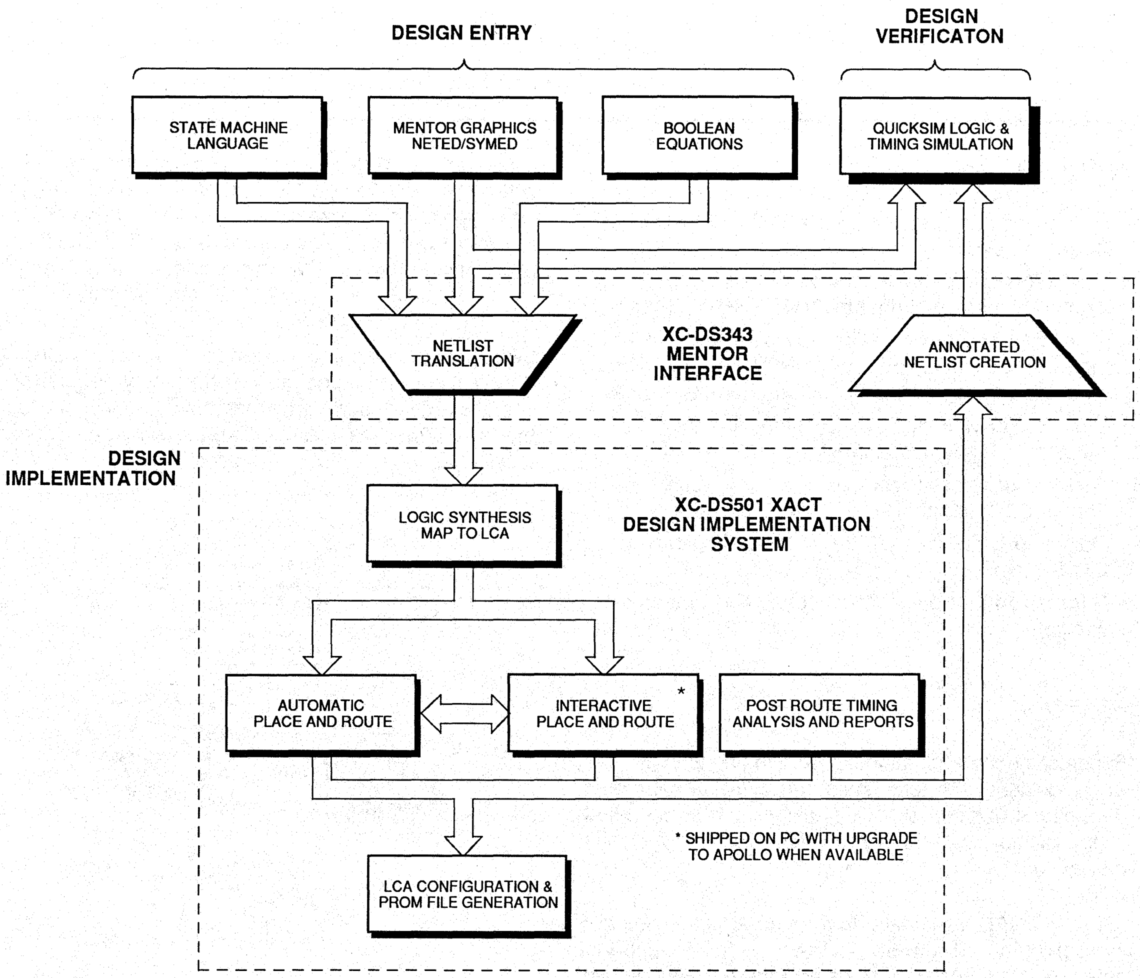
SYSTEM REQUIREMENTS

Mentor IDEA V6.0 or greater is required for schematic entry and simulation. The XC-DS343 Mentor Interface software requires the APOLLO SR9.6 or DOMAIN/IX operating system.

File transfer from the Mentor system to other PC-based Xilinx software may be accomplished using any existing PC/Apollo link, such as the DOMAIN/PCI software available from Mentor and Apollo.

The XC-DS501-AP1 Developement System is partially resident on Apollo (DN3000, DN3500, DN4000, and DN4500) and IBM PC System platforms. The Mentor interface, Automated Placement/Routing, timing analysis, and bitstream compilation are resident on the Apollo system, while the XACT Design Editor is currently resident on the IBM PC system.

1958


DESIGN ENTRY
DESIGN VERIFICATON
STATE MACHINE LANGUAGE
MENTOR GRAPHICS NETED/SYMED
BOOLEAN EQUATIONS
QUICKSIM LOGIC & TIMING SIMULATION
NETLIST TRANSLATION
XC-DS343 MENTOR INTERFACE
ANNOTATED NETLIST CREATION
DESIGN IMPLEMENTATION
LOGIC SYNTHESIS MAP TO LCA
XC-DS501 XACT DESIGN IMPLEMENTATION SYSTEM
AUTOMATIC PLACE AND ROUTE
INTERACTIVE PLACE AND ROUTE *
POST ROUTE TIMING ANALYSIS AND REPORTS
* SHIPPED ON PC WITH UPGRADE TO APOLLO WHEN AVAILABLE
LCA CONFIGURATION & PROM FILE GENERATION

1958 02

Xilinx LCA Design Flow on Mentor Graphics Workstations

XC-DS35 OrCAD SDT Schematic Entry Interface

Product Brief

FEATURES

- Library and translator for users with the OrCAD SDT Schematic Editor
- Macro library of over 100 standard logic family equivalents derived from the XACT™ Macro Library
- Primitive library of logic symbols includes all two-input, three-input and four-input AND, OR and XOR gates plus storage, input/output and clock elements
- 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 an IBM® PC/AT™ or compatible personal computer

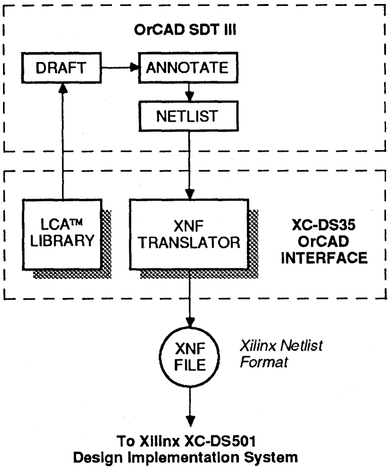
GENERAL

Schematic entry and automatic partitioning of Logic Cell Array designs shortens logic reduction and product development times. Complex designs can be specified schematically and quickly implemented for in-circuit design verification.

Xilinx's OrCAD Schematic Entry Interface provides the symbol library and conversion utility to permit designers to enter Logic Cell Array 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.

Xilinx provides ongoing support for users of the OrCAD Schematic Entry Interface. For the first year, software updates are included. After that, the user may purchase the XC-SC35 Annual Support Agreement to continue to receive the latest software releases.



1966 01

XC-DS351 OrCAD VST Simulator Interface

Product Brief

FEATURES

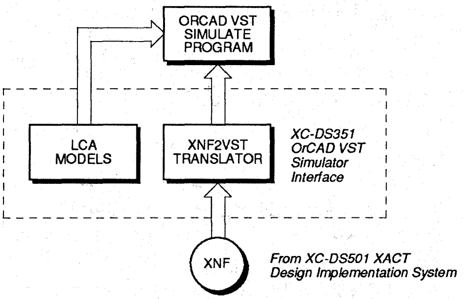
- Model library and netlist translator for users of the OrCAD VST Simulator
- Supports full timing simulation of routed LCA designs, and unit-delay simulation of unrouted designs
- Permits simulation of schematics which include PAL logic defined with PALASM, ABEL, CUPL, Log/IC, or PLDesigner.
- Input compatible with XACT Design Implementation System
- Includes unique automated Trace and Stimulus file generation
- Runs on an IBM 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 Logic Cell Array speed grade for a particular application.

Network inputs for Logic Cell Array 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.


ORCAD VST SIMULATE PROGRAM
LCA MODELS
XNF2VST TRANSLATOR
XC-DS351 OrCAD VST Simulator Interface
XNF
From XC-DS501 XACT Design Implementation System

1970 01

XILINX

XC-DS22 and XC-DS221 PC-SILOS™ Simulators

Product Brief

FEATURES

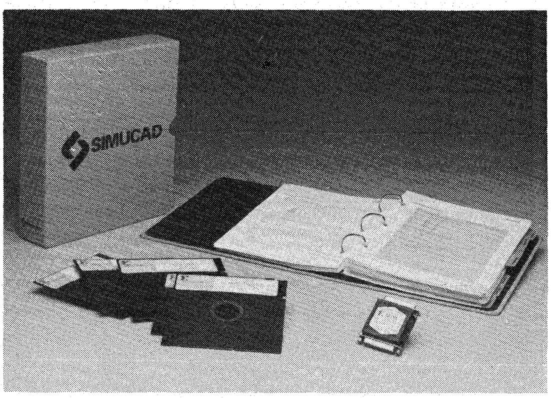
- PC based simulator for LCA design verification
- Simulates any LCA design, regardless of design input format (combined logic schematics, Boolean equations, and state machine descriptions)
- General purpose event driven logic and timing simulator
- Input automatically generated from XNF file
- Control 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
- Runs on an IBM® PC/AT™ or compatible personal computer
- Simulates logic complexities up to 16,000 gates (DS22) or 5,000 gates (DS221)

GENERAL

PC-SILOS is a powerful PC based simulator that provides event driven logic and timing simulation of Logic Cell™ Array 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 Logic Cell Array designs are automatically created by the XNF2SILO utility. The network includes logic and routing delay parameters and setup 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.



195

XC-DS112 Configuration PROM Programmer

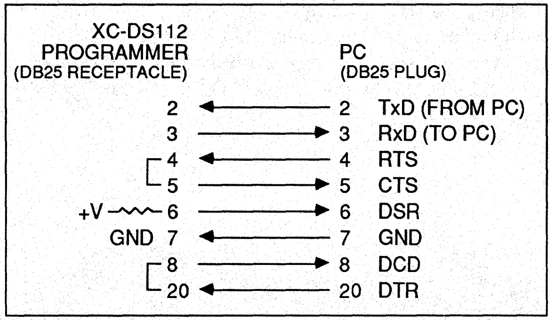
Product Brief

FEATURES

- Programs XC1736 Serial Configuration PROM
- Connects to serial port of IBM PC/AT or compatibles
- Operates from PC via software provided with programming unit
- Accepts HEX format data files created by the DS501 XACT Design Implementation System

GENERAL

Designers using the Xilinx XC1736 Serial Configuration PROM—an 8-pin mini-DIP PROM used to configure programmable gate arrays—can program the XC1736 with the XC-DS112 Configuration PROM Programmer.



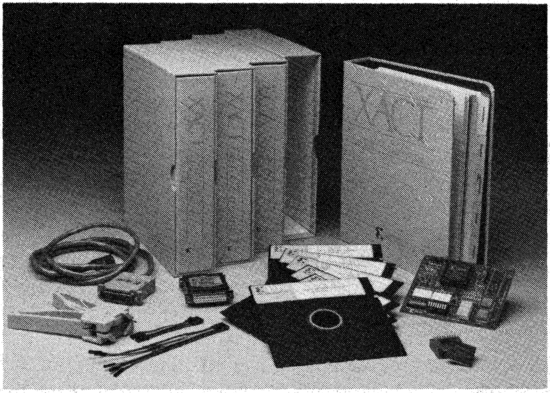
1960 01

XC-DS112 Interface to PC

The programming unit connects to a serial port of an IBM PC/AT or compatible and is controlled using PC-based software included with the XC-DS112.

Designers compile their Logic Cell Array designs into a standard HEX format file using the PC-based XACT development system. The programming software provided with the XC-DS112 is then used to download the HEX file into the programming unit and program a XC1736.

Since the XC-DS112 software is separate from the XACT software, the XC-DS112 software and programming unit can be installed on a PC other than the one used for design development—such as a PC located in a manufacturing area.



XC-DS26, XC-DS27, XC-DS28 XACTOR™ In-Circuit Design Verifier

Product Brief

FEATURES

- Real time in-circuit verification in user's target system
- Concurrent emulation of up to four devices
- Readback and display of Logic Cell™ Array 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 an IBM® 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 setup 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.

1968

XILINX

Development Systems Ordering Information

Further information is available from your local distributor sales office or the nearest Xilinx sales representitive.

ORDERING GUIDELINES

To design with Xilinx™ programmable gate arrays, designers should have *as a minimum*:

- XC-DS501 XACT™ Design Implementation System, *plus*
- LCA Interface and library for their schematic editor (see DS300 Optional Libraries and Interfaces, below)

PC-based designers who wish to use the Xilinx-provided FutureNet® DASH-LCA schematic editor can purchase all of the above from Xilinx as the "XC-DS53 XACT Design Implementation System with FutureNet DASH-LCA."

Workstation-based DS501 customers will initially received the PC version of the interactive XACT Design Editor; the workstation version of the XACT Design Editor will be provided as a no charge upgrade when available.

DS500 DESIGN IMPLEMENTATION DEVELOPMENT SYSTEMS

XC-DS501 XACT Design Implementation System for **IBM PC/AT**
XC-SC501 Annual Software Support Agreement

Bundled system equivalent to DS21 plus DS23—contains graphical interactive LCA editing tool plus all non-graphical, automated implementation tools.

- Design Manager provides convenient front-end for all Xilinx programs
- Accepts any combination of schematic netlists (XNF format) and PALASM-compatible text files (also accepts input form ABEL, CUPL, PLDesigner, and Log/IC)
- Automatic placement and routing program reduces design implementation time
- Logic synthesis software efficiently optimizes designs for LCA
- Automatically eliminates unused, disabled logic
- Bitstream compiler compiles design into LCA programming bitstream
- Graphical LCA design editor permits interactive editing of design placement/routing
- Download cable facilitates in-circuit debugging of LCA designs

XC-DS501-AP1 XACT Design Implementation System for **Apollo workstation** and PC/AT
XC-SC501-AP1 Annual Software Support Agreement

XC-DS501-SN1 XACT Design Implementation System for **Sun® 3 workstation** and PC/AT
XC-SC501-SN1 Annual Software Support Agreement

AP1 and SN1 versions contain PC-based XACT Design Editor and workstation-based enhanced automated design implementation software (includes bitstream compiler). Designer must have access to a PC/AT to run the XACT Design Editor for interactive placement/routing editing. An update to the workstation version of XACT Design Editor will be shipped when available.

- Accepts any combination of schematics netlists (XNF format) and PALASM-compatible text files (also accepts input from ABEL, CUPL, PLDesigner, and Log/IC)
- Automatic placement and routing program reduces design implementation time
- Logic synthesis software efficiently optimizes designs for LCA
- Automatically eliminates unused, disabled logic
- Bitstream compiler generates LCA programmable bitstream on workstation system.

DS500 DESIGN IMPLEMENTATION DEVELOPMENT SYSTEMS (Con't)

XC-DS53	XACT Design Implementation System with FutureNet DASH-LCA
XC-SC53	Annual Software Support Agreement *Includes **all** software required to enter programmable gate array schematics, place/route designs, and compile designs into configuration bitstreams to program LCA devices* • Includes DS501 XACT Design Implementation System ***plus*** a FutureNet DASH-LCA schematic editor (a version of FutureNet DASH4 solely for LCA designs) ***plus*** a DS31 FutureNet library and interface • Design Manager provides convenient front-end for all Xilinx programs • Includes support for PALASM®-compatible Boolean equation entry (also accepts input from ABEL, CUPL, PLDesigner, and Log/IC) • Runs on IBM® PC/AT™ or compatible

DS100 OPTIONAL HARDWARE PRODUCTS

XC-DS112*	Enhanced Serial Configuration PROM Programmer
XC-SC112	Annual Software Support Agreement

DS200 OPTIONAL DESIGN IMPLEMENTATION & VERIFICATION TOOLS

XC-DS21	XACT Design Editor for IBM PC
XC-SC21	Annual Software Support Agreement • Graphics-based interactive LCA design editor • Timing calculator, bitstream compiler, and macro library • Download cable, demo board and 2 LCA devices
XC-DS23	Automated Design Implementation (A.D.I.) software for IBM PC
XC-SC23	Annual Software Support Agreement • Accepts any combination of schematic netlists (XNF format) and PALASM-compatible text files • Logic synthesis software efficiently optimizes designs for LCA • Automatically eliminates unused, disabled logic • Automatic placement and routing program reduces design implementation time
XC-DS22	P/C-SILOS™ Simulator (16,000 gates) with LCA interface
XC-SC22	Annual Software Support Agreement
XC-DS221	P/C-SILOS™ Simulator (5,000 gates) with LCA interface
XC-SC221	Annual Software Support Agreement
XC-DS28	XACTOR™ In-Circuit Design Verifier (Controller Hardware and Software) *Order one XC-DS28 Controller, from one to four XC-DS26 Emulation Pods, and one XC-DS27 Emulation Header (of desired package type) per Emulation Pod* • Runs on IBM PC/AT or compatible (connects to serial port) • Controller supports up to four XC-DS26 Emulation Pods • XC-DS26 Emulation Pod accepts any XC-DS27 Emulation Header • XC-DS27 Emulation Header connects Emulation Pod to LCA socket in target system
XC-DS26	Emulation Pod for XACTOR (accepts any XC-DS27 Emulation Header)
XC-DS27-PD48	Emulation Header, 48 Pin DIP package
XC-DS27-PC68	Emulation Header, 68 Pin PLCC package
XC-DS27-PG68	Emulation Header, 68 Pin Grid Array package
XC-DS27-PC84	Emulation Header, 84 Pin PLCC package
XC-DS27-PG84	Emulation Header, 84 Pin Grid Array package
XC-DS27-PG132	Emulation Header, 132 Pin Grid Array package
XC-DS27-PC175	Emulation Header, 175 Pin Grid Array package

* XC-DS112 is an enhanced version of former XC-DS81 serial configuration PROM programmer

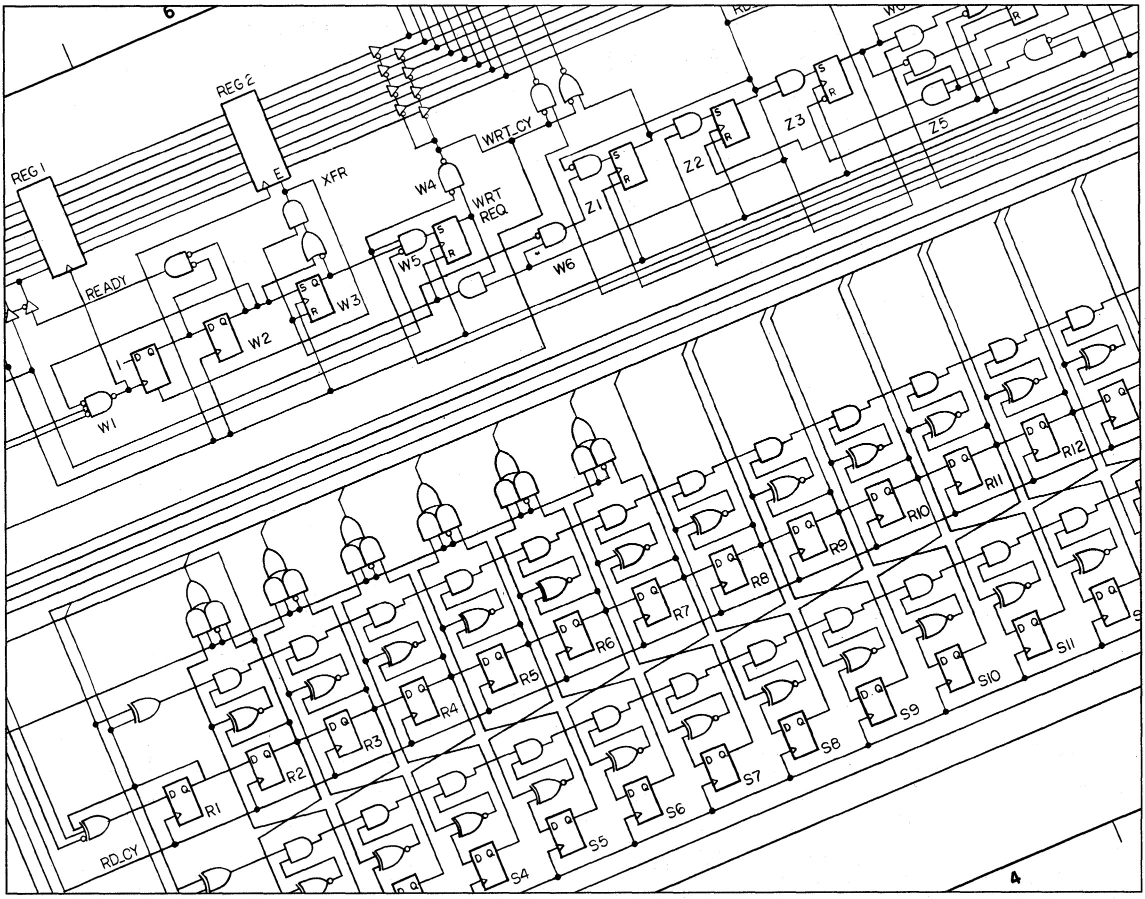
DS300 OPTIONAL LIBRARIES AND INTERFACES

XC-DS31	FutureNet DASH Schematic Interface and Library
XC-SC31	Annual Software Support Agreement
XC-DS311	TTL Library for FutureNet DASH and DASH-LCA
XC-SC311	Annual Software Support Agreement
XC-DS32	Schema II+ Schematic Interface and Library
XC-SC32	Annual Software Support Agreement
XC-DS33	Daisy® Schematic and Unit-delay Simulation Interface and Library
XC-SC33	Annual Software Support Agreement
XC-DS343*	Mentor®-certified Schematic and Full-timing Simulation Interface and Library
XC-SC343	Annual Software Support Agreement
XC-DS35	OrCAD/SDT® Schematic Interface and Library
XC-SC35	Annual Software Support Agreement
XC-DS351	OrCAD/VST® Simulation Interface
XC-SC351	Annual Software Support Agreement
XC-MANUAL	Additional Xilinx Development System Documention • Complete XACT LCA Development System Reference Manuals (Volumes I, II) • LCA User's Guide Manual (including several Application Notes) • XACT Macro Library Manual

* XC-DS343 is an enhanced version of former XC-DS34 which now includes full timing simulation

XILINX

The Programmable Gate Array Company



XC3090 Die

SECTION 6

Applications

XILINX

Applications

1962

XILINX

Applications

INTRODUCTION

The following pages show examples of systems and subsystems solutions using Xilinx 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 devices' capabilities, to highlight special advantage, to emphasize the best design methods, and in general to stimulate the designer's imagination.

A Xilinx programmable gate array 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 architectual details of the Xilinx programmable gate 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 family programmable gate arrays 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 a maximum of 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 their 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.

Programmable Gate Arrays offer a very large number of flip-flops (128 in the CLBs and another 128 in the IOB of the 3020, a total of more than 800 in the 3090). Different from the situation with Gate Arrays, these LCA flip-flops cannot, or do not have to be, traded off against logic. Logic coexists with the flip-flops in the form of function generators. The function generators are surprisingly versatile pieces of logic, unlimited in their flexibilty, 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 Xilinx Programmable Gate Arrays are extremely efficient. Certain high fan-in functions like ALUs tend to be less efficient, and bus-oriented designs have to be routed carefully to take advantage of the long lines and 3-state drivers of the 3000-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. Finally, there is interest in converting existing PAL designs to LCAs. All these subjects are covered in the beginning of this chapter.

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 of 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 3000 series CLB.

Address decoding is the classical strength of PAL devices. It is done efficiently in LCAs if the complete function includes the combination of several addresses or groups of addresses.

ALUs consume many LCA 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

LCAs 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 advantage, 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, but which 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.

SYSTEMS DESCRIPTIONS

LCAs are universal programming building blocks, that are used in a wide variety of systems.

An 8-digit frequency counter implemented in a 2064 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.

Article reprints from the trade press indicate the broad range of LCA applications.

The purpose of this applications chapter is not to provide cookbook solutions, but rather to stimulate the imagination, convey ideas and demonstrate that LCAs offer a better solution for a large variety of digital designs.

XILINX

Estimating Size and Performance

BY DAVE LAUTZENHEISER

INTRODUCTION

Programmable gate arrays are available in a range of densities and speed grades. Before committing resources to design implementation, the user should estimate which Programmable Gate Array best fits the specific application. Such 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 will always be 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 the fit of a system into a Programmable Gate Array can be made by counting the required input and output pins and internal storage elements. Table 1 lists Xilinx's XC2000 and XC3000 series Logic Cell Array (LCA) 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, one 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 programmable gate array 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 number of storage elements required is less than the number of logic storage elements, the desired functions can probably be performed in the chosen Logic Cell Array.

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 Xilinx 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.

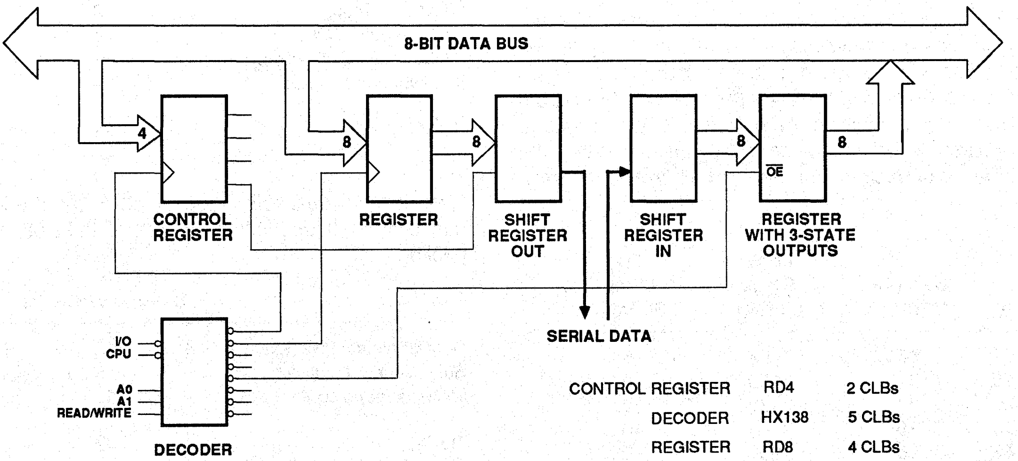
Function	Storage Elements
32 bit DMA (two-32 bit Counters)	64
Refresh generation (minimum)	10
32 bit control register	32
32 bit processor memory Address	32
Error check and correct	44
Miscellaneous control	20
TOTAL	202

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 pincount device a better choice. Second, high performance requires making the best use of a device's features. The 32-bit bus may impose critical performance requirements. Only the XC3064 and XC3090 allow 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 Configurable Logic Block (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. Where there are differences between XC2000 and XC3000 family block counts, these 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.



CONTROL REGISTER	RD4	2 CLBs
DECODER	HX138	5 CLBs
REGISTER	RD8	4 CLBs
SHIFT REGISTER OUT	RS8PR	4 CLBs
SHIFT REGISTER IN	RS8	4 CLBs
REGISTER IN 3-STATE	HX374	4 CLBs
TOTAL		23 CLBs

1133 01

In many schematics there are collections of random gates which 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 4 for XC2000 family devices, or 5 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.

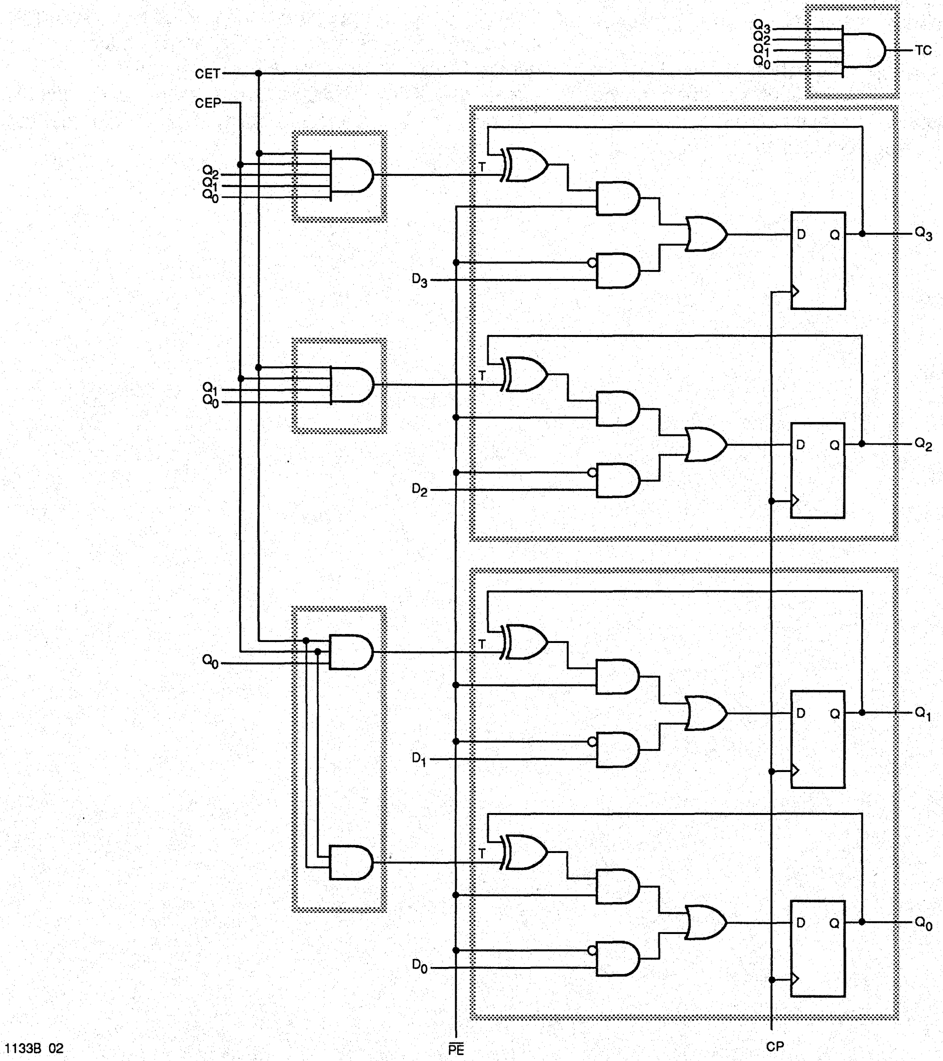


Figure 2. Six CLBs Are Required to Implement a 74161 Binary Counter

Estimating Performance

After selecting the right programmable gate array 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, and the LCA datasheet specifies the block delay for each level.

