AT&T Field-Programmable Gate Arrays

Data Book

July 1992

A Word About Trademarks. . .

The following AT&T trademarks are used in this document: *SoftPath*™, *Syscad*™, WE®.

Trademarks owned by entities other than AT&T are also presented. *Xilinx, XACTOR, LCA, XC1700, XC3020, XC3030, XC3042, XC3064, and XC3090* are trademarks and *XACT* is a registered trademark of Xilinx, Inc. *IBM, AT, PS/2*, and *Micro Channel* are registered trademarks and *PC/XT* is a trademark of International Business Machines Corporation. *ABEL* and *MESAII* are trademarks and *DATA I/O* is a registered trademark of Data I/O, Inc. *Microsoft, MS-Windows*, and *MS-DOS* are registered trademarks of Microsoft, Corp. *Mouse Systems* is a trademark of Mouse Systems, Inc. *PAL* and *PALASM* are registered trademarks of Advanced Micro Devices, Inc. *CUPL* is a trademark of Logical Devices, Inc. *Apollo* is a trademark and *Domain* is a registered trademark of Apollo Computer, Inc. *Quicksim, NETED,* and *IDEA* are trademarks and *Mentor Graphics* is a registered trademark of Mentor Graphics, Inc. *DECstation* is a trademark and *VMS* is a registered trademark of Digital Equipment Corporation. *Logitech* is a registered trademark of Logitech Inc. *Sun-3, Sun-4, SunOS,* and *SPARCstation* are registered trademarks of Sun Microsystems, Inc. *Kovar* is a registered trademark of Carpenter Technical Corporation. *NEC* is a trademark of NEC Electronics, Inc. *FutureNet* and *DASH* are registered trademarks of FutureNet Corporation. *Viewlogic, Viewsim, Viewdraw, Viewwave, and System VHDL* are trademarks of Viewlogic Systems, Inc. *Verilog* is a registered trademark of Cadence Design Systems, Inc. *SILOS* is a trademark of Simucad Corporation. *OrCad, VST,* and *SDT* are trademarks of OrCad Systems Corporation. *X-Windows* is a trademark of Massachusetts Institute of Technology. *Synopsys* is a registered trademark of Synopsys, Inc. *Valid* is a trademark of Valid Logic Systems. *Open Look* is a trademark and *UNIX* is a registered trademark of UNIX Systems Laboratories Inc. *PLDesigner* is a registered trademark of MINC Incorporated. *Advantest* is a registered trademark of Advantest Corporation. *Dasix* is a registered trademark of Daisy Cadnetics, Inc.

Contents

Contents Page

1. General Information

2. FPGA Qualification Handbook

3. AT&T FPGA Product Specifications

Contents Page

4. Software Development Systems

5. Applications

Contents **Page**

5. Applications (continued)

6. Migration to Gate Arrays

7. Package Information

Contents Page

8. Sales Offices

9. Ordering Information

AT&T Quality Policy

Policy

Quality excellence is the foundation for the management of our business and the keystone of our goal of customer satisfaction. It is, therefore, our policy to:

- Consistently provide products and services that meet the quality expectations of our customers.
- Actively pursue ever-improving quality through programs that enable each employee to do his or her job right the first time.

Intent

Quality will continue to be a major, strategic thrust in AT&T. It lies at the heart of everything we do.

Through active planning in every function in the company, we will strive to provide products and services that consistently meet all quality, schedule, and cost objectives. Furthermore, we will dedicate ourselves to continually improving the quality of our products and services by focusing on our processes and procedures.

Every employee is a part of our quality system.

- Each of us will strive to understand and satisfy the quality expectations of our customers (meaning the next internal organization in the process as well as the eventual end-customer).
- Each of us will strive to identify and eliminate the sources of error and waste in our processes and procedures.
- Each of us will aid the quality-planning and improvement efforts of others for the good of the corporation as a whole.

Responsibilities

Each business group president, entity head, and senior staff officer is responsible for:

- Communicating our quality policy to each employee.
- Clarifying specific responsibilities for quality.
- Developing and reviewing strategic quality plans and objectives on an on-going basis.
- Implementing a quality management system to carry out the plans and achieve objectives.
- Monitoring and continually improving the level of customer satisfaction.
- Monitoring and continually improving the defect and error rate of internal processes and systems.
- Developing joint quality plans with suppliers and other business partners.
- Implementing, funding, and reviewing specific quality improvement programs.
- Providing education and training in quality disciplines for all employees.

Robert E. Allen
Chairman of the Board and Chief Executive Officer

Section 1.
General Information

1. General Information

Table of Contents

Contents **Page**

AT&T Microelectronics

AT&T Field-Programmable Gate Arrays Data Book

Semiconductor Services

AT&T Microelectronics, a major business unit in AT&T's Network Systems Group, is one of the world's largest manufacturers of semiconductors and electronic components.

AT&T offers its customers a broad range of integrated products and services supported by a comprehensive methodology of high component quality and reliability. The company is considered an industry leader in technological advances in the manufacturing process.

AT&T's primary customer groups include:

- Internal customers among AT&T's business units and divisions: major buyers and users in their own right.
- External customers: manufacturers which demand AT&T quality and performance in the components used in their own product lines.

One of AT&T Microelectronics' fundamental advantages in meeting customer needs for quality semiconductors is its historic link with Bell Laboratories, the corporate research arm. Since AT&T's earliest years, Bell Laboratories' basic and applied research superiority has combined with AT&T's manufacturing strengths to produce components of exceptional quality and reliability, fabricated using processes and technologies perfected within the company.

As AT&T Microelectronics has evolved into its present position in today's global market, the company has applied its experience serving internal customers to the world semiconductor market. The company is already a market leader in several types of components, and is in demand for its expertise on advanced manufacturing processes.

Its mission statement is succinct: to become **the customers' vendor of choice** through excellence in customer satisfaction, technology, design, production, and applications.

Strategic Applications

Some of AT&T Microelectronics' key applications include:

- Advanced PCs and workstations
- Telecommunications
- Wireless communications

Among the key components used in advanced PCs and workstations is AT&T's extensive family of Metal Oxide Semiconductor (MOS) ICs. The MOS business unit is headquartered at AT&T's Allentown Works, Allentown, PA, and is strategically allied with sizeable resident Bell Laboratories organizations, product and system support groups, and related marketing divisions.

AT&T ASICs: The Logical Solution

In the steadily advancing world of digital design, the application-specific IC is widely embraced as a critical element in high-density, logic applications.

Given its popularity, however, what was once perceived as a product line limited to custom and standard devices today emerges at AT&T as a comprehensive ASIC family.

With the addition of semicustom devices, AT&T offers designers the opportunity to devise the ideal solution from among multiple ASIC architectures, including the forerunning field-programmable gate array as well as the mask-programmed gate array.

The benefits abound. Integral to AT&T ASICs are the economy of high-volume production, the vital advantage of quick prototyping and speedy movement to the marketplace, expert design assistance, proven process technolgy, and exclusive manufacturing superiority.

The Total Solution

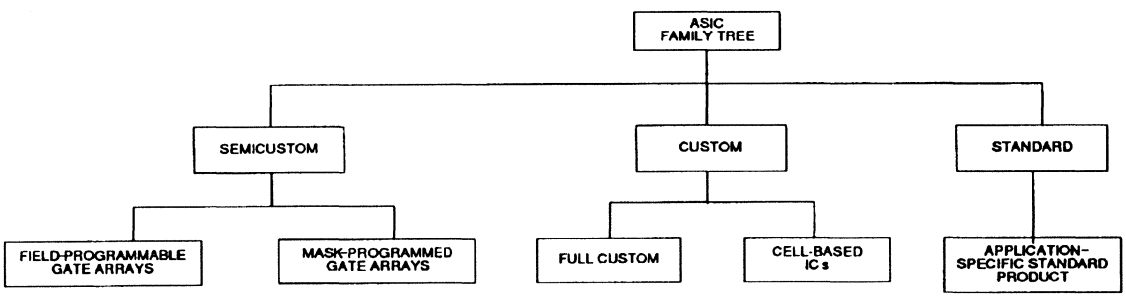
Whatever your needs in the ASIC family, you can look to AT&T today as the the only vendor offering the total ASIC design and manufacturing solution, including the capability to migrate FPGA designs quickly and efficiently between technologies.

In addition to FPGA development, the migration process is discussed in depth in this book along with quality and reliability considerations, product profiles and applications, design tools, including hardware and software requirements, and a complete cell library with logic, symbol, and sample placement information.

Low Risk

Choosing the right technology depends on a number of variables. Foremost among them are density, volume, and time to market.

Once circuit designers accrue a device history that resolves the uncertainties— that is, once an FPGA is used successfully to quickly prototype and verify their systems and achieve fast time to market—they can capitalize on AT&T's newest ASIC software tool to safely advance the original design, this time with low implemetation risk.



AT&T's Expanding ASIC Product Line

AT&T ASICs: The Logical Solution
(continued)

Extra Dividends

SoftPath™ is an exclusive AT&T software development that pays off in technology resynthesis, or in this case, the capability to economically fashion a functionally equivalent CMOS gate array or standard cell from a field-proven FPGA netlist.

So, as your needs change and demands for your products escalate, the choice of an AT&T FPGA at the onset proves even more practical.

The *SoftPath* benefits are substantial:

- Minimized gate count
- Cost-effective design
- Automatic I/O buffer mapping
- Pin-for-pin and footprint compatibility with the parent device
- Low-cost volume production
- Minimal engineering overhead
- Reduced development time
- Reduced power consumption
- Improved system performance
- One, single-source vendor

Design Cycles

AT&T offers two convenient FPGA design approaches.

For schematic capture, designers can use the AT&T-supplied libraries along with a choice of commercially available software packages, including *Viewdraw, OrCad,* and *Data I/O. Viewsim, Quicksm, and Silos* are used during simulation.

Customers normally supply a completed netlist, test vectors, pinout diagrams, and specifications. In return, AT&T provides the necessary technical support and consultation and guides the design through manufacture. We manage the physical layout using automatic placing and routing, extract parasitics, back annotate the customer's CAD system (if applicable), check design and electrical rules, and generate test programs. (For further information, refer to the software developement information in Section 4.)

Optionally, AT&T can perform the entire front-end design on a turnkey basis.

Training

A four-day FPGA-design training course is available through AT&T at convenient regional locations. Please contact with your local AT&T account manager for further details.

ATT3000-Series FPGA Cross-Reference Guide

AT&T Part Number	Description	*Xilinx* Part Number
	Commercial Temperature Range (0 °C to 70 °C)	
ATT3020-70M68	2000 Gate 70 MHz 68-Lead PLCC	*XC3020*-70PC68C
ATT3020-100M68	2000 Gate 100 MHz 68-Lead PLCC	*XC3020*-100PC68C
ATT3020-125M68	2000 Gate 125 MHz 68-Lead PLCC	*XC3020*-125PC68C
ATT3020-150M68	2000 Gate 150 MHz 68-Lead PLCC	N/A
ATT3020-70M84	2000 Gate 70 MHz 84-Lead PLCC	*XC3020*-70PC84C
ATT3020-100M84	2000 Gate 100 MHz 84-Lead PLCC	*XC3020*-100PC84C
ATT3020-125M84	2000 Gate 125 MHz 84-Lead PLCC	*XC3020*-125PC84C
ATT3020-150M84	2000 Gate 150 MHz 84-Lead PLCC	N/A
ATT3020-70J100	2000 Gate 70 MHz 100-Lead PQFP	*XC3020*-70PQ100C
ATT3020-100J100	2000 Gate 100 MHz 100-Lead PQFP	*XC3020*-100PQ100C
ATT3020-125J100	2000 Gate 125 MHz 100-Lead PQFP	N/A
ATT3020-150J100	2000 Gate 150 MHz 100-Lead PQFP	N/A
ATT3020-70R84	2000 Gate 70 MHz 84-Lead CPGA	*XC3020*-70PG84C
ATT3020-100R84	2000 Gate 100 MHz 84-Lead CPGA	*XC3020*-100PG84C
ATT3020-125R84	2000 Gate 125 MHz 84-Lead CPGA	N/A
ATT3020-150R84	2000 Gate 150 MHz 84-Lead CPGA	N/A
ATT3020-70N100	2000 Gate 70 MHz 100-Lead CQFP	*XC3020*-70CQ100C
ATT3020-100N100	2000 Gate 100 MHz 100-Lead CQFP	*XC3020*-70CQ100C
ATT3020-125N100	2000 Gate 125 MHz 100-Lead CQFP	N/A
ATT3030-70M44	3000 Gate 70 MHz 44-Lead PLCC	*XC3030*-70PC44C
ATT3030-100M44	3000 Gate 100 MHz 44-Lead PLCC	*XC3030*-100PC44C
ATT3030-125M44	3000 Gate 125 MHz 44-Lead PLCC	N/A
ATT3030-150M44	3000 Gate 1505 MHz 44-Lead PLCC	N/A
ATT3030-70M68	3000 Gate 70 MHz 68-Lead PLCC	*XC3030*-70PC68C
ATT3030-100M68	3000 Gate 100 MHz 68-Lead PLCC	*XC3030*-100PC68C
ATT3030-125M68	3000 Gate 125 MHz 68-Lead PLCC	*XC3030*-125PC68C
ATT3030-150M68	3000 Gate 150 MHz 68-Lead PLCC	N/A
ATT3030-70M84	3000 Gate 70 MHz 84-Lead PLCC	*XC3030*-70PC84C
ATT3030-100M84	3000 Gate 100 MHz 84-Lead PLCC	*XC3030*-100PC84C
ATT3030-125M84	3000 Gate 125 MHz 84-Lead PLCC	*XC3030*-125PC84C
ATT3030-150M84	3000 Gate 150 MHz 84-Lead PLCC	N/A
ATT3030-70J100	3000 Gate 70 MHz 100-Lead PQFP	*XC3030*-70PQ100C
ATT3030-100J100	3000 Gate 100 MHz 100-Lead PQFP	*XC3030*-100PQ100C
ATT3030-125J100	3000 Gate 125 MHz 100-Lead PQFP	N/A
ATT3030-150J100	3000 Gate 150 MHz 100-Lead PQFP	N/A
ATT3030-70R84	3000 Gate 70 MHz 84-Lead CPGA	*XC3030*-70PG84C
ATT3030-100R84	3000 Gate 100 MHz 84-Lead CPGA	*XC3030*-100PG84C
ATT3030-125R84	3000 Gate 125 MHz 84-Lead CPGA	N/A
ATT3030-150R84	3000 Gate 150 MHz 84-Lead CPGA	N/A
ATT3042-70M84	4200 Gate 70 MHz 84-Lead PLCC	*XC3042*-70PC84C
ATT3042-100M84	4200 Gate 100 MHz 84-Lead PLCC	*XC3042*-100PC84C
ATT3042-125M84	4200 Gate 125 MHz 84-Lead PLCC	*XC3042*-125PC84C
ATT3042-150M84	4200 Gate 150 MHz 84-Lead PLCC	*XC3042*-125PC84C
ATT3042-70J100	4200 Gate 70 MHz 100-Lead PQFP	*XC3042*-70PQ100C
ATT3042-100J100	4200 Gate 100 MHz 100-Lead PQFP	*XC3042*-100PQ100C
ATT3042-125J100	4200 Gate 125 MHz 100-Lead PQFP	N/A
ATT3042-150J100	4200 Gate 150 MHz 100-Lead PQFP	N/A
ATT3042-70H132	4200 Gate 70 MHz 100-Lead PPGA	*XC3042*-70PP132C
ATT3042-100H132	4200 Gate 100 MHz 100-Lead PPGA	*XC3042*-100PP132C
ATT3042-125H132	4200 Gate 125 MHz 100-Lead PPGA	N/A
ATT3042-150H132	4200 Gate 150 MHz 100-Lead PPGA	N/A
ATT3042-70R84	4200 Gate 70 MHz 84-Lead CPGA	*XC3042*-70PG84C
ATT3042-100R84	4200 Gate 100 MHz 84-Lead CPGA	*XC3042*-100PG84C
ATT3042-125R84	4200 Gate 125 MHz 84-Lead CPGA	N/A
ATT3042-150R84	4200 Gate 150 MHz 84-Lead CPGA	N/A
ATT3042-70N100	4200 Gate 70 MHz 100-Lead CQFP	*XC3042*-70CQ100C
ATT3042-100N100	4200 Gate 100 MHz 100-Lead CQFP	*XC3042*-100CQ100C
ATT3042-125N100	4200 Gate 125 MHz 100-Lead CQFP	N/A
ATT3042-70R132	4200 Gate 70 MHz 132-Lead CPGA	*XC3042*-70PG132C
ATT3042-100R132	4200 Gate 100 MHz 132-Lead CPGA	*XC3042*-100PG132C
ATT3042-125R132	4200 Gate 125 MHz 132-Lead CPGA	N/A
ATT3042-150R132	4200 Gate 150 MHz 132-Lead CPGA	N/A
ATT3064-70M84	6400 Gate 70 MHz 84-Lead PLCC	*XC3064*-70PC84C
ATT3064-100M84	6400 Gate 100 MHz 84-Lead PLCC	*XC3064*-100PC84C
ATT3064-125M84	6400 Gate 125 MHz 84-Lead PLCC	N/A
ATT3064-150M84	6400 Gate 150 MHz 84-Lead PLCC	N/A
ATT3064-70R132	6400 Gate 70 MHz 132-Lead CPGA	*XC3064*-70PG132C
ATT3064-100R132	6400 Gate 100 MHz 132-Lead CPGA	*XC3064*-100PG132C
ATT3064-125R132	6400 Gate 125 MHz 132-Lead CPGA	N/A
ATT3064-150R132	6400 Gate 150 MHz 132-Lead CPGA	N/A
ATT3064-70H132	6400 Gate 70 MHz 132-Lead PPGA	*XC3064*-70PP132C
ATT3064-100H132	6400 Gate 100 MHz 132-Lead PPGA	*XC3064*-100PP132C
ATT3064-125H132	6400 Gate 125 MHz 132-Lead PPGA	N/A
ATT3064-150H132	6400 Gate 150 MHz 132-Lead PPGA	N/A
ATT3064-70J160	6400 Gate 70 MHz 160-Lead PQFP	*XC3064*-70PQ160C

ATT3000-Series FPGA Cross-Reference Guide (continued)

AT&T Part Number	Description	*Xilinx* Part Number
	Industrial Temperature Range (–40 °C to +85 °C)	
ATT3064-100J160	6400 Gate 100 MHz160-Lead PQFP	*XC3064*-100PQ160C
ATT3064-125J160	6400 Gate 125 MHz 160-Lead PQFP	N/A
ATT3064-150J160	6400 Gate 150 MHz 160-Lead PQFP	N/A
ATT3090-70M84	9000 Gate 70 MHz 84-Lead PLCC	*XC3090*-70PC84C
ATT3090-100M84	9000 Gate 100 MHz 84-Lead PLCC	*XC3090*-100PC84C
ATT3090-125M84	9000 Gate 125 MHz 84-Lead PLCC	*XC3090*-125PC84C
ATT3090-150M84	9000 Gate 150 MHz 84-Lead PLCC	N/A
ATT3090-70J160	9000 Gate 70 MHz 160-Lead PQFP	*XC3090*-70PQ132C
ATT3090-100J160	9000 Gate 100 MHz 160-Lead PQFP	*XC3090*-100PQ132C
ATT3090-125J160	9000 Gate 125 MHz 160-Lead PQFP	N/A
ATT3090-150J160	9000 Gate 150 MHz 160-Lead PQFP	N/A
ATT3090-70N164	9000 Gate 70 MHz 164-Lead CQFP	*XC3090*-70CQ164C
ATT3090-100N164	9000 Gate 100 MHz 164-Lead CQFP	*XC3090*-100CQ164C
ATT3090-125N164	9000 Gate 125 MHz 164-Lead CQFP	N/A
ATT3090-70H175	9000 Gate 70 MHz 175-Lead PPGA	*XC3090*-70PP175C
ATT3090-100H175	9000 Gate 100 MHz 175-Lead PPGA	*XC3090*-100PP175C
ATT3090-125H175	9000 Gate 125 MHz 175-Lead PPGA	*XC3090*-125PP175C
ATT3090-150H175	9000 Gate 150 MHz 175-Lead PPGA	N/A
ATT3090-70R175	9000 Gate 70 MHz 175-Lead CPGA	*XC3090*-70PG175C
ATT3090-100R175	9000 Gate 100 MHz 175-Lead CPGA	*XC3090*-100PG175C
ATT3090-125R175	9000 Gate 125 MHz 175-Lead CPGA	*XC3090*-125PG175C
ATT3020-70M68I	2000 Gate 70 MHz 68-Lead PLCC	*XC3020*-70PC68I
ATT3020-100M68I	2000 Gate 100 MHz 68-Lead PLCC	N/A
ATT3020-70M84I	2000 Gate 70 MHz 84-Lead PLCC	*XC3020*-70PC84I
ATT3020-100M84I	2000 Gate 100 MHz 84-Lead PLCC	N/A
ATT3020-70R84I	2000 Gate 70 MHz 84-Lead CPGA	*XC3020*-70PG84I
ATT3020-100R84I	2000 Gate 100 MHz 84-Lead CPGA	N/A
ATT3020-70R100	2000 Gate 70 MHz 100-Lead PQFP	*XC3020*-70PQ100I
ATT3020-100R100	2000 Gate 100 MHz 100-Lead PQFP	N/A
ATT3030-70M44I	3000 Gate 70 MHz 44-Lead PLCC	*XC3030*-70PC44I
ATT3030-100M44I	3000 Gate 100 MHz 44-Lead PLCC	N/A
ATT3030-70M68I	3000 Gate 70 MHz 68-Lead PLCC	*XC3030*-70PC68I
ATT3030-100M68I	3000 Gate 100 MHz 68-Lead PLCC	N/A
ATT3030-70M84I	3000 Gate 70 MHz 84-Lead PLCC	*XC3030*-70PC84I
ATT3030-100M84I	3000 Gate 100 MHz 84-Lead PLCC	N/A
ATT3030-70J100I	3000 Gate 70 MHz 100-Lead PQFP	*XC3030*-70PQ100I
ATT3030-100J100I	3000 Gate 100 MHz 100-Lead PQFP	N/A
ATT3030-70R84I	3000 Gate 70 MHz 84-Lead CPGA	*XC3030*-70PG84I
ATT3030-100R84I	3000 Gate 100 MHz 84-Lead CPGA	N/A
ATT3042-70M84I	4200 Gate 70 MHz 84-Lead PLCC	*XC3042*-70PC84I
ATT3042-100M84I	4200 Gate 100 MHz 84-Lead PLCC	N/A
ATT3042-70J100I	4200 Gate 70 MHz 100-Lead PQFP	*XC3042*-70PQ100I
ATT3042-100J100I	4200 Gate 100 MHz 100-Lead PQFP	N/A
ATT3042-70R84I	4200 Gate 70 MHz 84-Lead CPGA	*XC3042*-70PG84I
ATT3042-100R84I	4200 Gate 100 MHz 84-Lead CPGA	N/A
ATT3042-70H132I	4200 Gate 70 MHz 132-Lead PPGA	*XC3042*-70PP132I
ATT3042-100H132I	4200 Gate 100 MHz 132-Lead PPGA	N/A
ATT3042-70R132I	4200 Gate 70 MHz 132-Lead CPGA	*XC3042*-70PG132I
ATT3042-100R132I	4200 Gate 100 MHz 132-Lead CPGA	N/A
ATT3064-70R132I	6400 Gate 70 MHz 132-Lead CPGA	*XC3064*-70PG132I
ATT3064-100R132I	6400 Gate 100 MHz 132-Lead CPGA	N/A
ATT3064-70H132I	6400 Gate 70 MHz 132-Lead PPGA	*XC3064*-70PP132I
ATT3064-100H132I	6400 Gate 100 MHz 132-Lead PPGA	N/A
ATT3090-70H175I	9000 Gate 70 MHz 175-Lead PPGA	*XC3064*-70PP175I
ATT3090-100H175I	9000 Gate 100 MHz 175-Lead PPGA	N/A
ATT3090-70R175I	9000 Gate 70 MHz 175-Lead CPGA	*XC3090*-70PG175I
ATT3090-100R175I	9000 Gate 100 MHz 175-Lead CPGA	N/A
ATT–DS22–PC1	PC–*SILOS* Simulator – 16K Gates	XC–DS222–PC1
ATT–DS290–PC1	*Viewlogic Viewsim* Simulator with XNF Timing Delay Interface	XC–DS290–PC1
ATT–DS295–SN2	*Sun4 Viewdraw–LCA, Viewsim, Viewwave* with XNF Interface & Library	N/A
ATT–DS31–PC1	XNF Interface & Library for *FutureNet DASH* Schematic Editor	XC–DS31–PC1
ATT–DS310–PC1	*FutureNet DASH–LCA* Schematic Editor, XNF Interface, & Library	XC–DS310–PC1
ATT–DS343–AP1	Interface & Library for *Mentor Graphics NetED* Seatic Editor and *Qsim* Simulator	XC–DS343–AP1
ATT–DS35–PC1	XNF Interface & Library for *Orcad SDT* Schematic Editor	XC–DS35–PC1
ATT–DS390–PC1	*Viewlogic Viewdraw–LCA* Schematic Editor, XNF Interface, & Library	XC–DS390–PC1
ATT–DS391–PC1	PC XNF Interface & Library for *Viewlogic Viewdraw* & *Viewsim*	XC–DS391–PC1
ATT–DS391–SN1	*Sun3* XNF Interface & Library for *Viewlogic Viewdraw* & *Viewsim*	XC–DS391–SN1
ATT–DS391–SN2	*Sun4* XNF Interface & Library for *Viewlogic Viewdraw* & *Viewsim*	XC–DS391–SN2
ATT–DS501–PC1	PC *XACT* Design Implementation System for ATT3000 Series	XC–DS501–PC1
ATT–DS501–AP1	*Apollo XACT* Design Implementation for ATT3000 Series	XC–DS501–AP1
ATT–DS501–SN1	*Sun3 XACT* Design Implementation for ATT3000 Series	XC–DS501–SN1
ATT–DS501–SN2	*Sun4 XACT* Design Implementation for ATT3000 Series	XC–DS501–SN2
DS–MANUALS	3K Series Complete Documentation	XC–Manual–3K