Some routing delay must be added to the block delay. Such routing delays can add 25–50% to the block delays.

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 nsec block delays, this gives 30 nsec delay from the first clock to the setup required for the next clock. Allowing 30% for routing (10 nsec) and 8 nsec for setup gives a total delay of 48 nsec. This should allow 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 Programmable Gate Arrays, estimating logic capacity and performance should precede device selection. If the design fits, the programmable gate array development system, and the simplicity of in-system design verification allow cost effective and rapid design implementation.

Of course, specifications sometimes change during the execution of a design. Logic changes may result in different requirements for I/O and logic blocks. In such cases the Xilinx products product line simplifies the migration to a compatible array that meets the new requirements.

Incorporating PLD Equations into LCAs

Application Brief BY THOMAS WAUGH

The ideal design environment allows entering a design in many different ways, independent of the target technology, then mapping that design into the technology that best suits the application. The design entry system should support the integration of several design entry methods. For example, a designer should be able to enter a combination of schematics, Boolean equations, and state machines, using whichever method best expresses the logic in any part of the design. Once the design is entered, the designer should then be able to optimize it for the particular technology in which it is to be implemented. Xilinx's Programmable Gate Array design tools support the integration of both schematics, state machines and Boolean equations into a design.

Some designs or parts of designs are more easily expressed in Boolean equations or state machines than in schematics. A simple yet effective illustration of this is a seven segment decoder design, which expressed schematically requires many AND and OR gates, but expressed with equations requires only a text file. The XC-DS23 Automatic Design Implementation (ADI) software allows designers to integrate designs entered using a schematic editor with Boolean equations expressed in the PALASM2 format. Designers can enter Boolean equations directly using a text editor or they can create them with ABEL or CUPL using their PALASM translators. The ability to create a design using schematics, equations, state machines, or a combination of the three makes the design entry process more flexible and more powerful.

There are four programs in the XC-DS23 that help the designer incorporate Boolean equations into a design. A translator program called PDS2XNF translates PALASM2 design files into Xilinx Netlist Format (XNF) files. XNF is a standard interface between the Xilinx Development System and various design entry packages, e.g., schematic editors. An optimization program called XNFOPT optimizes the logic for Xilinx's Logic Cell Array (LCA) architecture for highest efficiency in terms of density or speed. A program called XNFMERGE, merges various hierarchically related design files such as schematics and PLD designs into one flattened file. After merging, a program called XNF2LCA partitions a design's logic and assigns the logic to the elements of the LCA: configurable logic blocks (CLBs) and input/output blocks (IOBs). These four programs comprise the tools necessary to create designs that integrate schematics and equations.

INCORPORATING BOOLEAN EQUATIONS INTO SCHEMATICS

There are a number of ways in which PLD equations can be used in designing LCAs. Perhaps the best way is to incorporate PLDs into a schematic (see Figure 1).

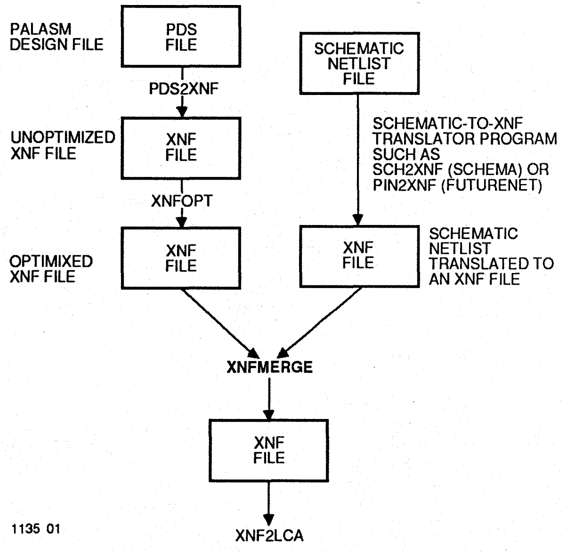


Figure 1. Design Flow of PLDs in a Schematic

Xilinx provides PLD symbols for both Schema II+ and DASH-LCA, so that designers can call up PLD symbols into their schematics. The user attaches a special attribute called a "FILE=" parameter to each PLD symbol. This parameter (FILE= *file_name*) references an optimized XNF file that contains the translated Boolean equations for that particular PLD (see Figure 2).

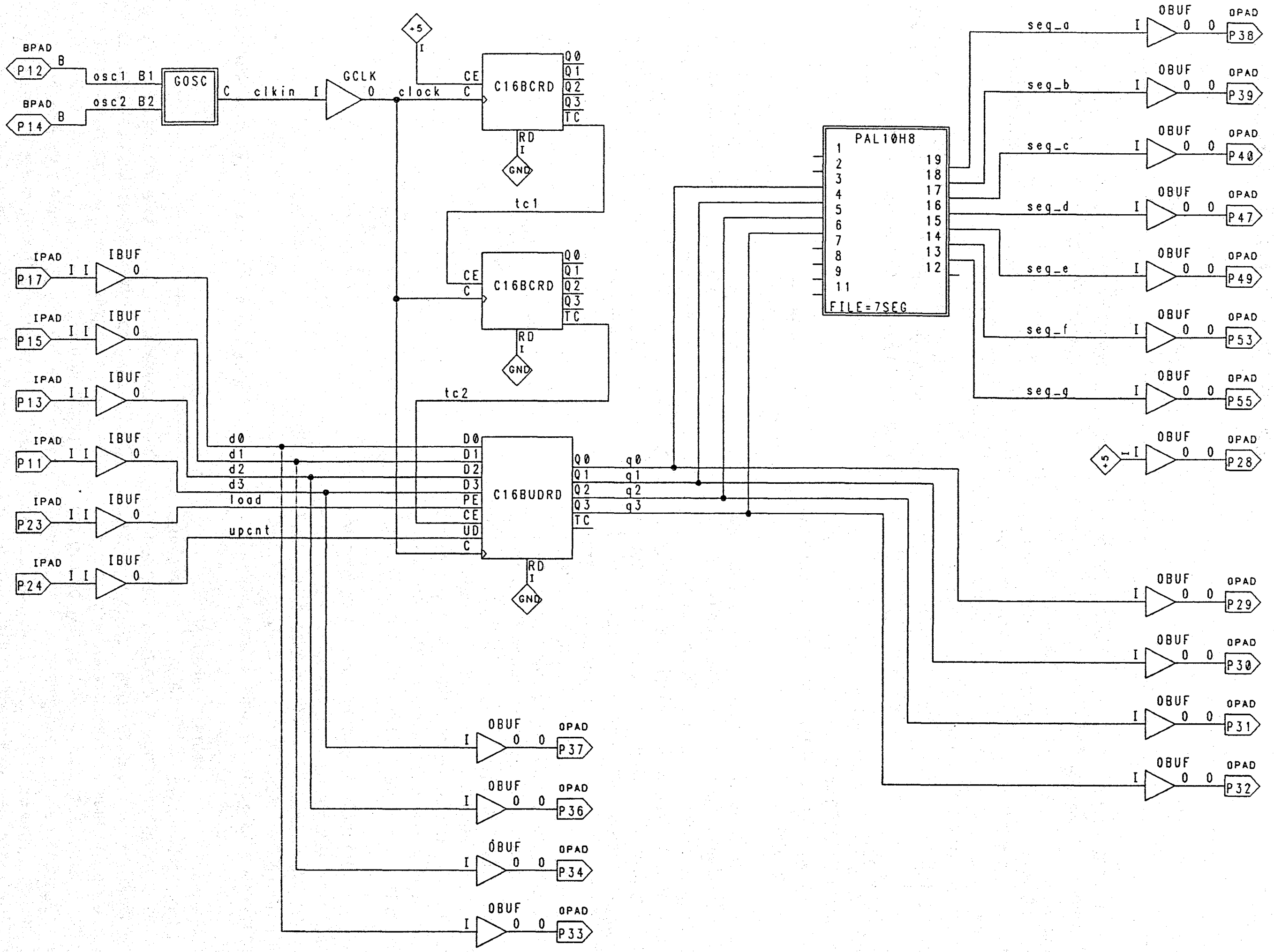


Figure 2. FutureNet Schematic of a Counter with a 7-Segment Decoder

```
TITLE          7SEG.PDS
AUTHOR         BART  REYNOLDS  AND  THOMAS  WAUGH
COMPANY               XILINX
DATE           APRIL  8,  1988

CHIP           7SEG     PAL10H8

;Input Pins     1  2  3  4  5  6  7  8  9 10
               NC NC D0 D1 D2 D3 NC NC NC NC
;Output Pins   11 12 13 14 15 16 17 18 19 20
               NC NC  G  F  E  D  C B  A  NC
;
;Input combinations
;
STRING ZERO         '/D3   * /D2   * /D1   * /D0'
STRING ONE          '/D3   * /D2   * /D1   *  D0'
STRING TWO          '/D3   * /D2   *  D1   * /D0'
STRING THREE        '/D3   * /D2   *  D1   *  D0'
STRING FOUR         '/D3   *  D2   * /D1   * /D0'
STRING FIVE         '/D3   *  D2   * /D1   *  D0'
STRING SIX          '/D3   *  D2   *  D1   * /D0'
STRING SEVEN        '/D3   *  D2   *  D1   *  D0'
STRING EIGHT        ' D3   * /D2   * /D1   * /D0'
STRING NINE         ' D3   * /D2   * /D1   *  D0'
STRING TEN          ' D3   * /D2   *  D1   * /D0'
STRING ELEVEN       ' D3   * /D2   *  D1   *  D0'
STRING TWELVE       ' D3   *  D2   * /D1   * /D0'
STRING THIRTEEN     ' D3   *  D2   * /D1   *  D0'
STRING FOURTEEN     ' D3   *  D2   *  D1   * /D0'
STRING FIFTEEN      ' D3   *  D2   *  D1   *  D0'

EQUATIONS
A =  ZERO + TWO + THREE + FIVE + SIX + SEVEN
     + EIGHT + NINE + TEN + TWELVE + FOURTEEN
     + FIFTEEN
B =  ZERO + ONE + TWO + THREE + FOUR + SEVEN
     + EIGHT + NINE + TEN + THIRTEEN
C =  ZERO + ONE + THREE + FOUR + FIVE + SIX
     + SEVEN + EIGHT + NINE + TEN + ELEVEN
     + THIRTEEN
D =  ZERO + TWO + THREE + FIVE + SIX + EIGHT
     + ELEVEN + TWELVE + THIRTEEN + FOURTEEN
E =  ZERO + TWO + SIX + EIGHT + TEN + ELEVEN
     + TWELVE + THIRTEEN + FOURTEEN + FIFTEEN
F =  ZERO + FOUR + FIVE + SIX + EIGHT + NINE
     +  TEN  +  ELEVEN  +  TWELVE  +  FOURTEEN
     +  FIFTEEN
G  =  TWO  +  THREE  +  FOUR  +  FIVE  +  SIX  +  EIGHT
     + NINE + TEN + ELEVEN + THIRTEEN
     + FOURTEEN + FIFTEEN
```

Figure 3. PALASM2 Design File Referenced in the Schematic

The PLD design file, entered in the PALASM2 format (see Figure 3), is translated into the Xilinx Netlist Format using the PDS2XNF program and then optimized for the LCA architecture using the XNFOPT program. PLD equations like the ones shown below are in a sum-of-products form that does not always map efficiently into the CLBs of an LCA. The logic represented by Boolean equations such as these must be optimized using the XNFOPT program before it can be translated into an LCA. The schematic portion of the design is also translated into an XNF file.

The schematic XNF file is considered the top level of the design hierarchy, since it contains references to other XNF files like optimized XNF files from PLD designs. The XNFMERGE program merges lower level XNF files into the top level XNF file from the schematic to form one flattened file that can be partitioned into the LCA architecture, then placed and routed just like any other design.

DESIGNING LOGIC CELL ARRAYS WITH BOOLEAN EQUATIONS

Another method of using equations to design LCAs is to express the LCA completely in terms of Boolean equations without regard for a particular PLD type (shown in Figure 4). The PDS2XNF program will translate PLD equations into an XNF file, even if the PLD design file does not correspond to an existing PLD device. The PDS2XNF program recognizes some features that are not permissible in the standard PALASM2 format, including unlimited signal names in the design pinlist, internal signals not named in the pinlist, arbitrarily complex equations, and 32 character signal names. LCA design with equations does not allow the use of all the features of the LCA (e.g., input flip-flops on XC3000 IOBs), but it is a useful and convenient way of designing LCAs completely and quickly using Boolean equations.

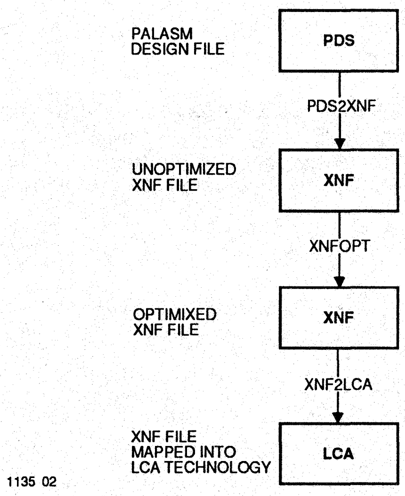


Figure 4. Flow Chart of Direct Translation of PLD Equations

DESIGN OPTIMIZATION

The cornerstone of the PLD conversion software is a program called XNFOPT. PLD equations are usually expressed in a sum-of-products form that is not optimally suited for the CLB architecture, with blocks of 4 or 5 inputs that perform any function of those inputs. The XNFOPT program is used to optimize logic so that the logic fits efficiently into the CLBs and IOBs of the LCA. XNF files translated from PLD equation design files should always be optimized using the XNFOPT program; in some situations XNFOPT may be used to optimize a design from a schematic as well, although there are some caveats.

There are some designs for which optimization using the XNFOPT program is inappropriate. XNFOPT is meant for synchronous designs, not asynchronous ones. XNFOPT significantly alters logic, so it may not preserve a design's asynchronous timing dependencies. Furthermore, although optimized logic is functionally equivalent to the original design, the structure may be significantly changed making debugging more difficult. It is generally a good idea to first simulate a design functionally to insure its correctness before optimizing it. Logic in an XNF file that is expressed as CLB or CLBMAP primitives cannot be optimized, as it is already partitioned into the LCA architecture.

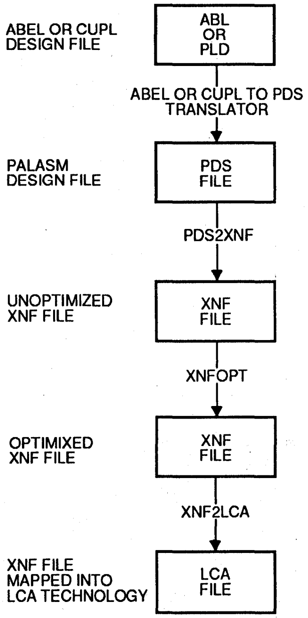
The XNFOPT program can be directed to optimize a design for speed or for density. Optimizing for density reduces the number of CLBs used by the design, while optimizing for speed reduces the number of levels of logic used by the design. In its default mode of operation, XNFOPT optimizes for density, using the -L option directs XNFOPT to optimize for speed.

STATE MACHINES

Xilinx LCA designs can be entered as state machines through either the ABEL or CUPL PLD entry language. Data I/O's ABEL software and Logical Devices' CUPL software offer options that convert their source design files into the PALASM2 Format. This PALASM2 format file can then be translated into an XNF file and optimized like a PLD equation file originally entered in PALASM2.

SUMMARY

In many cases, designs are most easily expressed using a combination of logic equations and schematics. Software tools in Xilinx's ADI package give designers the capability to integrate equation and schematic entry methods. The benefits of these logic synthesis tools are simpler and more powerful design entry, more efficient use of LCA logic, and clearer design documentation.



1135 03

Figure 5. Converting PLD Designs from ABEL or CUPL

XILINX

The XC2000 User's Guide to the XC3000 Family.

In late 1987 Xilinx introduced the XC3000 family which, within a year grew to five members: the XC3020, XC3030, XC3042, XC3064 and XC3090.

Designers already familiar with the original XC2000 family may be interested in the following synopsis of the differences between the 2000 and 3000 families.

COMBINATORIAL LOGIC:

- Combinatorial logic is wider (5 inputs vs. of 4 in XC2000 family).

CLB STORAGE ELEMENTS:

- There are two storage elements in a CLB.
- Storage element is a flip-flop only, the latch option is eliminated.
- Data input of either flip-flop can come from the combinatorial logic output, (F or G,) or from the one direct data input, DIN of each CLB.
- Clock input can be driven from local lines in addition to global and alternate clock lines.
- Enable Clock signal eliminates need for clock gating.
- Asynchronous set is removed but asynchronous reset is retained, consistent with TTL-MSI design practices.
- No flip-flop inputs are shared with combinatorial logic.
- Clock, Enable Clock, and Reset are common to both flip-flops.

IOBs

- Both direct input (I) and registered input (Q) are available simultaneously.
- Input storage element can be flip-flop or latch.
- Output signal and 3-state control signal can be inverted.
- Output signal can be either direct or registered.
- Output buffer has a programmable passive pull-up to prevent floating input.
- Output buffer has option to slow down the output transition to reduce transient noise.

ROUTING RESOURCES:

1 additional horizontal local line per row.

1 additional horizontal long line per row.

1 additional vertical clock line per column.

- All CLB inputs and outputs have access to both horizontal and vertical routing channels.
- Capability of magic box is enhanced.
- Both horizontal long lines have 3-state bus drivers.
- Bi-directional buffers are evenly distributed and used only when needed to boost a signal.

OTHER DIFFERENCES

- Crystal oscillator has a "÷ 2" option (50% duty cycle)
- Oscillator output can be connected to the general interconnect.
- Added fast, dedicated CMOS input buffer for the global and alternate clocks.
- Capability of the edge magic box is enhanced.
- Each edge has two IOCLKS, clock source multiplexers are located in each corner.
- IOCLK can be synchronized to the global and alternate clock. IOB on adjacent chip edges can be driven from the same clock source.
- All IOCLK clock drivers can be inverted.
- Two vertical long-lines in each column and one long-line on each horizontal and vertical edge can each be split into two half long-lines. (On the XC3020 this can only be done with one long-line on each edge.)

DIFFERENCE IN CONFIGURATION

- Parallel peripheral mode is added.
- Modified timing for slave mode avoids hold time.
- No RCLK output in master serial mode, use CCLK.
- Modified pinout for master and peripheral mode, all data pins are on the right hand edge.
- INIT state brought out as output.
- Done and Reset, programmable one clock early or late.
- PWRDN does not affect "House Cleaning".

Designing with the XC3000 Family

Application Brief BY THOMAS WAUGH

CLOCKING

Global and Alternate Clocks Buffers

There are two high-fanout, 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 bypasses 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 fanout 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.

THREE-STATE BUFFERS

Active High three-state is the same as active Low enable.

In other words: A "1" on the T pin of a TBUF or an OBUFZ three-states the output, and a "0" enables it.

Input/Output Blocks (IOBs)

Unused IOBs should be left unconfigured. They default to inputs pulled High with an internal resistor.

IOB pullup 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 on the opposite page

Designing with the XC2000 Family

Application Brief BY THOMAS WAUGH

CLOCKING

Global and Alternate Clocks Buffers

There are two high-fanout, 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 fanout 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 three-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 XILINX 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

These values were measured with the node pulled LOW. At a logic HIGH the resistor value is 5 to 10 times higher.

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 pin 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 Vcc decoupling.

OUTPUTS

All 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 Vcc:	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 nspF
Additional fall time for 812 pF	50 ns	64 ns
normalized	0.06 ns/pF	0.08 nspF

[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/MHz)
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 Vcc, varying only 0.6% for a 10% change in Vcc, but is very temperature dependent, increasing 40% when the temperature drops from 25°C to –30°C, decreasing 40% when the temperature increases from 25°C to +130°C.

Vcc	T	Freq
4.5V	25°C	687 kHz
5.0V	25°C	691 kHz
5.5V	25°C	695 kHz
4.5V	–30°C	966 kHz
4.5V	+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 4 MΩ.

A series-resonant crystal Y1 and additional phase-shifting components R2, C1, C2 complete the circuit.

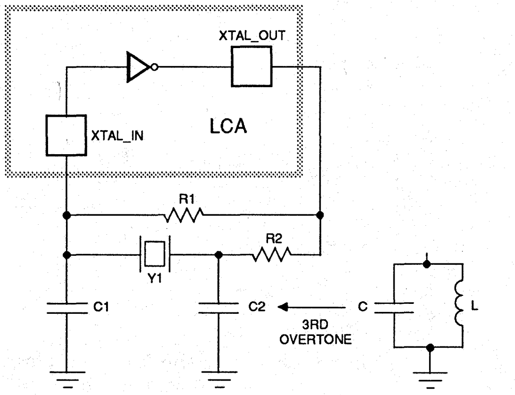
Fundamental Frequency Operation up to 24 MHz:

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)
32.00	1	60	20.6	430	23
35.00	1	44	24.0	310	23
49.00	1	31	28.6	190	23
72.00	1	18	37.5	150	12



1158 02

Crystal Oscillator

XILINX

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
clock-to-output	12 ns	8 ns	7 ns
routing	1 ns	1 ns	1 ns
logic set-up	12 ns	8 ns	7 ns
clock period	25 ns	17 ns	15 ns
clock frequency	40 MHz	59 MHz	67 MHz

2. A similar design with flip-flops several rows or columns apart would add routing delay:

	-50	-70	-100
clock-to-output	12 ns	8 ns	7 ns
routing	12 ns	8 ns	6 ns
logic set-up	12 ns	8 ns	7 ns
clock period	36 ns	24 ns	20 ns
clock frequency	28 MHz	42 MHz	50 MHz

3. An additional level of combinatorial logic plus routing reduces performance further:

	-50	-70	-100
clock-to-output	12 ns	8 ns	7 ns
routing	12 ns	8 ns	6 ns
logic delay	14 ns	9 ns	7 ns
routing	1 ns	1 ns	1 ns
logic set-up	12 ns	8 ns	7 ns
clock period	51 ns	34 ns	28 ns
clock frequency	20 MHz	29 MHz	36 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.

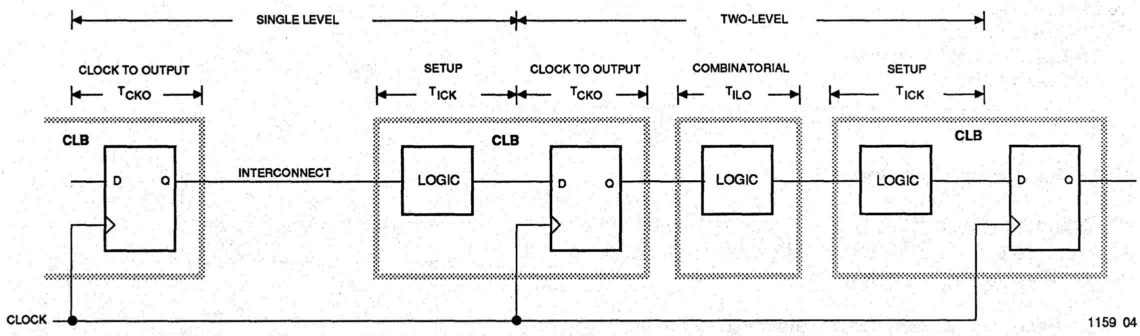


Figure 1. Critical Timing Parameters for Clocked CLB Driving Clocked CLB Directly (Single Level) and Driving it Through Additional Combinational Logic (Two-Level)

DESIGNING FOR HIGHEST DATA TRANSFER RATE BETWEEN 3000 FAMILY LCAs

Worst case analysis of a synchronous data transfer between 3000 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

3 + 6 + 13 = 22 ns for the -70 part
5 + 9 + 18 = 31 ns for the -50 part

On the receiving LCA, the input-pin to clock-pin set-up time is the specified I/O pad input set-up time (parameter TPICK 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:

20–3–6 = 11 ns for the -70 device
30–5–9 = 16 ns for the -50 device

Under these assumptions, the worst case (shortest) value for the clock period is:

22 + 11 = 33 ns, i.e., max 30 MHz for the -70 device
31 + 16 = 47 ns, i.e., max 21 MHz for the -50 device

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 -70 device, by 12 ns for the -50 device.

If this is not fast enough, there are design methods that can improve the performance. Let us assume a -70 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 4 ns to the clock delay, thus subtracts at least 3 ns 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 Vcc) 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.

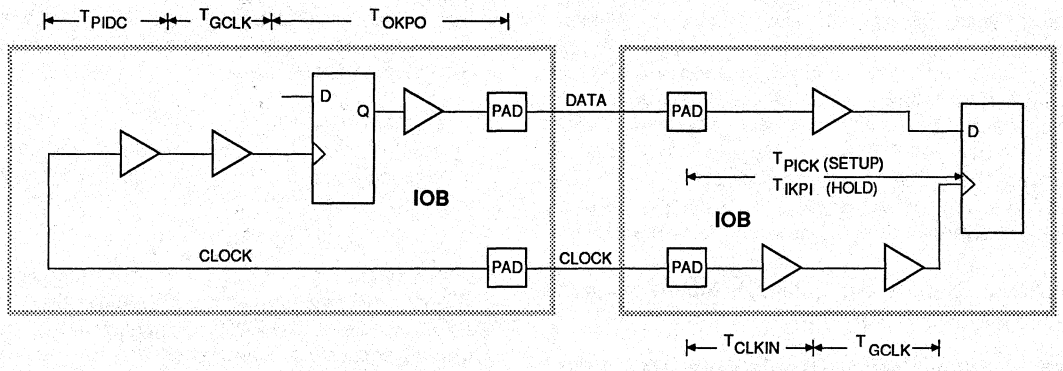


Figure 2. Critical Timing Parameters for Data Transfer Between LCAs

INPUT SET-UP TIME ON A 3000 SERIES LCA IS BETTER THAN THE SPECIFICATION.

The Xilinx 3000 series data sheet specifies a worst case input set-up time of 20 ns for the -70 speed grade (parameter #1 on page 41), 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 9 ns (XC3000-70), the sum of 3 ns for pad-to-CLKIN plus 6 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:

> 20 ns set-up time for a slow data input with an infinitely fast clock path
>
> 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, 15 ns (20 ns minus 54% of 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, 1.1 ns (10% of 20 ns minus 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 roughly inverse proportional to Vcc.

3. *Test Guardband.* The max delay test is performed at a temperature well above TMAX and a supply voltage well below Vcc 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 min. 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 2ns 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.

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 "LDC" pin as I/O.

During Configuration:

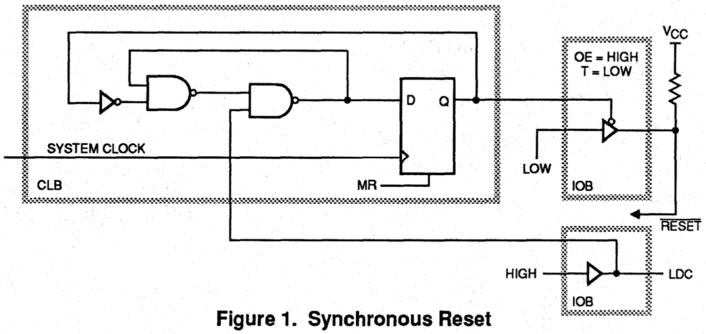
LDC (Low) holds D High, but Q is held Low by internal reset.
$\overline{\text{RESET}}$ is pulled High by internal and external resistors.

End of Configuration before first System Clock:

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.



1971 01

Figure 1. Synchronous Reset

XILINX

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

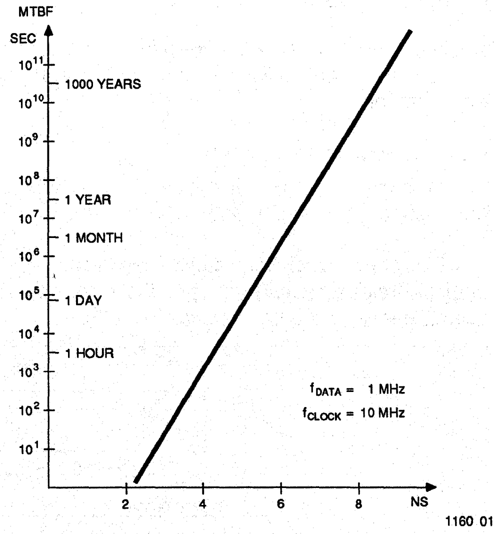
6.6 ns statistically once per year

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.

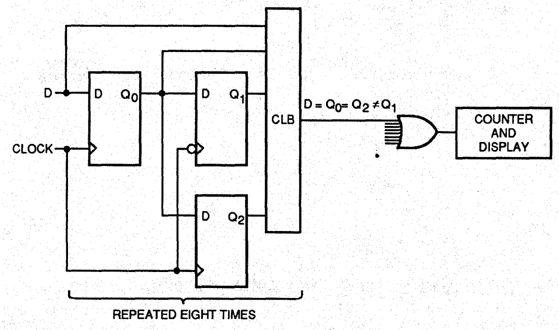
The MTBF curve was then normalized to a reasonable combination of clock and asynchronous data rates, using the generally accepted theory that, everything else being equal, the frequency of metastable events is proportioned to the product of the two frequencies at the D and CLK inputs of the flip-flop under test.