Section 2.
FPGA Qualification Information

2. FPGA Qualification Information

Table of Contents

Contents **Page**

Contents **Page**

AT&T Microelectronics

FPGA Qualification Information
Quality and Reliability Information

Introduction

AT&T Microelectronics' quality and reliability programs and resources, featured in this section, support state-of-the-art IC manufacturing. Our research, design, and fabrication processes are combined with rigorous sampling and testing procedures to pursue absolute reliability in every device we ship.

To bring our commitment full circle, AT&T Microelectronics' emphasis on customer service and feedback ensures immediate response and timely solutions in case of device shortcomings after delivery.

TQM and the QML Approach

Total quality management (TQM) is a business methodology that involves all employees in the process of satisfying customers. The AT&T Microelectronics MOS Business Unit has set a goal of **zero defects and 100% on-time delivery to customer requirements**. It is committed to an excellence of service from initial customer contact, through development and design, into manufacturing, and, to sustained support after the sale. MOS is using TQM to drive customer satisfaction.

The qualified manufacturer's list (QML) concept is one of the latest trends in the electronics industry. It is modeled on the benchmark publication *General Specification of Integrated Circuits (Microcircuits) Manufacturing,* MIL-I-38535, produced in December, 1989, through a joint project involving AT&T, General Electric, and Honeywell. AT&T Microelectronics was the first company to meet the new government standards specified in this document.

QML is currently practiced for products destined for government markets, but most of its disciplines are also applied to AT&T's commercial products.

QML's fundamental concept is to instill TQM throughout the product delivery process. This requires the full commitment of everyone involved at each stage of the process, and from every organization participating.

An important component of the QML approach is the technical review board (TRB). AT&T's TRB is responsible for the control, stabilization, monitoring, and improvement of the qualified technology. It is accountable to both the customer and government for all QML products shipped. Qualification plans are developed for new products and for design, material, and manufacturing enhancements to existing products.

AT&T's TRB includes representatives from marketing, development, design, and manufacturing. The TRB develops a Total Quality Management plan (TQMP) which specifies how the manufacturing operation for a given technology is controlled, monitored, and improved throughout its life cycle. The TQMP includes:

- TRB membership and responsibilities
- Review of customer requirements
- Design methodology
- Controlled manufacturing processes
- Statistical Process Control (SPC)
- Qualification
- Quality conformance inspection (QCI)
- Training
- Change control
- Customer notification
- Reliability/quality/yield monitoring and improvement
- Auditing
- Failure Mode Analysis (FMA)
- Corrective action

Several features of AT&T's QML process are of special interest. Process controls include AT&T's QML line monitoring program and incorporate specialized software and manufacturing tools, such as the Standard Evaluation Circuit (SEC), Tester for Reliability and Yield Components (TRYC), and the Process (Parametric) Monitor (PM), which enable the TRB to control and improve the technology of the manufacturing process on an ongoing basis.

Quality Improvement

Quality improvement, as achieved by the yield enhancement system (YES), is an aggressive, proactive program aimed at improving yields and reliability of the MOS manufacturing product lines. The YES project's goals include measuring and modeling yield in order to provide process engineers with the tools to efficiently collect data at significant points of the manufacturing process, and guiding quality improvement activities. AT&T's 1.25 µm and 0.9 µm MOS two-level metal technologies are part of the YES program.

Key tools designed for the YES program include a line monitor, zone monitors, a data management system, and a yield model. Each is described below.

Line Monitor

The primary element of the YES program is the line monitor, a device which receives and evaluates the full integrated circuit process from design through packaging.

The line monitor has three components: a Standard Evaluation Circuit (SEC), a Tester for Reliability and Yield Components (TRYC), and a Process (Parametric) Monitor (PM).

AT&T's SEC is a specially designed, highly diagnosable memory with DRAM and SRAM arrays whose yield can be monitored. The SEC is also run with monitored burn-in and extended life tests for periodic reliability evaluation of MOS product lines.

The TRYC contains structures for measuring defect densities to arrive at a correlation between direct measurement of defect density and SEC yield. It also contains testers for evaluating intrinsic reliability. These structures provide for the characterization of electromigration of metal 1 and metal 2 runners and contacts, time-dependent dielectric breakdown of gate oxides, hot carrier aging, and mobile ion contamination.

The PM consists of structures in the kerf or grid which measure electrical parameters such as threshold voltage, linear gain, and leakage current.

Zone Monitors

Although the line monitor provides an excellent evaluation of the complete fabrication process, often faster feedback is required. This is achieved by the zone monitors, test structures fabricated in parts (zones) of the process to measure the defect density of a particular processing segment.

Each zone monitor measures the defect density for particular failure modes. Since it addresses specific performance, isolated defects are identified, and the overall sensitivity of the measurement is enhanced. All YES tools permit fast turnaround and are used for defect density reduction experiments plus routine monitoring of portions of the process.

Yield Model

A yield model has been developed to analyze the effects of defect densities on product yields. The results this model provides also help to prioritize defect reduction projects.

Summary

The YES program plays a vital role in the quality control program of AT&T's wafer fabrication facilities. Feedback data on process defect density and reliability obtained from the YES tools is part of the continuous quality improvement program in place at AT&T Microelectronics.

YES concepts have been approved by government centers with which AT&T has a manufacturing supply relationship. In fact, AT&T became the first company to achieve Qualified Manufacturer's List (QML) status as a government supplier of 1.25 µm CMOS integrated circuits. AT&T has completed QML qualification for 0.9 µm CMOS devices as well. AT&T's use of YES tools will continue to contribute significantly to the fulfillment of the QML concept.

Additional AT&T process controls include the SPC program, equipment checks, and zone monitors. The SPC program is detailed in a following section; equipment checks include particle generation and calibration. The use of these process controls, combined with in-line product monitoring and testing, helps ensure continual quality improvement and detection and removal of defective material. The result: optimum performance and yield.

Quality Improvement (continued)

Summary (continued)

In summary, the QML process effectively promotes the control and improvement of product yield, quality, and reliability. It is a highly disciplined process, featuring certification, qualification, and auditing by the government.

QML is successfully concluded when two stages have been completed: certification and qualification of the technology. Certification is attained when the government recognizes evidence, involving validation reviews (audits), that the manufacturer is capable of designing and producing microcircuits of high quality.

Qualification is the actual demonstration of the certified manufacturer's capabilities: the actual production of first-pass microcircuits. Ongoing monitoring, including government reviews, is required to maintain QML status.

Thus, AT&T's position as an innovator and leader in QML programs helps ensure product quality and solid performance benefit to our customers. QML results in built-in quality, consistent manufacturability, and yield, while simultaneously boosting cost-effectiveness in programs and operations, and producing improved shipping performance to further enhance customer satisfaction.

Statistical Process Control (SPC)

Statistical process control (SPC) is a key component of AT&T Microelectronics' quality program. As the flowchart in Figure 2-1 illustrates, the SPC plan progresses in three phases: data collection (monitoring), process control (performance studies), and process improvement (process capability). Typical areas under SPC monitoring during wafer fabrication and package assembly are illustrated in Table 2-1, Table 2-2, Table 2-3, and Table 2-4. Quality improvement teams (QITs), composed of the process engineers, shop supervisors, and quality control engineers, identify the critical nodes (processes). The quality control engineer is responsible for completing performance studies on all nodes, and calculating process capability (Cp) indices for each.

The QIT seeks to reduce variation at all critical nodes beginning with those having process capability indices of less than 1.33. Critical nodes are identified by engineering judgment and customer feedback. The team then applies its energies to reducing variation at nodes having values of less than 2.0.

AT&T's goal is to become a "6 sigma" manufacturer by 1995. To accomplish this goal, the team uses experimental design techniques, Pareto analyses, distributions, correlation studies, control chart patterns, and process capability studies, as detailed in the *AT&T Statistical Quality Control Handbook*.

In addition to driving process improvements, the team has procedures for corrective action on all quality-related problems within its area of responsibility. It evaluates the potential impact of each problem in terms of customer satisfaction, performance, reliability, safety, and cost. Cause and effect are determined, and significant variables are identified.

The team's control extends to remedial action both on work in progress and on devices already shipped. If indicated, recall procedures and decision processes are implemented without delay, in order to preserve customer confidence.

Based on its findings, the team recommends and implements changes to manufacturing, packing, shipping, and storage processes, or revises specifications or the quality system itself. The team then tracks the successful implementation of its recommendations and monitors the results.

Statistical Process Control (SPC) (continued)

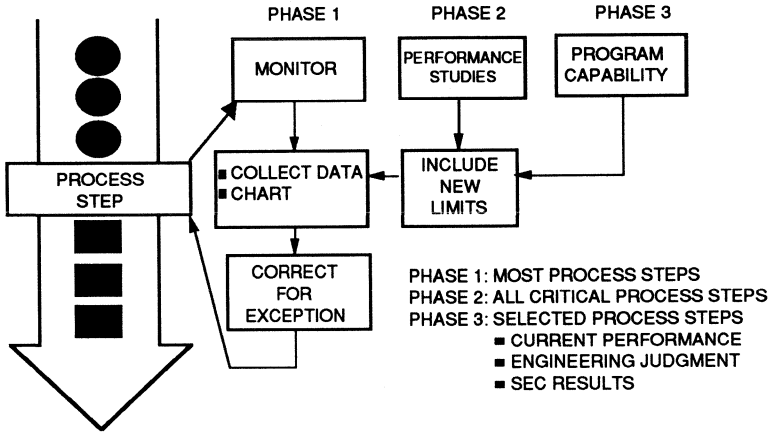


Figure 2-1. The SPC Process Flow

To accomplish these goals, the team uses experimental design techniques, distributions, correlation studies, control chart patterns, and process capability studies, as detailed in the *AT&T Statistical Quality Control Handbook*.

In addition to driving process improvements, the team has procedures for corrective action on all quality-related problems within its area of responsibility. It evaluates the potential impact of each problem in terms of customer satisfaction, performance, reliability, safety, and cost. Cause and effect are determined, and significant variables are identified.

The team's control extends to remedial action both on work in progress and on devices already shipped. If indicated, recall procedures and decision processes are implemented without delay, in order to preserve customer confidence.

Based on its findings, the team recommends and implements changes to manufacturing, packing, shipping, and storage processes, or it revises specifications or the quality system itself. The team then tracks the successful implementation of its recommendations and monitors the results.

Change Control

The process by which changes are implemented in AT&T Microelectronics is called change control. It is communicated through the change control document, also called the change order (CO), and administered by the quality management team. The change control process flow is shown in Figure 2-2.

Major changes in the design, fabrication, assembly, and packaging of devices are evaluated. In addition, changes to test facilities are also evaluated and tracked by the CC process. Figure 2-2 illustrates the process.

A new or changed process qualification plan is developed by a qualification review board (QRB) composed of a group of engineers and designers for each product in question. The group conducts the plan with the support and concurrence of the other group members.

Change Control (continued)

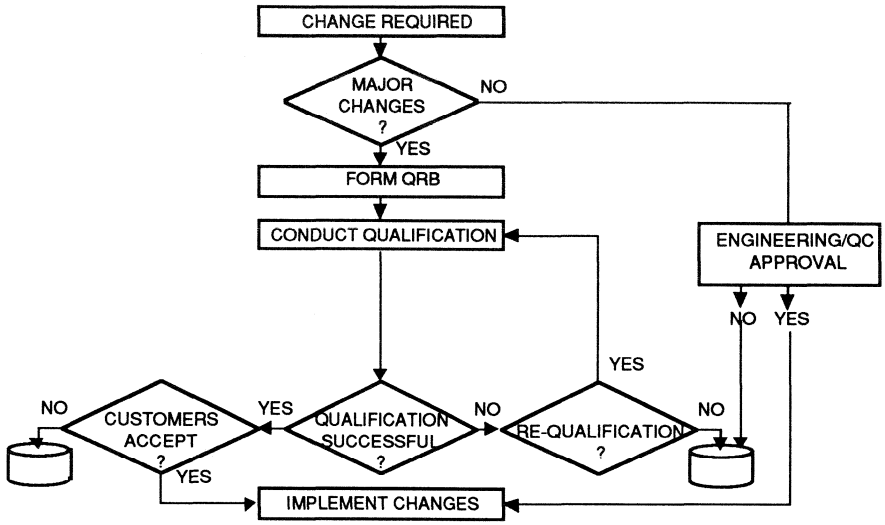


Figure 2-2. Change Control Process Flow

Failure Mode Analysis (FMA)

Failure mode analysis (FMA) is a comprehensive procedure which determines the cause of IC failures that occur during manufacturing, qualification, or reliability monitoring. Special attention and high priority are assigned to customer returns.

AT&T Microelectronics' FMA laboratory is responsible for conducting FMA analysis. The laboratory is staffed by highly trained personnel who, through training and experience, have distinguished themselves as specialists.

Devices sent to the FMA lab are initially tested to confirm failure and determine test failure signature. The FMA engineer analyzes the results and determines the nature of the failure (opens or shorts, parametric, or functional). The engineer may elect to perform full characterization and schmoo plots as well. When test results are complete, the engineer proceeds with the FMA.

Parametric failures and opens, shorts, or leakage are verified by a curve tracer or parametric analyzer.

Functional failures of operating devices include pattern sensitivity, loss of voltage range, and timing failures. Bit maps are used to pinpoint these failures.

Functional failures are subjected to analysis of the die surface. The die is exposed using a decapsulation technique known not to compromise the bond pad or wire bond integrity.

After decapsulation, inspections are performed in the suspected areas. Layout and circuit schematics are used to confirm defects. If further analysis is needed, a systematic removal of the device layers is done, with a detailed inspection performed after each etch.

Failure Mode Analysis (FMA)
(continued)

The Analytical Lab

When more sophisticated analytical techniques are needed, the AT&T/Bell Laboratories analytical lab is used to conduct further FMA.

The analytical lab's facilities span the fields of optical and electron microscopy, ion beam techniques, and traditional analytical chemistry. Its staff is eminently qualified and equipped to perform any of a battery of more than 50 complex tests under controlled conditions.

Customer Service and the Device Quality Issue (DQI) Process

Customer satisfaction is a top priority in AT&T's quality effort. AT&T has created a customer technical support center (CTSC) to administer the DQI process. The DQI process, shown in the flowchart in Figure 2-3, was created to respond to customer-reported problems and achieve rapid and complete solutions. The DQI process is a full-circle approach to closing the customer feedback loop. It ensures that a customer-indicated device arriving for FMA will be fully monitored and documented, and that marketing organizations and the customer are included as vital links in the feedback process.

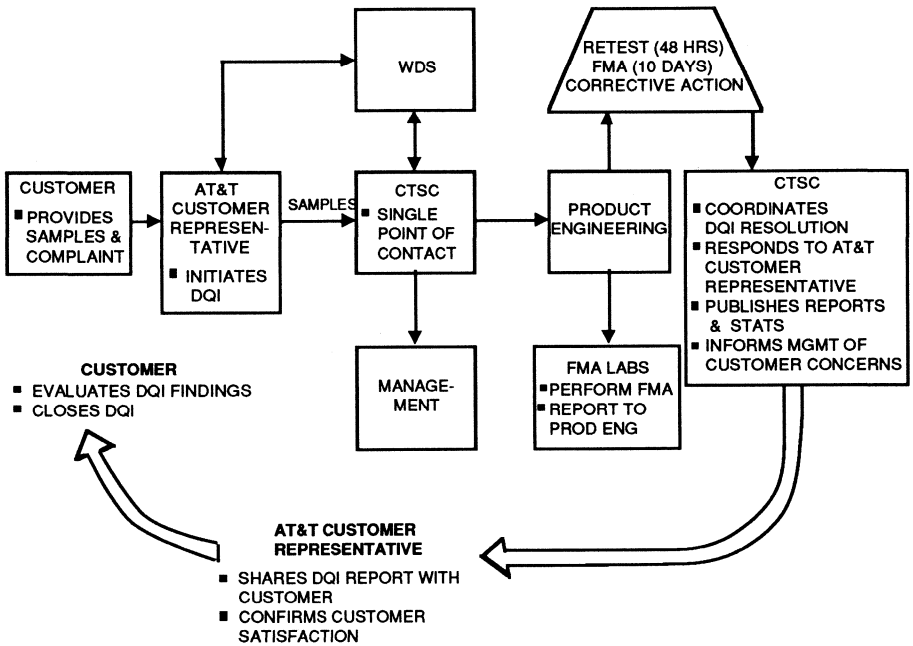


Figure 2-3. The Device Quality Issue (DQI) Full-Circle Process Flow

Product Qualification Plan

AT&T's product qualification plan for ATT3000 Series FPGAs involves rigorous environmental, mechanical, and electrical testing to confirm product soundness.

Industry-standard tests are applied to test devices. Specialized tests to examine electrostatic discharge and latch-up parameters are also incorporated.

ATT3000 Series FPGAs are qualified by a three-phase process: AT&T's 0.9 µm CMOS process, the design of the device itself, and the performance of a given die in a given package.

Since all ATT3000 Series FPGAs are manufactured using the AT&T 0.9 µm CMOS process, the process is carefully monitored and tested. Regarding device design, since each of the five devices in the ATT3000 Series FPGAs is a matrix of repetitive elements, the qualification review board (QRB) determined that qualifying one of the five dies would qualify the design for the product family.

Finally, the QRB determined the qualification testing necessary to qualify each die in a given package. Industry-standard tests were chosen, based on prior qualifications with the given package for prior 0.9 µm CMOS designs.

For example, if a prior qualification of a 0.9 µm CMOS design had been performed with a larger die than the one in question, the only new tests needed would be electrostatic discharge (ESD) and latch-up (LU). However, if the die to be qualified was the largest 0.9 µm CMOS design to be put in that package, more extensive testing would be required. A table of the tests performed is included in the section on package qualification.

0.9 µm CMOS Process Qualification Results

Introduction

This section presents quality and reliability information for AT&T's 0.9 µm advanced CMOS process, which was transferred to SEMATECH as its baseline process.

AT&T's 0.9 µm CMOS process employs N- and P-channel LDD MOS transistors with a typical electrical gate length of 0.75 µm. The process also uses two levels of metal and one level of polycide. Source and drain regions are silicided for low resistance.

This technology has been rigorously tested for reliability and manufacturability. Prior to qualification and manufacture of devices using this process technology, approximately 10,000 devices were subjected to extended reliability tests and extensive generic reliability tests.