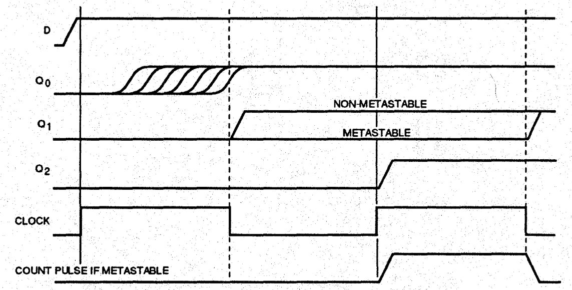
Assuming that the metastable window is 0.1 ns wide, and the clock is 10 MHz, one data change in 1000 will fall into the metastable window. A 1 MHz data rate gives an MTBF of 1 ms for an additional delay of zero. Each additional ns of acceptable delay increases the MTBF by a factor 40, see diagram on 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.





Battery Backup for Logic Cell Arrays

Application Brief BY DAVID P. LAUTZENHEISER

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 ($\overline{\text{PWRDWN}}$) pin and
- Switching between primary Vcc supply and battery.

Important considerations are:

- Insure that $\overline{\text{PWRDWN}}$ is asserted logic Low prior to Vcc falling, held Low during the primary Vcc loss time, and returned High after Vcc has returned to a normal level. $\overline{\text{PWRDWN}}$ 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 Volts ± 5% (±10% for Industrial and Military).

Figures 1 and 2 show two circuits which satisfy the above requirements. In Figure 1, discrete components are used to perform the switching while the power sensing is performed by a linear circuit, the TL7705A, made by Texas Instruments. This circuit lets the user adapt the Vcc pass-transistor to the load required. The same switching arrangement might also be used to battery back-up additional circuits, such as RAMs. The user also has control over the timing of the $\overline{\text{PWRDWN}}$ signal after Vcc has returned. While the LCA only requires microseconds to return to a normal state, other circuits may want to have a longer RESET period.

Figure 2 uses a single chip solution, the DS1259 from Dallas Semiconductor Corporation,
4350 Beltwood Parkway South, Dallas, Texas 75244,
Telephone (214) 450-0400.
The user only needs to supply the device and battery to accomplish the desired function. The DS1259 provides up to 250 mA of Icc with a drop of less than 0.2 Volt.

In both circuits, the user must supply both bulk and high speed decoupling. Use a large bulk capacitor in conjunction with a high speed 0.1 µF capacitor for each Vcc pin. Both capacitors should have low leakage to insure long battery life.

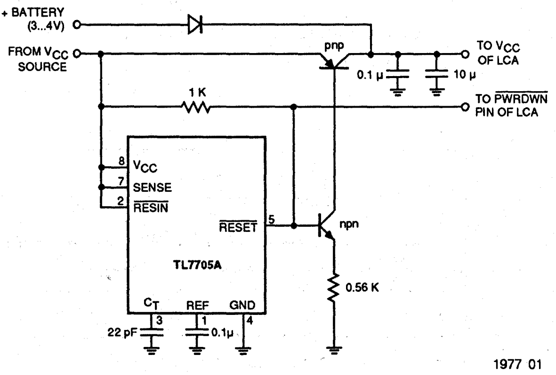


Figure 1. Discrete Battery Backup Circuit

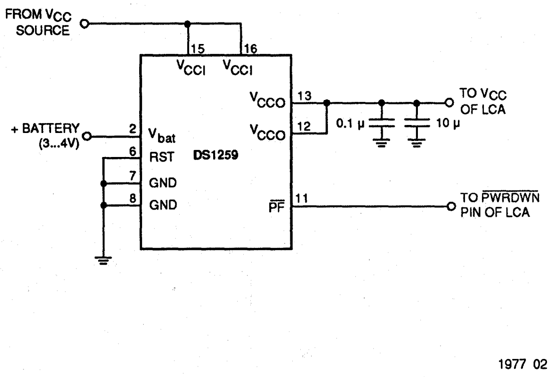


Figure 2. Integrated Battery Backup Circuit

Compact Multiplexer and Barrel Shifter

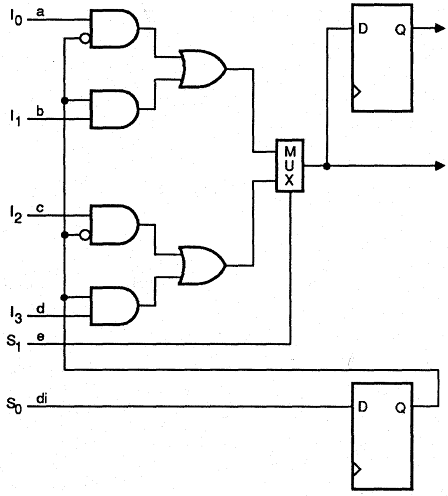
Application Brief

FOUR-INPUT MULTIPLEXER IN ONE CLB

Since the function generator in the XC3000 series CLB has only five inputs, it cannot directly implement a 4 input multiplexer which needs 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 4-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 4-input multiplexer by 50%, this approach is definitely worth considering.



1978 01

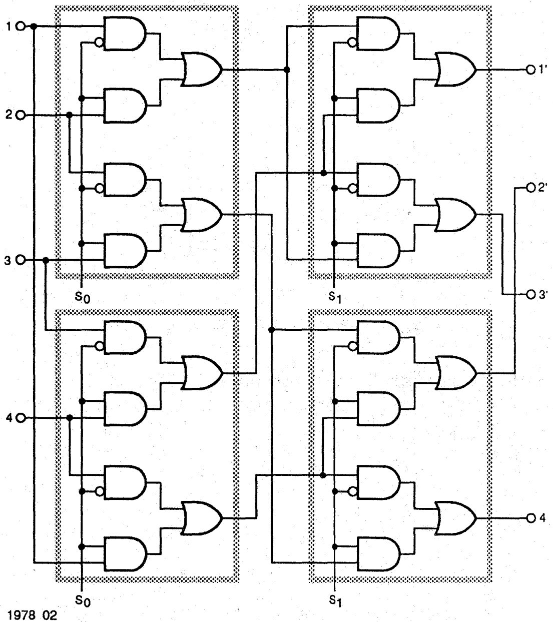
FOUR-BIT BARREL SHIFTER IN ONLY FOUR CLBs

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 4-input multiplexers, since each output can receive data from any input. Each four 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 cross-overs 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 02

Majority Logic, Parity

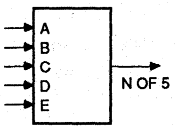
Application Brief

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.

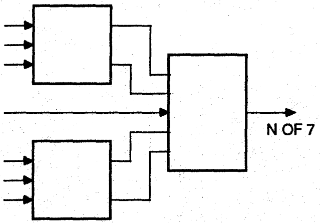
We can define the output F, G to be Low for 0, 1, or 2 inputs High, to be High for 3, 4, or 5 inputs High.

Majority logic is a special case of "N-of-X Decoding." A 3000-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.



5-INPUT MAJORITY FUNCTION:

F = ABC + ABD + ABE + ACD + ACE
+ ADE + BCD + BDE + BCE + CDE

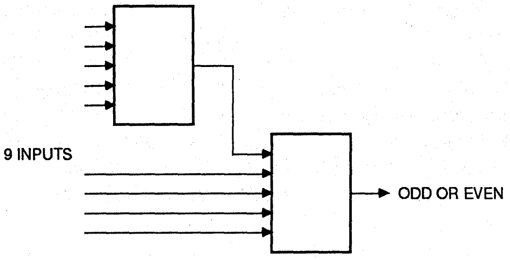


1139 01 **Figure 1. N of X, e.g., Majority Logic**

The first-level blocks can only have 3 inputs, since the two outputs can only encode 4 different states: none, one, two, or three active.

PARITY

Two CLBs can generate the parity for nine inputs, or can check a nine-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.

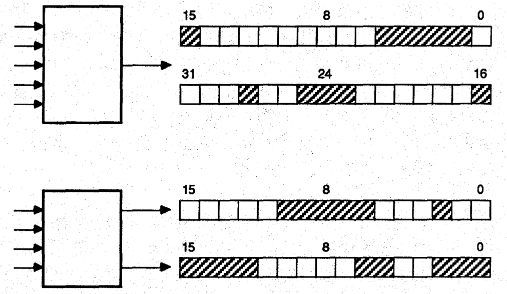


1139 02 **Figure 2. Parity**

Multiple Address Decoding

Application Brief

A 3000-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.



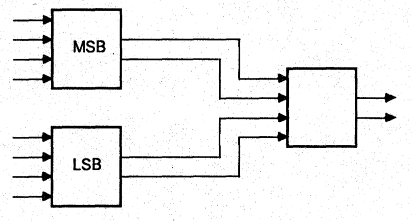
1140 01

Figure 1. Multiple Address Decoding

Three 3000-series CLBs can decode three distinct addresses in an 8-bit address field: One CLB decodes the lower 4 bits and encodes the result on its two outputs (00 = no match). The second CLB decodes and encodes the upper 4 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.



1140 02

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.

XILINX

Binary Adders, Subtractors and Accumulators

Application Brief

There are many different ways to implement binary adders, subtractors and accumulators in Xilinx 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 shiftregister, 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 8 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 6 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 an accumulator 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. Xilinx 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.

1979

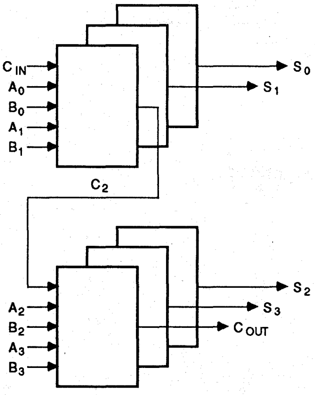
Adders and Comparators

Application Brief BY PETER ALFKE

The LCA-structure accommodates 1-bit and 2-bit adders very efficiently. A 1-bit adder with 3 inputs (A ,B, Cin) generating 2 outputs (S, Cout) fits exactly in one 2000 series CLB, where the flip-flop might be used for storing the carry in a bit serial adder. A 3000-series CLB can even include an additional control input, either ADD/SUBTRACT or ADD ENABLE.

A two-bit adder requires three 3000 series CLBs. The five inputs A0, B0, A1, B1, and Cin are common to all three CLBs, the outputs are S0, S1, and Cout. The propagation delay is only one CLB combinatorial delay, as little as 10 ns. Two such adders can be cascaded to form a four bit adder in 6 CLBs with a through-delay of two CLBs, i.e., 25 ns (allowing for some interconnect delay).

Four two-bit adders can be cascaded to form a byte-wide adder, using 12 CLBs with a through-delay of 4 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 Cin). 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 Cin 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 Cout.

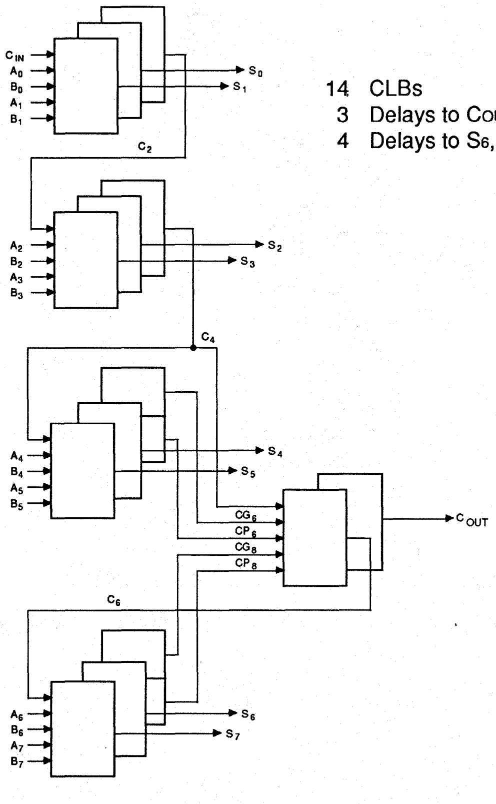


1141 01

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 6-24.

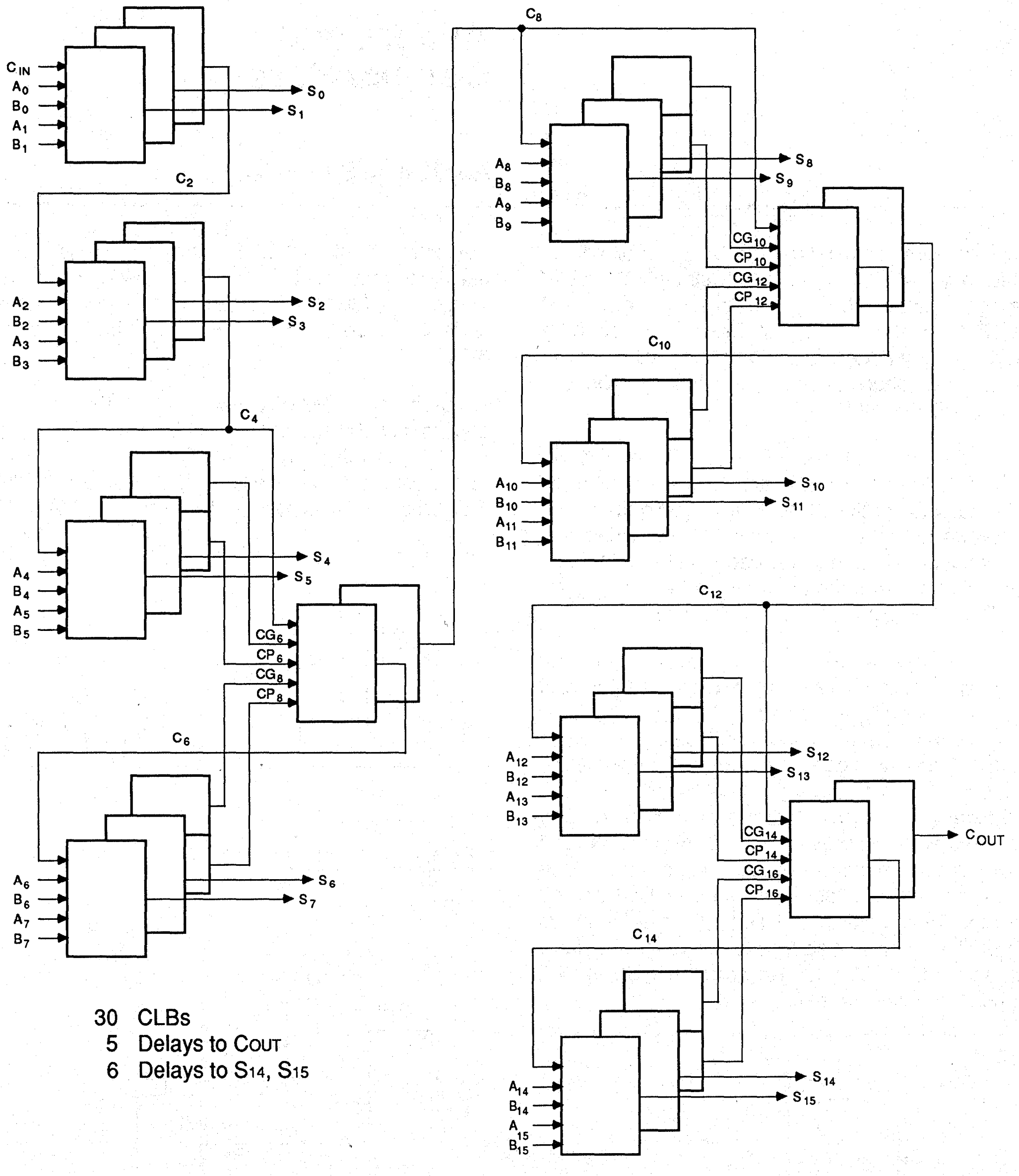
A 16-bit adder benefits from carry-lookahead. Simply cascading di-bit adders uses 24 CLBs at a max propagation delay of 8 CLBs from Cin to Cout or to S14, 15. A look-ahead carry scheme uses 30 CLB at a max prop delay of 5 CLBs from Cin to Cout (6 delays to S14, 15).



14 CLBs
3 Delays to COUT
4 Delays to S6, S7

1141 02

Figure 2. 8-Bit Adder with Carry Lookahead



30 CLBs
5 Delays to COUT
6 Delays to S14, S15

1141 03

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.

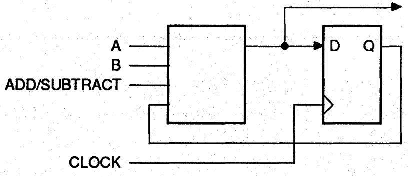
	Inputs				
Outputs	C_0	A_0	B_0	A_1	B_1
S_0=1	1	0	0	x	x
	0	1	0	x	x
	0	0	1	x	x
	1	1	1	x	x
S_1=1	x	1	1	0	0
	1	x	1	0	0
	1	1	x	0	0
	x	0	0	1	0
	0	x	0	1	0
	0	0	x	1	0
	x	0	0	0	1
	0	x	0	0	1
	0	0	x	0	1
	x	1	1	1	1
	1	x	1	1	1
	1	1	x	1	1
C_2=1	x	x	x	1	1
	x	1	1	1	0
	1	x	1	1	0
	1	1	x	1	0
	x	1	1	0	1
	1	x	1	0	1
	1	1	x	0	1

	Inputs				
Outputs	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	1	1	1	0
	x	1	0	0	1

	Inputs				
		Lower		Higher	
Outputs	C_{IN}	CP	CG	CP	CG
Lower Carry	x	x	1	x	x
	1	1	x	x	x
Higher Carry	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.



1141 04

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

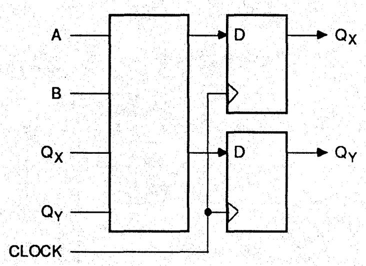
MSB first: Start with both flip-flops reset

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



1141 05

Figure 5. Serial Magnitude Comparator

Conditional Sum Adder Adds 16 Bits in 33 ns

Application Brief BY MATT KLEIN, HEWLETT-PACKARD

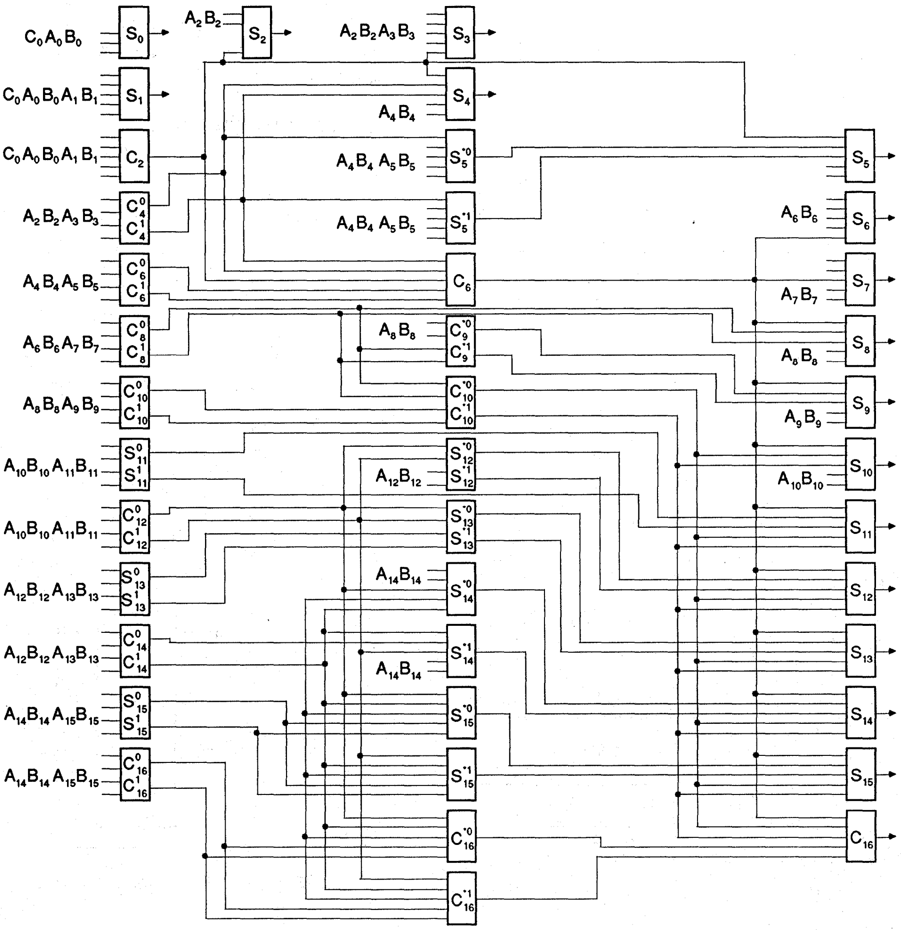
This circuit is based on a 1960 paper by J. Sklansky (see page 6-26). With careful placement and routing the total delay can be kept below 33 ns.

The block diagram below shows each CLB and its inputs and outputs.

27 of the CLBs each generate one function of up to 5 variables, 14 of the CLBs each generate two functions of four varibles. 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

A complete LCA file is available from Xilinx Applications and will be published in XCELL.



1988 01

16-Bit Conditional Sum Adder

Building Latches Out of Logic

Application Brief

Since the 3000-series, unlike the 2000-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 five-input logic structure allows an amazing diversity of latch designs: four examples are documented in the 3000-series Macro Library on page 15 and 16. Here are additional ideas:

With F fed back to close the feedback path, there are four control inputs left. We might call them Set, Reset, Data and Enable, defining them such that S and R are independent of E, but D is activated by E. We can still define any of these four inputs as active High or active Low. That gives us 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. Enable width = 19/14 ns assuming 5 ns interconnect delay.

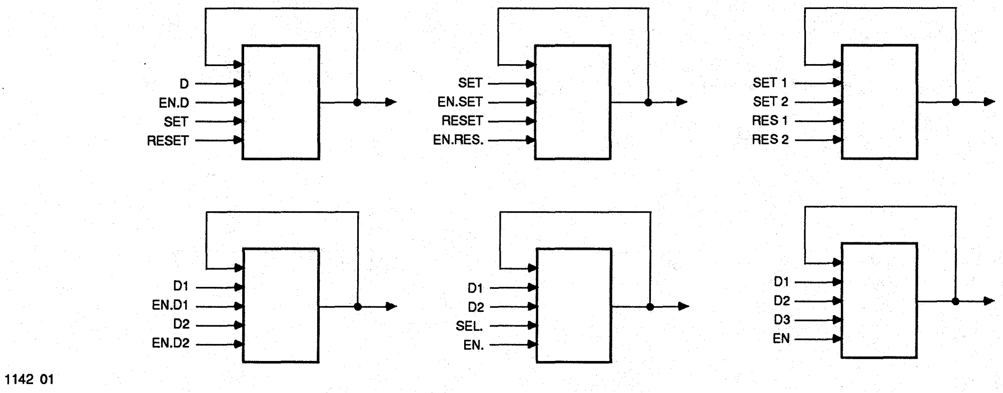


Figure 1. Latched Logic

Synchronous Counters, Fast and Compact

Application Brief

FULLY SYNCHRONOUS 4-BIT COUNTER USES ONLY TWO CLB'S TO COUNT ANY CODE

This four-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.

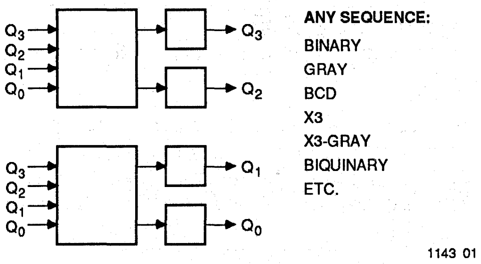


Figure 1. Synchronous 4-Bit Counter in 2 CLBs

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 3000-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, Q5, but with complete freedom to define its waveform

Q0 through Q4 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, Q5 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.

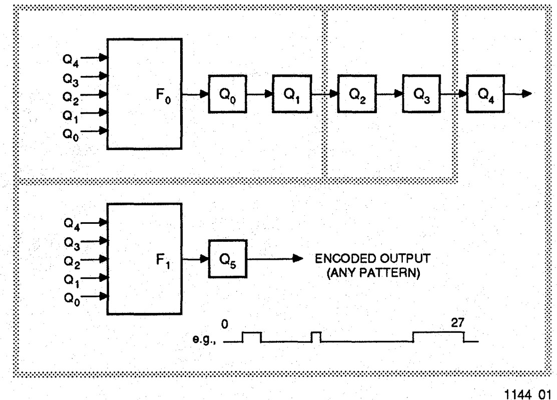


Figure 2. Synchronous 5-Bit Counter in 3 CLBs

XILINX

30 MHz Binary Counter Uses Less Than One CLB per Bit

Application Brief BY PETER ALFKE

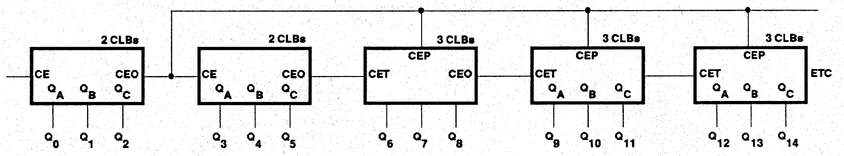
Borrowing the concept of Count Enable Trickle/Count Enable Parallel that was pioneered in the popular 74160 TTL-MSI counter, a fast non-loadable synchronous binary counter of arbitrary length can be implemented efficiently in the XC3000 series CLBs. For best partitioning into CLBs, the counter is segmented 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.

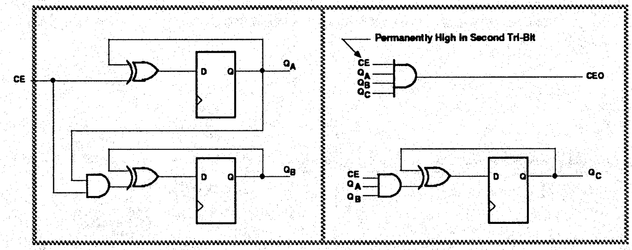
Each Count Enable Output from any other tri-bit drives the next more significant Count Enable Trickle (CET) input. The clock causes any tri-bit to increment if all its Count Enable inputs are active. CEO is active when all three bits are set AND CET is High. CEP does not affect CEO.

The least significant tri-bit thus stops the remaining counter chain for 7 out of 8 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 (TCKO + TILO), plus the CEP input set-up time for all other tri-bits (TICK), 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.

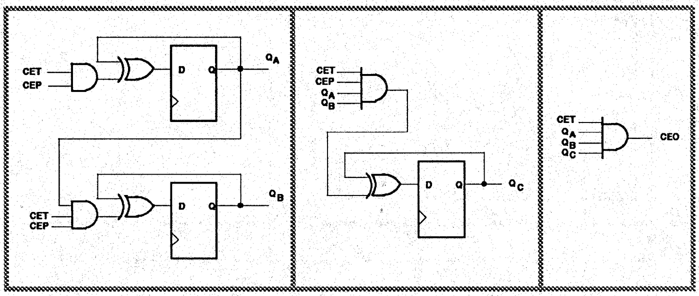
The two least significant tri-bits each have a single Count Enable input; they fit, therefore, in only two CLBs each. The higher tri-bits have two Count Enable inputs (CEP and CET) and require three CLBs.


2 CLBs
2 CLBs
3 CLBs
3 CLBs
3 CLBs
CE
CEO
CE
CEO
CEP
CET
CEO
CEP
CET
CEP
CET
ETC
Q_A Q_B Q_C
Q_0 Q_1 Q_2 Q_3 Q_4 Q_5 Q_6 Q_7 Q_8 Q_9 Q_{10} Q_{11} Q_{12} Q_{13} Q_{14}

1980 01 **30 MHz Non-Loadable Binary Counter, Expandable up to 60 Bits**



1980 02 **First and Second Tri-Bits Use Two CLBs Each**



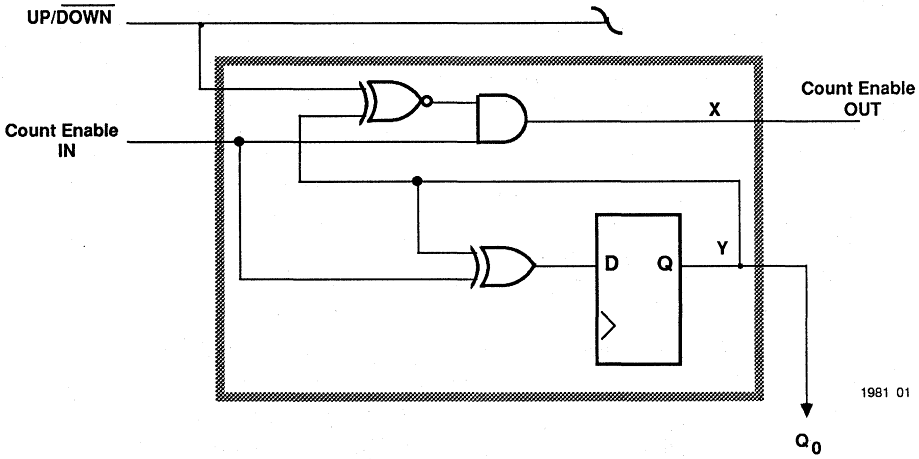
1980 03 **All More Significant Tri-Bits Use Three CLBs**

Up/Down Counter Uses Less Than One CLB per Bit

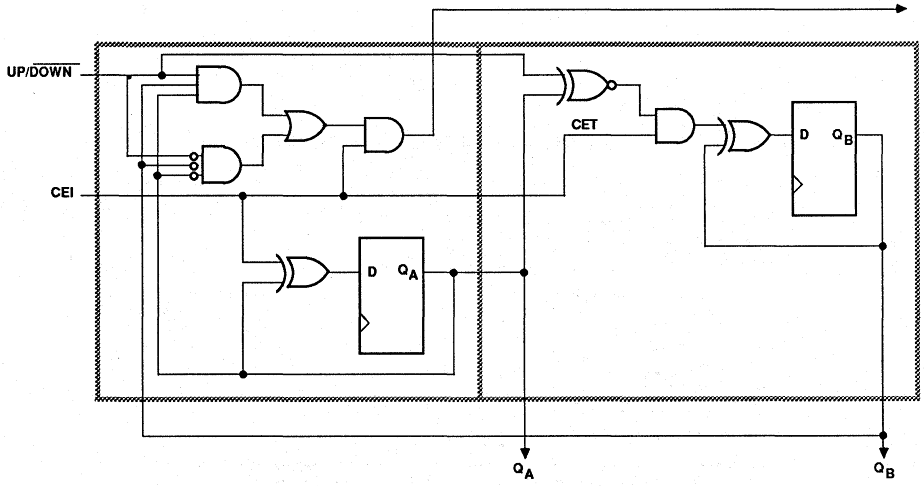
Application Brief BY PETER ALFKE

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 01



1981 02

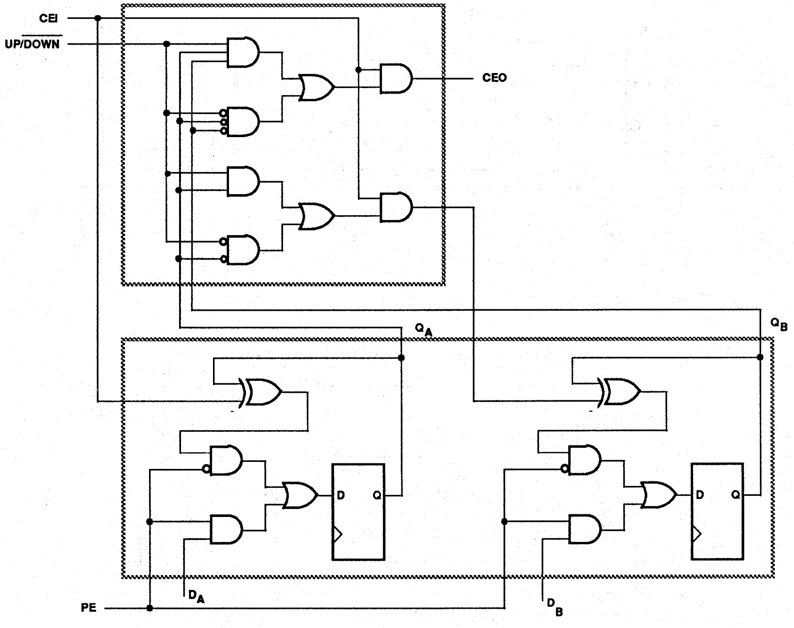
Loadable Up/Down Counter Uses One CLB per Bit

Application Brief BY PETER ALFKE

The five-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.



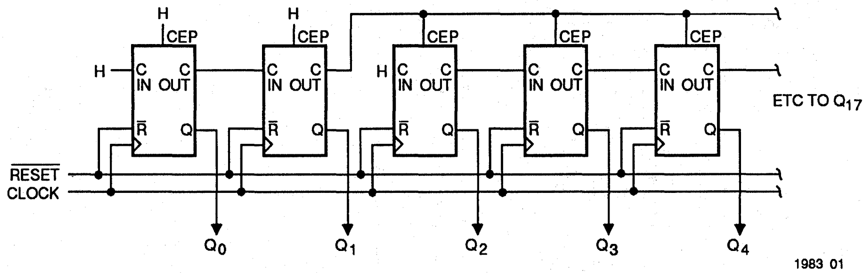
1982 01

30 MHz Binary Counter with Synchronous Reset

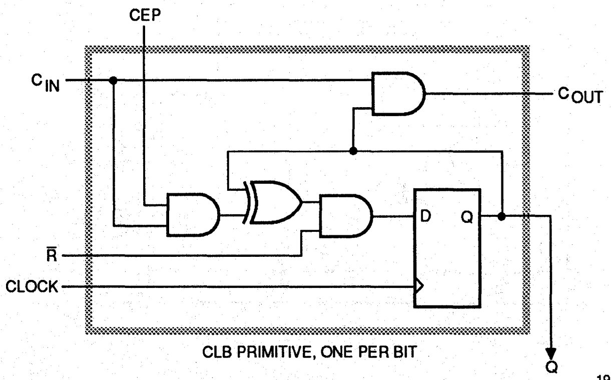
Application Brief BY PETER ALFKE

In many applications, design modularity is more important than highest clock speed and best space efficiency. This applications brief describes a counter design 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 one-bit block size. Any block increments only if both Count Enables are High, but the outgoing count enable (COUT) is not a function of CEP. The CEP input thus prevents erroneous counts while the ripple carry chain is settling.

A shorter counter (6 bits or less) drives the CEP net from the Qo 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. In order to achieve this performance, CEP and R̄ must be driven by long lines.



1983 01



CLB PRIMITIVE, ONE PER BIT

1983 02

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 Logic Cell Array, 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 microsecond.

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 4 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.

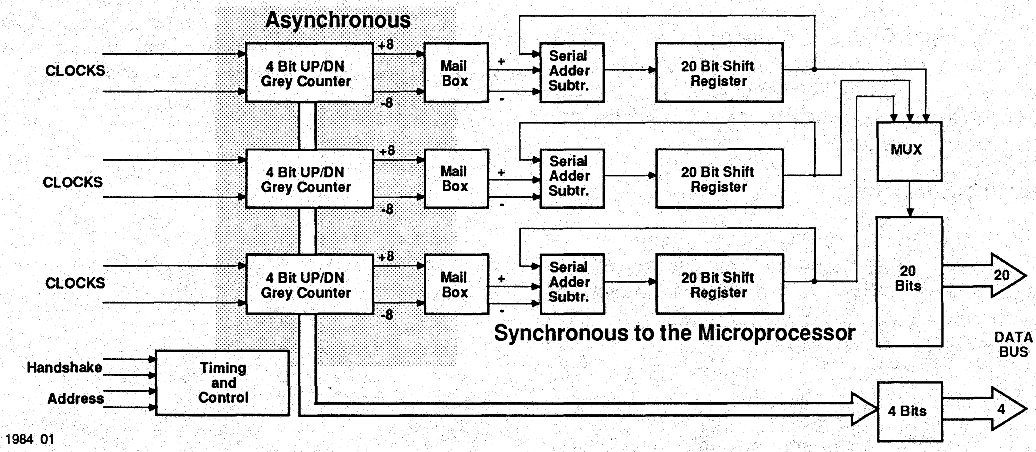


Figure 1. Triple 22-Bit Up/Down Counter with Microprocessor Interface

40 MHz Presettable Counter

Application Brief BY PETER ALFKE AND PERRY WU

This application note describes a new counter architecture used to implement a very high speed presettable, up to 40- bit long binary counter in an XC3020 Programmable Gate Array. 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.

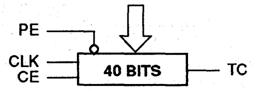
Synchronous counters allow decoding and presetting in one clock period, but pay for this with complex carry logic. Carry propagation is always the limiting factor in the traditional design of presettable synchronous counters, since the complete carry chain must reach a steady state before the next incoming clock edge. Brute force parallel decoding of all previous states becomes unmanageable beyond 8 stages, but cascaded decoding introduces additional delays. Either approach reduces the inherent resolution of the counter.

Decoding Terminal Count (TC) in order to preset the counter again, poses a similar problem. The design described in this paper separates the two functions of the carry chain into:

- One which decodes the terminal count of the whole counter and generates a Parallel Enable signal
- One which propagates the carry signal from the less significant to the more significant bit positions, and causes the appropriate flip-flop to toggle.

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 before. TC can, therefore, be decoded in a slow gating chain that starts at the most significant end of the counter.



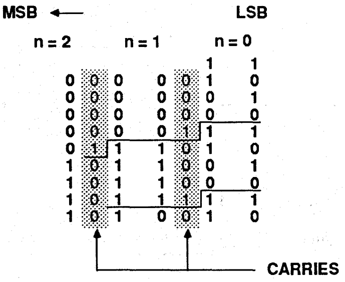
1145 01

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.

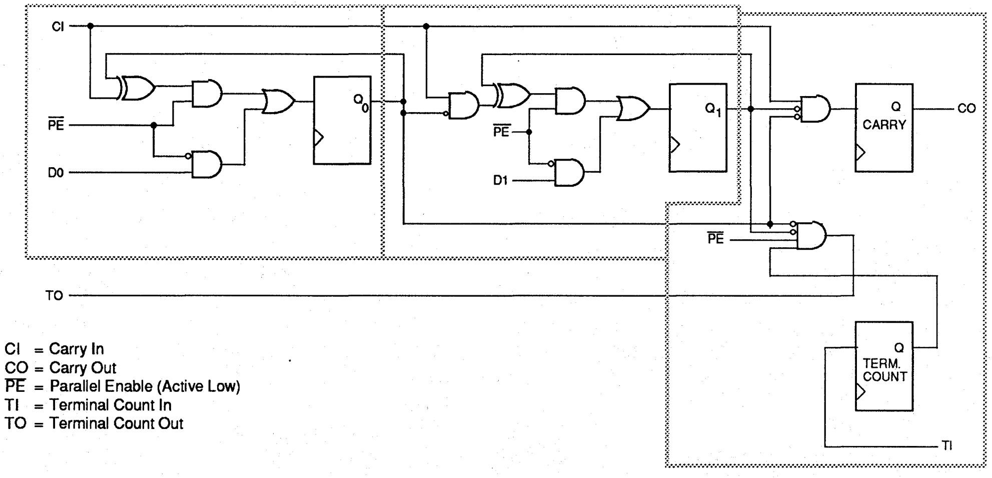
The counter is divided into a number of small sections, each two bits (a dibit) long, implemented as a synchronous presettable down-counter, with carry-in (=count enable), parallel enable and two data inputs. Terminal count (0.0) is decoded with an additional input coming from the next higher section. 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 dibit 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 dibit, except the first, makes its transition n clock pulses later than required by the binary code sequence (n is the relative position of the dibit, n=0 for the input dibit). 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, e.g., as the frequency determining counter in a phase-locked-loop frequency synthesizer.

Different from 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 9ns, which means that a -70 device can count at a 40 MHz rate (worst case).



n = 2			n = 1		n = 0	
					1	1
0	0	0	0	0	1	0
0	0	0	0	0	0	1
0	0	0	0	0	0	0
0	0	0	0	1	1	1
0	1	1	1	0	1	0
1	0	1	1	0	0	1
1	0	1	1	0	0	0
1	0	1	1	1	1	1
1	0	1	0	0	1	0

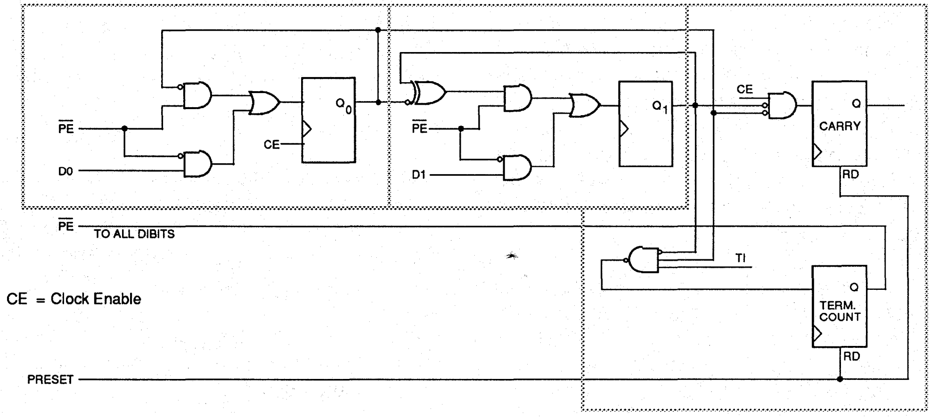
1145 02



CI = Carry In
CO = Carry Out
PE = Parallel Enable (Active Low)
TI = Terminal Count In
TO = Terminal Count Out

Any Dibit Except the Least Significant

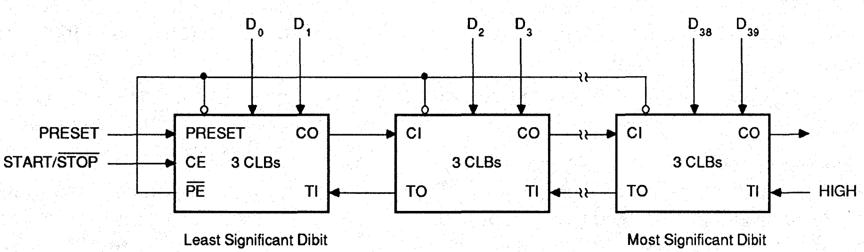
1145 04



CE = Clock Enable

Least Significant Dibit

1145 05



Synchronous Presettable Counter—40 Bits in 60 CLBs

1145 03

Since this circuit was first published in mid 1988, several designers have used it to create fast counters.

There have also been many questions about the rather terse description and about an error in the schematic drawing (the AND gate generating PE) that has been corrected in this printing.

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, 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 PE in the AND gate that detects TC, reliably de-activate TC and thus 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 dibit different?

In order to achieve a 40 MHz clock rate, the 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 dibit can be like all the other dibits, but PE must be excluded from the AND gate generating 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, however, that the flip-flop will now come alive 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 6-31).

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 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

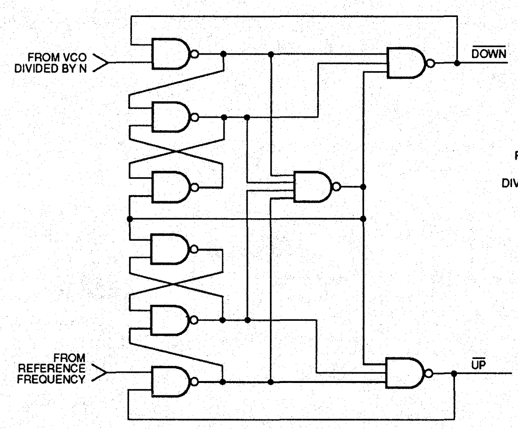
A Phase-Locked-Loop (PLL) manipulates a local voltage-controlled oscillator (VCO) so that it is in phase with a reference signal. One popular application is a programmable frequency synthesizer for radio communications. Here a crystal oscillator is divided down to a low reference frequency of 5 kHz, 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.

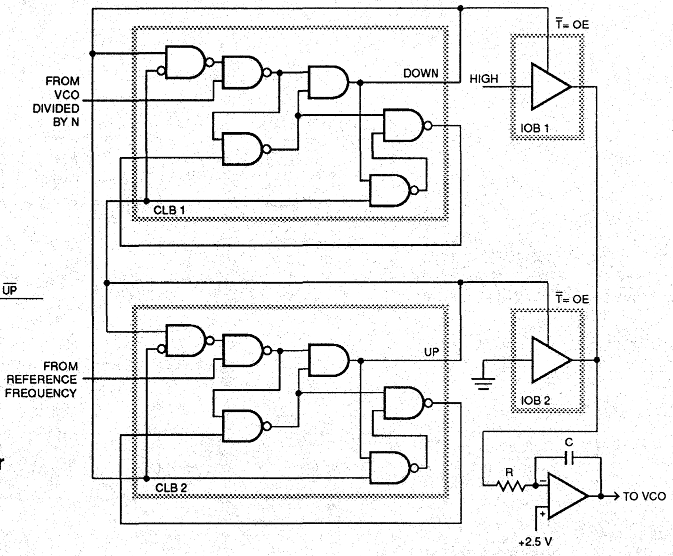
The well-known circuit shown in Figure 1 performs this function. It generates "pump-up" pulse when the VCO frequency is too low, "pump-down" when its too high. The multiple feedback network assures proper operation even at large frequency errors.

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.



1985 01

Figure 1. Digital Frequency/Phase Detector



1985 02

Figure 2. Frequency/Phase Detector Using Four Blocks

XILINX

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.

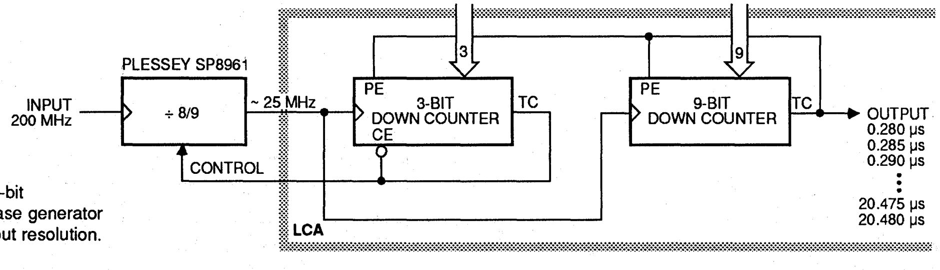
The prescaler divides by either n or n + 1, depending on the state of the control input. In other words, it "swallows" one additional clock pulse if told so by the control input. By keeping the control input active for the appropriate number of prescaler output periods, the LCA can fine tune the total 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).

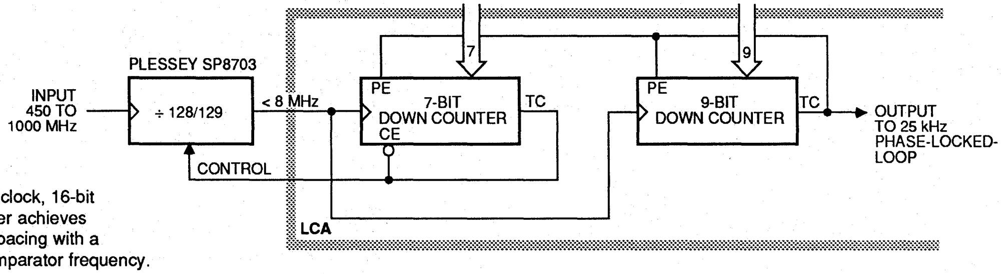
An 8/9 prescaler has blind spots below 56
A 64/65 prescaler has blind spots below 4,032
A 128/129 prescaler has blind spots below 16,256

This limitation is usually of no practical consequence in a real design.

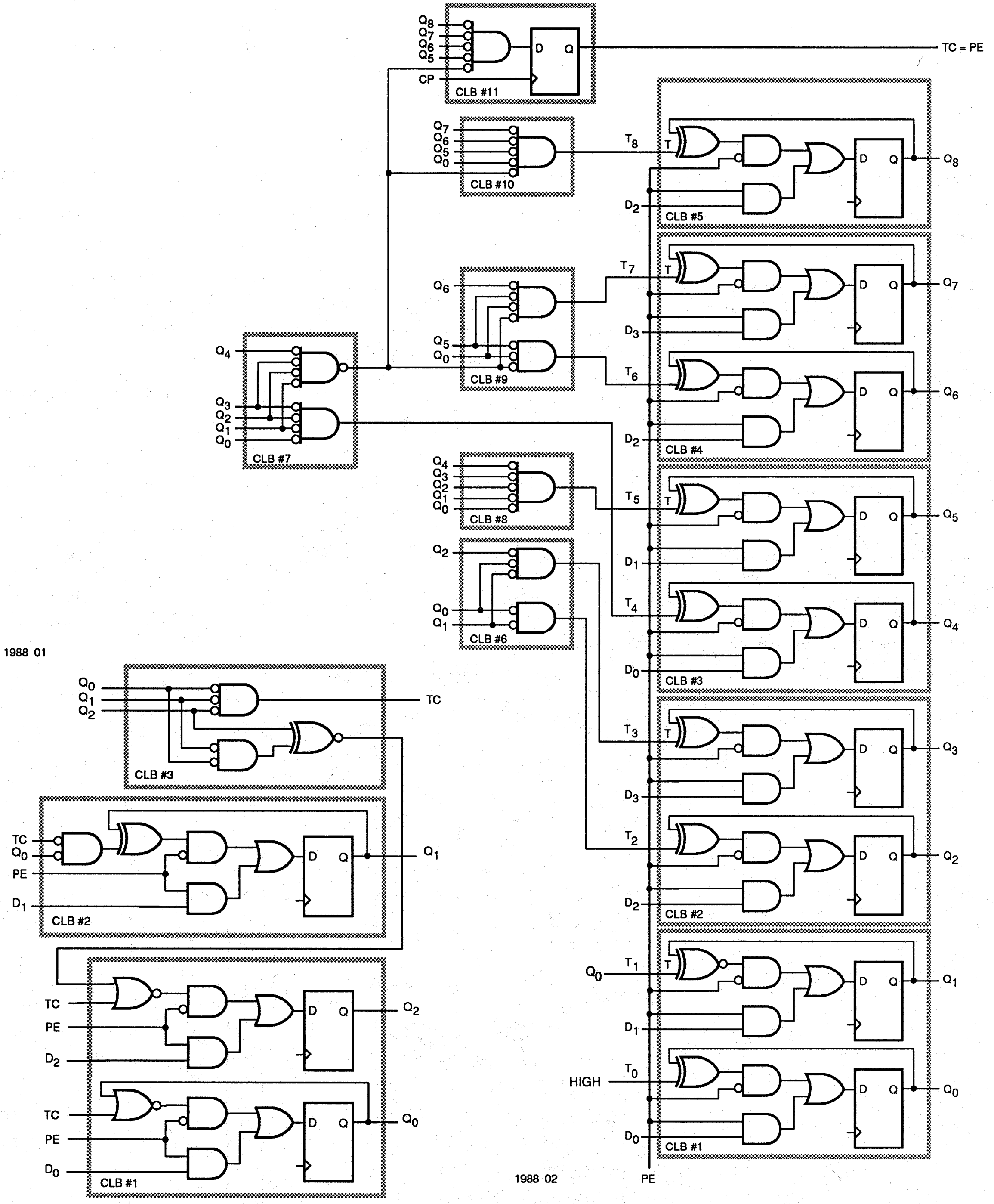
The prescaler-LCA combination can divide by **any** integer number higher than the values above.



Example 1:
200 MHz clock, 12-bit presettable time base generator achieves 5 ns output resolution.



Example 2:
450 to 1000 MHz clock, 16-bit presettable counter achieves 25 kHz channel spacing with a 25 kHz phase comparator frequency.



3-Bit Presettable Down Counter with Pipelined Terminal Count, Locking Up on TC

9-Bit Presettable Down Counter with Decoded Terminal Count (TC)

100 MHz Frequency Counter

Application Brief BY PETER ALFKE

The block diagram below describes a complete 100 MHz frequency counter in an XC3020.

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 multiplexed 8-digit, 7 segment LED or LCD display.

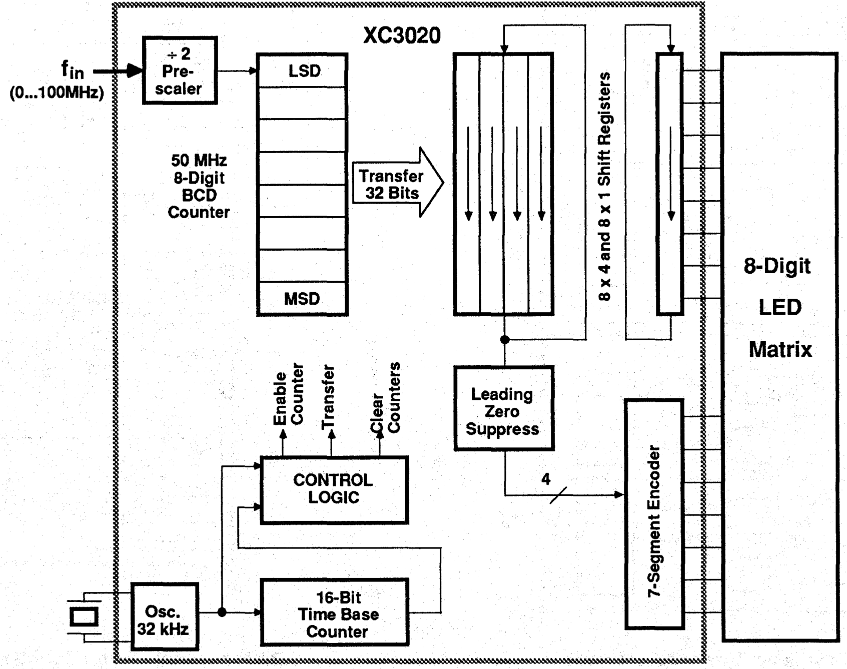
The oscillator uses three IOBs, since the dedicated crystal oscillator input is already used as signal input.

The time base is already generated by a 16 bit binary counter consisting of four asynchronously cascaded 2-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 51 of the 64 CLBs in an XC3020:

Time Base	8 CLBs
BCD Counter	16 CLBs
5 Shift Registers	20 CLBs
7-Segment Encoder	4 CLBs
Leading Zero Suppressor	1 CLB
Control	2 CLBs



1129 01

Figure 1. Block Diagram

XILINX

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.

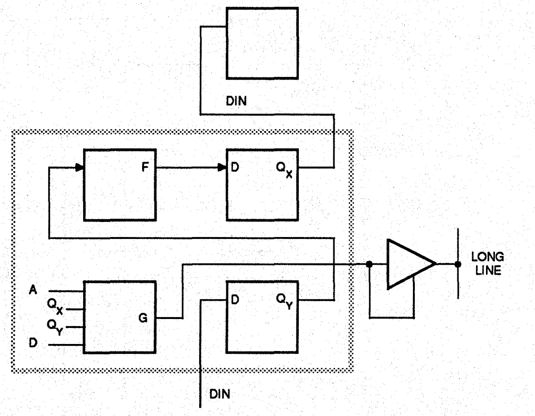
The lower half of the combinatorial array compares the content of the two flip-flops against data supplied on the A and D inputs. A match is indicated on the G output and routed to a 3-state buffer driving a long line.

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.

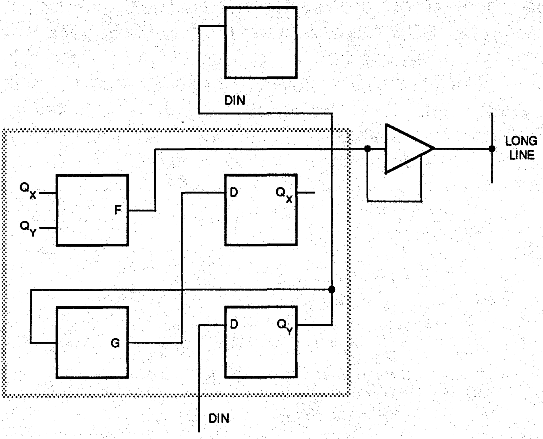
When the complete pattern has been shifted in, it is transferred with one clock pulse to the X-flip-flops, using the lower half of the function generator. Data to be detected is then shifted in through the DIN input into the Y-flip-flop, and from there to the DIN input of the next CLB. The upper half 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.



1147 01

Figure 1. Fixed Pattern Detector

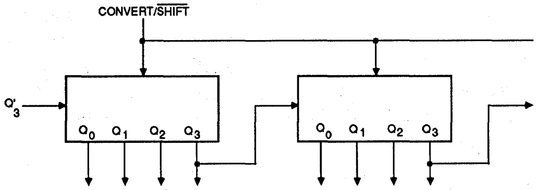


1147 02

Figure 2. Serial Comparator Finds Pattern Match or Correlates Patterns

Serial Code Conversion Binary to BCD

Application Brief BY PETER ALFKE



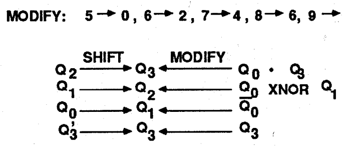
1146 03

Figure 1. Binary to BCD (MSB First)

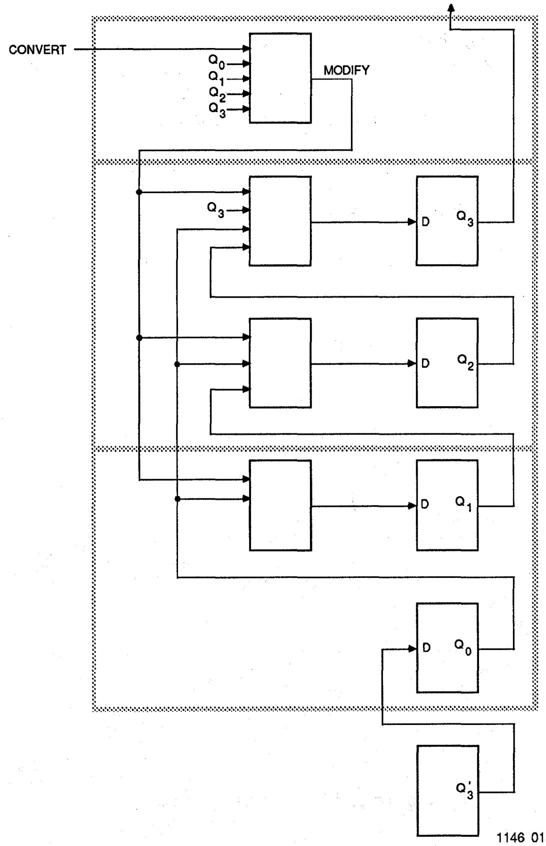
The LCA architecture with its powerful function generators evenly interspersed between flip-flops lends itself very well to serial code conversion, where data is shifted into a register in one format, and shifted out of the same register in a converted format.

A binary to BCD converter requires 3 CLBs for every 4 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.

In order to stay 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.