After introduction into manufacture, a continuous quality improvement program was begun. This report concentrates on the latest version of the technology, Issue 9. New improvements are introduced into the technology, with new issues of the technology put into effect at a rate of about one per year.

Test Codes

Reliability studies have been performed on two codes: a standard evaluation circuit (SEC) and a test chip TC19A .

The SEC contains a 16K DRAM and 32K of SRAM. It also contains large test structures to measure the defect densities at various levels, such as metal.

In this test chip, the total yield of the SEC can be correlated to the yield components measured in the large test structures. These structures are called TRYC (tester for yield and reliability components) circuits. This circuit was fabricated in a large, plastic dual-in-line package.

The TC19A is a 64K SRAM which has been used extensively in the development of the 0.9 µm technology and was used as the initial manufacturing demonstration vehicle by SEMATECH. The design rules in this chip are representative of design rules used on memory products.

0.9 µm CMOS Process Qualification Results (continued)

Testing

Much of the evaluation of this technology is done with a standard evaluation circuit (SEC). The SEC is tested with a very rigorous methodology, intended to ensure that the device functions over a full voltage range from 4 V to 6.5 V at 85 °C.

A 7 V stress test is included to detect breakdown of any weak oxides. The device is required to be functional during this stress test. Functional testing is done before and after stress testing to determine which parts, if any, succumbed to the stress test. Leakage currents are tested with an upper limit of 10 µA. Other tests are performed to detect low-level leakage at a transistor level; these include a variety of hold time tests with various patterns.

The same test routine is performed at least three times: at wafer probe, after packaging, and at the end of reliability testing. Hold time and voltage guard band limits are built into the testing.

Generic Reliability Data

Electromigration

Electromigration is a well-known failure mechanism which affects continuity or isolation of interconnect. The interconnect is allowed a maximum failure rate budget for all interconnect of 10 FITS (maximum) at a junction temperature of 85 °C over 40 years.

Failure mechanisms follow a log-normal distribution for this failure mechanism. Activation energies, sigmas, and median times to failure are measured for each level separately in the interconnect structure. For failure rate calculations, it was assumed that the chip was dissipating 2 W of power.

Table 2-1 summarizes measurements of metal reliability of Issue 9 of the 0.9 µm process, using the SEC as a test vehicle. These measurements were made with current levels 10 to 20 times higher than the maximum allowed by the current density design rules.

For Issue 7 of the technology, a refractory metal layer has been included as part of the Metal 1 structure. As a result, the median times to failure have increased by a factor of four to 10 for contacts and Metal 1. Also, activation energies for these levels have increased significantly. As a result, current density limits are higher and predicted failure rates are lower for Issue 7 of the 0.9 µm technology. No statistically significant difference was observed for the sigma of the failure distributions of Issue 7 compared to Issue 6.

Metal Notching

The effect of notching on electromigration was studied for the technology. Metal notching was induced by a 90-hour bake at a temperature of 180 °C and 210 °C. No degradation in the electromigration performance was noted on the samples studied.

Hot Electron Effects

Device aging on the SEC showed greater than 10 years to 10% G_m degradation under worst-case substrate current conditions. Actual devices should not be affected by hot electron induced failure mechanisms.

Mobile Ion Contamination

Mobile ion contamination was checked on the SEC at wafer level on two lots per the standard shop procedure. Both lots passed this test with no signs of contamination.

Table 2-1. Summary of Electromigration for Issue 9 of the 0.9 µm CMOS Process

Structure	Activation Energy (eV)	Temperature (°C)	MTTF* (hours)	σ	λ (FITS)†
Metal 1	0.92	200	55,925	0.93	8E-11
Metal 2	0.70	250	21,306	1.18	7E-4
Via	0.70	250	2,431	0.48	3E-10
Contact	1.00	270	3,775	0.42	9.5E-20
Total					7E-4

* MTTF is the median time to fail at use conditions (80 °C, 2E+5 A/cm² for M1, 1E+5 A/cm² for M2, i = 1.1 mA for contacts and vias).

† Failure rates for vias and contacts are multiplied by a factor of 1000 to scale tester results to circuit. λ is calculated under the same conditions at 40 years.

Process Qualification

High Temperature Operating Bias (HTOB)

The main vehicle for studying HTOB is the standard evaluation circuit (SEC). This circuit contains two 16K blocks of DRAM (N and P type) and two 16K blocks of SRAM. Results on this code are given in Table 2-2.

After testing, the failed parts were examined with physical FMA techniques. On the SEC, electrical failure modes are easily correlated with failure locations. Table 2-3 through Table 2-5 show the results of physical failure mode analysis.

The failing parts were taken through electrical and physical FMA. The results of the physical and electrical FMA are given in Table 2-6. Gate-level defects are caused by silicon particles at the gate level. Several programs are now in place to attack gate-level defects.

Table 2-2. SEC Infant Mortality and Long-Term Reliability Results

Issue	Sample Size	Burn-In Failure Rate (24 hours, 6.2 V, 150 °C)	Sample Size	Long-Term Failure Rate (1,000 hour, 5.8 V, 150 °C)
7	599	0.5%	221	0.5%

Table 2-3. SEC Burn-In and HTOB FMA

# Devices = 221 Plastic T = 150 °C Vcc = * Start Date = 12/27/89 End Date = 2/2/90		
t Failure Hours	**% Fail**	**Failure Mode Analysis**
†	0.45	**Elec. FMA:** P-DRAM—High standby currents **Phys. FMA:** Isolated to one finger of protection circuit, M2 bridging

* 6.2 V for 24-hour burn-in; then lowered to 5.8 V.
† Electrical test failure at 1002 hours; devices were not Teradyne tested after 24 hours of burn-in.

Table 2-4. SEC Infant Mortality FMA

# Devices = 378 Plastic T = 150 °C Vcc = 6.2 V Start Date = 1/18/90 End Date = 1/19/90		
t Failure Hours	**% Fail**	**Failure Mode Analysis**
2	0.26	**Elec. FMA:** SRAM1—Full COL # 71 fails 1s **Phys. FMA:** Particle shorting M2 to M1 col true to Vss
*	0.53	**Elec. FMA:** High standby currents **Phys. FMA:** Cannot be localized for physical FMA

* Failed electrical test at 24 hours.

Table 2-5. TC19A Infant Mortality and Long-Term Reliability Results

Issue	Sample Size	Burn-In Failure Rate (24 hours, 6.2 V, 150 °C)	Sample Size	Long-Term Failure Rate (1,000 hour, 5.8 V, 150 °C)
7	283	1.1%	280	1.1%

Process Qualification (continued)

High Temperature Operating Bias (HTOB) (continued)

Table 2-6. Failure Mode Analysis

# Devices = 285 Plastic, T = 150 °C, Vcc = 6.2 V/5.8 V		
t Failure Hours	**% Fail**	**Failure Mode Analysis**
0.0	0.35	**Elec. FMA:** Upper memory array dead **Phys. FMA:** Nothing found
0.0	0.70	**Elec. FMA:** Single column **Phys. FMA:** 1.5 µm particle at M2
0.9	1.05	**Elec. FMA:** Standby array current 160 µA at 85 °C **Phys. FMA:** Nothing found
310	0.35	**Elec. FMA:** Single bit **Phys. FMA:** Gate-level defect
344	0.70	**Elec. FMA:** Single bit **Phys. FMA:** Gate-level defect
654	1.05	**Elec. FMA:** Single column **Phys. FMA:** Nothing found

Temperature Humidity Bias (THB)

Temperature humidity bias testing was performed on both the TC19A and the SEC per AT&T requirements of static bias at 85 °C and 85% relative humidity. One processing lot of the SEC showed two failures. THB is known to induce mechanical stress effects due to expansion of the plastic due to water absorption.

Physical FMA of these parts showed the failures were related to die attach, similar to the failures for temperature cycling. Table 2-7 summarizes these results.

Table 2-7. Temperature Humidity Bias Results

Code	Sample Size	Failures	Accept Number
SEC	59	0	—
	150	2	—
Total	209	2	2
TC19A (360 hrs)	225	0	3

Temperature Cycling

Temperature cycling was performed per MIL-STD-883C, Method 1010, Condition C. The 100 cycle results are given in Table 2-8.

Table 2-8. Temperature Cycling Results

Code	Sample Size	Failures	Accept Number
SEC	105	2	3
TC19A	105	0	2

Physical FMA of the failures showed crazing of the aluminum metal 2. According to FMA, this was due to package stresses. These stresses were higher than normal due to an irregularity with die attach.

Process Qualification (continued)

Thermal Shock (TS)

Thermal shock was performed per MIL-STD-883 Method-1011, which requires –55 °C to +125 °C, liquid to liquid, for 15 cycles. A sample of 35 SEC samples was tested. This lot had only one failure.

In addition, 34 samples of the TC19A were evaluated. This sample had no failures. Details of the test results are given in Table 2-9.

Table 2-9. Thermal Shock Results

Code	Sample Size	Failures	Accept Number
SEC	20	1	—
	15	0	—
Total	35	1	2
TC19A	34	0	2

FMA of the thermal shock failure showed crazing of the passivation similar to that reported in the segment on temperature cycling failures.

Steam Bomb

Steam bomb testing was run on both the SEC and TC19A test chips. No failures were observed out of 137 devices. Details are summarized in Table 2-10. See Autoclave in the Device Reliability Test Results section for a description of the test conditions.

Table 2-10. Steam Bomb Results

Code	Sample Size	Failures	Accept Number
SEC	60	0	—
	72	0	—
Total	132	0	3
TC19A	105	0	2

Electrostatic Discharge (ESD)

ESD testing was done on the test chip TC19A. Twelve devices were stressed. For the human-body model (HBM) procedure, only the worst-case pin combination was tested.

All six test devices passed 2,000 V, but failed 4,000 V. Six devices were tested for charged-device model (CDM), and all passed both positive and negative stressing up to 2,000 V.

Latch-Up (LU)

The TC19A device was checked for latch-up according to normal procedures. No latch-up was found on the three parts tested. See Latch-Up as per AT&T Method L757185 in the Device Reliability Test Results section.

Bond Strength (BS)

Bond strength was tested per the usual procedure in the models packaging shop at Allentown, PA. All packages for reliability evaluation were packaged at this shop. Results are given in Table 2-11.

Since the minimum allowed value is 5 grams, this test was passed.

Table 2-11. Bond Strength Test Results

Sites Measured	Mean Value (grams)	Standard Deviation (grams)	Minimum Measured Value (grams)
32	12.5	1.2	8.0
29	13.8	1.3	12.0
29	14.0	1.6	12.0

Die Shear

Die shear strength was measured on the same lots as listed in the section on bond strength. For 15 values measured, the minimum value was 54 grams, and the maximum value was 78 grams. Since the minimum allowed value is 30 grams, this test was passed.

Summary

This latest issue of the 0.9 µm CMOS technology meets or exceeds requirements for qualification in all critical areas.

Infant mortality and long-term failure rates have improved significantly with the lots processed with this log, and it is expected that further improvements will be made as part of a continuous quality improvement program.

Device Qualification Testing

Overview

AT&T's product quality program for ATT3000 Series FPGAs requires that the devices undergo a rigorous testing program prior to introduction. The program includes a series of life and environmental tests designed to accelerate failure probabilities in the die and package. As part of device qualification, the following tests are performed:

Environmental Tests

- High Temperature High Bias (HTHB) 150 °C, 6 V
- Component Lead Assembly Simulation Sequence (CLASS)
- Temperature Humidity Bias (THB) 85 °C/85% RH
- Autoclave (SB) 121 °C, 2 ATM
- Temperature Cycling (TC) –65 °C to +150 °C (air to air)
- Thermal Shock (TS) –55 °C to +125 °C (liquid to liquid)
- Moisture Resistance
- Salt Atmosphere (Corrosion)
- Low-Temperature Aging (LTA) –10 °C, 7 V

Mechanical Tests

- Flammability and O_2 Index
- Solvent Resistance
- Physical Dimensions (PD)
- Solderability
- Bond Strength
- Die Shear Strength
- X-ray

Electrical Tests

- Electrostatic Discharge (ESD) 1,000 V Each Pin
- Latch-up (LU)

The specific tests used to qualify devices such as the ATT3000 FPGA product family are described below. Test characteristics, parameters, allowed failure rates, and other details are included.

Failure mode analysis (FMA) is used as an integral part of quality testing and surveillance to assess any performance flaw, determine corrective measures, and implement these measures to eliminate recurrence. In a following section, the test results for the chosen FPGA device are presented.

Environmental Tests

High Temperature High Bias (HTHB)

High temperature high bias (HTHB) testing, also called dynamic life testing, is performed at 150 °C and 6 V to provide acceleration over the condition of use.

Dynamic operating life stress is considered to be more representative than static stress because the continual switching of internal circuit nodes more closely approximates the operation of the device in an actual system.

Component and Lead Assembly Simulation Sequence (CLASS)

This test involves heating, infrared solder simulation, and lead bending. It is a preconditioning test for THB.

Temperature Humidity Bias (THB)

Because a plastic package is inherently nonhermetic, the penetration of ambient moisture could damage the device due to galvanic action. Thus, temperature humidity bias (THB) and autoclave (steam bomb) tests are used to accelerate moisture ingression in order to determine the tolerance of the die to its presence.

THB testing is performed at an ambient temperature of 85 °C and a relative humidity of 85%. The sample devices are tested for 1,000 hours at 5 V, and go through presoak at 85/85 for 96 hours.

Device Qualification Testing (continued)

Environmental Tests (continued)

Autoclave

Autoclave, also known as steam bomb, is a storage test employing the environmental conditions of T_A = 121 °C, 100% relative humidity, and 15 psig. for 96 hours. This test is used as an additional stringent test to measure the moisture resistance of the packaging system and the susceptibility of the die to corrosion. As with THB testing, both package integrity and actual die construction play major roles in the results.

Temperature Cycle

The compatibility of materials used in the fabrication of any device is essential to that device's reliability. Any appreciable mismatch in physical properties, such as thermal expansion coefficients, can cause long-term device failures.

In the temperature cycle test, devices are subjected to temperature extremes of –65 °C to +150 °C in nitrogen-filled chambers. One test cycle consists of a 10-minute dwell at each temperature extreme plus a transition time of approximately five minutes.

The gradual change of temperature and relatively long dwell times in an air ambient tend to uncover problems related to expansion rate differentials. Devices are electrically tested after 100 cycles.

Thermal Shock

Just as with the temperature cycle test, the thermal shock test is designed to reveal differences in expansion coefficients for components of the packaging system. However, thermal shock creates a more severe stress in that the device is exposed to a sudden change in temperature due to the high thermal conductivity and capacity of the liquid ambient.

Devices are placed in a fluorocarbon bath cooled to –55 °C. After remaining in the cold chamber for at least five minutes, the sample is transferred to an adjacent chamber filled with fluorocarbon at 125 °C and is held for an equivalent time. After 15 cycles, thermal shock end-point testing is performed.

Moisture Resistance

This test evaluates, on an accelerated schedule, how well component parts and constituent materials resist deterioration from high temperature and humidity typical of tropical environments. The test differs from the steady-state humidity test and derives its added effectiveness by employing temperature cycling, which alternates periods of condensation and drying. This is essential to developing the corrosion process and, in addition, produces a breathing action of moisture into partially sealed containers. The test is carried out per MIL-STD-883 (method 1004).

Salt Atmosphere (Corrosion)

This test is an accelerated laboratory corrosion test. It simulates the effects of seacoast atmosphere on devices and package elements. The test is carried out per MIL-STD-883 (method 1009).

Low-Temperature Aging

This test studies device aging due to hot carrier effects. Since it is well known that hot carrier effects are more pronounced at lower temperatures, the devices are aged at –10 °C for 1,000 hours.

Mechanical Tests

Flammability and O_2 Index

This test follows UL94 and ASTM 2863-77 methods.

Solvent Resistance

This test is performed per MIL-STD-883 (method 2015). Its purpose is to verify that the marking on the component parts will not become illegible when the parts are subjected to solvents. It also seeks to ensure the solvents will not cause deleterious mechanical or electrical damage, or deterioration of the materials or finishes used in the part.

Device Qualification Testing (continued)

Mechanical Tests (continued)

Physical Dimensions

This test is performed to verify that the external physical dimensions of the device are in accordance with the applicable procurement document. The test is carried out per MIL-STD-883 (method 2016).

Solderability Test

This test is conducted to determine the solderability of all terminations normally joined by soldering. The determination is made on the basis of the ability of these terminations to be wetted or coated by solder.

Test procedures verify whether treatment during the manufacturing process to facilitate soldering is satisfactory, and that such treatment has been applied to the required portion of the part which is designed to accommodate a solder connection.

The test includes an accelerated aging test which simulates at least six months' natural aging under a combination of storage conditions, each designed to produce particular deleterious effects. This test is carried out per MIL-STD-883 (method 2003).

Bond Strength

This test measures bond strength and evaluates bond strength distributions, and can therefore be used to determine compliance with specified bond strength requirements of the product's acquisition document. The test is carried out per MIL-STD-883 (method 2011).

Die Shear Strength

This test determines the integrity of materials and procedures used to attach semiconductor die or surface-mounted passive elements to package headers or other substrates. It is carried out per MIL-STD-883 (method 2019).

X-Ray

This examination is performed to nondestructively detect defects within the sealed case, especially those resulting from the sealing process. It is also performed to discover internal defects such as foreign objects, improper interconnecting wires, and voids in the die attach material or in the glass when glass seals are used. The test is carried out per MIL-STD-883 (method 2012).

Electrical Tests

ESD Human-Body Model as per AT&T Method X-19435
(described later in this section)

This qualification requires that each pin in the package be protected up to 1000 V from electrostatic discharge caused by human contact.

ESD Charged-Body Model as per AT&T Method X-19435
(described later in this section)

This qualification requires that each pin in the package be protected up to 1000 V from electrostatic discharge from any charged surface.

Latch-Up as per AT&T Method L757185
(described later in this section)

Latch-up testing includes three components:

- dc stressing of all inputs and I/O pins
- power supply slew rate (dv/dt)
- power supply overvoltage

Three unstressed, fully functional devices are used for each of the three tests. A device is considered latched up if, due to the application of stress, the Icc current exceeds the manufacturer's maximum Icc current and remains at that level after the stress is removed.

Device Qualification Testing (continued)

Electrical Tests (continued)

Table 2-12. Latch-Up Qualification Summary

Test	Conditions	Limits	Susceptibility Level*
1	+dc Stimuli	0 to 2Vcc or +500 mA 0 to Vss – 5 V dc or – 500 mA	4
2	dv/dt	0 to 0.63 Vcc in 100 ns	4
3	Power Supply	>2Vcc	4

*Refer to the latch-up test details on the following pages.

Details of ESD Tests

Electrostatic Discharge (ESD) Test Methods and Requirements

Source: AT&T Bell Laboratories Specification X-19435, Issue 3 (July 1991)

Purpose

This specification describes a uniform method for establishing electrostatic discharge (ESD) withstand thresholds. It also includes threshold requirements and reporting procedures.

Issue 2 includes the following AT&T drawings:

Drawing Number	Figure 3-	Issue
L-224227	1	2
L-224228	2	2
L-224229	3	2
L-224230	4	2

Scope

All packaged semiconductor devices, thin film circuits, surface acoustic wave (SAW) devices, optoelectronic devices, and hybrid integrated circuits (HICs) containing any of these devices are to be evaluated according to this specification. The device thresholds are to be reported in the product design information (PDI) document. Device thresholds and corner pin thresholds are to be reported in the appropriate qualification documents.

Product Design Information (PDI)

The PDI is the official AT&T document in which the responsible design and manufacturing organizations agree that the product information contained therein satisfies manufacturing, legal, and regulatory requirements; it also places certain manufacturing documents under formal change order control.

ESD testing is conducted before transmitting new PDIs and when existing PDIs are reissued due to modification in process, design, packaging, or specification. Tests are conducted on the device and package that represent the product in all details presented in the PDI.

Types of Testing

Two types of testing are required:

- Human-body Model (HBM)
- Charged-device Model (CDM)

Pin Combinations to be Tested

For the human-body model, test three pin combinations:

- Stress each input (or output) pin while grounding all power supply pins
- Stress each power supply pin while grounding each differently named supply pin (or group of pins)
- Stress each input (or output) pin while grounding all output (or input) pins.

Details of ESD Tests (continued)

Electrostatic Discharge (ESD) Test Methods and Requirements (continued)

Pin Combinations to Be Tested (continued)

The power supply pins include V_{DD}, V_{CC}, V_{SS}, V_{BB}, GND, +Vs, –Vs, and V_{REF}. Pins such as offset adjust, compensation, clocks, controls, address, data, and input are considered input pins. Output and input/output pins are considered output pins. In addition, do not test no connects (NCs). For the charged-device model, test each pin. For CDM testing of pin grid array (PGA) devices, however, it is permissible to limit testing to the outer pins.

Voltage levels for HBM ESD testing include:

100 V	1,000 V
200 V	2,000 V
500 V	4,000 V

Voltage levels for CDM ESD testing include:

100 V	1,000 V
200 V	2,000 V
500 V	

Any devices which fail at 500 V are further tested at 100 mV increments to determine threshold value.

Test Procedure Overview

At least six devices are needed to obtain the HBM and CDM thresholds. Prepare at least three of these for HBM testing, and at least three more for the CDM tests. Carry out all testing at room temperature. Circuit schematics for HBM and CDM testing are shown in Figure 2-4 and Figure 2-5 respectively.

Specific Test Procedure: HBM

Insert the first device under test (DUT) into the socket as shown in Figure 2-4. Start with the recommended voltage or with any desired level, as discussed above. At each voltage level, stress all pin combinations as described earlier in this section.

At each voltage level, apply five pulses of each polarity with a one-second interval between pulses to the DUT. Test the device, using the failure criteria specified later in this section. Record the PASS/FAIL results for both the device and the corner pins. Use the device result for the next step, described below.

If the result is PASS, stress the same device or a new device at the next higher level as shown in the tables earlier in this section (when you reach the highest level, stop testing). If the result is FAIL, decrease the voltage to the next lower level, select a new device, and stress only those pins that failed at the previous level. If there is no lower level, stop testing. Repeat these steps until the highest passing voltage for the device and the corner pins is determined.