MODIFY: 5 → 0, 6 → 2, 7 → 4, 8 → 6, 9 → 8
SHIFT
MODIFY
$Q_2 \rightarrow Q_3 \leftarrow Q_0 \cdot Q_3$
$Q_1 \rightarrow Q_2 \leftarrow \overline{Q_0}$ XNOR Q_1
$Q_0 \rightarrow Q_1 \leftarrow \overline{Q_0}$
$Q'_3 \rightarrow Q_3 \leftarrow Q_3$

1146 01b



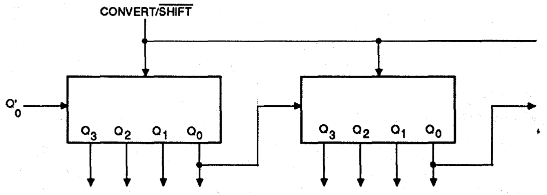
1146 01a

Figure 2. Binary to BCD converter 3 CLB's per 4 Bits: MSB First

XILINX

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 3 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.

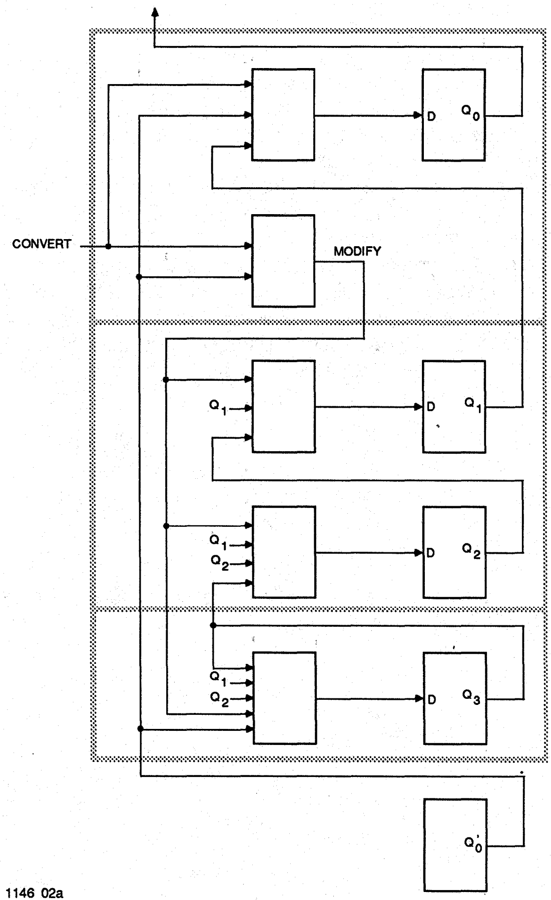
Each shift divides the content by two. When the LSB of a BCD digit is a "one", shifting it one position down would give it a weight of 8 in the lower decade instead of the weight of 5 appropriate for a 10 divided by 2. A value of 3 is therefore subtracted from the content of the decade whenever a "one" 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
$Q_1 \longrightarrow$	Q_0	$\longleftarrow \overline{Q}_1$
$Q_2 \longrightarrow$	Q_1	$\longleftarrow Q_1$ XOR Q_2
$Q_3 \longrightarrow$	Q_2	$\longleftarrow \overline{Q}_3$ AND $(\overline{Q}_1$ OR $\overline{Q}_2)$
$Q'_0 \longrightarrow$	Q_3	$\longleftarrow Q_3$ OR $(Q_1 \cdot Q_2)$

1146 02b



1146 02a

Figure 2. BCD to Binary converter 3 CLB's per 4 Bits: LSB First

XILINX

"Corner Bender" or 8-Bit Format Converter

Application Brief BY PETER ALFKE

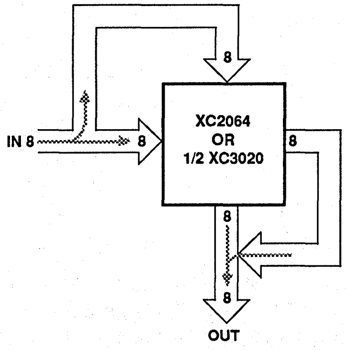
Pulse Code Modulation (PCM) has become the dominating encoding method in digital telephony. Analog signals are sampled at 8 kHz and represented by their 8-bit digital equivalent, using a logarithmic encoding scheme, μ-law in the US and Japan, A-Law in the rest of the world using the CCITT standard.

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.

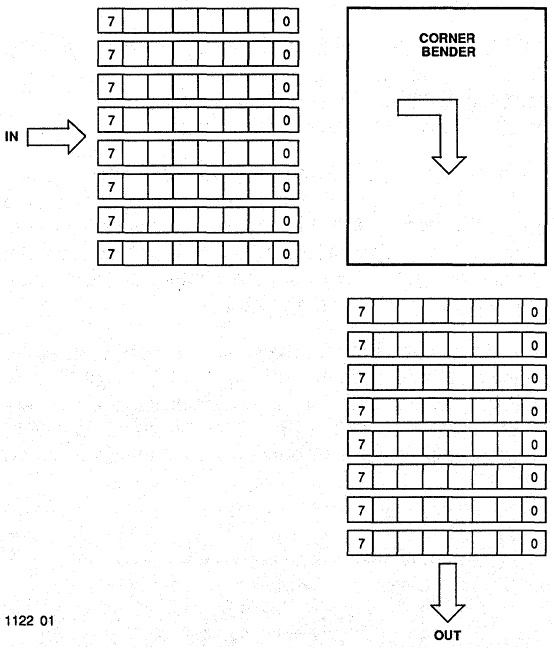
In the central office or PBX, however, the eight bits representing one particular sample must be routed together. The telephone system thus uses a large number of serial-to-parallel and parallel-to-serial converters, all operating on 8-bit words, all running synchronously. Eight S-P converters with 8 data inputs and 8 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 newserial bits can be shifted 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.

This design can be simplified and made to fit into an XC2064 or half an XC3020.



1122 03



1122 01

The LCA implementation of a 2-dimensional shift register is straight forward:

A common clock drives all flip-flops, organized in an 8x8 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 8 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.

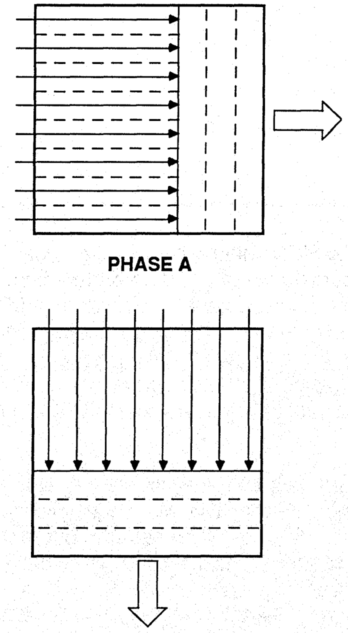
New serial data can be shifted in from one side while old parallel data is being shifted out at the opposite side. There is no need for any of the additional flip-flops required by the older designs.

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 must externally be wire-ored with the eight right-hand outputs.

The design fits exactly into one XC2064 or into half of an XC3020 and can run at up to 35 MHz.



1122 02

Megabit FIFO in Two Chips: One LCA and One DRAM

Application Brief BY PETER ALFKE

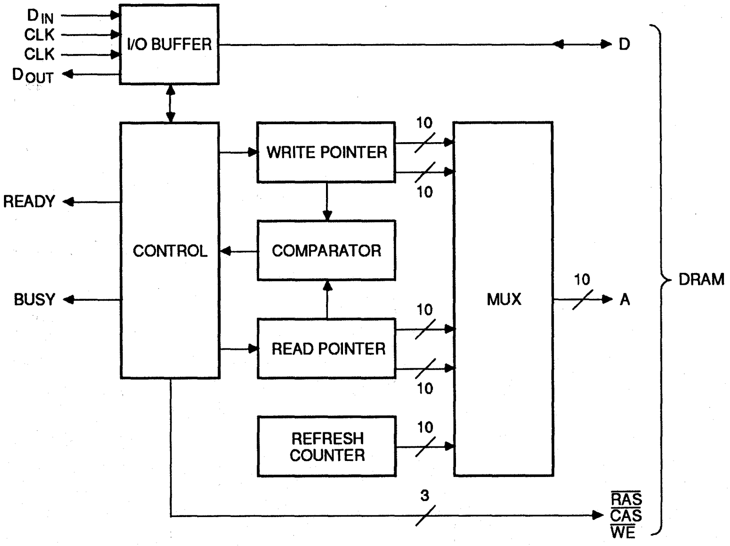
A bit serial FIFO buffer is a general-purpose tool to relieve system bottlenecks, e.g., in LANs, in communications, and in the interface between computers and peripherals. Small FIFOs are usually designed as asynchronous shift registers, but a larger FIFO with more than 256 locations is better implemented 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 bit capacity, a DRAM offers the lowest cost solution, if the controller can be implemented in a compact and cost-effective way. A Xilinx 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 10-bit refresh counter
- A 5-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 getting full or empty.



1130 01

Figure 1. Megabit FIFO Controller in an XC3020

When write and read pointer become identical as a result of a write operation, then 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, then 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 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. The loss of one memory location is more than compensated by the two bits (one incoming, one outgoing) stored in the controller.

This design fits two shift register counter bits in one 3000-series CLB, 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).

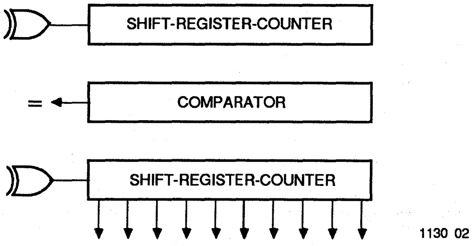


Figure 2. Shift-Register-Counter and Free Row-Column MUX

Both 20-bit pointers, plus their 20-bit identity comparator, plus the Row/Column multiplexer thus fit into only 20 CLBs; refresh timer and refresh address counter and multiplexer use another 15 CLBs and the data buffer plus control and arbitration logic might take another 15 CLBs, for a total of 50 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, or 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 Megabytes per second byte-parallel.

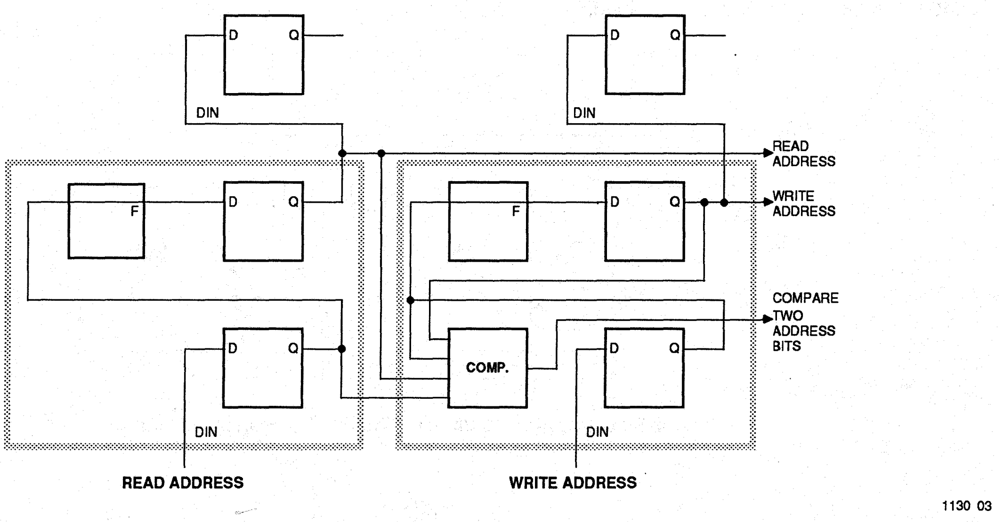


Figure 3. 2-Bit Slice of Two Counters and Comparator in Two CLBs

XILINX

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. Xilinx LCAs 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 5 registered CLB outputs drive the 5 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 6-23.

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

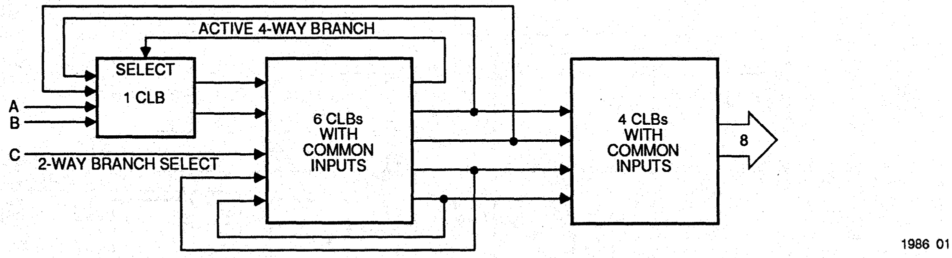
This simple state machine uses only eleven CLBs. It has up to 16 states, and eight outputs, each decoding/encoding any combination of states. It performs a 2-way branch from any state to any one of two freely assigned states, (possibly including the present state) determined by control input C. (Avoid 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



30 MHz State Machine, 16 States, 2-Way/8-Way Branch, 8 Outputs

XILINX

Complex State Machine in One LCA

Application Brief BY PETER ALFKE

Simple and fast state machines can easily be implemented in a Xilinx 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 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 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.

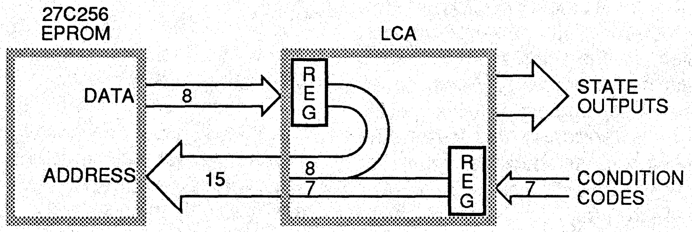
This rudimentary state machine can thus have 240 different states, and can jump from any state to any one of 128 arbitrarily defined next states, controlled by the 7 bit condition code.

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. For most states, almost all of the 128 possible next states will be programmed to be identical.

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 as low as 100ns. For higher speed- at a higher cost- the EPROM can be shadowed by fast SRAMS.



1987 01

Programmable Gate Arrays and Self-Diagnosing Hardware

Application Brief BY RICHARD B. RAVEL

SELF-DIAGNOSING HARDWARE

Most designers would agree that it is desirable to incorporate self-diagnostics into their circuit boards, and they would do it more often if the cost of additional components and board space were acceptable. It is obviously best to consider diagnostics at the beginning of a project so that the board is designed with testability in mind. This not only makes a board more manufacturable, but it makes it easier to find failures in the end-user environment.

Programmable Gate Arrays are used in many different applications, and have the unique capability of having their specific functions defined by the systems in which they reside. Xilinx Logic Cell Arrays (LCAs) can also be reprogrammed, in-system, as many times as necessary. This ability to dynamically re-configure the logic of the LCA makes board-level self-diagnostics a practical goal. An LCA can perform diagnostic functions at power-up or in test modes, and perform normal functions when the board is determined to be operational. (See the application note on "Configuring Xilinx Logic Cell Arrays" in the Programmable Gate Array User's Guide.) This approach to diagnostics based on reprogrammable gate arrays adds no additional cost to the circuit board.

This concept is really not new. Board-level self-diagnostics became popular with the advent of microprocessors. A special diagnostic program written for the microprocessor and stored in its normal EPROM could be invoked at power-up, by the press of a button, or by a special command. This approach adds little cost to the system because it requires only a small amount of EPROM storage. For example, when a PC is initially powered-up, all of the system RAM is tested. Further initialization of the PC will not take place unless this memory is 100% functional. If there is a memory failure, it can be isolated to the specific IC. The LCA allows an extension of this idea. The microprocessor will still have some special programs for diagnostics, but now the diagnostics can extend well beyond the immediate reach of the microprocessor, without adding circuitry just for this purpose.

DIAGNOSTIC HARDWARE

The circuitry which is most easily diagnosable is that which is immediately accessible by the microprocessor. In many cases, however, some of this circuitry cannot be tested directly due to the specific design. Testing this logic, as well as other unrelated circuitry, requires additional logic on the board specifically for the purpose of diagnostics. This is typically not done because of board space and cost considerations.

The logic which typically surrounds a microprocessor is the I/O control, memory control, bus control, and interrupt control logic. The peripheral control logic is the most difficult to diagnose using the microprocessor. I/O control functions are usually implemented with dedicated peripheral controller chips, since they are cost effective and readily available. If they do not give the designer enough flexibility to perform all of the required functions, logic is added to the board. Some peripheral controller chips allow the designer to include diagnostic readback firmware. This firmware, however, would usually not include access to the supplemental circuitry which might be required. Even when this readback capability is available, its scope is limited.

Peripheral control logic is usually diagnosable by writing special programs for the microprocessor, and adding special circuitry that the microprocessor can access specifically for this purpose. Adding special logic for this purpose is certainly not desirable and, with the use of Logic Cell Arrays, not necessary.

The Logic Cell Arrays can perform many different peripheral control functions, and can also be the primary interface between a microprocessor and its peripherals. In these microprocessor designs, as previously mentioned, there would usually be circuitry on the board to which the microprocessor may or may not have immediate access. The LCA, as the bus and I/O controller for example, would easily be able to access this logic as an extension of the microprocessor.

SELF-TEST TECHNIQUES

LCAs can be used to implement hardware diagnostics. When the board is initially powered-up, the Logic Cell Array can be programmed with a special diagnostic configuration. The LCA can then be used in conjunction with the microprocessor to test the peripheral circuitry. This LCA configuration can include the ability to communicate status information about the peripherals and other circuitry to the microprocessor that might otherwise require additional logic.

The remainder of this paper will focus on several examples of how the Logic Cell Array can be used to perform board-level self-diagnostics.

LOOPBACK

Many designs have special drivers and receivers on a board to which there is no direct access. A special loopback test connector is usually needed to test these drivers and receivers. This connector must be installed before and removed after testing. This usually means that this test is only performed as a last resort, when all other tests have failed to find a fault. The Logic Cell Array's user-configurable interconnections allow the drivers and receivers to be tested without any additional test connectors or manual intervention. To fully test this circuitry, it is only necessary to connect traces on the printed circuit board from the drivers and receivers to I/O pins on the Programmable Gate Array. This allows the LCA, with a diagnostic configuration loaded, to drive the receivers and to read the data from the drivers without the use of a loopback test connector. Refer to Figure 1.

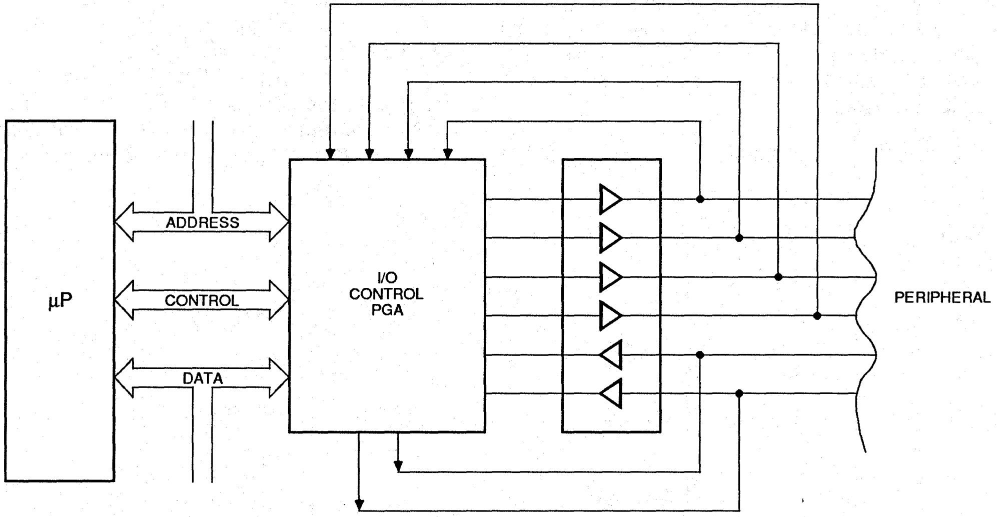


Figure 1. Diagram of LCA Used in LOOPBACK Testing

TESTING I/O AND MEMORY ERROR DETECTION CIRCUITRY

A microprocessor can use the LCA to drive the peripherals and additional supporting logic in non-standard ways. This is often valuable in diagnosing circuitry. For example, during normal operation of a serial communications channel, it is not possible to force an error in the transmission of the data. This can easily be done, however, when an LCA is used as an I/O controller. It can be programmed with a special diagnostic configuration that can force parity errors, overrun errors, and CRC/checksum errors in the data stream which should be caught by the error detection circuitry. This, along with special diagnostic firmware for a microprocessor, allows full testing of serial (or parallel) communications channels.

This same concept can be applied to testing standard memory when the LCA is performing memory control functions or has write access to the memory. With a diagnostic configuration in the LCA, it can force data into the memory with incorrect parity or check bits. The normal circuitry and firmware for reading the memory should then detect the errors, and the operation of the error correcting logic can be verified.

As soon as the microprocessor has determined that the board is fully functional, it can re-program the LCAs for their normal functions and the board can begin normal operation.

INTERRUPT VECTORS

Interrupt circuitry is also difficult to test, as there is not always a straight-forward method of generating all of the interrupt requests. With a Logic Cell Array as the interrupt controller in the system, it can be configured with a special test configuration that can sequence through all of the different interrupts. It can even generate multiple interrupts in sequences that test interrupt prioritization logic. Refer to Figure 2.

TIMEOUT INTERRUPTS

Some systems include timeout interrupts such as watchdog timers that do not occur in normal operation. This type of timeout interrupt is difficult to generate during normal operation. Again, using a Programmable Gate Array as an interrupt controller allows the designer to easily force this type of interrupt and verify that the detection circuitry and error recovery routines are functioning correctly.

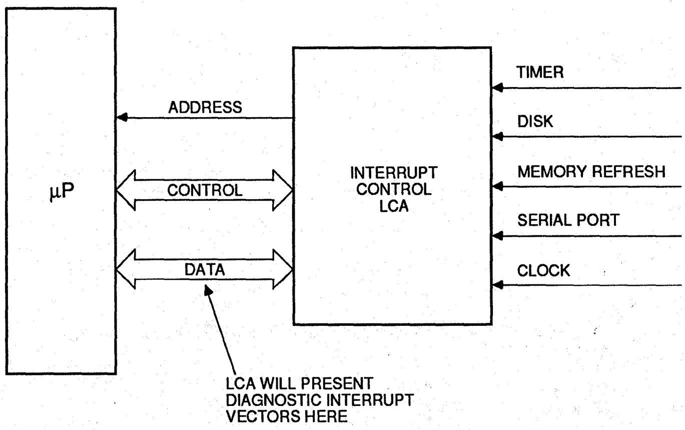


Figure 2. LCA Used as an Interrupt Controller

1131 02

XILINX

PS/2 Micro Channel Interface

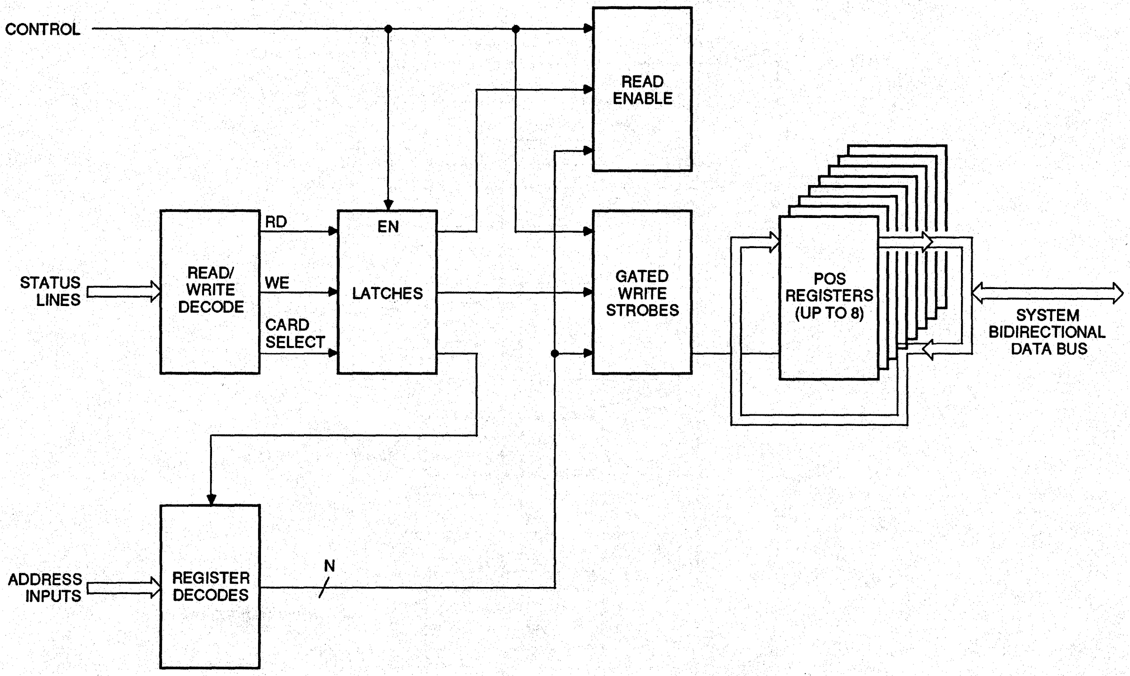
Application Brief BY ROB STRANSKY

IBM's new 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 new and 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 Programmable Gate 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.



1119 01

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 Programmable Gate Array 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 PGA 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 Programmable Gate Arrays.

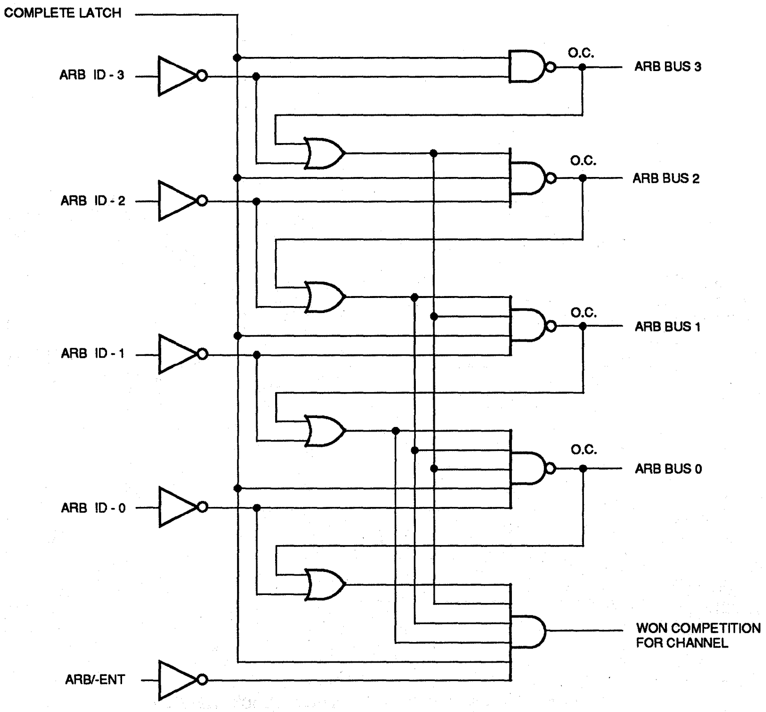


Figure 2. Local Arbiter Logic 1119 02

XILINX

High-Speed Bar Code Reader Interface

Application Brief BY DAVE LAUTZENHEISER

Bar code readers have become familiar to the average consumer due to their widespread use in point-of-sale systems, such as those used in most grocery stores. Additionally, many industrial applications are now using bar codes to identify materials in various phases of manufacturing, inventory and distribution. Bar codes can be automatically read and processed to provide for computerized control and optimization of virtually all phases of a manufacturing process. In many industrial applications, the performance requirements of the bar code system are significantly higher than those of the point-of-sale system.

As an example, consider an industrial application where material on a conveyor is being scanned to read bar code data. If the conveyor is moving at 5 meters/second, and a bar code label with 30 bars is 25 mm long, that label passes a point scanner in 5 milliseconds, or 166 microseconds per bar. If the minimum bar width is 12% of the average time period, it must be detected and its width determined in 20 microseconds. For a maximum bar width five times the minimum, the time period is 100 microseconds. If a multiple scan system is being used, the active period for a bar is reduced proportional to the scan rate.

For this type of scanning system, some pre-processing of the scan data is required to insure that a processor can decode the label into code numbers and process them accordingly. In addition, systems may be required to operate with a variety of different bar code scanners or sensors for different objects or labels, at different scan rates. Traditionally, different logic and micro computer coding dedicated to each application was designed.

Programmable Gate Arrays provide a method of implementing the scanner specific interface logic in a cost effective, compact manner. At the same time, system design flexibility is provided while meeting the performance constraints of the system. Designers of bar code processing systems can design a single interface card that utilizes the programmability of the PGA to define the logic for a specific scanner in software as part of the system initialization.

The block diagram in Figure 1 shows the functions performed in the PGA. For each type of sensor, the specific controls and data signals have different functions and timing relationships. In addition, the amount of data can vary depending on the specific application. In different scanner interfaces, the counter control logic, edge detect circuitry and the operation of the black area and white area counters can be modified to fit the needs of the particular scanner. The bus control logic and the data loading into the interface memory remain virtually unchanged, providing a consistent interface to the processor.

Implementation of the interface logic can be accomplished with either an XC2000 or XC3000 family device. Because there are processor and memory data and address bus structures, the XC3000 family would provide a simpler solution. Even with a master clock rate of 10 or 15 MHz, the 16-bit counters for the black and white areas are easily implemented as up counters with reset capability. Edge detection is performed with either simple gating, or synchronization to the master clock. Memory FIFO control also involves counters, but these operate at a much slower rate. A new word is written to memory only after a bar in the target has been passed.

At the completion of a scan an End Of Label is signalled by forcing a data entry in the RAM of FF hex. An interrupt to the processor can be generated based on the End Of Label or End Of Scan condition. The microprocessor then normalizes the absolute timing information for black and white bars stored in the FIFO in order to extract the encoded data. This allows for variations in scan speed, scan angle, etc.