Then, stress and test two new devices at the highest passing voltage level for all the pin combinations required for the DUT, as listed earlier in the section. If **both** devices pass, this voltage level is the HBM withstand threshold. Otherwise, lower the voltage level and repeat the above testing steps until the HBM threshold is obtained. If the weakest pin is not at a corner, start testing with the voltage one level higher than the highest voltage passed by all the corner pins. Then, determine their threshold by using the same procedure applied to the corner pins.

If one device fails and the other passes, it is permissible to further test the voltage level for three (or multiples of three) new devices. **If no more than** one out of six (or two out of 12, etc.) devices fails, this voltage level is the HBM threshold. This section also applies to the testing of the corner pins.

During device testing procedures, it is allowable to separate polarity; in other words, to stress a device with one polarity and a new device with the opposite polarity. First, determine the threshold for each polarity, as described above. Then, report the threshold value with the lower magnitude.

These test procedures can be modified to fit special circumstances. For example, the withstand threshold is the highest level at which three out of three (or five out of six, or 10 out of 12) stressed devices pass. If the device does not pass any level, its threshold is 0 V.

Details of ESD Tests (continued)

Electrostatic Discharge (ESD) Test Methods and Requirements (continued)

Specific Test Procedure: CDM

Prepare at least three samples. Place the first device on the *Teflon* * base with the leads up. Start with the recommended (or any desired) voltage level, as discussed earlier in the section. All pins will be tested.

First, charge the device with a positive potential by touching all leads with the charging probe (manual testers) or the charging pad (semiautomatic testers). Then discharge the package within 5 s by activating the vacuum relay. Repeat this procedure five times consecutively with a >0.1 s interval between discharges. Repeat the procedure for the negative polarity.

Follow the procedure detailed earlier in this section to determine the CDM threshold. This completes the CDM testing procedure.

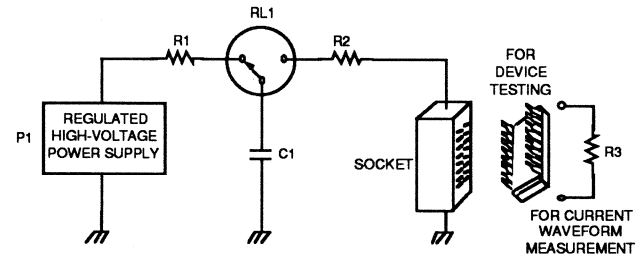


Figure 2-4. Circuit Schematic of Human-Body ESD Simulator

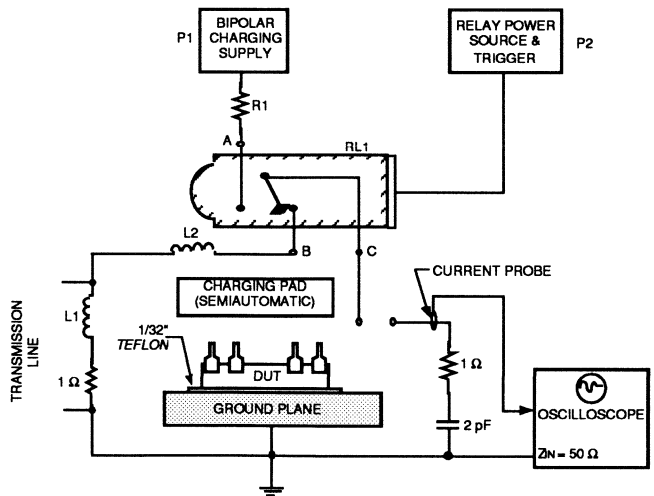


Figure 2-5. Circuit Schematic for CDM Simulator and Waveform Measurement

* *Teflon* is a registered trademark of the Dupont Co.

Details of ESD Tests (continued)

Electrostatic Discharge (ESD) Test Methods and Requirements (continued)

Summary of Pass Criteria

All pins should pass 1,000 V.

Failure Criteria

Parameters identified in the device specification are monitored for ESD testing. If a device cannot pass its own specifications, it is considered to have failed.

Other Information

The devices used for each of the above tests will not be used for any prior or future qualification tests. In addition, the test devices are handled with extreme care, using ESD preventative measures so as not to influence the test results. All operators wear grounding straps when handling the devices. The devices are transported in appropriate ESD protective packaging.

Details of Latch-Up Tests

Integrated Circuit Latch-Up Test Procedure per AT&T Method A88AL1006

Purpose

This section describes the testing method AT& T uses to determine the latch-up susceptibility of CMOS integrated circuits.

Data in this section applies to devices requiring power supply voltages not exceeding ±15 Vdc for normal operation.

Test Procedure

The latch-up testing procedure includes three tests:

- dc stressing of all inputs and I/O pins
- power supply slew rate ($\Delta V/\Delta t$)
- power supply overvoltage

Three unstressed, fully functional devices are used for each of the three tests. Each DUT is heated to the maximum recommended operating case temperature during each test. During testing, if the DUT incurs obvious permanent damage such as if the input opens due to excess input current or Icc > Icc (max), a new device is used to test the remaining pins. The substituted device is not considered as the second or third device of the three-device-per-test requirement.

Unless specifically required, power supply voltages and stresses are applied at a sufficiently slow rate so that the DUT is not adversely affected. If an adverse effect is noted, the rate of application is reduced until the effect is eliminated.

Testing Pre- and Post-Vcc/Vss dc Stressing of Input and I/O Pins

Overview

Input pins not under test are connected to ground. I/O pins are left floating. This pin configuration applies, as long as it does not cause the DUT to malfunction, as in an Icc > Icc max condition.

If the DUT does malfunction due to the pin configuration, an alternate pin configuration may be determined by the responsible device qualification engineer. The engineer will record the alternate configuration in the comments section of the results report form.

The ±dc stimuli limits, detailed later in this procedure, state a voltage and current limit. Voltage is applied to the pin under test via the curve tracer, until either the indicated voltage or current limit is attained or latch-up occurs as defined above.

The duration of the dc stimulus applied to the DUT is less than two seconds, and power supply levels are set to the manufacturer's maximum recommended operating level. Figure 2-6 illustrates equipment hookup.

Details of Latch-Up Tests (continued)

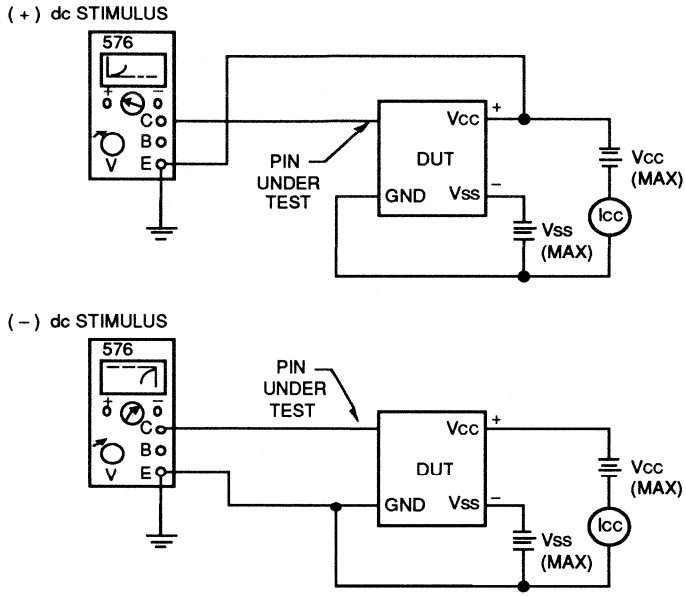


Figure 2-6. Test 1: Equipment Hookup: + or – dc Stimuli

Pre-Vcc/Vss Stress

The DUT is placed in the test socket, and the pin under test is connected to the dc source. The dc stimulus is applied according to the level and polarity being tested, and then Vcc and Vss are applied.

The dc stimulus is then removed from the input pin under test, and the Icc is observed to determine whether latch-up has occurred. If latch-up occurs, the device is immediately removed from the test socket; if it does not occur, the DUT is left in the test socket.

In either case, Vcc and Vss are removed, and the pin under test is connected back to its pretest level. The DUT is resocketed, if necessary, and the next pin is tested. The procedure is repeated until all inputs and I/O pins have been tested. Results are recorded on the results report form.

Post-Vcc/Vss Stress

The DUT is placed in the test socket, and the Vcc and Vss pins are connected to their respective potentials first. Then, the input pin under test is connected to the dc source. The dc stimulus is applied according to the level and polarity being tested.

The dc stimulus is then removed from the input pin under test and the Icc is observed to determine whether latch-up has occurred. If latch-up occurs, the device is immediately removed from the test socket; if it does not occur, the DUT is left in the test socket. In either case, Vcc and Vss are removed, and the pin under test is connected back to its pretest level. The DUT is resocketed if necessary, and the next pin is tested. The procedure is repeated until all inputs and I/O pins have been tested. Results are recorded on the results report form.

Details of Latch-Up Tests (continued)

I/O pins are stressed at all possible output states. The DUT is placed in the test socket, and Vcc and Vss pins are connected to their respective potentials. The I/O pin under test is then connected to the dc source, and the dc stimulus is applied according to the level and polarity being tested.

The dc stimulus is then removed from the I/O pin under test, and the Icc is observed to determine whether latch-up has occurred. If latch-up occurs, the device is immediately removed from the test socket; if it does not occur, the DUT is left in the test socket. In either case, the I/O pin under test is connected back to its pretest level. The DUT is resocketed, if necessary, and the next I/O pin is tested. The procedure is repeated until all I/O pins have been tested.

Power Supply Slew Rate (ΔV/Δt) Limit

If the DUT has multiple power supplies, each power supply pin is stressed separately. The power supply pin(s) not under test are set to the manufacturer's recommended nominal level (Vnom) before the power supply pin under test is stressed. The rate that voltage is applied to the power supply pin under test is described by V(t):

$$V(t) = Vmax\ [1 - exp\ (-t/RC)]$$

where Vmax = manufacturer's recommended maximum supply voltage,
R = 10 Ω, 5%, and C = 0.1 μF, 5%

See Figure 2-7 for equipment hookup. The DUT is placed in the test socket, and the voltage is raised on the power supply pin not under test to Vnom. I/O pins are left floating for each test. Voltage is applied to the power supply pin under test, at the specified rate (see ΔV/Δt limit), as follows:

1. Twelve times with all input pins tied to Vcc
2. Twelve times with all input pins grounded
3. Twelve times with the device in the Icc (active) mode
4. Twelve times with the device in the Icc (standby) mode.

If the Icc active mode and the Icc stand-by mode are reached by conditions 1 and 2, conditions 3 and 4 are not performed.

Each of the 12 times that power is applied to the DUT, Icc is observed to determine whether latch-up has occurred. If latch-up has occurred, the DUT is removed from the test socket immediately and the results and pin configuration are recorded on the results report form. The procedure is repeated for each power supply pin.

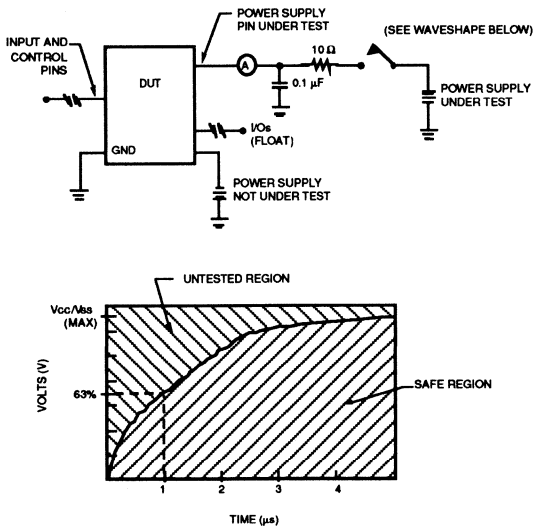


Figure 2-7. Test 2: Power Supply (ΔV/Δt) Limit

Details of Latch-Up Tests (continued)

Power Supply Overvoltage Test

If the DUT has multiple power supplies, each power supply pin is stressed separately. The power supply pin(s) not under test are set to the manufacturer's recommended nominal level (Vnom) before the power supply pin under test is stressed.

The input and I/O pin configuration during this test is the same as for the tests detailed earlier in this section. See Figure 2-8 for equipment hookup. The DUT is placed in the test socket, and the voltage is raised on all power supply pins to Vnom. The voltage on the power supply pin under test is then raised to the power supply overvoltage limit. See the list of power supply overvoltage limits later in this section.

The voltage on the power supply pin under test is returned to Vnom, and the Icc is observed to determine whether latch-up has occurred. If it occurs, the DUT is removed from the test socket immediately. This procedure is repeated for each power supply pin. The results are recorded on the results report form.

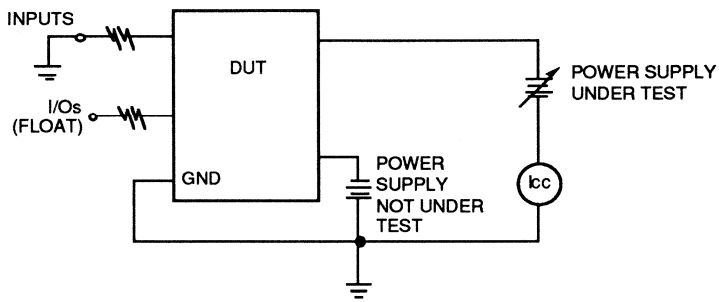


Figure 2-8. Test 3: Power Supply Overvoltage Test Equipment Hookup

Details of Latch-Up Tests (continued)

Susceptibility Levels

A device is considered latched up if, due to the application of a stress, the Icc current exceeds the manufacturer's maximum Icc current and remains at that level after the stress is removed. Susceptibility levels are summarized in the table below. Each device is stressed at level 1 conditions before a higher level is attempted. Devices that latch up due to level 1 conditions or less are considered very susceptible to latch-up and are so noted on the results report form. The device qualification engineer is responsible for determining whether a product should be rejected due to performance on this and all tests. In any case, a device is categorized into the lowest level in which the DUT incurs latch-up during any one of the three tests.

Unless otherwise requested, the maximum stress applied is the maximum conditions described by level 2. Levels 3 and 4 have been designated for the benefit of those whose applications may require a more robust device.

Table 2-13. Limits Employed in Testing

V_{CC} = Manufacturer's maximum operating V_{CC} voltage; V_{SS} = Manufacturer's maximum operating V_{SS} voltage.

Susceptibility Level	Test 1		Test 2	Test 3
	Current	Voltage	Slew Rate	PS Overvoltage
0	<50 mA	V_{CC} + 5 V V_{SS} + 5 V	>0.63 V_{DD} in 10 µs	<1.50 V_{DD}
1	>50 mA <150 mA	V_{CC} + 5 V V_{SS} + 5 V	>0.63 V_{DD} in 5 µs	>1.50 V_{DD} <1.75 V_{DD}
2	>150 mA <250 mA	V_{CC} + 5 V V_{SS} + 5 V	>0.63 V_{DD} in 1 µs	>1.75 V_{DD} <2.00 V_{DD}
3	>250 mA <500 mA	V_{CC} + 5 V V_{SS} + 5 V	>0.63 V_{DD} in 500 µs	>2.00 V_{DD} <2.25 V_{DD}
4	>500 mA	V_{CC} + 5 V V_{SS} + 5 V	>0.63 V_{DD} in 100 µs	>2.25 V_{DD}

Qualification Test Procedures

Introduction

The purpose of this section is to document the qualification test procedures and results of the ATT3000 Series FPGA product family. The qualification plan for this FPGA product family involves rigorous environmental, mechanical, and electrical testing to confirm product soundness. Industry-standard tests are applied to test devices, as well as additional specialized tests to examine electrostatic discharge and latch-up parameters.

Failure mode analysis (FMA) is a comprehensive procedure which determines the cause of any failure encountered during the qualification testings. FMA is used as an integral part of quality testing and surveillance to assess any performance flaw, determine corrective measures, and implement these measures to eliminate recurrence.

FPGA Product Family

There are five arrays in this FPGA product family: ATT3020 (die size 4660 µm x 5570 µm), ATT3030 (5500 µm x 6543 µm), ATT3042 (6160 µm x 7500 µm), ATT3064 (7056 µm x 9441 µm), and ATT3090 (7600 µm x 11000 µm).

These arrays are then assembled in a number of different plastic or ceramic packages as shown below. These packages are plastic leaded chip carrier (PLCC), plastic pin grid array (PPGA), plastic quad flat pack (QFP-EIAJ), ceramic pin grid array (CPGA), and ceramic quad flat pack (CQFP).

ATT3090-H175
ATT3064-H132
ATT3042-H132
ATT3020-N100
ATT3042-N100
ATT3090-N164
ATT3030-M44
ATT3030-M68
ATT3020-M68
ATT3020-M84
ATT3042-M84
ATT3030-M84
ATT3064-M84
ATT3090-M84
ATT3090-R175
ATT3020-R84
ATT3030-R84
ATT3042-R84
ATT3042-R132
ATT3064-R132
ATT3090-J160
ATT3042-J100
ATT3020-J100
ATT3030-J100
ATT3064-J160

Qualification Strategy

The qualification of ATT3000 Series FPGAs can be broken into three specific groups: process technology, silicon design, and package.

Process Technology: These arrays are processed at Allentown using the 0.9 µm CMOS technology. AT&T's 0.9 µm CMOS process employs N- and P-channel LDD MOS transistors with a typical electrical gate length of 0.75 µm. The process also uses two levels of metal and one level of polycide. Source and drain regions are silicided for low resistance. This technology has been rigorously tested for reliability and manufacturability and is fully qualified.

Silicon Design: Since the five arrays in the ATT3000 Series FPGAs are a matrix of repetitive elements, the qualification review board (QRB) decided to qualify one die which would qualify the silicon design methodology of all five arrays. Therefore, HTOB test was performed on the ATT3020 die in the 84PLCC package. The result is given in Table 2-14.

Package: The QRB determined the qualification requirements of each die in a given package. The results are given in Table 2-14 through Table 2-18.

Qualification Test and FMA Results

Table 2-14. PLCC Packages

Qualification Information	Device							
	3030-M44	3020-M68	3030-M68	3020-M84	3030-M84	3042-M84	3064-M84	3090-M84
ESD-HBM[1]	>2000 V	3	>2000 V	>1000 V	>2000 V	>1000 V	>3000 V	>3500 V
ESD-CDM[1]	>3000 V	3	>3000 V	>3000 V	>3000 V	>1000 V	>2500 V	>3000 V
1000 hrs. THB	3	3	3	3	3	3	4	5
1000 hrs. HTOB	1/166, Pass[2]	3	3	3	3	3	4	3020M84 for HTOB
CLASS	3	3	3	3	3	3	4	0/134, Pass
1000 Hrs. THB	3	3	3	3	3	3	4	0/134, Pass
Autoclave (96 hrs. SB)	3	3	3	3	3	3	4	1/105, Pass[2]
100 c/s TC	3	3	3	3	3	3	4	1/105, Pass[2]
15 c/s TS	3	3	3	3	3	3	4	0/25, Pass
Moisture Resistance	3	3	3	3	3	3	4	5
Corrosion	3	3	3	3	3	3	4	5
Solvent Resistance	3	3	3	3	3	3	4	5
Physical Dimensions	3	3	3	3	3	3	4	5
Solderability	3	3	3	3	3	3	4	5
Bond Strength	3	3	3	3	3	3	4	5
Die Shear Strength	3	3	3	3	3	3	4	5
X-ray	3	3	3	3	3	3	4	5
LU Class	III	3	IV	III	III	III	IV	IV
Reference for Other Tests	1042L.BR	1042L.BR	1042L.BR	1042L.BR	1042L.BR	1042L.BR	3090M84	409AT[5]
Status	Fully qualified	Fully qualified	Fully qualified	Fully qualified	Fully qualified	Fully qualified	Fully qualified	Fully qualified

1. HBM = human-body model and CDM = charged-device model; see page 2-15 for a description.
2. FMA reported two failures at 320 hours. One failure was due to EOS damage induced either during testing or during HTOB operation and was discounted. The other failure was due to metal 2 to metal 1 leakage as a result of metal filaments caused by PETEOS trenching.
3. See 1042L.BR.
4. See 3090-M84.
5. See 409AT, Q89027.

Qualification Test and FMA Results (continued)

Table 2-15. Plastic QFP-EIAJ Packages

Qualification Information	Device				
	3020-J100	3030-J100	3042-J100	3064-J160	3090-J160
ESD-HBM[1]	>2000 V	>3000 V	>3000 V	3	4
ESD-CDM[1]	>2500 V	>3000 V	>3000 V	3	4
1000 hrs. THB	2	2	2	3	4
1000 hrs. HTOB	2	2	2	3	4
CLASS	2	2	2	3	4
1000 hrs. THB	2	2	2	3	4
Autoclave (96 hrs. SB)	2	2	2	3	4
100 c/s TC	2	2	2	3	4
15 c/s TS	2	2	2	3	4
Moisture Resistance	2	2	2	3	4
Corrosion	2	2	2	3	4
Solvent Resistance	2	2	2	3	4
Physical Dimensions	2	2	2	3	4
Solderability	2	2	2	3	4
Bond Strength	2	2	2	3	4
Die Shear Strength	2	2	2	3	4
X-ray	2	2	2	3	4
LU Class	III	IV	III	3	4
Reference for Other Tests	409ATX or S1116X[2]	409ATX or S1116X[2]	409ATX or S1116X[2]	3090-160QFP-EIAJ[3]	*WE*-DSP32C-F32[4]
Status	Fully qualified	Fully qualified	Fully qualified		—

1. HBM = human-body model and CDM = charged-device model; see page 2-15 for a description.
2. See 409ATX, Q90001 and S1116X, Q90111.
3. See 3090-J160.
4. See *WE*-DSP32CF32, Q89129.1.