This type of interface represents only one of several methods that can be used to pass the information to the host processor. For other types of systems, different techniques may be used. Regardless of the technique, the flexibility of the PGA provides significant advantages over a fixed logic solution.

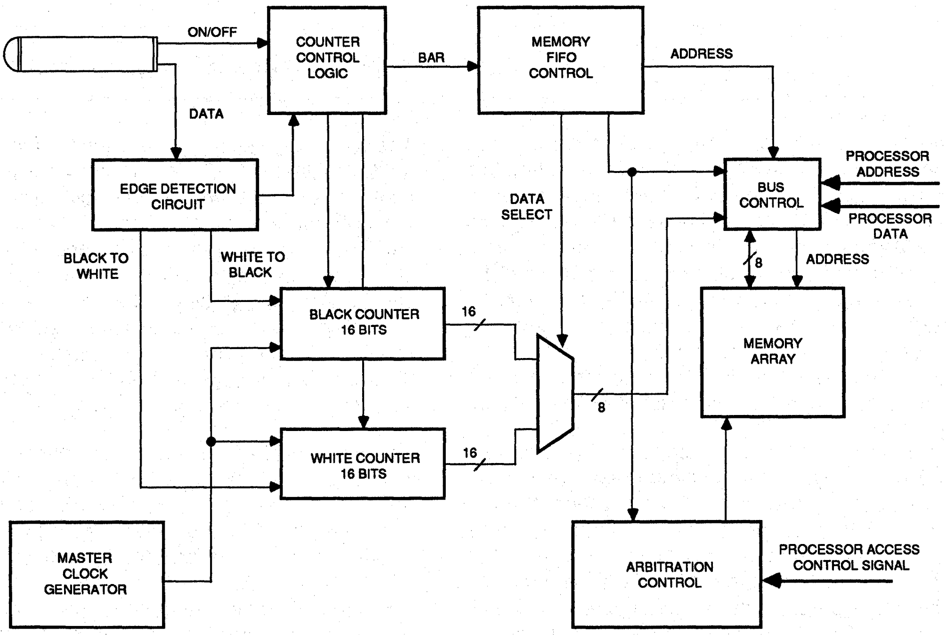


Figure 1. Bar Code Reader Interface

XILINX

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. Non-recurring 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 user programmable logic have created a third option. Xilinx's 3000 family of programmable gate arrays brings unprecedented density to programmable logic, with devices containing as many 9000 usable gates. The architecture of the 3000 family devices make them particularly well-suited to memory controller applications.

WHY IMPLEMENT A DRAM CONTROLLER WITH A PROGRAMMABLE GATE ARRAY?

There are several reasons why one would want to design a DRAM controller with a Xilinx Logic Cell™ Array. First, the true programmability of the LCA 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. 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 MB of memory comprised of forty-four 256 KB DRAM chips: thirty-two for data and twelve for the correction bits. A single LCA 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 (IOB).

DESIGN OVERVIEW

The DRAM Controller/ECC uses a 16 MHz clock synchronized with the processor's 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 (S2, S1, S0) 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.

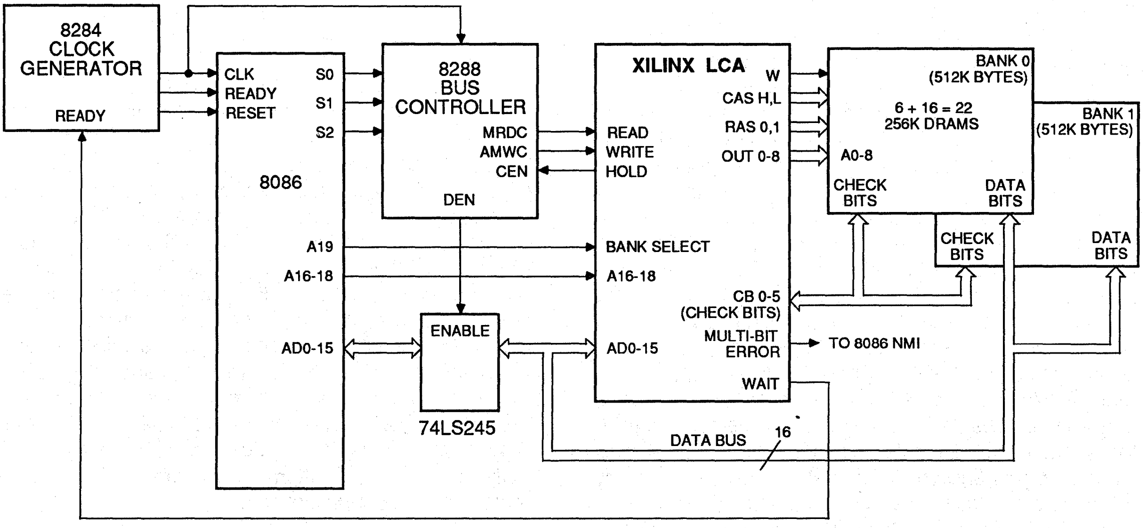


Figure 1. System Overview
System overview of DRAM Controller with error correction and detection.

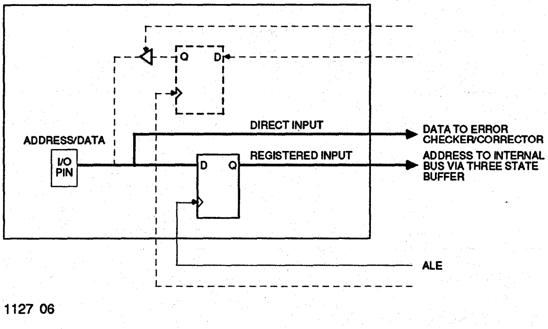
SYSTEM TIMING

Figures 3a–3c show the timing involved in some of the different memory cycles. The Word Write (Figure 3a) requires no wait states as shown. The check bits from the ECC are written to memory along with the data. The Read cycle (Figures 3b & 3c) 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 allow time to correct the error. The insertion of the two additional wait states effects 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

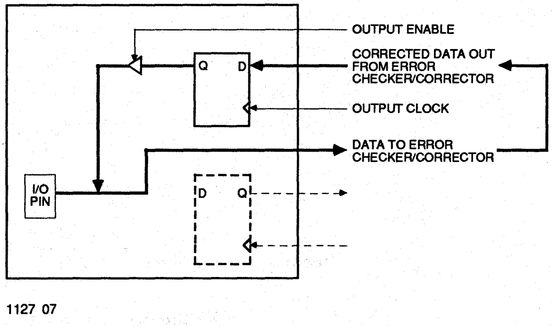
The 3000 family LCA architecture has a number of features that are essential to the DRAM controller design. The first such feature is the dual data input paths in the IOBs, one registered and one direct. This structure permits the address and data on a multiplexed bus to be latched from the same I/O pin. Figure 4 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's 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 5 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 three-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 three-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 three-state buffer. This allows the corrected data to be read by the 8086 at the same time it is being written back to the DRAM.



Q
D
ADDRESS/DATA
I/O PIN
DIRECT INPUT
REGISTERED INPUT
D
Q
DATA TO ERROR CHECKER/CORRECTOR
ADDRESS TO INTERNAL BUS VIA THREE STATE BUFFER
ALE
1127 06

Figure 4. 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.



OUTPUT ENABLE
Q
D
CORRECTED DATA OUT FROM ERROR CHECKER/CORRECTOR
OUTPUT CLOCK
DATA TO ERROR CHECKER/CORRECTOR
I/O PIN
D
Q
1127 07

Figure 5. Data in and out through ECC
Data Flow through the 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.

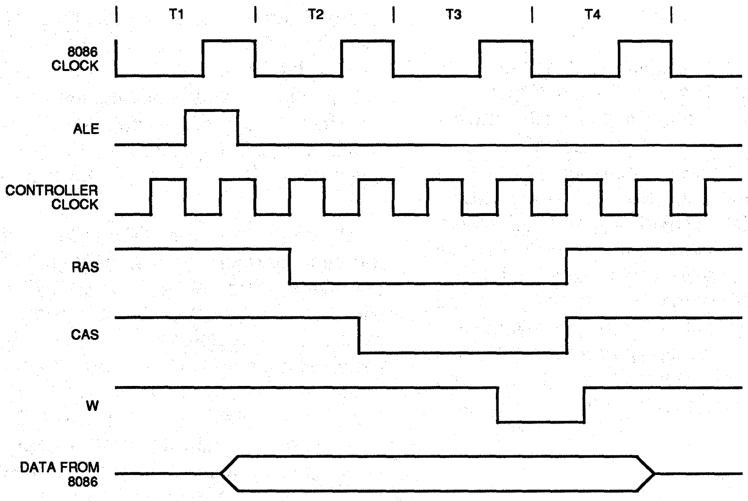


Figure 3a. Word Write Timing

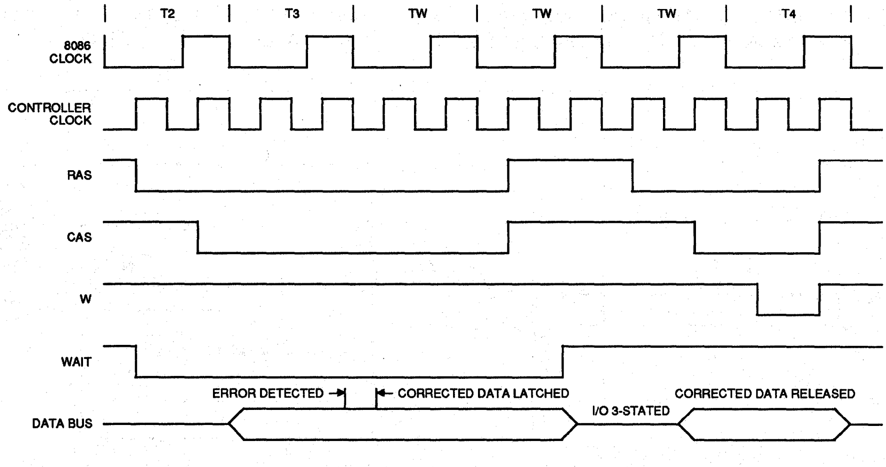


Figure 3b. Word Read Timing with Errors Detected

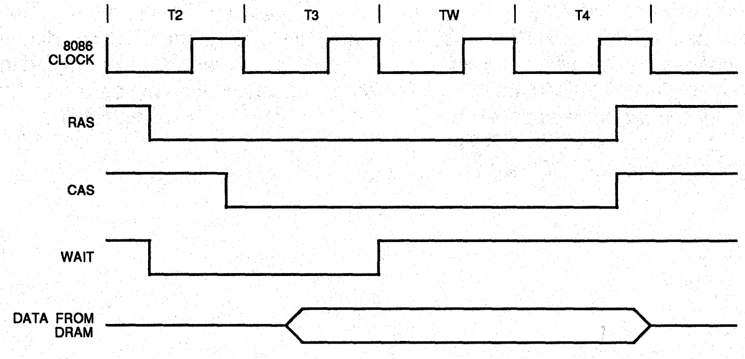


Figure 3c Word Read Timing with No Errors Detected

Figure 2 is a block level diagram of the DRAM Controller and ECC that reside in the LCA. A functional description of each block follows:

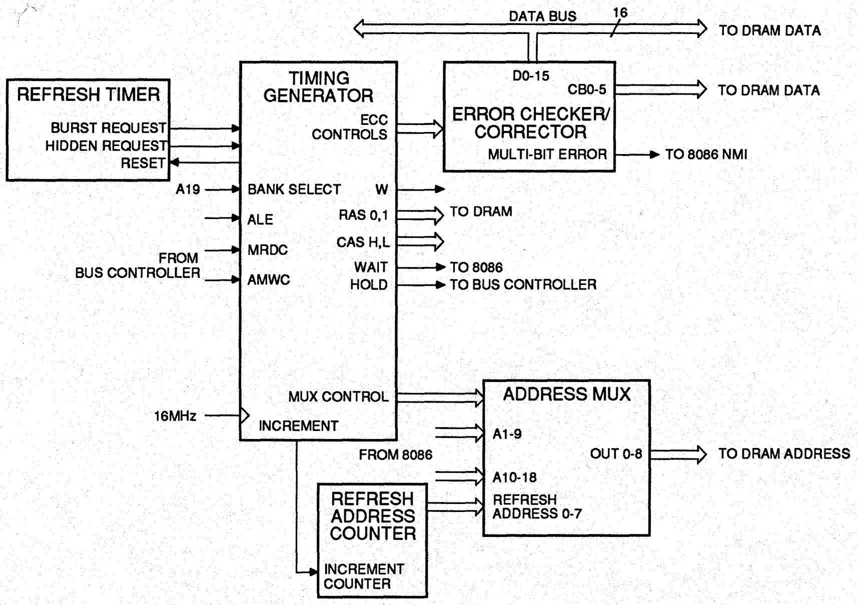
The *refresh timer* is driven by the 16 MHz clock to provide a signal that tells the DRAM controller that the memory needs refreshing. Each of the 256 rows of memory in this system must be refreshed every 4 ms. The controller attempts to refresh eight rows every 125 µsec, so that all 256 rows are refreshed 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 µsec 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 in order 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 eight bit counter that provides the eight 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 sixteen 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.



1127 02

Figure 2. 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 tradeoffs 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 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.

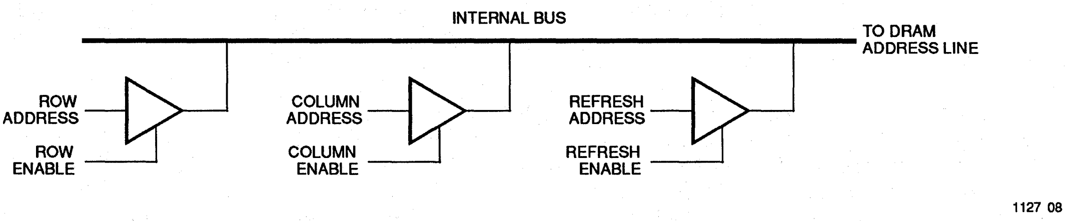


Figure 6. Address Multiplexing Using Three-State Enables onto Internal Buses

XILINX

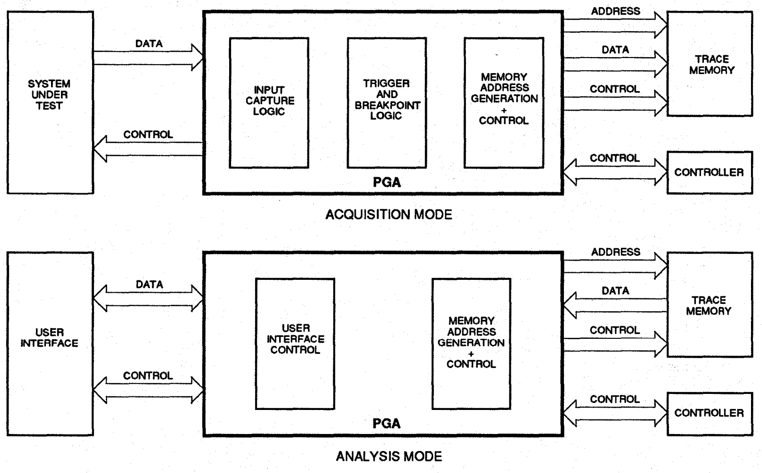
Logic Analyzer/In-Circuit Emulator

Application Brief BY BRAD FAWCETT

Logic analyzers and in-circuit emulators are similar types of electronic test equipment. Each involves the monitoring of certain digital signals within the system being tested. For general-purpose logic analyzers, these signals can be any nodes on the board being tested that the user selects. Another category of logic analyzers, sometimes called bus analyzers, are designed to plug into a specific microcomputer bus (such as a PC-bus or Multibus) and monitor and control the activity on that bus. In-circuit emulation of a specific component (usually a microprocessor) involves monitoring and controlling the signals input and output by the device being emulated.

A record of the activity on the nodes being tested by the analyzer is stored in a memory buffer called "trace memory"; this activity is called "acquisition mode." In some cases, the analyzer might also be controlling some of the target system's functions, such as single stepping a clock or interrupting a processor. After tracing is completed, the trace memory is then read and displayed by control logic in the analyzer; this is the "analysis mode." Trigger and breakpoint logic is integral to this operation; the user specifies when the tracing of signals in the target system should begin (triggering) and end (the breakpoint). Hence, a description of the combination of signals that make up the triggers and breakpoints must be stored in the analyzer before beginning operation, and the state of the target system must be continuously monitored to determine when a trigger or breakpoint occurs. Triggers and breakpoints might be defined as a single event in the target system, or some ordered sequence of multiple events. Usually, trigger and breakpoint conditions are simply stored in registers and then continuously compared to the target system signals that are being monitored. Typically, logic analyzers and in-circuit emulators use many SSI/MSI latches and comparators to perform these functions.

The large number of logic functions and registers available in the Xilinx Programmable Gate Arrays make them ideal for implementing these trigger and breakpoint functions within a single device. Furthermore, the interface logic for controlling the trace memory can also be integrated into the PGA. This application can take advantage of the re-configurable nature of the PGA architecture. (Effectively, the Programmable Gate Array can perform different functions within the same system at different times by loading different configuration programs. This can lead to fewer packages on the printed circuit board and increased reliability.) One configuration of the PGA could be used for acquisition mode operation, capturing data, searching for the triggers and breakpoints, and filling the trace memory. When the breakpoint is reached, a different configuration could be loaded into the same PGA for the analysis mode; the PGA now controls the reading of the captured data from the trace memory. Additional functions, such as control of the user interface, also could be incorporated into the PGA.



1128 01

Figure 1. Logic Analyzer/In-Circuit Emulator

XILINX

The Programmable Gate Array Company

Electronic Design
EXECUTIVE VIEWPOINT
ASIC
Controller design
Buildi
disk
EDN
Programmable array serves as a controller for dynamic RAMs
Technology Update: STANDARD & PROGRAMMABLE LOGIC
tural Subtleties Make The Difference In
ction Of A Programmable Logic Device
Electronic Engineering TIMES
Circuits In Systems
Using Program
In a Sate
Logic Cell Arrays
on
DESIGN ENTRY
Printer buffer proves
sed logic's
d versatility
Technology Update: CUSTOM/SEMICUSTOM ICs
Logic Cell Arrays: RAM-Programmable Logic Devices
Architectural Overview
ESD
Xilinx makes hardware
San Fra
XILINX

Article Reprints

Article Reprints

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

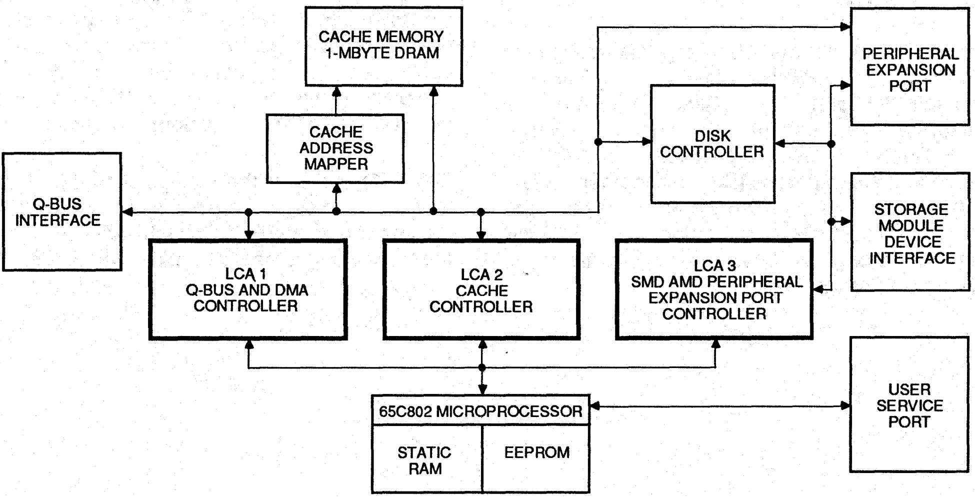
Computer manufacturers historically have relied on advances in CPU and semiconductor memory technology for increasing system throughput. At the same time, they accepted as inevitable the hardware-bound I/O bottleneck. This position is becoming untenable with recent advances in magnetic disk technologies, 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.

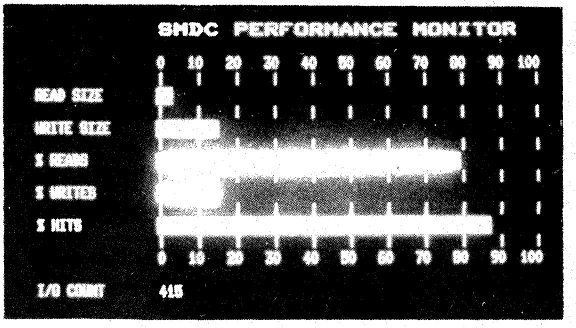


Figure 2. The user service port can create color bar graphs that dynamically show various attributes of the data cache, such as read times, forward block reads, and I/O completion rates.

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.

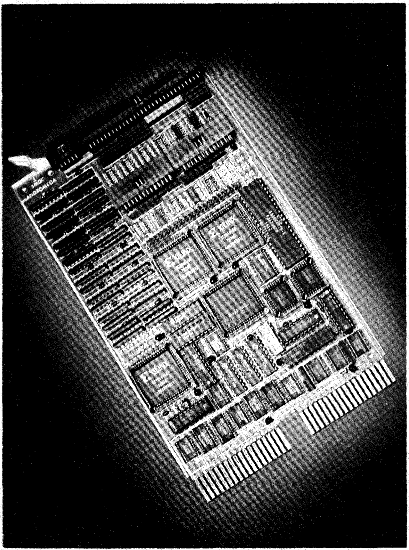
The SMDC achieves more performance and flexibility than did previous generations of disk controllers. It includes data caching, high data-transfer rates, a peripheral expansion port, field-loadable microcode, and a user service port. State-of-the-art VLSI components and packaging techniques fit the entire 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. On-board 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.

$$$$$

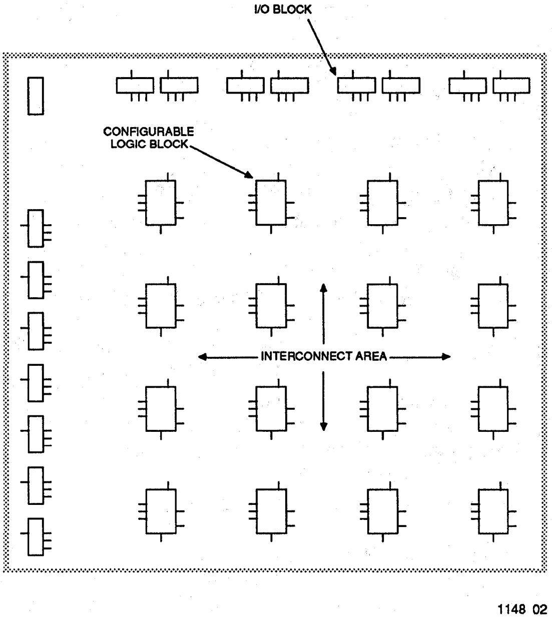
Andromeda Systems' Storage Module Device Controller is available now for $2,195. (The company's ESDI controller is available for $1,995.) For more information, call Don Talmadge at 818-709-7600, or circle 336 for the SMDC and 337 for the ESDC.

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.



Reprinted with permission of *Electronic Products.*

Electronic Engineering
TIMES

The Acid Test

By Steven K. Knapp, Field Applications Engineer, Xilinx Inc.

Using an EPLD vendor's application, the same circuit was built with a Logic Cell Array. The comparison provides real insight to the differences between these two technologies.

As with a processor, a PLD's architecture determines its functional logic density and its performance.

Among processors, general-purpose microprocessors and application-oriented devices like digital signal processors share many similar internal logic structures and functions. DSPs have a more rigid architecture, designed for particular applications, while general-purpose microprocessors address a much wider set of applications. The same situation exists in the world of PLDs.

Until recently, the sum-of-products (SOP, also known as AND-OR) structure was the only one available for PLDs. Its simple, fixed architecture fits a variety of low-density, high-speed applications. These include fast, wide logic functions found in decoders, multiplexers and counters.

Some recently introduced high-density PLDs are based on extensions of the conventional AND-OR architecture. Examples include the MMI 64R32 MegaPAL and the Altera EP1800 Erasable Programmable Logic Device (EPLD).

Like a DSP processor, the sum-of-products architecture of these devices efficiently meets certain needs. However, this architecture is not suitable for higher-density, register-intensive random logic structures commonly found in logic designs.

LOGIC CELL ARRAY

The Programmable Gate Array differs vastly from the conventional sum-of-products architecture. Its flexible, register- and I/O-rich, array-style architecture addresses a wider set of common logic design problems.

Its Logic Cell Array architecture consists of three basic programmable elements: input/output blocks, configurable logic blocks and programmable interconnects.

Each I/O block can be individually configured as a direct or registered input, a direct or three-state output or as a bidirectional I/O.

The configurable logic blocks contain combinatorial logic plus storage elements. The combinatorial logic within each block implements any possible single function of four variables, or any two functions of up to three variables. The storage element is configurable as an edge-triggered flip-flop, or as a level-sensitive latch, both with asynchronous SET and RESET inputs. The programmable interconnect allows each of the storage elements to be clocked either synchronously or asynchronously.

Three levels of programmable interconnect resources provide the interconnection between blocks:

- Direct interconnect allows fast connections between adjacent blocks.

- General-purpose signals travel through an array of switching matrices that yield an efficient means of connecting scattered random logic.

- Long metal lines that traverse the chip distribute clock or other signals with high fan-out or requirements for minimum skew.

The best way to illustrate the difference between the Logic Cell Array and the more conventional AND-OR architectures is through a design example.

An X-Y position controller design was originally developed by Altera as an application example for its EP1800. The design is fairly simple, and illustrates the capability of programmable logic to address the problems faced by logic designers. The example can be used to demonstrate the capabilities of both the Altera EPLD, and the Xilinx Logic Cell Array.

X-Y position controllers are employed in a variety of design applications to control motors for printers, plotters, robotics, and numerical controllers. The controller compares the desired location loaded from an external processor with the present motor position stored within the PLD. Based on the results of the comparison, the controller drives two four-phase stepper motors (an X- and a Y-position motor) to its desired position.

In this design, X- and Y-position data is loaded into the device from an external microprocessor. The 1,800-gate EPLD has no input storage elements, so the data must be held valid by some external device until both motors reach their final position. Since the Logic Cell Array has input flip-flops which hold the X- and the Y-position data, it offers a simpler interface to the processor.

The desired position data is compared against the present position data. The result of the comparison drives the 7-bit up/down counters which, in turn, drive the state-machine motor control. The stepper motors used in the application have 7.5 degree fullstep increments with optional 3.75 degree half-step increments available. A 7-bit binary up-down counter is required to keep track of the 96 motor steps required to rotate each motor a full 360 degrees.

In the EPLD designs, seven macrocells are used to implement the binary up/down counter for each half of the position controller. Because of the low-frequency clock (500 kHz), this same counter requires only seven Logic Cell Array logic blocks. If this were a high-speed counter, a few additional blocks would perform some level of carry-look-ahead.

If the desired position is greater than the present position, the state-machine-based motor control drives the stepper motor clockwise. If the position is less, the controller drives the motor counter-clockwise.

Four EPLD macrocells are required to implement the state-machine for each half of the position controller, while seven Logic Cell Array logic blocks are required to implement the same function.

To signal the processor that the entire operation is complete, an open drain interrupt signal is generated when both the X- and the Y-position motor have reached their final location. This also signals the processor that further motor action may be taken. A master RESET signal resets the present position to zero.

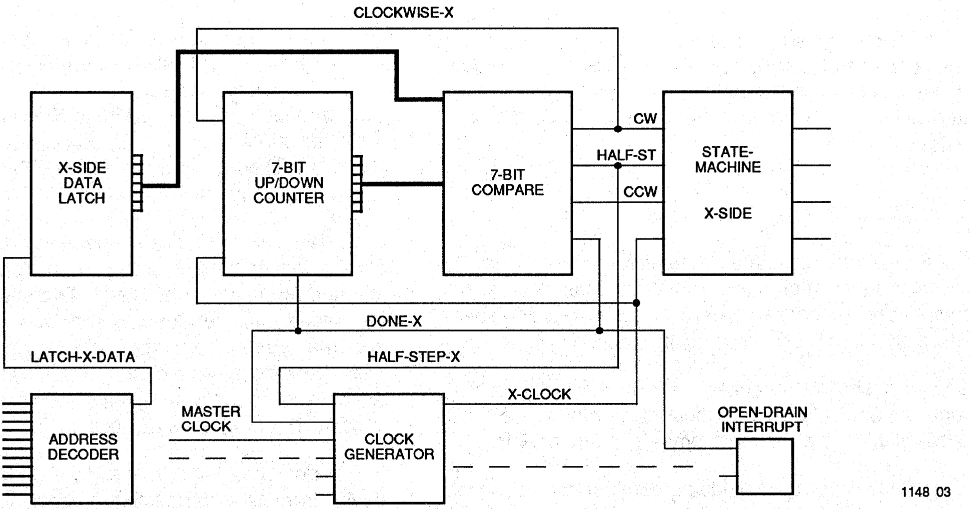
RESOURCE REQUIREMENTS

Based on the requirements of the design, the X-Y position controller example requires 47 of the 48 macrocells in the 1,800-gate EPLD and all of its 64 I/O pins. A number of the I/O pins are used simply because a register is hard-connected to the pin. The two 7-bit counter, for example, use 14 I/O pins because they cannot be buried. The X-Y controller design uses 98 percent of the macrocells and 100 percent of its I/O pins.

The same design (plus the input data registers) fits in 49 configurable logic blocks and 27 input/output blocks in a Logic Cell Array (the MASTER RESET input uses the Logic Cell Array's dedicated master reset pin). Fewer I/O pins are required, since the 7-bit up/down counters and the state-machines were buried in the Logic Cell Array design.

The X-Y controller design would occupy 77 percent of the 1,200-gate XC2064's logic blocks and 47 percent of its I/O pins. Or if the 1,800-gate Logic Cell Array were used, the X-Y controller would occupy just 49 percent of the logic blocks and 36 percent of its I/O. The remaining logic and I/O blocks can be used to build chip select logic to make a clean interface to the microprocessor.

The architectural differences between the two technologies allow a design which requires an entire 1,800-gate EPLD to fit in just a portion of a 1,200-gate Logic Cell Array. The EPLD can perform any one of the required tasks quite



The schematic above illustrates the X-axis half of the position controller used as the design example. The other half is identical to this circuit, and drives the Y axis instead of the X.

well. The sum-of-products architecture allows EPLDs to implement fast counter, decoders, and multiplexers in a single pass. Combining these elements into a larger, more complex system with additional passes through the array however, hurts both performance and density.

The flexible Logic Cell Array architecture allows entire complex systems to be implemented efficiently and quickly. Given the same 1,800 gates of logic, and 1,800-gate Logic Cell Array could implement an X-Y-Z position controller in the same density used to implement an X-Y controller in an EPLD.

CONCLUSION

The benefits of programmable logic devices within any design are generally undeniable. Both devices shown in the design example (the EP1800 EPLD, and the XC2064 Logic Cell Array) replace many SSI/MSI components.

However, not all PLDs are created equal. A designer should contemplate his system needs and goals when deciding which PLD to use. Like making the correct choice of a system processor, the correct PLD choice will have an impact on the final system performance.

Some guidelines and cautionary notes are necessary for designers new to the PLD approach. These are:

- Be wary of PLD gates counts. As mentioned before, not all PLDs are created equal. Equivalent gate counting is used by many manufacturers to compare their devices against a competitor's. Two devices with approximately equal gate counts will differ on functional density within a system because of differences in architecture. A better way to determine density is to count the types and amounts of various resources on the chip, and compare those with the needs of your application.

- Understand which PLD architecture best suits your application. The sum-of-products (AND-OR) architecture is well suited to applications that require ANDing of a large number of inputs, like those found in address decoders and some counters. The interconnect structure of an AND-OR PLD makes single-pass logic functions quick and efficient. However, the complex random logic requirements of higher-density logic design stretches the limits of conventional AND-OR devices because of the feedback requirements.

- Determine your I/O and register requirements. In most AND-OR PLDs, outputs are hard-connectors to the registers within the device. Therefore, each register either uses or wastes an I/O pin and vice versa. In the Logic Cell Array, I/O and registers are separated by programmable interconnects. Therefore, entire register resources like counters and shift registers can be buried without using or wasting I/O pins.

Major enhancements in high-density PLDs will stem from architectural innovations rather than process-related developments. Regardless of which production process is used, all PLD manufacturers are searching for the optimal mix of high performance, high density, and production cost to best meet designer's logic needs.

Reprinted with permission from *Electronic Engineering Times.*

	MegaPAL	MegaPAL	EPLD	EPLD	LCA	LCA
Claimed Gate Density	1,500	5,000	1,200	1,800	1,200	1,800
Architecture	AND-OR	AND-OR	AND-OR	AND-OR	Array	Array
Developed By	MMI	MMI	Altera	Altera	Xilinx	Xilinx
Inputs (max.)	32	64	36	64	58	74
Outputs (max.)	16	32	24	48	58	74
Storage Elements	16	32	28	48	64	100
Elements	0	0	12	0	58	74
Speed For 16-Input AND (ns, On-through-Off, fastest)	40	50	50	50	45	45
Register-To-Register Clock Frequency (Max. MHz)	16	16	20	23.2	58.8	58.8
Quiescent Power (W)	1.4	3.25	0.015	0.001	0.025	0.025
Packages	40 DIP 44 PLCC	84 PLCC 88 PGA	40 DIP 44 CJCC 44 PLCC	68 CJCC 68 PLCC 68 PGA	48 DIP 68 PLCC 68 PGA	68 PLCC 84 PLCC
Process	Bipolar	Bipolar	CMOS	CMOS	CMOS	CMOS
Technology	Fuse	Fuse	EPROM	EPROM	SRAM	SRAM

1148 04

Shown above is data on several types of high-density EPLDs. Though the MegaPAL devices offer the most gates, they are bipolar, fuse-based design, and so are not reprogrammable.

A DIGITAL DESIGN PUBLICATION
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.

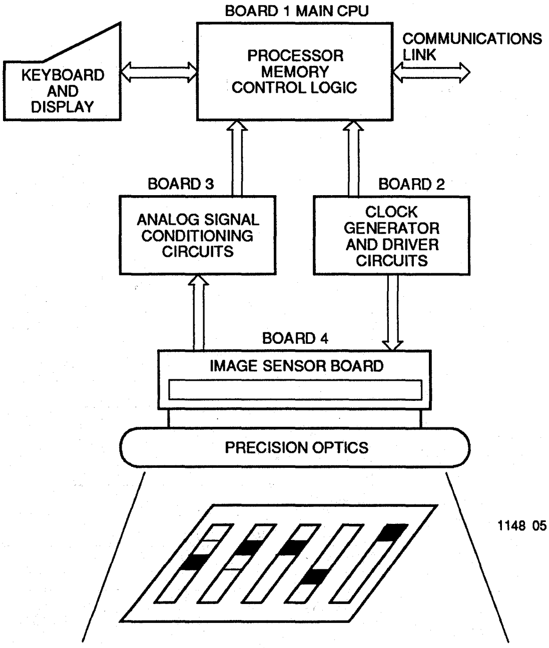
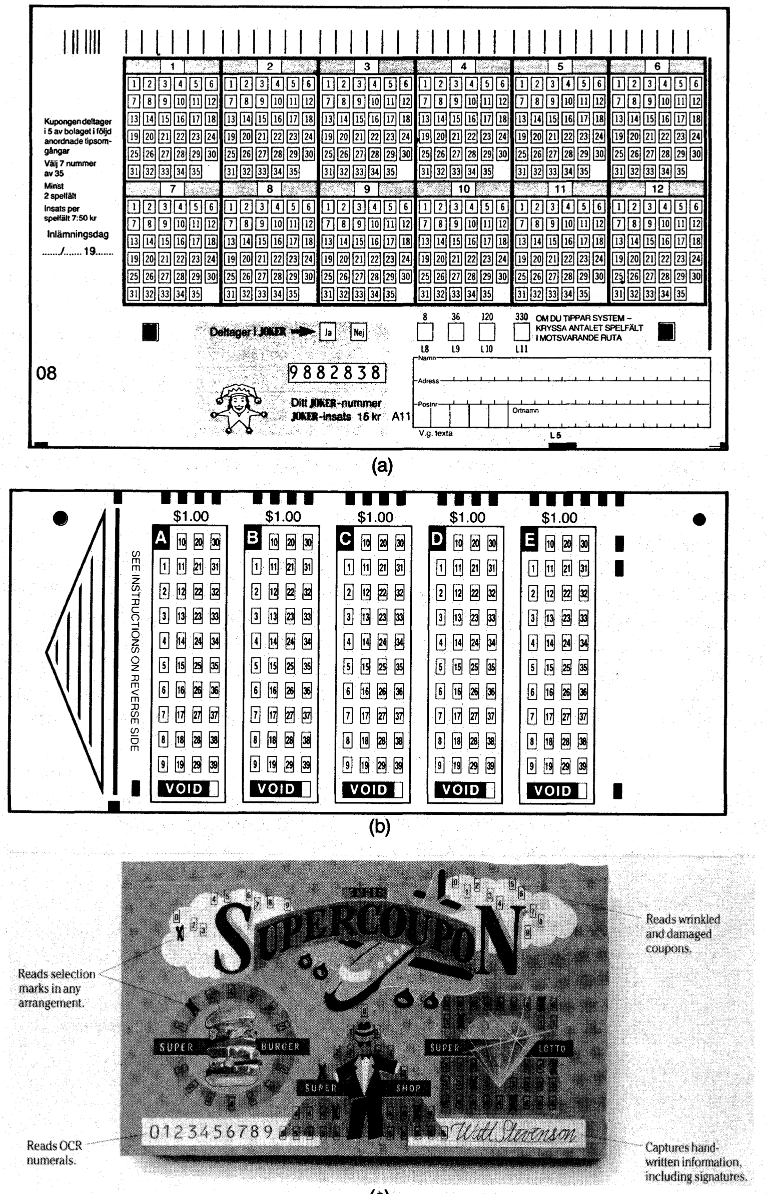


Figure 2. The goal of developing a prototype bet-slip processor (shown above) was to prove that handmade marks could be recognized in an imaging system. Four boards were initially developed for this modular design: CPU/memory, clock-driver, analog amplifier, and sensor mounting.

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 betslip 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.

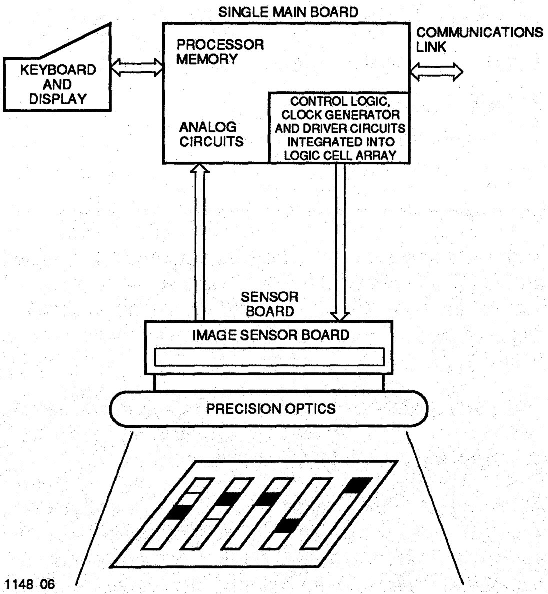


Figure 3. GTECH's Solid State Reader uses Logic Cell Arrays (LCAs) to maximize flexibility and minimize board space. Frame-grabber, memory addressing, and sensor clock driver functions are consolidated in the LCA. By reducing the number of chips, the required number of boards shrinks from four to two.

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.*

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.

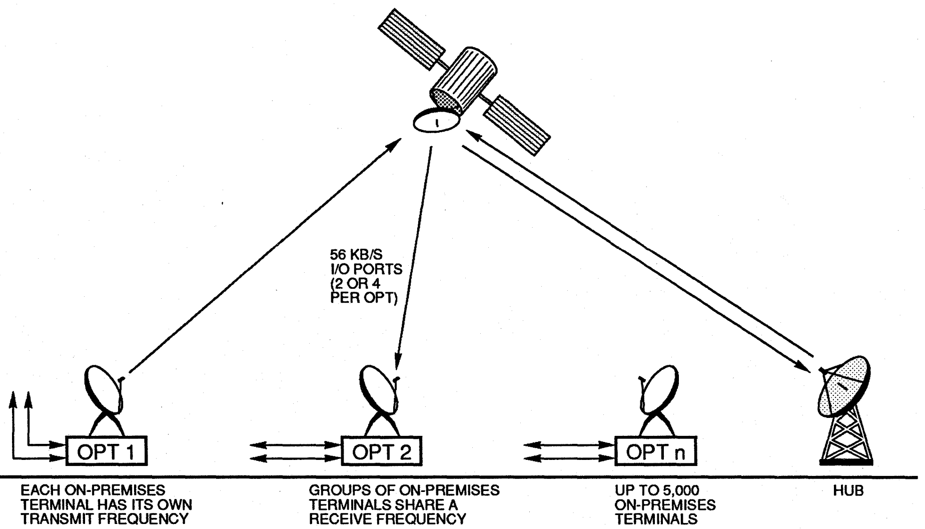

56 KB/S
I/O PORTS
(2 OR 4
PER OPT)
OPT 1
OPT 2
OPT n
EACH ON-PREMISES TERMINAL HAS ITS OWN TRANSMIT FREQUENCY
GROUPS OF ON-PREMISES TERMINALS SHARE A RECEIVE FREQUENCY
UP TO 5,000 ON-PREMISES TERMINALS
HUB
1148 07

Figure 1. Satellite System

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		

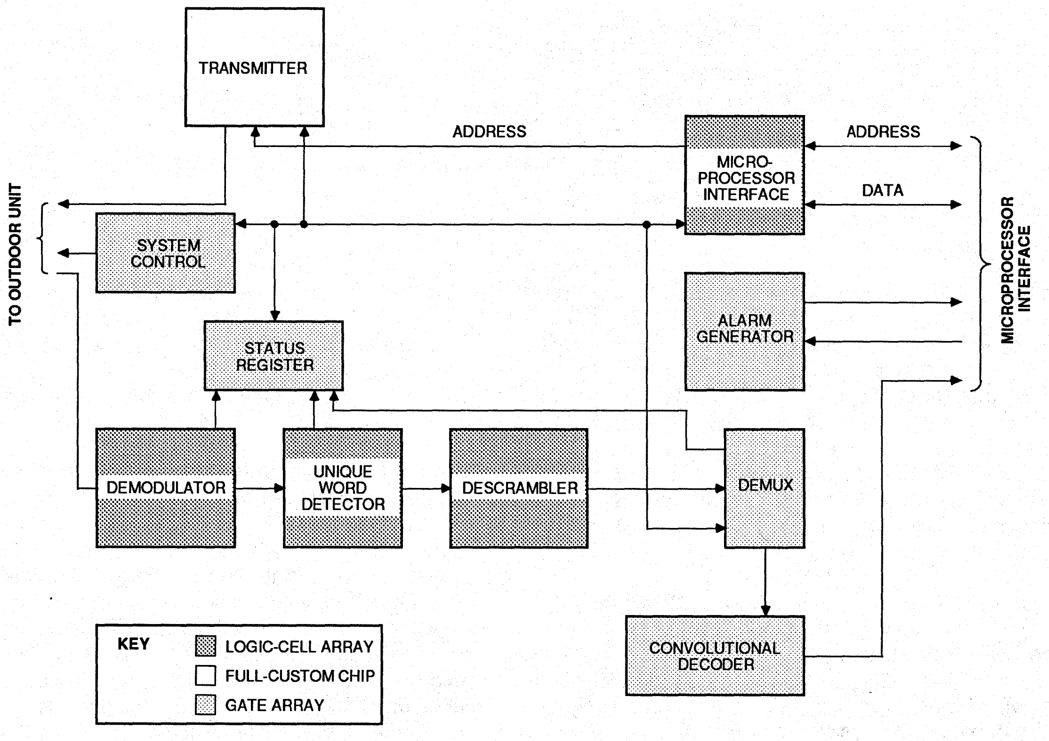
Table 1. Standard Off-the-Shelf Equivalents to the Logic Contained in the LCA.

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.

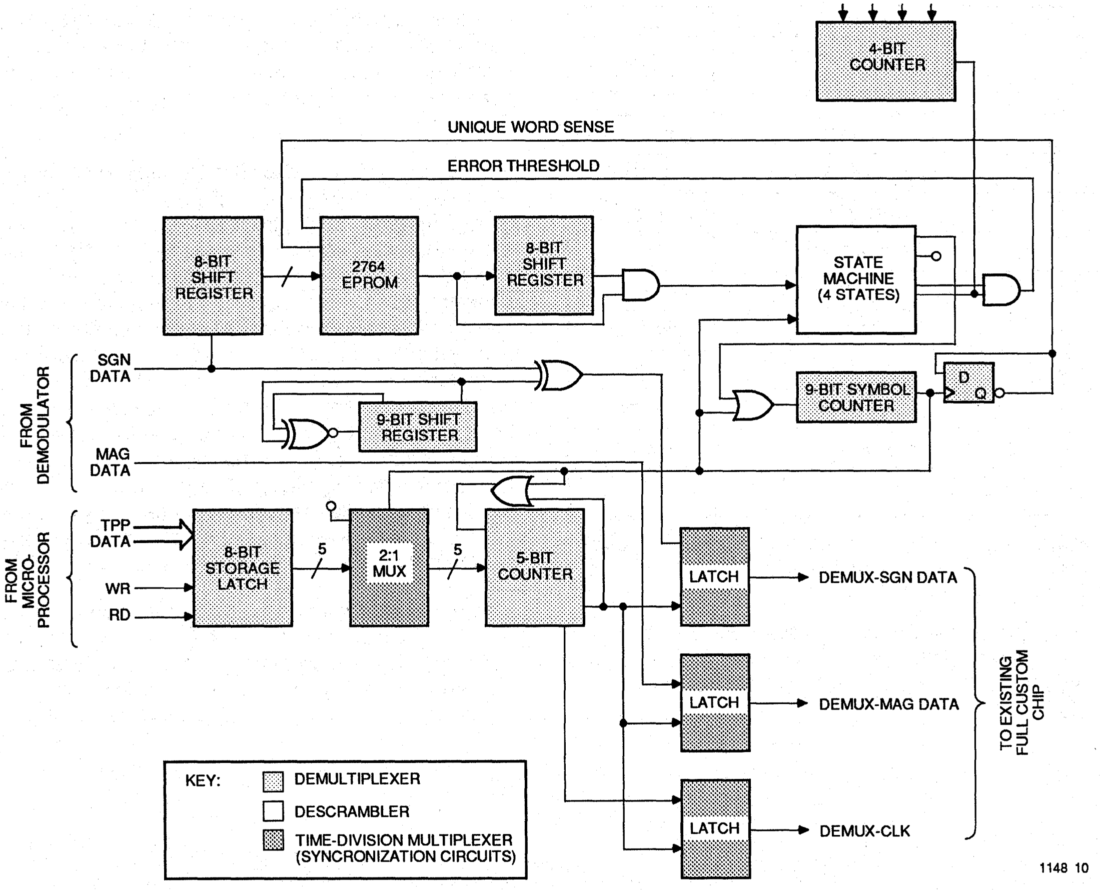


Figure 3. A Schematic of the Digital Systems Incorporated into the LCA.

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.

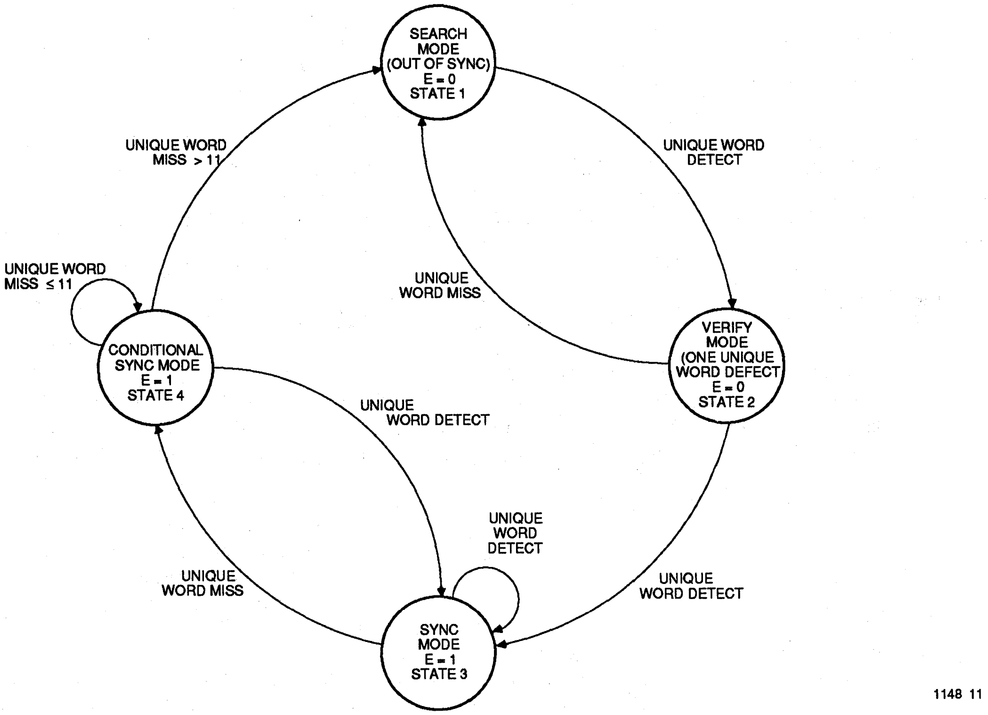


Figure 4. State Machine for the Time-division Multiplexer.

OPEN-END ĐEVELOPMENT

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.*

A DIGITAL DESIGN PUBLICATION
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

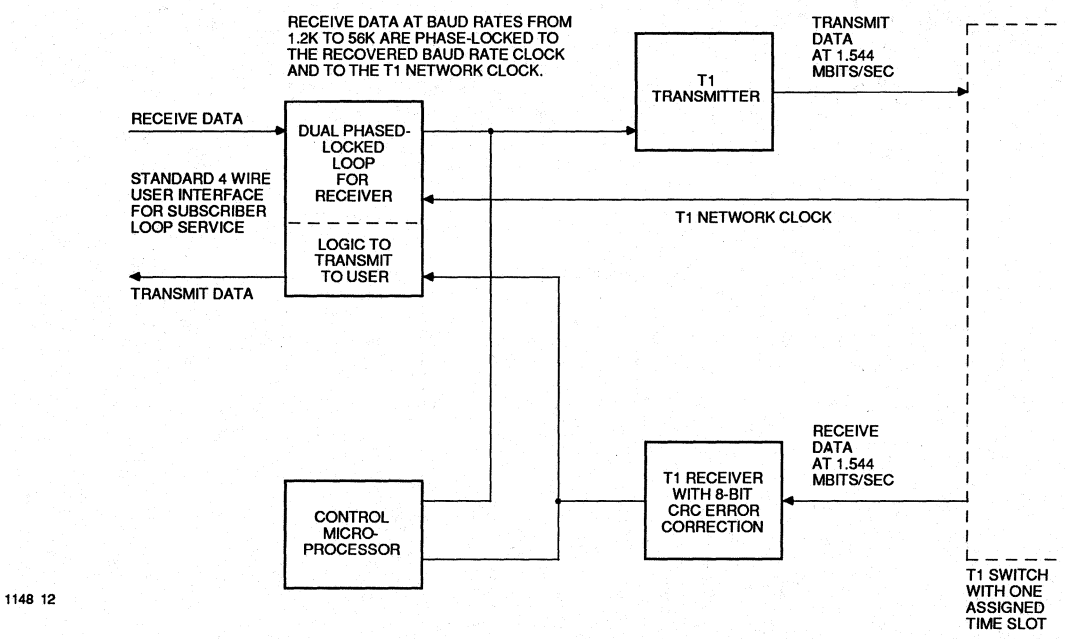

RECEIVE DATA AT BAUD RATES FROM 1.2K TO 56K ARE PHASE-LOCKED TO THE RECOVERED BAUD RATE CLOCK AND TO THE T1 NETWORK CLOCK.
RECEIVE DATA
STANDARD 4 WIRE USER INTERFACE FOR SUBSCRIBER LOOP SERVICE
TRANSMIT DATA
DUAL PHASED-LOCKED LOOP FOR RECEIVER
LOGIC TO TRANSMIT TO USER
T1 TRANSMITTER
TRANSMIT DATA AT 1.544 MBITS/SEC
T1 NETWORK CLOCK
CONTROL MICRO-PROCESSOR
T1 RECEIVER WITH 8-BIT CRC ERROR CORRECTION
RECEIVE DATA AT 1.544 MBITS/SEC
T1 SWITCH WITH ONE ASSIGNED TIME SLOT
1148 12

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.*

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.*

A DIGITAL DESIGN PUBLICATION

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.

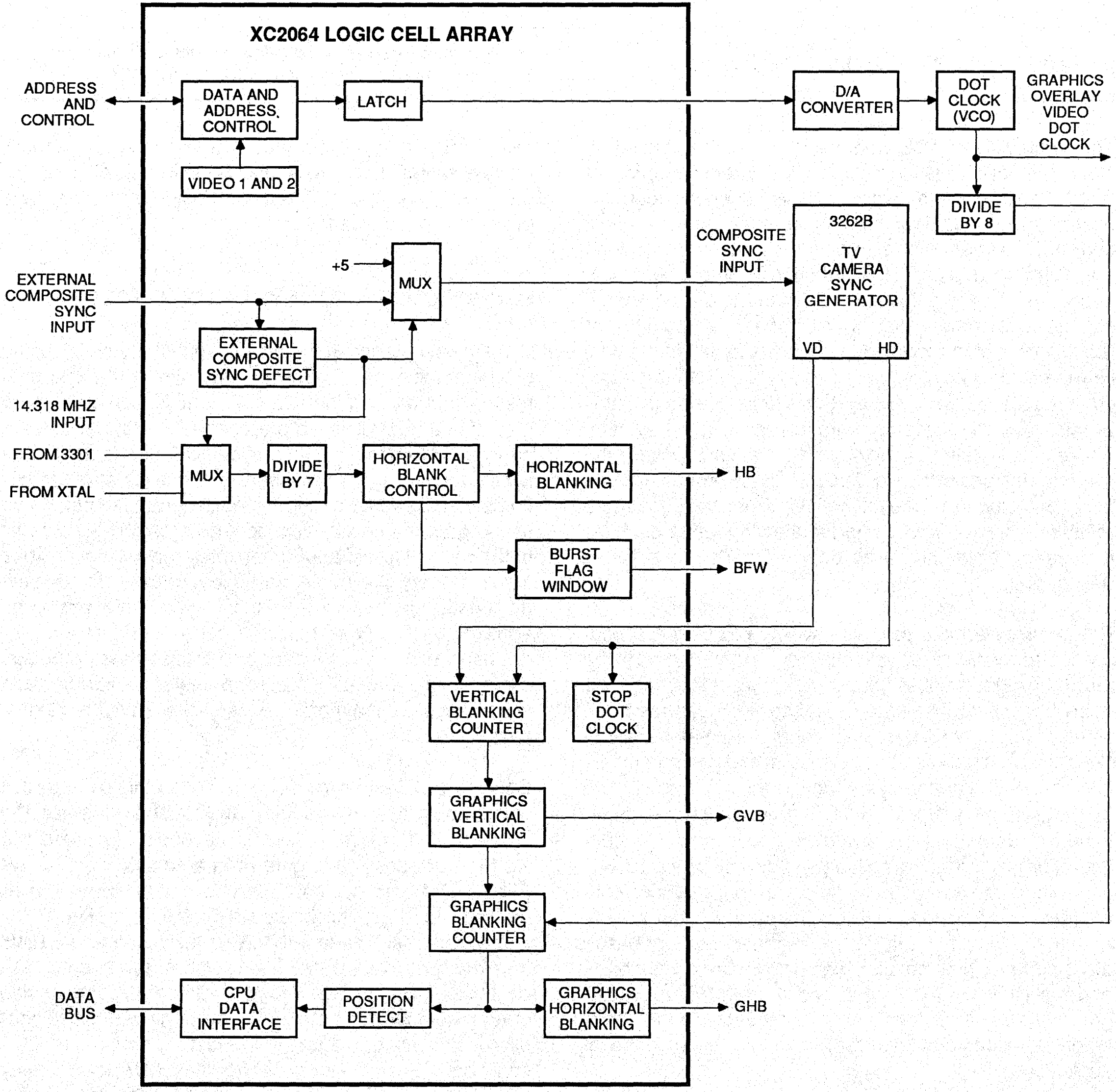
Two more LCAs implement a three-bit ALU. This technique achieves ultra-high-speed screen writes for both horizontal and vertical lines. For many applications, these are the most common lines drawn, so a special control bit is used to simultaneously modify pixels. Horizontal lines can be written at 14 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.

XILINX

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

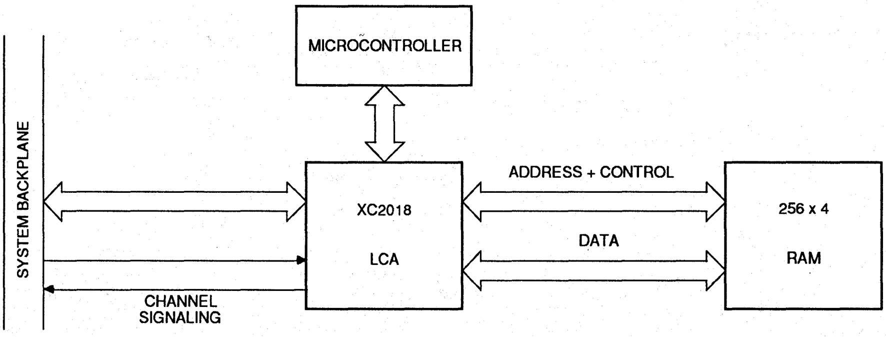
System self-diagnostics can be implemented by using programmable gate array configurations dedicated to testing functions. When the system is powered-up or placed in a test mode, its programmable gate arrays are configured with logic functions dedicated to testing other circuitry in the system. Once 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

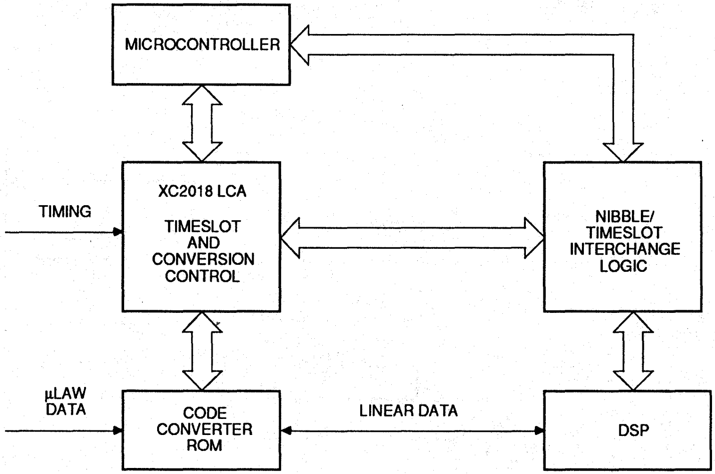
A similar use of reconfigurable logic is the implementation of a single hardware design that can be adapted for varying tasks or environments. In such systems, any of a number of potential configuration programs can be downloaded into a system's LCAs to alter the logic for particular applications or 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.