Qualification Test and FMA Results (continued)

Table 2-16. CPGA Packages

Qualification Information	Device					
	3020-R84	**3030-R84**	**3042-R84**	**3042-R132**	**3064-R132**	**3090-R175**
ESD-HBM[1]	>1000 V	>2500 V	>2500 V	>1000 V	>2500 V	>3500 V
ESD-CDM[1]	>3000 V	>3000 V	>3000 V	>2500 V	>3000 V	>3000 V
1000 hrs. THB	—	—	—	—	—	—
CLASS	—	—	—	—	—	—
1000 hrs. THB	—	—	—	—	—	—
Autoclave (96 hrs. SB)	—	—	—	—	—	—
100 c/s TC	—	—	—	—	—	—
15 c/s TS	—	—	—	—	—	—
Moisture Resistance	—	—	—	—	—	—
Corrosion	—	—	—	—	—	—
Solvent Resistance	—	—	—	—	—	—
Physical Dimensions	—	Pass	Pass	Pass	—	Pass
Solderability	—	—	—	—	—	—
Bond Strength	—	—	—	—	—	—
Die Shear Strength	—	—	—	—	—	—
X-ray	—	—	—	—	—	—
—	IV	IV	IV	IV	IV	III
Reference for Other Tests	*WE*-DSP32C-R32 1083A 3030R84	*WE*-DSP32C-R32 1083A	*WE*-DSP32C-R32 1083A	*WE*-DSP32C-R32 1083A	*WE*-DSP32C-R32 1083A	*WE*-DSP32C-R32 1083A[2]
Status	Fully qualified	Fully qualified	Fully qualified	Fully qualified	Fully qualified	Fully qualified

1. HBM = human-body model and CDM = charged-device model; see page 2-15 for a description.
2. See *WE*-DSP32CR32, Q88105.1 and 1083A, Q89175.

Qualification Test and FMA Results (continued)

Table 2-17. PPGA Packages

Qualification Information	Device		
	3042-H132	**3064-H132**	**3090-H175**
ESD-HBM[1]	>2500 V	>2500 V	>3000 V
ESD-CDM[1]	>2500 V	>3000 V	>2500 V
1000 hrs. THB	—	—	—
1000 hrs. HTOB	—	—	—
CLASS	—	—	—
1000 hrs. THB	—	—	—
Autoclave (96 hrs. SB)	—	—	—
100 c/s TC	—	—	—
15 c/s TS	—	—	—
Moisture Resistance	—	—	—
Corrosion	—	—	—
Solvent Resistance	—	—	—
Physical Dimensions	—	—	—
Solderability	—	—	—
Bond Strength	—	—	—
Die Shear Strength	—	—	—
X-ray	—	—	—
LU Class	IV	IV	IV
Reference for Other Tests	3090H175	3090H175	None
Status	In planning/In progress	In planning/In progress	In planning/In progress

1. HBM = human-body model and CDM = charged-device model, see page 2-15 for a description.

Qualification Test and FMA Results (continued)

Table 2-18. CQFP Packages

Qualification Information	Device		
	3020-N100	3042-N100	3090-N164
ESD-HBM[1]	—	—	>3000 V
ESD-CDM[1]	—	—	>2500 V
1000 hrs. THB	—	—	—
1000 hrs. HTOB	—	—	0/100, Pass
CLASS	—	—	—
1000 hrs. THB	—	—	—
Autoclave (96 hrs. SB)	—	—	—
100 c/s TC	—	—	—
15 c/s TS	—	—	—
Moisture Resistance	—	—	—
Corrosion	—	—	—
Solvent Resistance	—	—	—
Physical Dimensions	—	—	—
Solderability	—	—	—
Bond Strength	—	—	—
Die Shear Strength	—	—	—
X-ray	—	—	—
LU Class	—	—	IV
Reference for Other Tests	3090N164 3020R84 for ESD, LU	3090N164 3020N100 3042R84 for ESD, LU	1056D[2]
Status	In progress Initially qualified	In progress Initially qualified	In progress Initially qualified

1. HBM = human-body model and CDM = charged-device model; see page 2-15 for a description.
2. See 1056D, Q90132.3.

Qualification Test and FMA Results (continued)

Reference Tests Results

The following tests are used as references for ATT3000 Series qualification.

Table 2-19. Reference Test Results I

Qualification Information	Device			
	1042L.BR, Q89166	**409AT, Q89027**	***WE*-DSP32CF32, Q89129.1**	**409ATX, Q90001**
Technology	0.9 µm CMOS	0.9 µm CMOS	0.9 µm CMOS	0.9 µm CMOS
Chip Size (µm)	7410 x 7710	7040 x 6830	8210 x 10960	6830 x 7040
Package	68-pin PLCC	100-pin PLCC	164-pin PQFP-JEDEC	100-pin PQFP-EIAJ (assembled by AMKOR)
Steam Bomb	0/105 Pass	—	1/105 Pass	1/105 Pass
1000 hrs. HTOB	0/135 Pass	1/142 Pass	0/105 Pass	2/168 Pass
CLASS	2/135 Pass	1/134 Pass	1/137 Pass	0/132 Pass
1000 hrs. THB	0/132 Pass	1/133 Pass	0/135 Pass	0/132 Pass
100 c/s TC or 300 c/s TC	1/105 Pass	1/105 Pass	0/105 Pass	0/105 Pass
15 c/s TS or 100 c/s TS	0/24 Pass	0/25 Pass	0/25 Pass	0/25 Pass
Moisture Resistance	—	—	—	0/38 Pass
Corrosion	—	—	—	0/15 Pass
Solvent Resistance	—	—	—	0/8 Pass
Physical Dimension	0/15 Pass	0/15 Pass	0/15 Pass	0/15 Pass
Solderability	—	—	—	0/22 (3 devices) Pass
LI	0/15 Pass	0/15 Pass	0/15 Pass	0/15 Pass
Bond Strength	—	0/15 Pass	—	—
Die Shear Strength	—	0/5 Pass	—	—
X-ray	0/5 Pass	0/5 Pass	0/5 Pass	0/5 Pass
ESD	—	—	—	—
Latch-up	—	—	—	—

Qualification Test and FMA Results (continued)

Reference Tests Results (continued)

Table 2-20. Reference Test Results II

Qualification Information	Device			
	S1116X, Q90111	***WE*-DSP32CR32, Q88105.1**	**1083A, Q89175**	**1056D, Q89175**
Technology	0.9 µm CMOS	0.9 µm CMOS	0.9 µm CMOS	0.9 µm CMOS
Chip Size (µm)	6830 x 7040	10960 x 8210	12960 x 7460	12700 x 12720
Package	100-pin PQFP-EIAJ (assembled by SHINKO)	133-pin CPGA (assembled at Allentown)	180-pin CPGA (assembled at Allentown)	256-pin CQFP (assembled at Allentown)
Steam Bomb	0/105 Pass	—	—	—
1000 hrs. HTOB	1/105 Pass	3/185 Pass	0/105 Pass	0/79 Pass
Class	2/132 Pass	—	—	—
1000 hrs. THB	0/129	—	—	—
100 c/s TC or 300 c/s TC	1/105 Pass	0/105 Pass	0/105 Pass	0/77 Pass
15 c/s TS or 100 c/s TS	0/25 Pass	0/25 Pass	0/25 Pass	—
Moisture Resistance	0/38 Pass	—	—	—
Corrosion	0/15 Pass	—	—	0/15 Pass
Solvent Resistance	0/8 Pass	—	—	0/8 Pass
Physical Dimension	0/15 Pass	0/15 Pass	0/15 Pass	—
Solderability	0/22 (3 devices) Pass	—	—	0/22 Pass
LI	0/15 Pass	—	0/15 Pass	0/45 Pass
Bond Strength	—	—	—	0/22 Pass
Die Shear Strength	—	—	—	0/5 Pass
SQ	—	2/76 Pass	0/38 Pass	0/15 Pass
X-ray	0/5 Pass	—	0/5 Pass	0/5 Pass
ESD	—	—	—	—
Latch-up	—	—	—	—
Internal Water Vapor	—	0/5 Pass	—	0/3 Pass
Internal Visual	—	0/5 Pass	—	0/5 Pass
Thermal Sequence (TS, TC, MR, ELECT with Leak Test)	—	—	—	1/34 Pass

Inspection/Quality Assurance

This section summarizes the completed product audit for MOS products, comprised of procedures used by quality assurance (QA) final inspection. When manufacturing is completed and products are presented for shipment, QA selects samples and confirms that they meet specifications.

Electrical sampling is performed on a code-by-code basis, using a two-level system as shown in Figure 2-9. Normal level inspection is done lot-by-lot at 0.1% acceptable quality level (AQL). Actual quality is much better than 0.1%. For example, in 1990, QA measured fewer than 25 parts per million (ppm) defective during an electrical audit. Skip-lot inspection is performed on one lot of every four received by Quality Assurance using the same 0.1% AQL plan. Mechanical/visual inspection uses the same process, but lots may be grouped by package type, as well as by code.

Sticks (or tubes) of devices are randomly chosen until the required sample size is reached. To avoid mixing of products in final inspection, only samples from one lot are inspected at any one time and are returned to the parent lot before choosing samples from the next lot.

Codes may be moved to skip-lot inspection after ten consecutive lots pass lot by lot. Any rejected lot causes the code or package type to return (or remain) on lot-by-lot sampling.

Specific procedures exist for tighter sampling and inspection procedures when the QA engineering organization determines them to be necessary.

Final inspection electrical tests are made in accordance with the latest issue of the device specification. Mechanical/visual requirements are obtained from package drawings and internal workmanship standards. Nonacceptable production lots are returned to the operating shop for corrective action.

QA reports the results of the electrical and mechanical/visual inspections as a measure of outgoing quality expressed in parts per million (ppm). These reports are generated from a computerized data base, which gathers information entered by inspectors on each lot as inspection is performed.

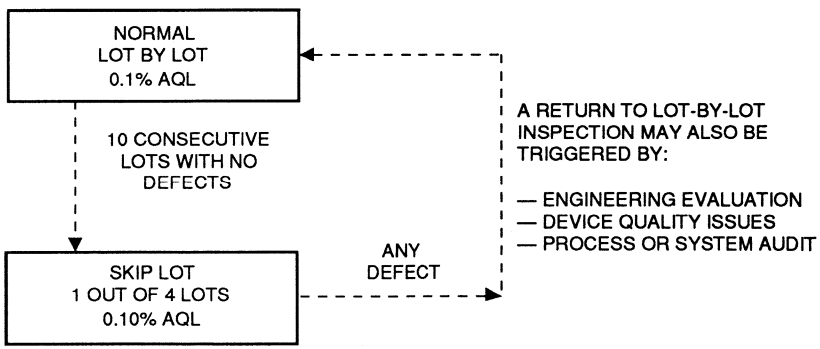


Figure 2-9. Electrical Test Sampling: Code-by-Code Basis

Product Reliability Monitoring Plan

Two-Tier Testing Program

A keystone of AT&T's reliability plan is its two-tier program of stress testing.

Level 1 represents AT&T's extended life testing program, in which samples of devices are taken for initial qualification—and on a regular monthly schedule thereafter—from each reliability testing group and stressed for periods simulating up to 40 years of product life.

Level 2 is AT&T's normal production monitoring program in which a sample of devices is chosen from each testing group and stressed for a period approximating ten years of product life. Level 2 samples are normally taken biweekly. Level 2 HTOB testing includes a test point at 24 hours which is intended as an infant mortality measurement.

Product Families and Sampling Coverage

To facilitate product sampling and ensure complete coverage of our reliability program, all AT&T products are grouped into families according to their basic design and manufacturing process. The resulting product families are shown here.

Bipolar, including:

- SFOXIL
- SBC
- GIMIC
- OXIL

MOS, including:

- Custom logic
- Memory devices
- Microprocessors
- Codec/analog
- Digital signal processors
- Field-programmable gate arrays

The Reliability Testing Plan

Each level of AT&T's two-tier testing plan has its own sampling approach, described in this section. As a special feature, products from test groups with excellent reliability test result histories (≤A over the last four samples) may be shipped after the sample is selected but before stress testing is completed.

Level 2 testing provides reasonable assurance of reliability at reasonable cost. Most Level 2 tests are completed within one week so any potential problems can be quickly identified. Level 1 tests are needed for ultimate assurance of long life reliability, but generally require several weeks to complete. Safeguards include provisions for holding product shipment if life test results are not excellent, and formally convening a reliability review board (RRB) to address and resolve any questionable test results.

Sample Selection

Level 2 samples are selected every two weeks from the testing universes shown in Table 2-27 and Table 2-28. When several clean rooms are represented in the same testing universe, samples are taken from each combination of clean room and assembly location. Successive samples are taken from different wafer lots to ensure the reliability data represents a continuous flow of product from each clean room.

Performance at Level B, as shown in Table 2-21, Table 2-22, and Table 2-23, triggers RRB action.

Product Reliability Monitoring Plan (continued)

The Reliability Testing Plan (continued)

Table 2-21. Sampling Plan: High Temperature Operating Bias

Test Level	Test Hours	Sampling Frequency	Stress Conditions	Sample Size	LTPD	Acceptance Numbers	
						A	B
2	160	Biweekly	125 °C	116	—	0	1
1	1,000	Monthly	125 °C	76	3.0	0	1
2	160	Biweekly	150 °C	100	4.0%	1	2
1	1,000	Monthly	150 °C	58	~6.7%	1	2

Table 2-22. Sampling Plan: Temperature Humidity Bias (THB)

Test Level	Test Hours	Sampling Frequency	Stress Conditions	Sample Size	LTPD	Acceptance Numbers	
						A	B
2	240	Monthly	85 °C/85% RH	130	3.0%	1	2
1	1,000	Monthly	85 °C/85% RH	76		0	1

Table 2-23. Sampling Plan: Temperature Cycle

Test Level	Test Cycles	Sampling Frequency	Stress Conditions	Sample Size	LTPD	Acceptance Numbers	
						A	B
2	100	Monthly	–65 °C/+150 °C	105	5.0%	2	3
1	300	Monthly	–65 °C/+150 °C	77		1	2

Failure Treatment Procedures

Devices which fail after 24 hours are treated as follows:

- Functional failures (plus bins recognized as contact defects) are sent to the product engineer for FMA.
- Parametric failures are data-logged, marked, and returned to the chamber for further stressing.

Data sheet specifications are the criteria for failures.

Level 1 HTOB stressing is considered destructive, and such devices are not shipped.

Logging Procedures

When failures occur, they are data-logged, and lot information is entered into an electronic data base named RELMS. A failure analysis request form is generated from RELMS, and the devices are given to engineering for failure analysis. This individual reviews the data log and retests the failures.

When failure analysis is complete, the responsible engineer returns the form to the reliability testing group, which enters the data into RELMS.

Product Reliability Monitoring Plan (continued)

Test Condition Tables

Table 2-24 through Table 2-26 represent conditions for the stress testing described in this section. When surface-mount devices are tested, they are preconditioned using CLASS prior to reliability stressing. CLASS is an AT&T procedure which simulates the thermal stresses a device will encounter during circuit board fabrication.

Table 2-24. Test Conditions: High Temperature Operating Bias (HTOB)

Temperature Option	Bias Conditions	Bias Voltage
125 °C	Fully Dynamic	Over Voltage (Approximately 10%)
150 °C	Fully Dynamic	Over Voltage (Approximately 10%)

Table 2-25. Test Conditions: Temperature Humidity Bias

Temperature Range	Bias Arrangement
85 °C/85% RH	dc Bias at Low-power State

Table 2-26. Test Conditions: Temperature Cycling

These conditions correspond to MIL-STD-883C, Method 1010, Condition C.

Temperature Range	Bias Arrangement
–65 °C to +150 °C	None

Reliability Testing Procedures

Specific Procedures for HTOB Testing

Sample devices, chosen according to the protocol in Table 2-27, are loaded into boards and placed into the HTOB oven set for 150 °C (if not otherwise specified). After testing, samples are cooled to 30 °C oven ambient temperature while under bias for a minimum of 30 minutes. They are then removed from the oven and electrically tested. Samples designated for Level 1 testing are then returned to the oven for an additional 840 hours—a total of 1,000 hours.

Specific Procedures for Temperature Humidity Bias (THB) Testing

THB is designed to test the integrity of the package-chip interface, as well as chip metallization. THB samples are selected monthly from testing universes shown in Table 2-28 in this section. Device selection is rotated within the testing group to give a representation of the different pinouts for different package sizes.

Samples are loaded into prescribed load boards, placed in the THB chamber for 240 hours (Level 2), and then electrically tested. Units which pass are reloaded and returned to the stressing chamber to complete 1,000 hours of testing.

Within one hour after this phase, the sample is placed in a moisture chamber, and posttesting is performed. Devices which fail in posttest are returned to the THB testing board for a 48-hour presoak (no bias), followed by a minimum of 48 hours with bias, and are then retested.

Product Reliability Monitoring Plan (continued)

Reliability Testing Procedures (continued)

Specific Procedures for Temperature Cycling (TC)

TC stressing is designed to detect thermal mismatches of materials. Devices are cycled through temperature extremes to accelerate such failure mechanisms.

TC samples are selected monthly from the testing universes depicted in Table 2-27. Device selection is rotated to ensure a distribution of different package types.

Samples are placed in the TC chamber with no bias for 100 cycles at the conditions listed in the accompanying tables. They are then tested electrically, and the data is entered into the data base. Good devices are then returned to TC stressing to complete 300 cycle stressing. Level 1 temperature cycling is considered destructive, and test devices are not shipped as normal products.

Failure recording is conducted as in the other stress testing routines.

Sampling Universes

Table 2-27. Sampling Universes by Technology: HTOB

Super Family	Process Technology	Layout Style
CMOS		
A	3.5/5.0 LC	Analog
	3.5/5.0 C	Analog
	3.5/5.0 C	Custom Digital
	3.5/5.0 C	Standard Cell Digital
B	2.5 C	Custom Digital
	2.5 C	Standard Cell Digital
	2.5 C	Memory
C	1.75 CT	Standard Cell Digital
	1.75 C	Analog/Custom Digital
	1.75 C	Standard Cell (1P 1M)
	1.75 C	Memory
C.1	1.75 LC	Analog/Standard Cell (2P 1M)
D	1.25 C	Custom Digital
	1.25 C	Standard Cell Digital
	1.25 C	Memory
D.1	1.25 CT (2-level metal)	Standard Cell Digital
E	0.9 LC	Analog
E.1	0.9 CT	Standard Cell Digital
E.2	0.9 M	Memory
NMOS		
F	2.9 N	Memory
	3.5 N	Standard Cell Digital
	5.0/7.5 N	Analog
	5.0/7.5 N	Standard Cell Digital
G	1.7/1.9/2.8 N	Custom Digital
	1.7/1.9 N	Memory
H	0.75 N	Custom Digital
	1.0 N	Custom Digital

Product Reliability Monitoring Plan (continued)

Sampling Universes (continued)

Table 2-28. Sampling Universes by Technology: THB and TC

Package Universe	Package Type	Package Variation
A	Ceramic DIP	300 MIL 600 MIL
B	Plastic DIP	300 MIL 600 MIL
C	Ceramic Chip Carrier	<40 Pins >40 Pins
D	Plastic Chip Carrier	<40 Pins >40 and <100 Pins ≥100 Pins
E	Plastic Quad, Fine Pitch	≥100 Pins
F	Pin Grid Array Multiple In Line	All All
G	Small Outline	SOJ SOG
H	Ceramic Leadless CC	All
I	Plastic Leadless CC	All

The Reliability Review Board (RRB)

AT&T's Reliability Review Board is responsible for monitoring reliability testing results. The membership of the board is determined by the issue being addressed. Members must include process engineers and product engineers, and may also include quality assurance engineers, design engineers, marketing personnel, and customer representatives. The board holds regular meetings and any member may convene a meeting for cause.

The RRB reviews reliability results and provides the engineering expertise and management judgment necessary to respond to and resolve issues generated by test results and research input. The board's single goal is to maintain and improve the reliability of AT&T's silicon integrated circuits.

The RRB's powers are far-reaching. They extend to product treatment, sampling procedures, test design, and device failures. The RRB reviews all FMA reports, determines the validity of test conclusions, and assists in determining corrective action. If necessary, it can recommend the recall of products already shipped.

The board is administered by the MOS reliability engineer, assisted by the reliability engineers at each assembly location. Each RRB member is specifically qualified to make recommendations for the processes or products under that individual's care.

The process engineer is responsible for monitoring the fabrication line, interpreting process data, and assisting with corrective action. The product designer is responsible for determining specifics for product testing, such as voltages, temperatures, bias arrangements, and load board configuration. The designer works with the RRB to resolve any questions concerning the appropriateness of testing parameters or procedures.

Process Features

This section describes AT&T's 0.9 µm CMOS processing technology. It focuses on process sequence, mask levels, and specific features of the technology.

An illustration showing a cross-sectional view of the 0.9 µm CMOS process concludes the section.

Process Characteristics

Overview

AT&T's 0.9 µm Twin-Tub V is a world-class, double-level metal CMOS technology with 0.75 µm minimum design rules and 0.6 µm minimum effective N- and P-channel length. It was transferred to SEMATECH (SEmiconductor MAnufacturing TECHnology)—a consortium of American semiconductor manufacturers working with the government and academia to sponsor and conduct research aimed at helping to ensure leadership for the U. S. semiconductor industry—and was used as the baseline process for initial manufacture.

The technology is designed to provide a manufacturable process along with high performance and high packing density, at 5.5 V maximum power supply voltage.

Depending on layout and application, this silicided technology reduces chip size by 15% to 30% over nonsilicided technologies. Silicide routing can reduce capacitive loading, resulting in faster circuits. Due to scaling at all levels, the speed of devices processed in this technology is increased by a factor of 2.0 over the previous 1.25 µm CMOS technology.

Technology Highlights

These packing density and speed improvements result from the following process features:

- Thirteen mask levels
- No channel stop mask
- 5 V operation with 0.75 µm electrical channel lengths
- Hipox field oxidation
- N-channel and P-channel LDD
- $TiSi_2$ on poly for low sheet resistance and low contact resistance
- $TiSi_2$ source/drain for low sheet resistance and low contact resistance
- Doped glass: low-temperature flow (dielectric 1)
- Undoped glass: (dielectric 2)
- Ti under M1
- Two-level metal technology
- Bilayer CAPS: Nitride and undoped glass

Process Outline

The specific series of mask steps used in fabricating the 0.9 µm CMOS technology is listed in Table 2-29.

Table 2-29. Mask Steps In 0.9 µm CMOS Process

Sequence	Mask Step
1	NTUB
2	GASAD
3	THRESHOLD ADJ I
4	THRESHOLD ADJ II
5	GATE
6	P SOURCE/DRAIN
7	N SOURCE/DRAIN
8	P SOURCE/DRAIN
9	WINDOW I
10	ALUMINUM I
11	WINDOW II
12	ALUMINUM II
13	CAPS

Process Cross Section

A cross-sectional illustration of the 0.9 µm CMOS process is shown in Figure 2-10. This is a schematic of the process and is designed to convey general concepts of areas and their placement. The schematic has been purposely enlarged in height (*y* axis) for maximum visual clarity, as shown by the scale in its upper left corner.