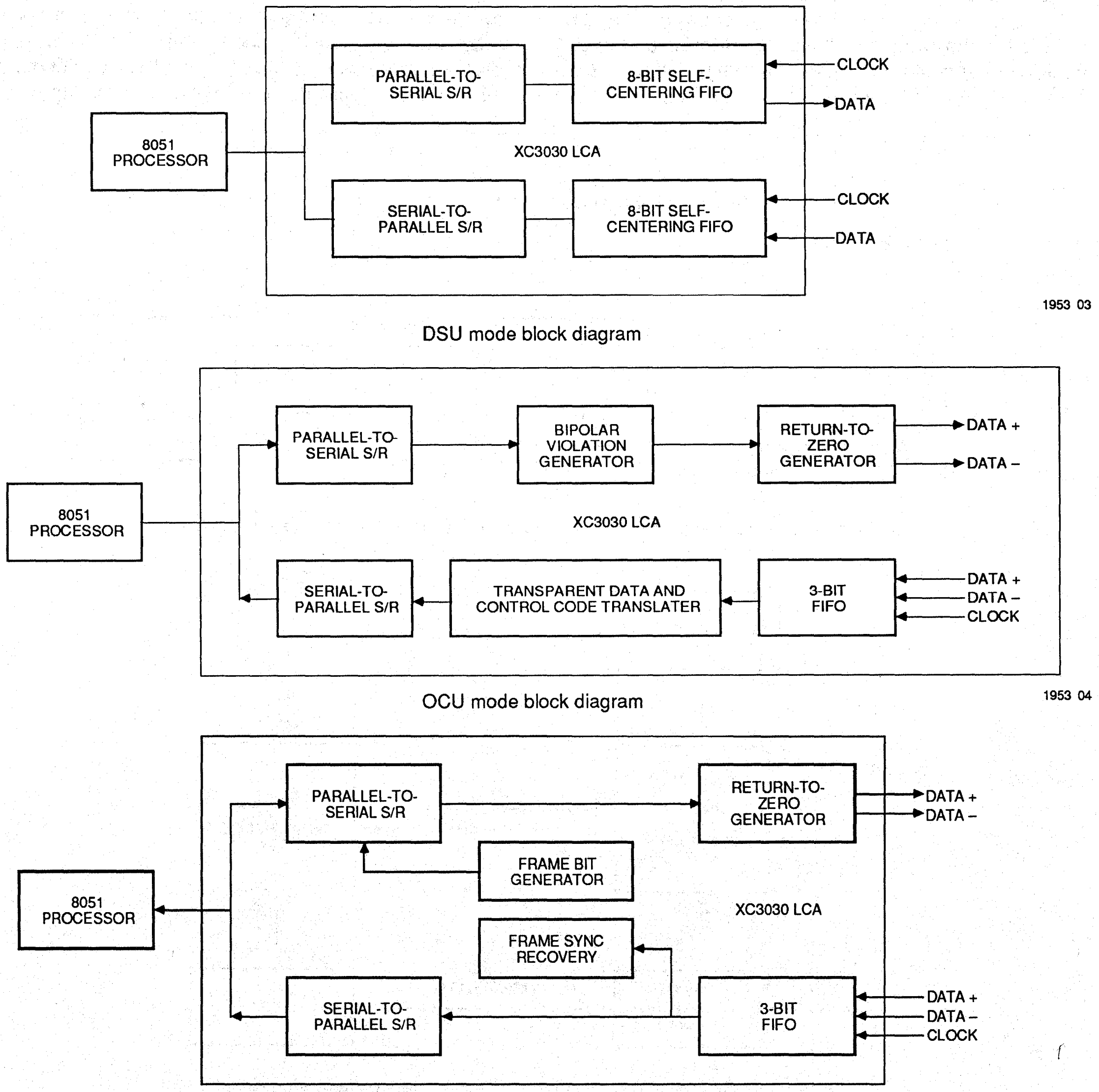
1953 02

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]

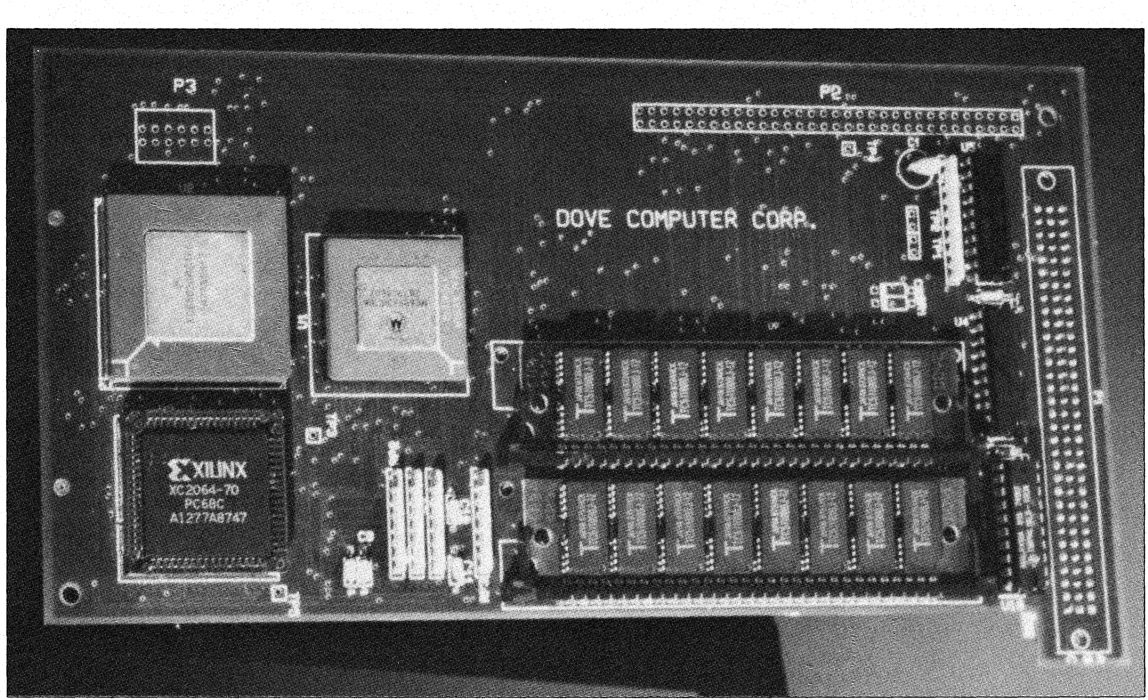


Figure 3. The Dove Computer Mach II/SE includes a micro-processor, floating-point co-processor, memory, bus drivers, and an LCA that holds all the interface logic.

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.

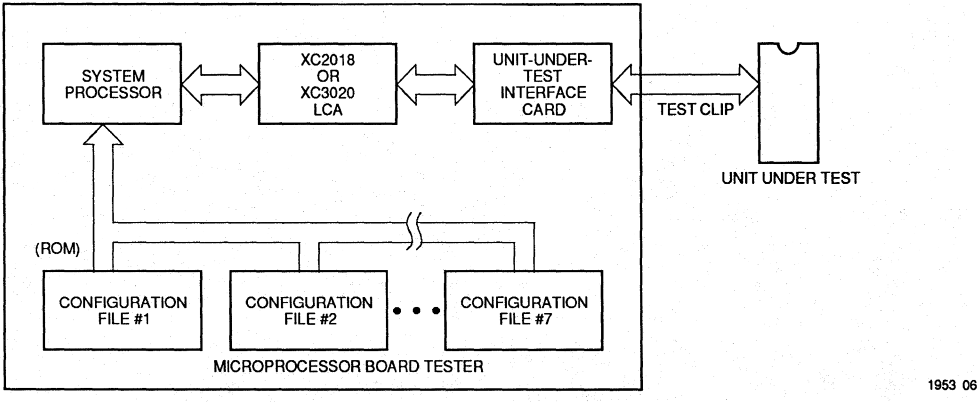


Figure 4. In Innovage Microsystem's microprocessor board tester, an LCA is configured for the appropriate microprocessor type and selected diagnostic test.

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-

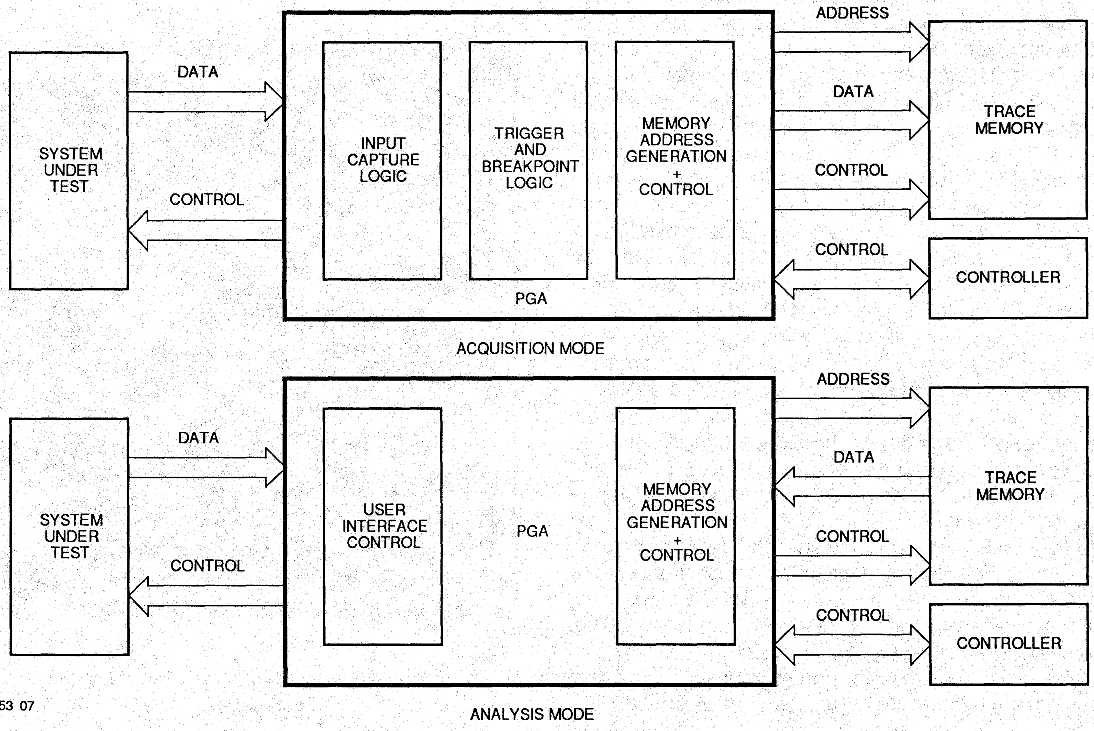


Figure 5. An LCA can be reconfigured to support both acquisition mode and analysis mode operations in a logic analyzer.

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-

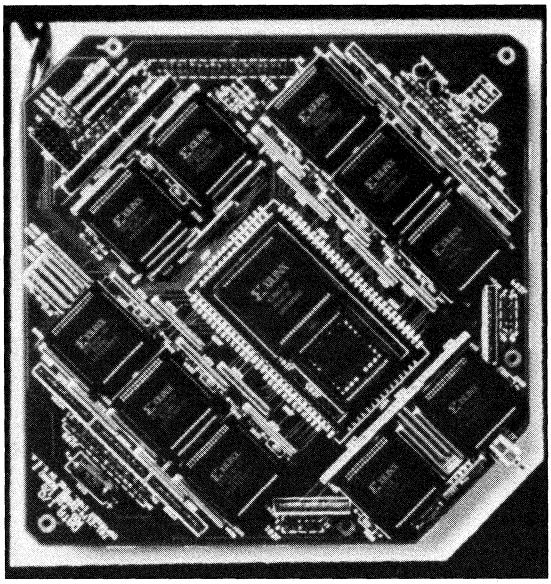


Figure 6. The internal logic of Data I/O's MESA-1 in-circuit debugger is implemented entirely in Xilinx programmable gate arrays.

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.

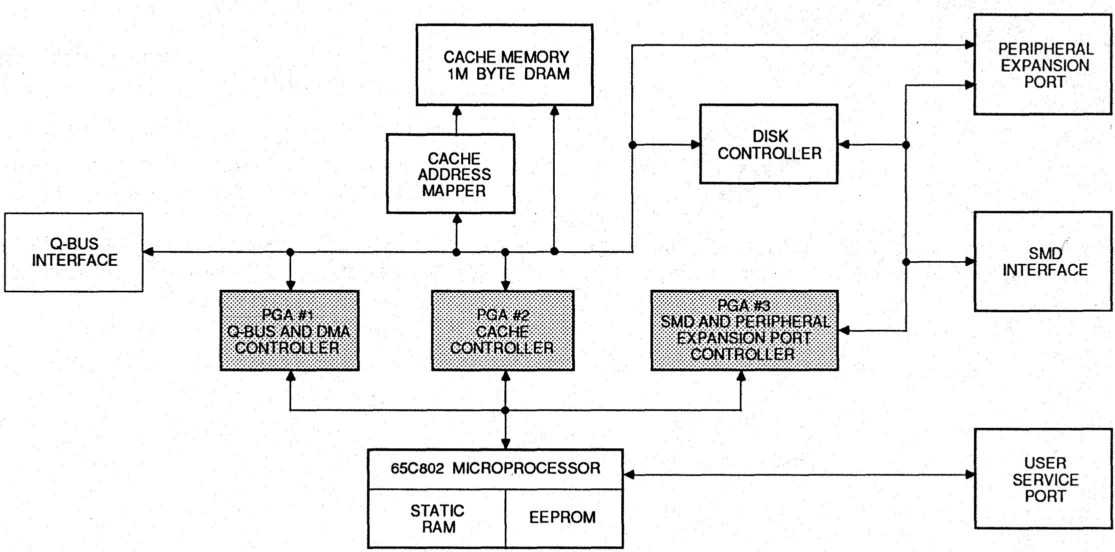
The ability to implement easily modifications to the logic enables and encourages experimentation during the design cycle, resulting in better designs. For example, the use of Xilinx LCAs allowed GTECH Corp. to evaluate different image sensors during the design of a bet-slip reader for the lottery industry.[4] 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



1953 08

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.

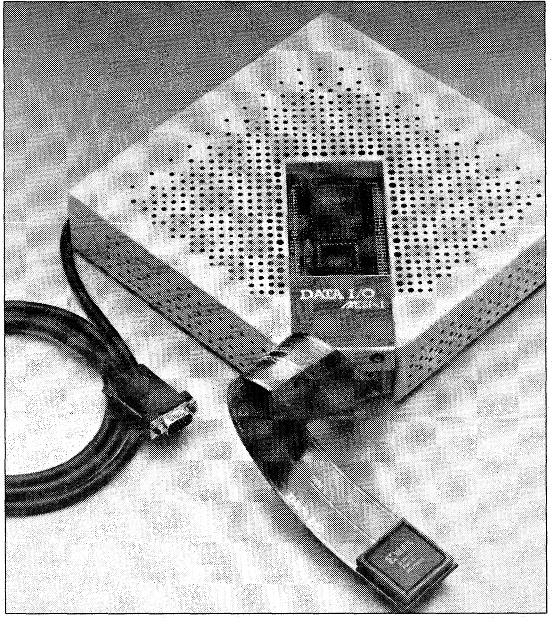


Figure 8. The reconfigurability of LCAs allows for the design of their own in-circuit verification tools, such as the MESA-1 from Data I/O.

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.

SECTION 8

Index

Index

Index

Sales Offices

SALES OFFICES

EUROPE

XILINX, Ltd.
Station House
Bepton Road, Midhurst
Sussex GU 29 9RE
England
Tel: 073081 6725
FAX: 073081 4910

JAPAN

XILINX K. K.
Kybashi No. 8
Nagaoka Bldg. 8F
20-9 Hatchobori Nichome
Chuo-ku, Tokyo 104, Japan
Tel: 03-297-9191
FAX: 03-297-9189

NORTH AMERICA

XILINX, INC.
2100 Logic Drive
San Jose, CA 95124
(408) 559-7778
TWX: 510-600-8750
FAX: 408-559-7114

XILINX, INC.
1270 Oakmead Parkway
Suite 201
Sunnyvale, CA 94086
(408) 245-1361
FAX: 408-245-0517

XILINX, INC.
2659 Townsgate Road
Suite 101
Westlake Village, CA 91361
(805) 494-5026
FAX: 805-496-0239

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
(312) 605-1972
TLX: 510-601-5973
FAX: 312-605-1985

DOMESTIC SALES

ALABAMA

Technology Marketing
Associates, Inc.
3315 So. Memorial Pkwy.
Suite 5
Huntsville, AL 35801
(205) 883-7893
FAX: 205-882-6162

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
4614 S. Knoxville Avenue
Tulsa, OK 74135
(918) 744-9964
TWX: 510-600-5274

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

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

Micro Comp, Inc.
1421 S. Caton Avenue
Baltimore, MD 21227
(301) 644-5700
TWX: 510-600-9460
FAX: 301-644-5707

FLORIDA

Technology Marketing
Associates, Inc.
8000 Orange Ave., Suite 111
Orlando, FL 32809
(407) 857-3760
TWX: 510-600-4721
FAX: 407-857-6412

Technology Marketing
Associates, Inc.
1239 E. Newport Center Dr.,
Suite 107
Deerfield Beach, FL 33442
(305) 427-1090
FAX: 305-427-1626

Technology Marketing
Associates, Inc.
11100 66 St. No., Suite 25
Largo, FL 34643
(813) 541-1591
FAX: 813-545-8617

GEORGIA

Technology Marketing
Associates, Inc.
6855 Jimmy Carter Boulevard
Suite 2420
Norcross, GA 30071
(404) 446-3565
FAX: 404-446-0569

IDAHO (Southwest)

Thorson Company Northwest
12301 N.E. 10th Place
Bellevue, WA 98005
(206) 455-9180

ILLINOIS

Beta Technology Sales, Inc.
1009 Hawthorne Drive
Itasca, IL 60143
(312) 250-9586
TWX: 62885853
FAX: (312) 250-9592

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
FAX: 913-782-8641
TWX: 910-350-6002

LOUISIANA (Northern)

Bonser-Philhower Sales
689 W. Renner Rd., Suite lII
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

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
TWX: 816-287-310
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
1810 Craig Road, Suite 213
St. Louis, MO 36146
(314) 878-2921
FAX: 314-878-1994

NEVADA

Norcomp
(Excluding Clark County)
3350 Scott Blvd., Suite 24
Santa Clara, CA 95054
(408) 727-7707
TWX: 510-600-1477

Quatra Associates
(Clark County)
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

Sales Offices

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.
3901 Commerce Avenue
Suite 180
Willow Grove, PA 19090
(215) 657-7250
TWX: 510-601-1856
FAX: 215-657-3781

NEW MEXICO

Quatra Associates
9704 Admiral Dewey N.E.
Albuquerque, NM 87111
(505) 821-1455

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

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, Inc.
5337 Trestlewood Lane
Raleigh, NC 27610
(919) 833-7771
TWX: 510-600-0558
FAX: 919-856-1644

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
3623 Brecksville Road
Richfield, OH 44286-0177
(216) 659-3131
FAX: 216-659-4823
TWX: 810-427-9100

Bear Marketing, Inc.
240 W. Elmwood Drive
Suite 1002
Centerville, OH 45459-4248
(513) 436-2061
FAX: 513-436-9137

OKLAHOMA

Bonser-Philhower Sales
2727 E. 21st Street
Suite 602
Tulsa, OK 74114
(918) 744-9964
FAX: 918-749-0497

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

Technology Marketing
Associates, Inc.
1280 S.W. 36th Ave.,
Suite 201
Pompano Beach, FL 33069
(305) 977-9006
TWX: 510-601-0120
FAX: 305-977-9044

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, Inc.
5337 Trestlewood Lane
Raleigh, NC 27610
(919) 833-7771
TWX: 510-600-0558
FAX: 919-839-0791

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
9704 Admiral Dewey N.E.
Albuquerque, NM 87111
(505) 821-1455

UTAH

Front Range Marketing
7050 Union Park Center
Suite 440
Midvale, UT 84047
(801) 566-2500
FAX: 801-566-2951

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.
1421 S. Caton Avenue
Baltimore, MD 21227
(301) 644-5700
TWX: 510-600-9460
FAX: 301-644-5707

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

WEST VIRGINA

Bear Marketing, Inc.
1563 East Dorothy Lane
Suite 104
Kettering, OH 45429
(513) 299-5877
FAX: 513-299-0756

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 304C
Milwaukee, WI 53227
(414) 543-6609

INTERNATIONAL SALES

AUSTRALIA

ACD/ITRONICS
106 Belmore Rd. North
Riverwood, N.S.W. 2210
Tel: Sydney 534-6200
FAX: Sydney 534-4910

ACD/ITRONICS
Unit 2, 17-19 Melrich Road
Bayswater VIC 3153
P.O. Box 139
Tel: Melbourne (03) 762 7644
Fax: Melbourne (03)762 5446

ACD/ITRONICS
55 Noreen Street
Chapel Hill QLD 4069
Tel: QLD 878 1488
Fax: QLD 878 1490

AUSTRIA

Eljapex Ges. m.b.h.
Eitnergasse 6
A-1232 Wien
Austria
(01) 86 3211
FAX: (01) 86 3211 200

BELGIUM & LUXEMBURG

Le Mar Rodelco
Limburg Stirum 243
1810 Wemmel
Belgium
(02) 460-0560
FAX: (02) 460-0271

BRAZIL

International Trade
Development Corporation
450 San Antonio Road
Suite 32
Palo Alto, CA 94306
(415) 856-6686
Telex: 650-282-9742

CANADA

BRITISH COLUMBIA (Vancouver)

Thorson Company Northwest
12301 N.E. 10th Place
Bellevue, WA 98005
(206) 455-9180

ONTARIO

Electro Source, Inc.
320 March Road, Suite 500
Kanata, Ontario K2K 2E3
(613) 592-3214
FAX: 613-592-4256

Electro Source, Inc.
230 Galaxy Boulevard
Rexdale, Ontario M9W 5R8
(416) 675-4490
TWX: 06-989271
FAX: 416-675-6871

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: (02) 4345 47
FAX: (02) 4345 67

FINLAND

OY SW-Instruments AB
Ala Portti IC
SF-02210 Espoo, Finland
Tel: (08) 41 10 41
FAX: (08) 03 19 55

FRANCE

Reptronic
11, Escalier des Ulis
91400 Orsay, France
Tel: (16) 928 8700
FAX: (16) 928 1750

R.T.F. Gentilly
9, rue d'Arcueil
94253 Gentilly Cedex,
France
Tel: (16) 4 66 41 11 0
TLX: 2010169F
FAX: (16) 4 66 44 19 9

R.T.F. (Radio Television
Francaise S.A.)
13, rue Lhote
33000 Bordeaux, France
Tel: 56 52 99 59
TLX: 560627
FAX: 56 48 17 83

R.T.F. Ouest
3, rue de Paris
35510 Cesson Sevigne,
France
Tel: 99 83 84 85
TLX: 741127
FAX: 99 83 80 83

R.T.F. Sud Ouest
Avenue de la Mairie
31320 Escalquens, France
Tel: 61 81 51 57
TLX: 520927F
FAX: 61 81 22 36

R.T.F. Rhone Alpes
St. Mury, Le Vaucanson
38240 Maylan, France
Tel: 76 90 11 88
TLX: 980796
FAX: 76 41 04 09

GERMANY

Metronik
Leonhardsweg 2
Postfach 13 28
8025 Unterhaching
München, Germany
Tel: (089) 611080
TLX: 897434
FAX: (089) 611 6468

Metronik
Semerteichstrasse 92
4600 Dortmund 30
Dortmund, Germany
Tel: (0231) 423037/38
TLX: 8227082

Metronik
Osterbrooksweg 61
2000 Schenefeld
Hamburg, Germany
Tel: (040) 8304061
TLX: 2162488

Metronik
Siemensstrasse 4-6
6805 Heddesheim
Mannheim, Germany
Tel: (06203) 4701-03
TLX: 465053

Metronik
Laufamholzstr. 118
8500 Nürnberg 30
Nürnberg, Germany
Tel: (0911) 590061/62
TLX: 626205

Metronik
Löwenstr. 37
7000 Stuttgart 70
Stuttgart, Germany
Tel: (0711) 764033/35
TLX: 7255228

HONG KONG

Excel Associates, Ltd.
1502 Austin Tower
22-26A Austin Avenue
Tsimshatsui, Kowloon
Hong Kong
Tel: 3-7210900
FAX: 3-696826
TLX: 30841

ISRAEL

E.I.M International Ltd.
8, Emil Zola Street
P.O. Box 4000
Petach Tiqva
Tel Aviv, Israel
Tel: (3) 92 33257
FAX: (3) 924 4857
TLX: 3811 44 E.I.M.I.L.

ITALY

ACSIS S.R.L.
Via Alberto Mario. 26
20149 Milano, Italy
Tel: (02) 4390832
TLX: 326566
FAX: (02) 4697607

Celdis Italiana S.P.A.
Via F.ill Gracchi 36
20092 Cinisello Balsamo
Milano, Italy
Tel: (02) 61 839 1
TLX: 334887
FAX: (02) 61 735 13

Celdis Italiana S.P.A.
Via Massarenti 219/4
40138 Bologna, Italy
Tel: (051) 53 333 6

Celdis Italiana S.P.A.
Via Savelli 15
35100 Padova, Italy
Tel: (049) 77 209 9

Celdis Italiana S.P.A.
Via G. Pitre´ 11
00162 Roma, Italy
Tel: (06) 42 897 1

Celdis Italiana S.P.A.
Via Mombarcaro 96
10136 Torino, Italy
Tel: (011) 32 993 88

JAPAN

Okura & Co., Ltd.
6-12, Ginza Nichome
Chuo-Ku
Tokyo, 104 Japan
Tel: (03) 566 6361
TWX: J22306
FAX: (03) 563 5447

KOREA

Excel-Tech
410-5 Hapjeong-Dong
Mapo-Gu
Seoul, Korea
Tel: 82-2-3357823
FAX: 82-2-3357825

Shinhwa Corp.
Room 902, Bang-hyoup Bldg.
43-8 Kwanchul-Dong
Chongro-ku
Seoul, Korea

THE NETHERLANDS

Rodelco BV Electronics
Takkebijsters 2
P.O. Box 6824
4802 HV Breda
Netherlands
Tel: (076) 784911
TLX: 54195
FAX: (076) 710029

NORWAY

B.I.T. Elektronikk AS
P.O. Box 36 Lerbyen
N-3401 Lier, Norway
Tel: (03) 84 70 99
FAX: (03) 84 55 10

SINGAPORE

Excel Associates Ltd.
Singapore Representative
Office
111 North Bridge Road
#11-04/06 Peninsula Plaza
Singapore 0617
Tel: 3366577
FAX: 3395291
Telex: RS 36033 WWBCS

SOUTHEAST ASIA

Excel Associates, Ltd.
1502 Austin Tower
22-26A Austin Avenue
Tsimshatsui, Kowloon
Hong Kong
Tel: 852-3-7210900
FAX: 852-3-696826
TLX:30841

SPAIN

ADM Electronica SA
Menorea No. 3 Entreplanta
Madrid 28009
Spain
Tel: (01) 409 4725
FAX: (01) 409 5215

SWEDEN

D.I.P. Electronics AB
P.O. Box 15046
S-104 65 Stockholm, Sweden
Tel: (08) 44 91 90
FAX: (08) 43 00 47

SWITZERLAND

Data Comp AG
Silbernstrasse 10
CH-8953 Dietikon
Switzerland
Tel: (01) 7405140
Telex: 827750
FAX: (01) 7413423

TAIWAN

Molecatex, Inc.
21F 258 Sec 3
Nanking East Road
Taipei, Taiwan R.O.C.
Tel: (02) 7410400
FAX: (02) 7217461
TLX: 29214 MTEX

Jeritron Ltd.
5/F Fu San Building
1182 Cheng-Teh Road
Taipei, Taiwan, R.O.C.
Tel: (02) 8823154
FAX: (02) 8820710

Sertek Int'l Inc.
15/F, 135 Sec 2
Chien Kuo N. Road
Taipei 10479
Taiwan R.O.C.
Tel: 2 501 0055
Fax: 2 501 2521
Telex: 23756 Sertek

UK AND IRELAND

Microcall, Ltd.
17 Thame Park Road
Thame
Oxon OX9 3XD
England
Tel: (084) 421 5405
TLX: 837457
FAX: (084) 421 4267

Microcall, Ltd.
The Genesis Centre
Birchwood Science Park
Garrett Field
Warrington WA3 7BN
England
Tel: (0925) 825065

Distributed By

Hamilton/Avnet
locations throughout
the U.S. and Canada.
1-800-HAM-ASIC
FAX: 408-743-3003

Marshall Industries
locations throughout
the U.S. and Canada.
(818) 459-5500
FAX: 818-459-5660

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

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 (______) ____________________

XILINX

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 (______) ____________________

XILINX

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

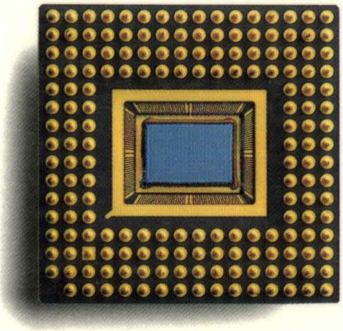
NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

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.

EasyLink 62916309 TWX: 5106008750 XILINX UQ FAX: (408) 559-7114

P/N 0010048 02