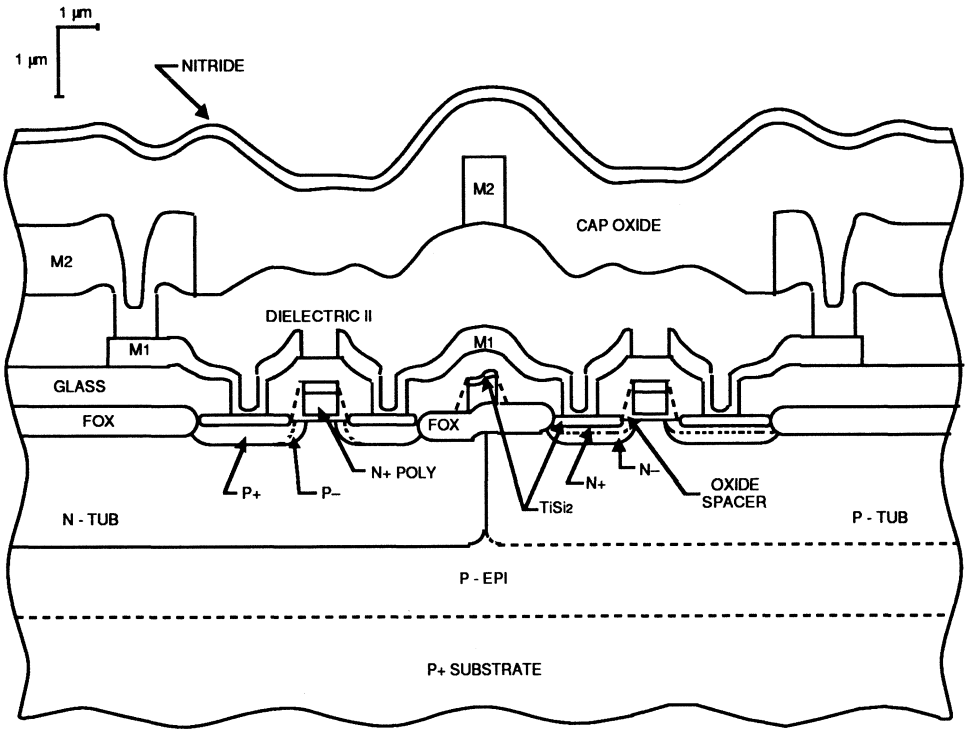


Figure 2-10. 0.9 µm CMOS Process Cross Section

Production Flow

This section, covering production flow for AT&T's ATT3000 Series FPGA, is presented in a series of specific segments of the production process. For each segment, one or more flowcharts explains the steps in the process.

Wafer flow is shown in Figure 2-11, and assembly flow is shown in Figure 2-12 and Figure 2-13. Figure 2-14 and 2-15 illustrate the subsequent test flow and finish stage. Segments on quality monitors, statistical process control, and statistical process control monitors complete this section.

Wafer Flow

Figure 2-11 shows the generic wafer flow for the CMOS FPGA product line.

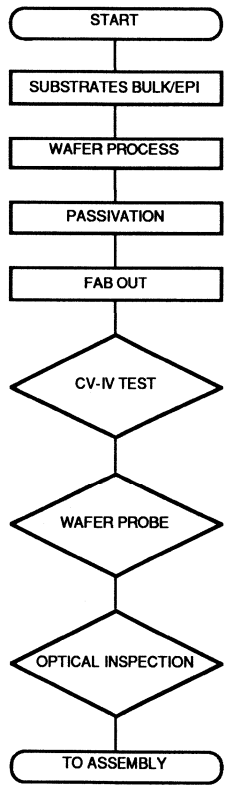


Figure 2-11. Wafer Flow

Assembly Flow

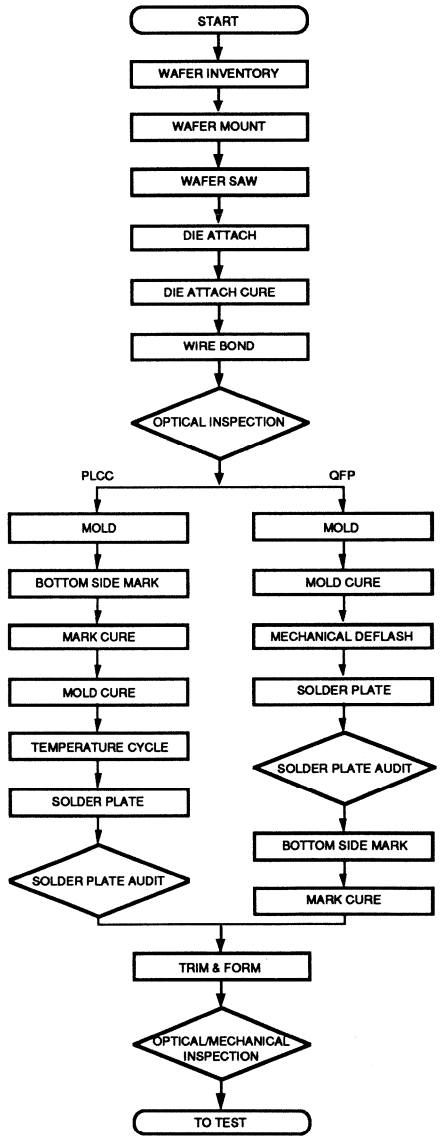


Figure 2-12. Assembly Flow (PLCCs and PQFPs)

Assembly Flow (continued)

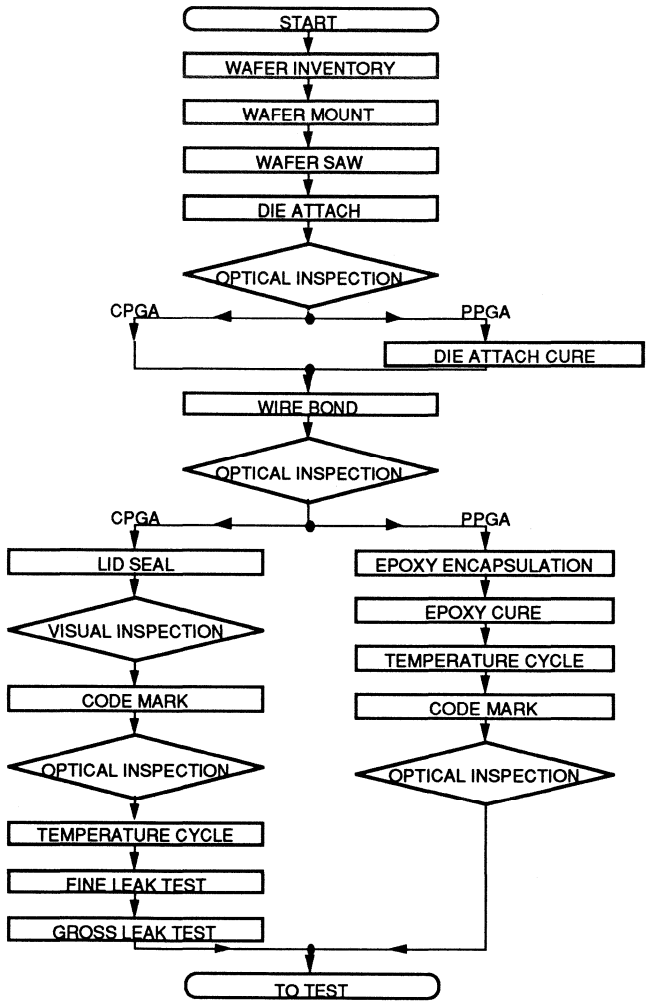
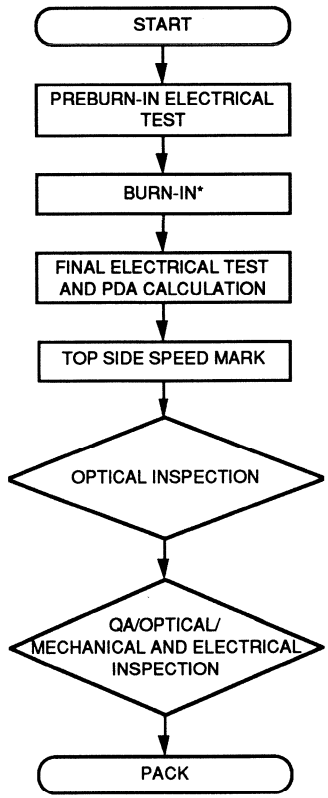


Figure 2-13. Assembly Flow (CPGAs and PPGAs)

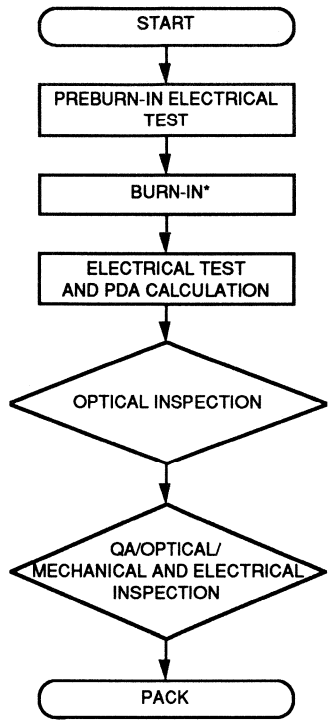
Test and Finish Flow



* As required to meet infant mortality requirements or to meet customer specifications.

Figure 2-14. Commercial Test and Finish Flow (PLCCs and QFPs)

Test and Finish Flow (continued)



* As required to meet infant mortality requirements or to meet customer specifications.

Figure 2-15. Commercial Test and Finish Flow (CPGAs and PPGAs)

Quality Monitors

Quality monitors during the production of AT&T's CMOS FPGAs are summarized in this section.

Table 2-30. Quality Monitors in Critical Fabrication Processes for 0.9 μm CMOS Technology

Operation	Characteristic	Frequency	Sample Size
Pad 1 Oxidation	Oxide Thickness	Each Run	Four (4) Wafers
Nitride 1 Deposition	Nitride Thickness	Each Run	Four (4) Wafers
Tub Oxidation	Oxide Thickness	Each Run	Four (4) Wafers
Tub Drive-In	Oxide Thickness	Each Run	Four (4) Wafers
Pad 2 Oxidation	Oxide Thickness	Each Run	Four (4) Wafers
Nitride 2 Deposition	Nitride Thickness	Each Run	Four (4) Wafers
Hipox—Field	Oxide Thickness	Each Run	Four (4) Wafers
Gate 0 Oxidation	Oxide Thickness	Each Run	Four (4) Wafers
Gate 1 Oxidation	Oxide Thickness	Each Run	Four (4) Wafers
Polysilicon Deposition	Polysilicon Thickness	Each Run	Four (4) Wafers
Phosphorous Diffusion	Sheet Resistance	Each Run	Two (2) Controls
Poly Photoresist	Linesize	Each Lot	Three (3) Wafers
LDD Push Oxidation	Oxide Thickness	Each Run	Four (4) Wafers
TEOS 1 Deposition	TEOS Thickness	Each Run	Five (5) Controls
Spacer Etch	Field Oxide Thickness	Each Run	Three (3) Wafers
Sputter Titanium 1	Titanium Thickness	Each Run	One (1) Control
RTA Silicide	Sheet Resistance	Each Lot	One (1) Control
RTA Ti-Silicide	Sheet Resistance	Each Lot	One (1) Control
TEOS 2 Deposition	TEOS Thickness	Each Run	Five (5) Controls
BPTEOS Deposition	BPTEOS Thickness Weight % Phosphorous Weight % Boron	Each Run Each Run Each Run	Four (4) Controls Four (4) Controls Four (4) Controls
RTA Window Reflow	Sheet Resistance	Each Lot	One (1) Control
Sputter Titanium 2	Titanium Thickness	2x/Shift	One (1) Control
RTA Ti-Aluminum	Sheet Resistance	Each Lot	One (1) Control
Sputter Aluminum 1	Aluminum Thickness	2x/Shift	One (1) Control
Aluminum Photoresist	Linesize	Each Lot	Three (3) Wafers
Sputter Aluminum 2	Aluminum Thickness	2x/Shift	One (1) Control
Metal 2 Photoresist	Linesize	Each Lot	Three (3) Wafers
Sincaps Deposition	Sincaps Thickness Index of Refraction	Each Run Each Run	Four (4) Controls Four (4) Controls
Aluminum Step Coverage	Windows/Vias Minimum Space	Weekly Every Fifth Lot	Four (4) Wafers Variable
Visual Inspection	Visual Defects	Each Lot	12% of Lot

Quality Monitors (continued)

Table 2-31. Quality Controls During FPGA Assembly (PLCCs)

Operation	Inspection	Frequency	Sample Size
Wafer Saw	Visual Inspection	2x/machine/shift	20 units
	Machine Inspection	1x/machine/shift	Each machine
Die Attach	Visual Inspection	Every magazine	3 or 4 strips
	Epoxy Resistivity	1x/month	2 glass slides
	Die Shear Strength	1x/machine/shift	3 units
	Epoxy Void	1x/machine/shift	10 units
	Machine Inspection	1x/machine/shift	Each machine
	Over Cure	1x/machine/shift	Each machine
Wire Bond	Visual Inspection	Every magazine	4 strips
	Bond Pull Strength	2x/machine/shift	5 or 10 wires
	Ball Sheer Strength	2x/machine/shift	5 or 10 wires
	Machine Inspection	1x/machine/shift	Each machine
Mold	Visual Inspection	Very 20 mold run	1 mold run
	X-ray	2x/machine/shift	12 units
	Machine Inspection	1x/machine/shift	Each machine
Code Mark	Visual Inspection	Every lot	40 units
	Mark Permanency	1 in 5 lots	4 units
	Machine Inspection	1x/day	Each machine
Trim/Form	Visual Inspection	Every 40 tubes	20 units
	Leadspread	2x/machine/shift	5 units
	Coplanarity	2x/machine/shift	5 units
Solder	Visual Inspection	1 lot/package type/ shipment	22 units
	Ionic Contamination	Each shipment	1 sample/package type
	Solder Thickness (C of C)	1 in 5 lots	3 readings
	Solderability (after plating)	2x/week	12 units
	Outgoing Solderability	1x/week	4 units

Table 2-32. Quality Controls During FPGA Assembly (EIAJ Standard QFPs)

Operation	Inspection	Frequency	Sample Size
Wafer Saw	Visual Inspection	4x/shift	25 dice/machine
	Machine Inspection	1x/shift	Each machine
Die Attach	Visual Inspection	4x/shift	2 strips/machine
	Epoxy Resistivity	N/A	N/A
	Pin Shear Strength	2x/shift	2 units/machine
	Epoxy Void	N/A	N/A
	Machine Inspection	1x/shift	Each machine
	Oven Cure	1x/shift	Each oven
Wire Bond	Visual Inspection	4x/shift	2 strips/machine
	Bond Pull Strength	2x/shift	10 wires/machine
	Ball Shear Strength	N/A	N/A
	Machine Inspection	1x/shift	Each machine

Quality Monitors (continued)

Table 2-32. Quality Controls During FPGA Assembly (EIAJ Standard QFPs) (continued)

Operation	Inspection	Frequency	Sample Size
Mold	Visual Inspection	4x/shift	1 shot/machine
	X-ray	2x/shift	5 units/machine
	Machine Inspection	1x/shift	Each machine
Code Mark	Visual Inspection	4x/shift	2 strips/machine
	Mark Permanency	2x/shift	5 units/machine
	Machine Inspection	1x/shift	Each machine
Trim/Form	Visual Inspection	4x/shift	2 strips/machine
	Leadspread	4x/shift	2 strips/machine
	Coplanarity	4x/shift	2 strips/machine
Solder	Visual Inspection	4x/shift	2 strips/machine
	Ionic Contamination	N/A	N/A
	Solder Thickness	1x/shift	5 units/machine
	Solderability (after plating)	1x/shift	5 units/machine
	Outgoing Solderability	N/A	N/A

Table 2-33. Quality Controls During FPGA Assembly (Ceramic and Plastic Pin Grid Arrays)

Operation	Inspection	Frequency	Sample Size
Wafer Saw	Visual Inspection	Each magazine of wafers	4 wafers/magazine
Die Attach	Visual Inspection	100%	All packages
	Epoxy Resistivity	1x/day/machine	One slide
	Die Shear	1x/shift/machine	One package
	Bonder Temperature	1x/shift/machine	Each machine
	Oven Cure Profile	1x/month/machine	Each oven
Wire Bond	Visual Inspection	100%	All packages
	Bond Pull Strength	1x/shift/wafer lot	One package
	Ball Shear Strength	1x/shift/wafer lot	One package
	Workholder Temperature	1x/shift/machine	Each machine
	Preheater Temperature	1x/shift/machine	Each machine
Encapsulate	Visual Inspection	100%	All packages
Lid Seal	Visual Inspection	100%	All packages
	Fine Leak Test	100%	All packages
	Gross Leak Test	100%	All packages
Code Mark	Visual Inspection	100%	All packages

September 1991
Application Note

AT&T Microelectronics

ATT3000 Series Field-Programmable Gate Array (FPGA) Test Methodology

Contents

Introduction

AT&T Microelectronics is committed to providing the highest level of quality and reliability for the Field-Programmable Gate Array (FPGA). Quality is best ensured by taking the necessary steps to achieve zero defects. Comprehensive testing helps ensure that every FPGA device is free from defects and conforms to the *ATT3000 Series FPGAs* Data Sheet specifications. The memory-cell design ensures 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 FPGA 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 control and observance for comprehensive testability. The design schedule must also include time for the development of test vectors and a test program specification. If gate array users require a comprehensive test program, then they 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.

Testing (continued)

Field-Programmable Gate Array (FPGA): The testability of the FPGA device is similar to other standard products, including microprocessors and memories. The following is the result of the design and 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 FPGA user can better appreciate the FPGA test procedure by examining each of the testing requirements:

- All configuration-memory bits must be exercised and then verified. This is performed using readback mode.
- All possible process-related faults, such as short circuits, must be detected. The FPGA is configured such that every metal line can be driven and observed directly from the input/output pads.
- All testing configurations must provide good controllability and observability. This is possible since all configurable logic blocks can be connected to input/output pads. This makes them easy to control by testing different combinations of inputs and easy to observe by comparing the actual outputs with expected values.

These points bring out an important issue: the FPGA was carefully designed to achieve 100% fault coverage. With the AT&T testing strategy, the number of design configurations needed to fully test the FPGA 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 FPGA 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 FPGA 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 FPGA can build significant added value into the design by providing higher quality levels.

Testing of the FPGA

The FPGA device is tested as a standard product. Every device is tested for 100% functionality, dc parametrics, and speed. This allows the end user to design and use the FPGA without worrying about testing for a particular application.

The strategy for testing the FPGA device is to test the functionality of every internal element. These elements consist of memory cells, metal interconnects, transistor switches, bidirectional buffers, inverters, decoders, and multiplexers. If each element is functional, then the user's design will also be functional if the proper design procedures are used.

The static memory cells and the symmetry of the FPGA make it 100% testable. The FPGA 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 consists of a set of test vectors that configures the FPGA device with a hardware design that utilizes specific elements, and a set of test vectors that exercises those specific elements. The symmetry of the FPGA device allows the test engineer to develop the test for one configurable logic block (CLB) or configurable I/O block (IOB) and then apply it to all others. All configuration/test patterns are exercised at both Vcc minimum and maximum.

Testing of the FPGA (continued)

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 to 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.

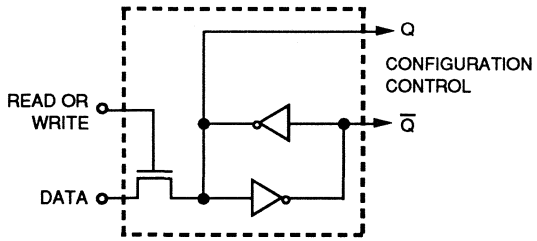


Figure 1. Configuration Memory Cell

The memory cells are directly tested in the FPGA with three test patterns that are equivalent to those used on a RAM device. The first test pattern writes 95% of all the RAM cells to a logic zero and then reads each RAM cell back to verify its contents. The second test pattern writes 95% of all the RAM cells to a logic one and also verifies the contents. The third pattern is used to verify that all I/O and configurable logic blocks can have their logic value read back correctly. All RAM cells are thus written to 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 test for continuity of each metal segment, test for shorts between metal segments, and 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, checks for continuity of the line.

Testing of the FPGA (continued)

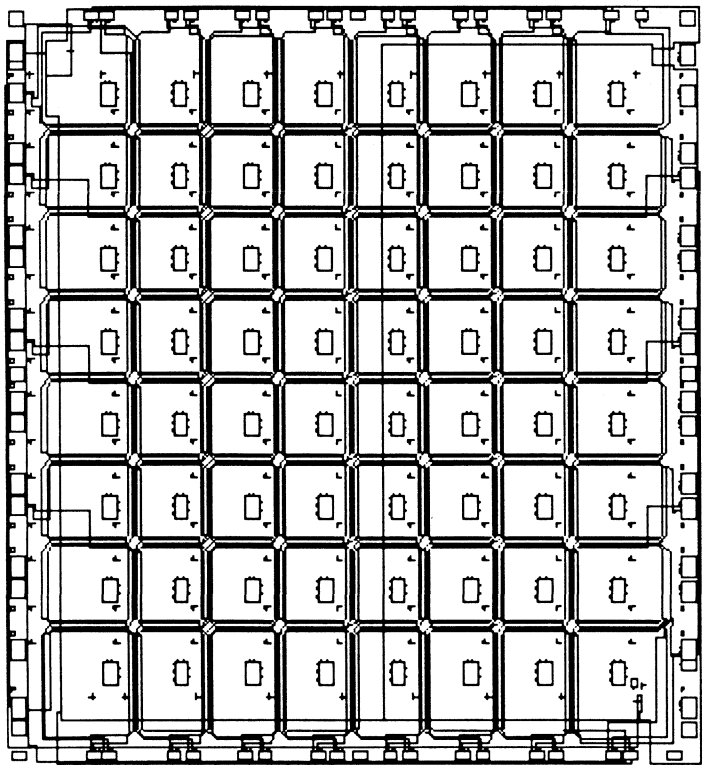


Figure 2. Interconnect Test Pattern

Testing of the FPGA (continued)

I/O Block Testing

Each I/O block includes registered and direct input paths and a programmable 3-state output buffer. The testing of these functions is accomplished by several configuration/test patterns that implement and test each option that is available to the user. One method used to test the I/O blocks is to configure them as a shift register that has a 3-state control (see Figure 3). This allows a test pattern to check the ability of each I/O block to latch and to output data that is derived from either another I/O block or from the tester. Several of these patterns are used to exercise different input and output combinations allowed for each I/O block. Configuration/test patterns are also used to precondition the device to test dc parameters such as V_{IH}, V_{IL}, V_{OH}, V_{OL}, TTL standby current, CMOS standby current, and input/output leakage. The V_{OH}/V_{OL} test is done while all outputs are either all low or all high.

Configurable Logic Block Testing

Each configurable logic block has a combinatorial-logic section, a flip-flop section, and an internal-control section. The combinatorial-logic section of the logic block uses a 32 x 1 array of RAM cells 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 two flip-flops of the logic block are tested with configuration/test patterns that configure the FPGA device as shift registers. Each shift register pattern will have different data in the look-up tables and will have a different pin used as the input to each shift register. Other configuration/test patterns are used to implement and test the internal-control section.

Testing the Speed of the FPGA

FPGA speed is checked with configuration/test patterns that have been correlated to ac values in the *ATT3000 Series FPGAs* Data Sheet.

Most of these patterns are shift registers with interconnect IOBs and CLBs in the data path (see Figure 4). They are designed with the idea that all elements in the path must be fast enough for the proper data to get to the next input of the shift register before the next clock occurs. If any element doesn't meet the specified ac value, then the shift register will clock in the wrong data and fail the test. The complexity of the logic between two shift register cells determines the maximum frequency required for the clock pulse input of the shift register. This can be used to reduce the performance requirement of the tester in use. The patterns used consist of a TCKO + TILO + INTERCONNECT + TICK for each shift register. This increases the shift register clock pulse separation time from 30 ns 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 FPGA Device

Currently, the FPGA is being tested on Advantest testers. The 3000 Series products are being tested on Advantest 3340 VLSI test system with one million vectors required for 3042—3090, and 512K vectors required for 3020—3030.

Hardware Testing Considerations for the FPGA Device (continued)

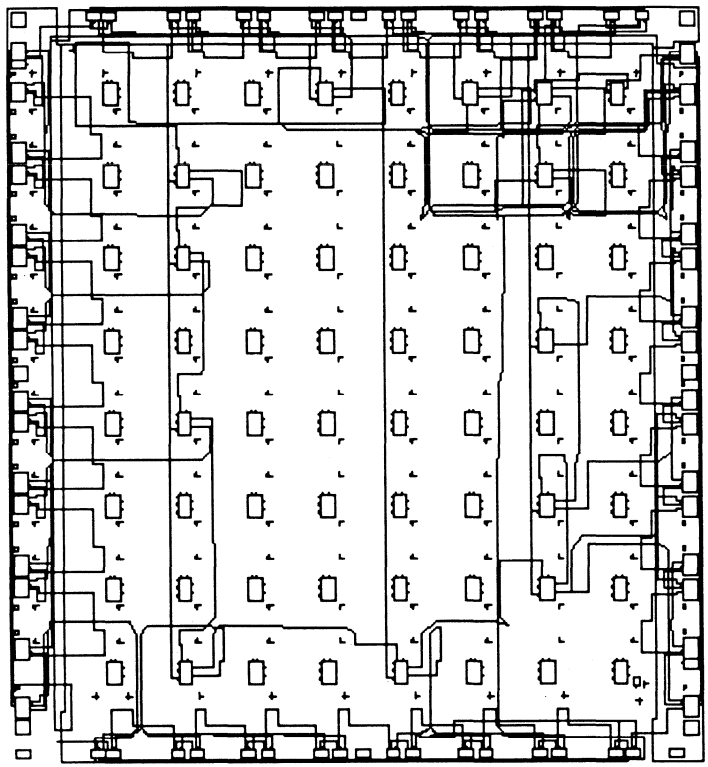


Figure 3. IOB Test Pattern

Hardware Testing Considerations for the FPGA Device (continued)

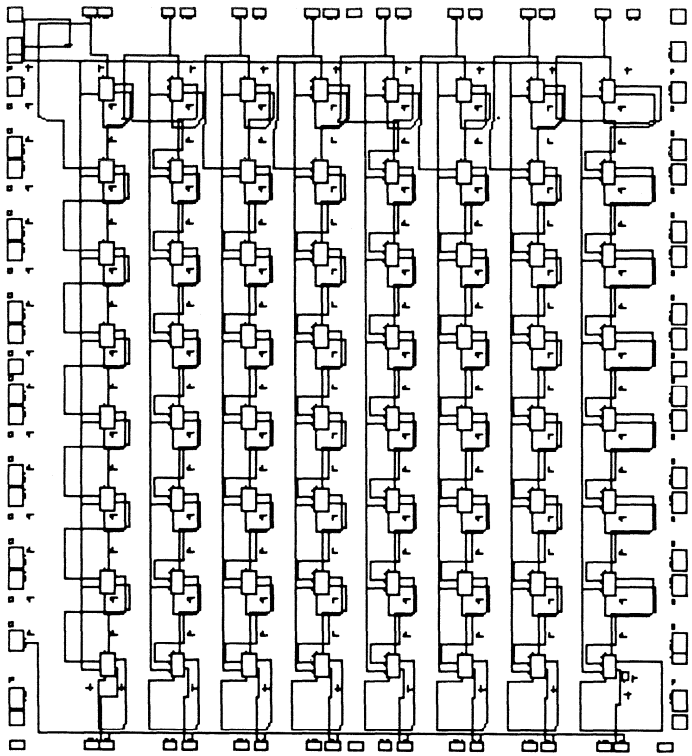


Figure 4. Speed Test Pattern

Section 3.
AT&T FPGA Product Specifications

3. AT&T FPGA Product Specifications

Table of Contents

Contents **Page**

Data Sheet
July 1992

AT&T Microelectronics

ATT3000 Series Field-Programmable Gate Arrays

FEATURES

- High Performance—up to 150 MHz Toggle Rates
- 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
  - Pin-for-Pin to *Xilinx* XC3000 Family
  - 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
  - Standard PROM File Interface

DESCRIPTION

The CMOS ATT3000 series Field-Programmable Gate Array (FPGA) 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 FPGA is shown in Figure 1 on the next page. The *XACT* development system provides schematic capture and auto place-and-route for design entry. Logic and timing simulation, and in-circuit emulation are available as design verification alternatives. The design editor is used for interactive design optimization, and to compile the data pattern which represents the configuration program.

The FPGA'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. A serial configuration PROM can provide a very simple serial configuration program storage.

Basic Array	Logic Capacity (usable gates)	Configurable Logic Blocks	User I/Os	Program Data (bits)
ATT3020	2000	64	64	14779
ATT3030	3000	100	80	22176
ATT3042	4200	144	96	30784
ATT3064	6400	224	120	46064
ATT3090	9000	320	144	64160

The ATT3000 series FPGAs are an enhanced family of Field 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 FPGA 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 FPGA at power-up and may be reloaded on command. The FPGA includes logic and control signals to implement automatic or passive configuration. Program data may be either bit serial or byte parallel. The *XACT* development system generates the configuration program bit-stream used to configure the FPGA. The memory loading process is independent of the user logic functions.

CONFIGURATION MEMORY

The static memory cell used for the configuration memory in the FPGA has been designed specifically for high reliability and noise immunity. Integrity of the FPGA configuration memory based on this design is assured even under various adverse conditions. Compared with other programming alternatives, static memory is believed to provide 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.

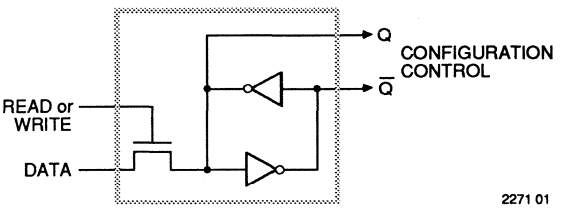


Figure 2. A static configuration memory cell is loaded with one bit of configuration program and controls one program selection in the FPGA.

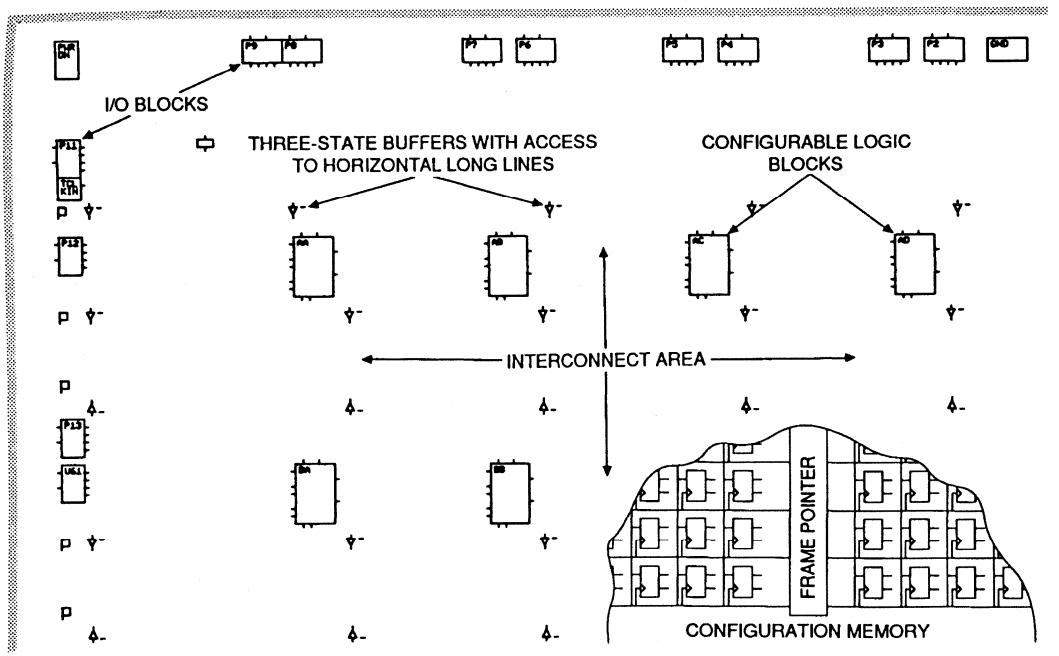


Figure 1. The structure of the Field-Programmable Gate Array consists of a perimeter of programmable I/O blocks, a core of configurable logic blocks and their interconnect resources. These are all controlled by the distributed array of configuration program memory cells.

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 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 AT&T 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 pro-

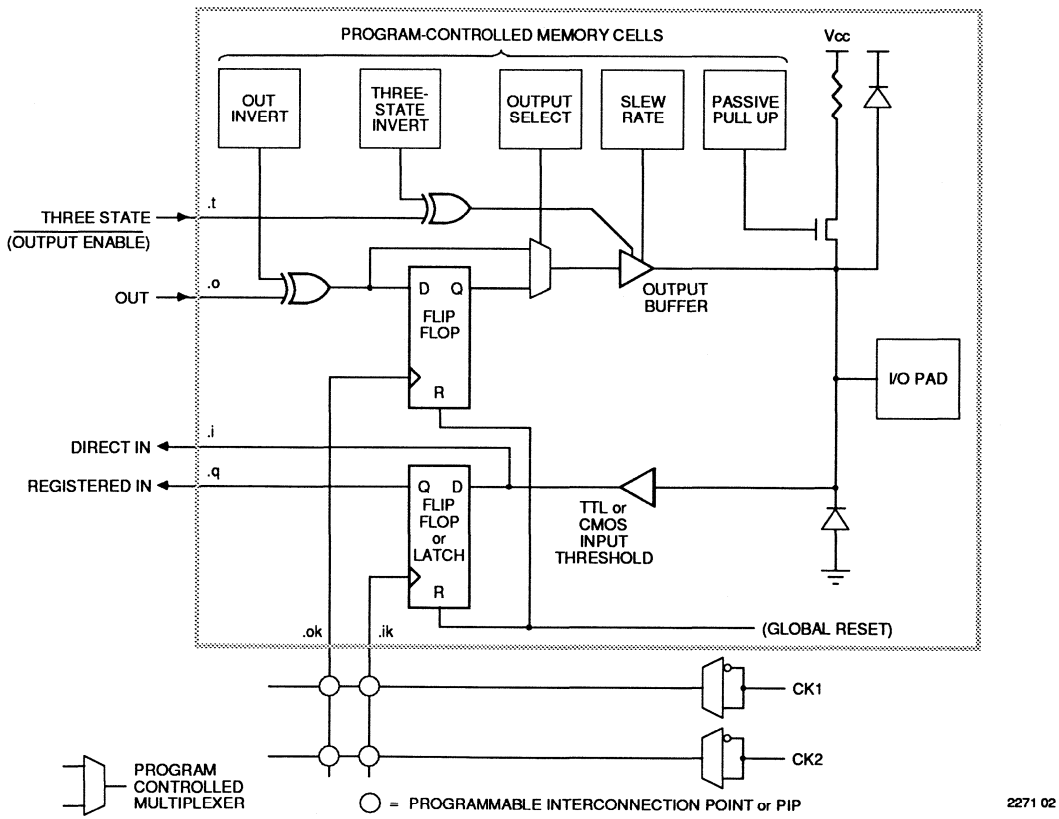


Figure 3. The Input/Output Block includes input and output storage elements and I/O options selected by configuration memory cells.
A choice of two clocks is available on each die edge.
The polarity of each clock line (not each flip-flop or latch) is programmable. A clock line that triggers the flip-flop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

vides 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 threshold 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 200 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 FPGA 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 FPGA, 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 ATT3020 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.

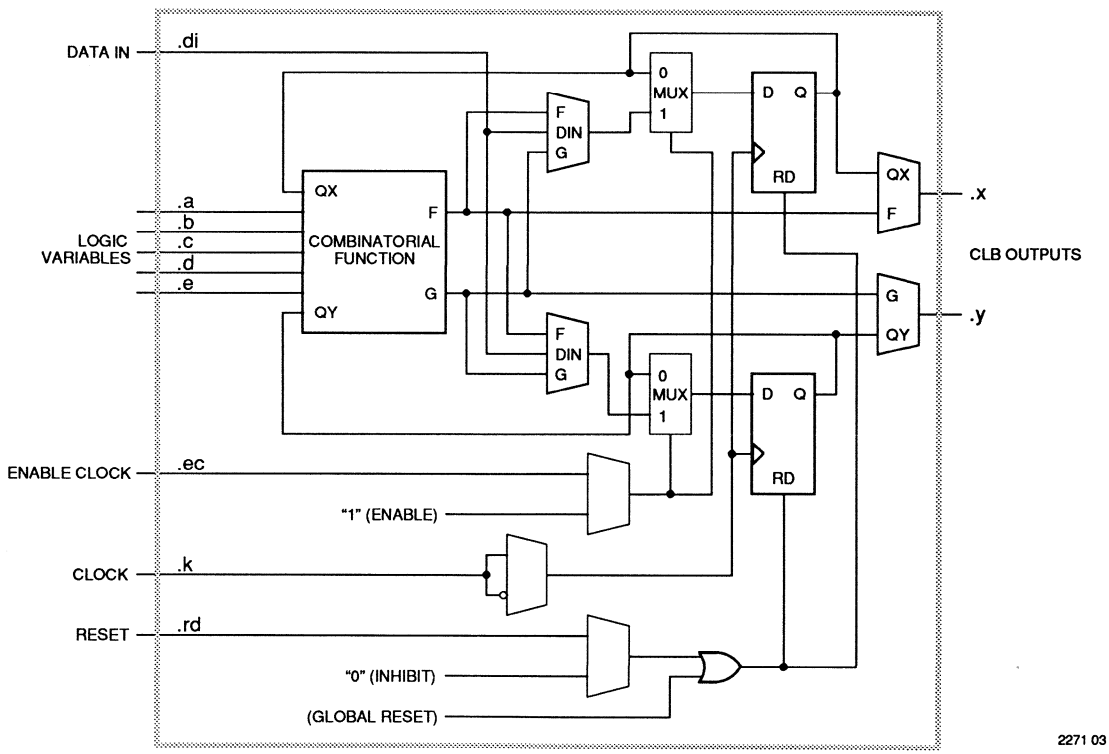


Figure 4. Each Configurable Logic Block includes a combinatorial logic section, two flip-flops and a program memory controlled multiplexer selection of function.
It has: five logic variable inputs .a, .b, .c, .d and .e.
a direct data in .di
an enable clock .ec
a clock (invertible) .k
an asynchronous reset .rd
two outputs .x and .y

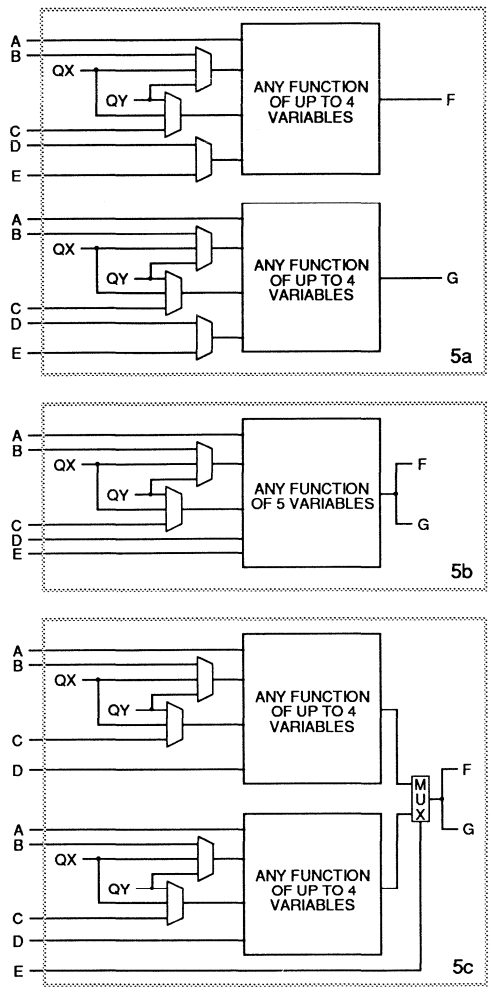


Figure 5

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.

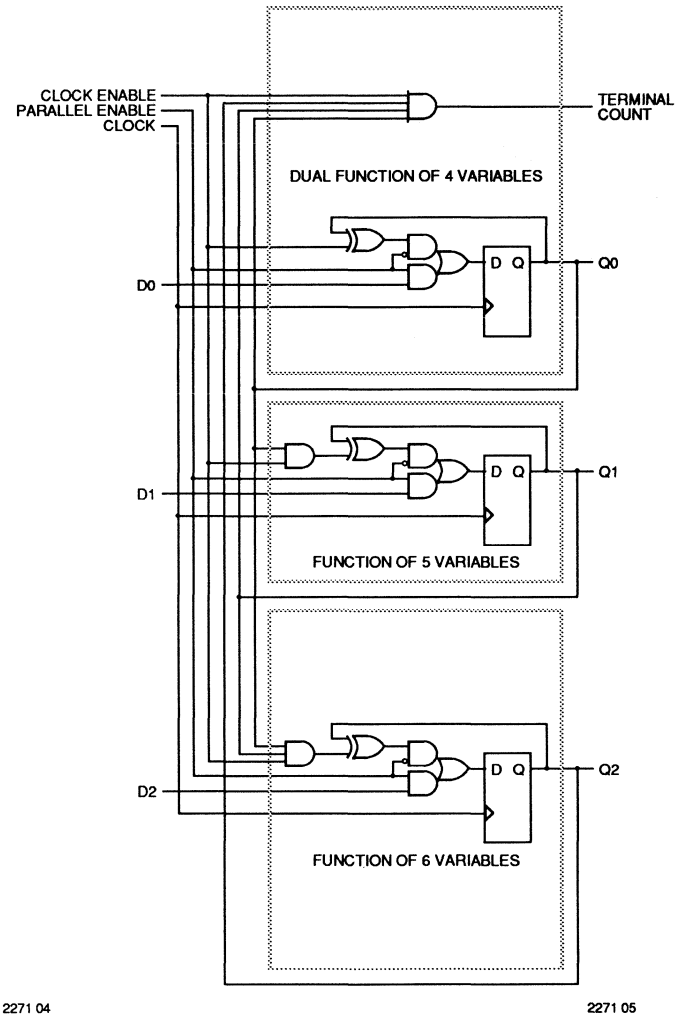


Figure 6. The C8BCP macro (modulo 8 binary counter with parallel enable and clock enable) uses one combinatorial logic block of each option.

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 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 FPGA 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 pro-

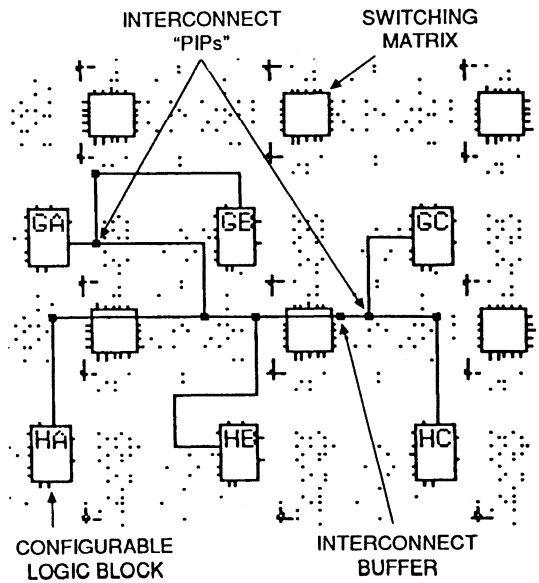


Figure 7. An *XACT* view of routing resources used to form a typical interconnection network from CLB GA.

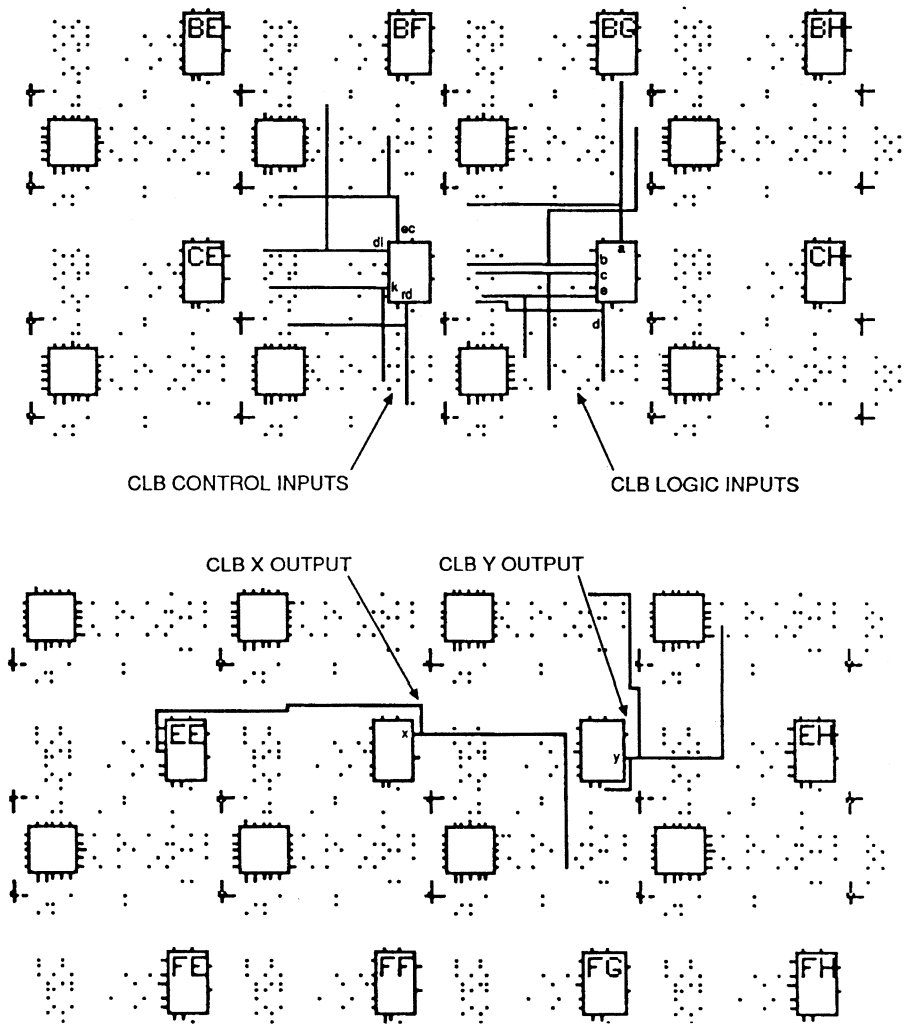


Figure 8. The AT&T *XACT* Development System view of the locations of interconnect access, CLB control inputs, logic inputs and outputs. The dot pattern represents the available programmable interconnection points (PIPs).

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."

grammed 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*.

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 bi-directional (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

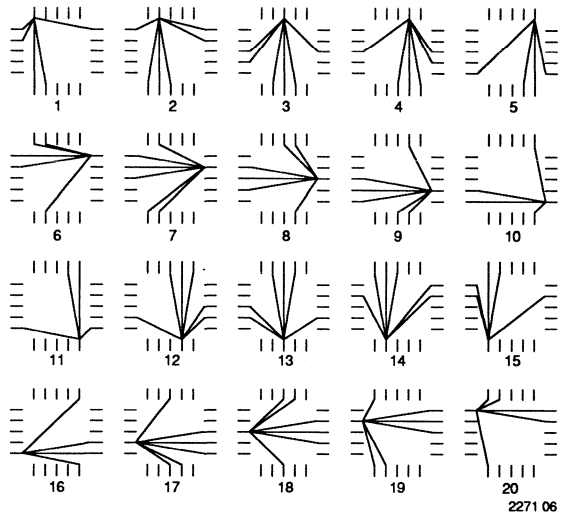


Figure 10. Switch matrix interconnection options for each "pin." Switch matrices on the edges are different. Use "Show Matrix" menu option in *XACT*.

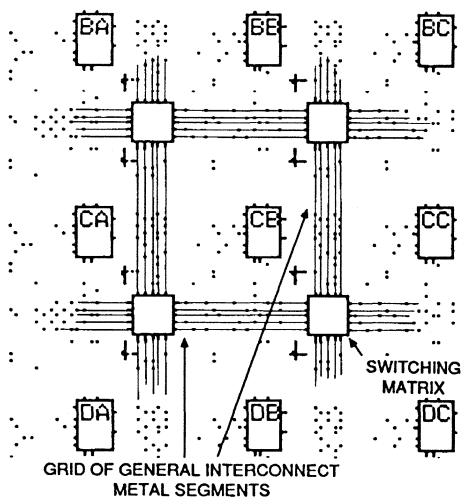


Figure 9. FPGA general-purpose interconnect is composed of a grid of metal segments which may be interconnected through switch matrices to form networks for CLB and I/O block inputs and outputs.

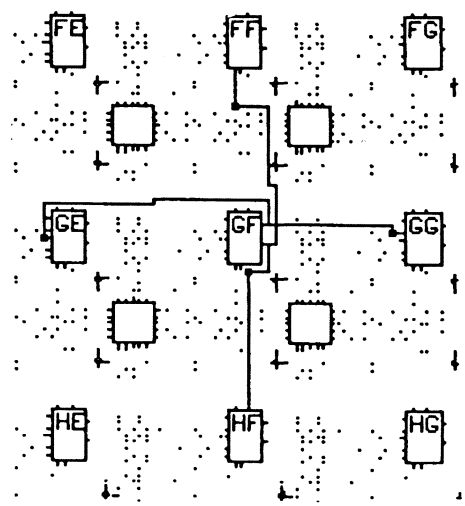


Figure 11. The .x and .y outputs of each CLB have single contact, direct access to inputs of adjacent CLBs.

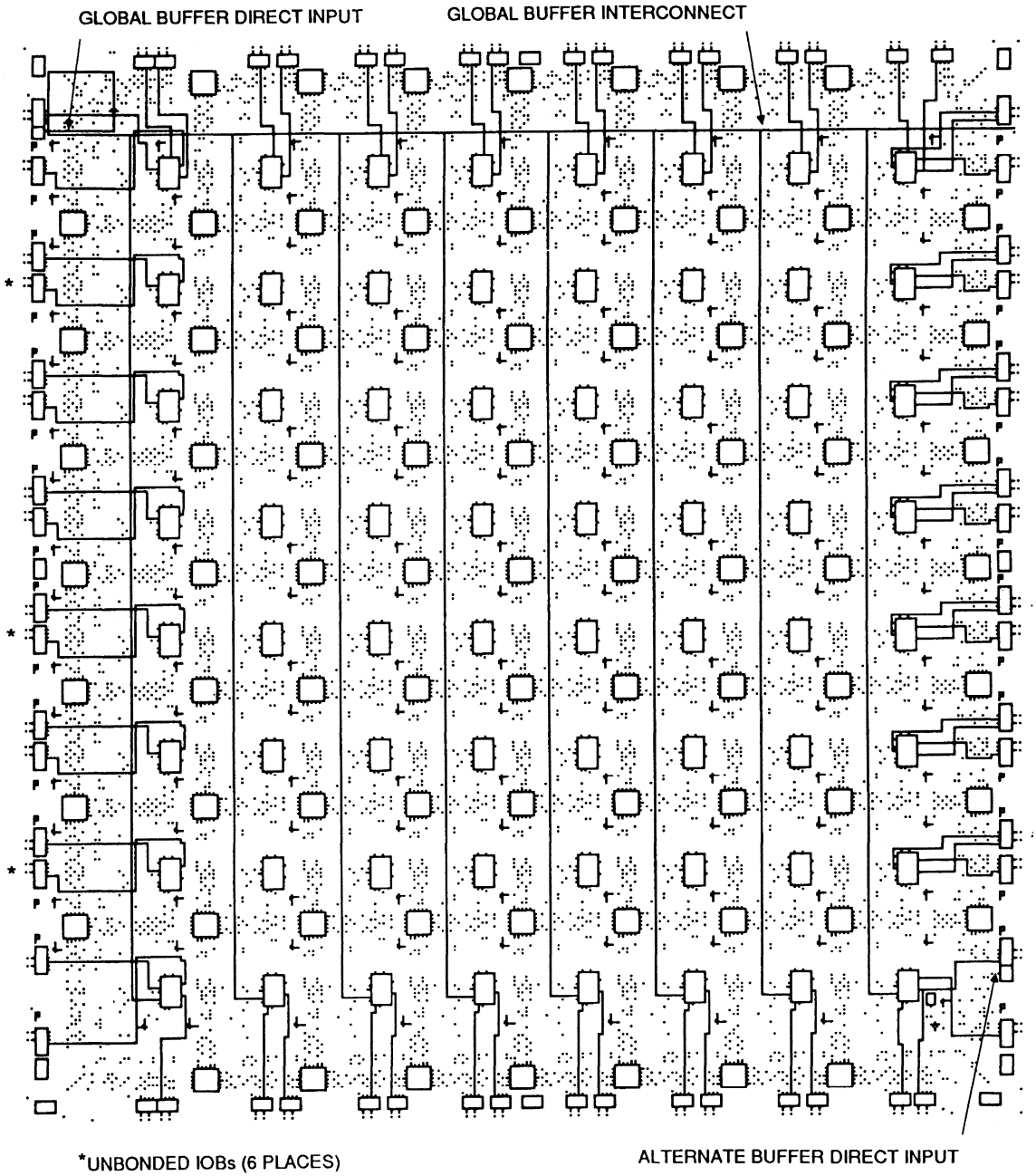


Figure 12. ATT3020 die edge I/O blocks are provided with direct access to adjacent CLBs.

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 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. Two vertical long lines in each column are connectible half-length lines, except on the ATT3020, where only the outer long lines serve that function.

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 FPGA chip drives a global net which is available to all *.k* inputs of logic blocks. Using the global buffer for a clock signal provides a skew-free, high fan-out, synchronized clock for use at any or all of the 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.

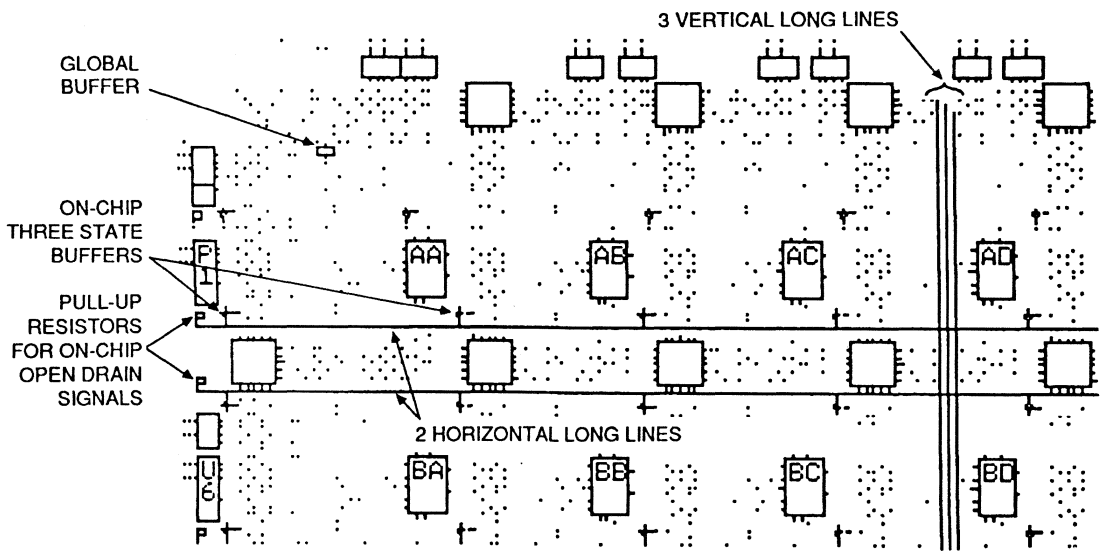


Figure 13. Horizontal and vertical long lines provide high fan-out, low-skew signal distribution in each row and column. The global buffer in the upper left die corner drives a common line throughout the FPGA.

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 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 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

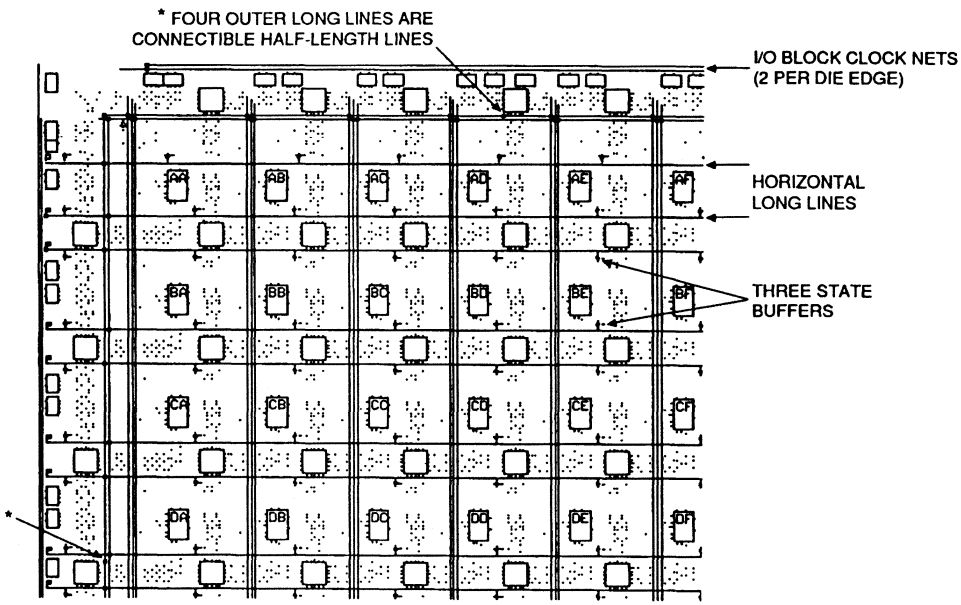
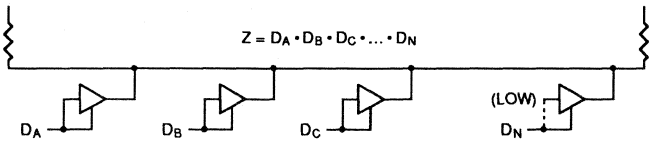
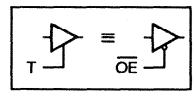


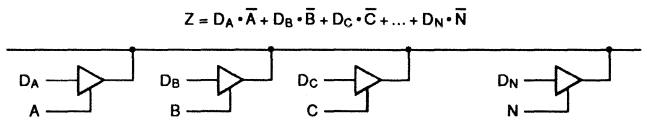
Figure 14. Programmable interconnection of long lines is provided at the edges of the routing area. Three-state buffers allow the use of horizontal long lines to form on-chip wired-AND and multiplexed buses. The left two vertical long lines per column (except ATT3020) and the outer perimeter long lines may be programmed as connectible half-length.



2271 07

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.





2271 08

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

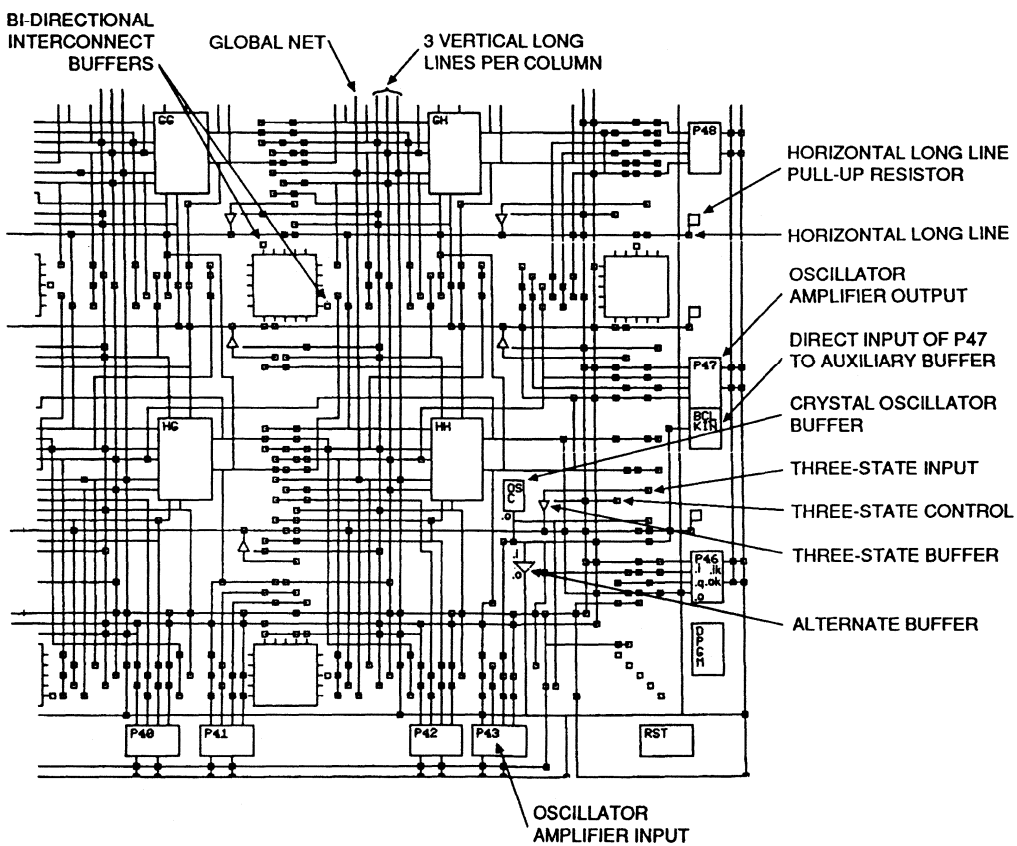


Figure 16. An *XACT* Development System extra large view of possible interconnections in the lower right corner of the ATT3020.

is designed to be used from 1 MHz to one-half the specified CLB toggle frequency. Use at frequencies below 1 MHz may require individual characterization with respect to a series resistance. Crystal oscillators above 20 MHz generally require a crystal which operates in a third overtone mode, where the fundamental frequency must be suppressed by 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

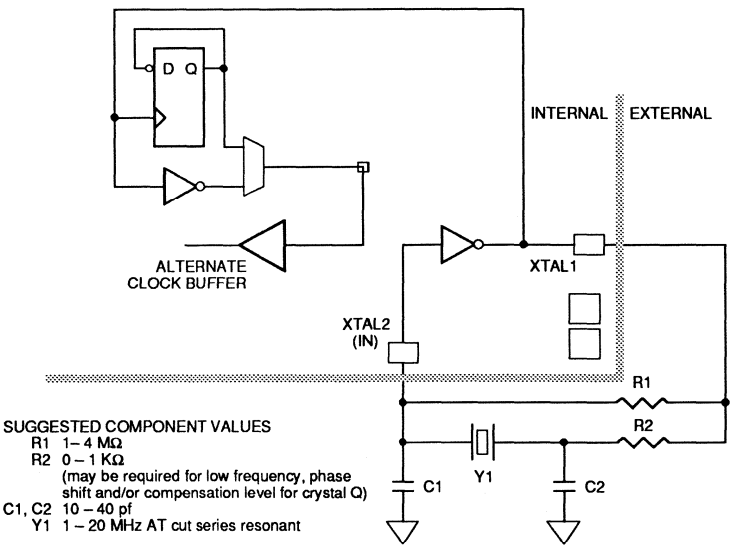
Initialization Phase

An internal power-on-reset circuit is triggered when power is applied. When Vcc reaches the voltage at which portions of the FPGA 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 FPGA 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 FPGA with mode lines se-

Table 1

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



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

2271 09

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.

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

If a configuration has begun, a re-assertion of $\overline{\text{RESET}}$ for a minimum of three internal timer cycles will be recognized and the FPGA will initiate an abort, returning to the Clear state to clear the partially loaded configuration memory words. The FPGA will then re-sample $\overline{\text{RESET}}$ and the mode lines before re-entering the Configuration state. A re-program is initiated when a configured FPGA senses a HIGH to LOW transition on the DONE/$\overline{\text{PROG}}$ package pin. The FPGA 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 FPGAs, 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 FPGAs connected in series read and shift preamble and length count in on positive and out on negative configuration clock edges. An FPGA which has received the preamble and length count then presents a HIGH Data Out until it has intercepted the appropriate number of data frames. When the configuration program memory of an FPGA is full and the length count does not compare, the FPGA shifts any additional data through, as it did for preamble and length count.

When the FPGA configuration memory is full and the length count compares, the FPGA will execute a synchronous start-up sequence and become operational. See Figure 20. Three CCLK cycles after the completion of loading configuration data the user I/O pins are enabled as configured. As selected in MAKEBITS, the internal user-logic reset is released either one clock cycle before or after the I/O pins become active. A similar timing selection is programmable for the DONE/$\overline{\text{PROG}}$ output signal. DONE/$\overline{\text{PROG}}$ 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 FPGA is in its Initialization, Clear or Configure states. They and DONE/$\overline{\text{PROG}}$ provide signals for control of external logic signals such as reset, bus enable or PROM enable during configuration. For

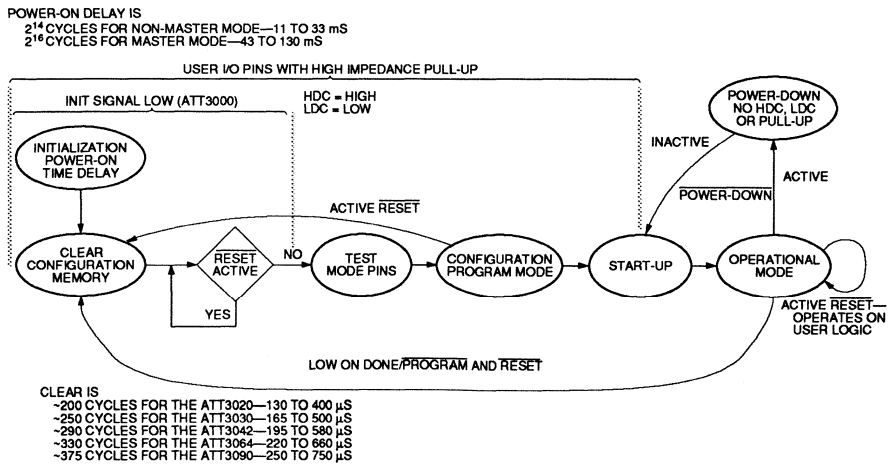


Figure 18. A state diagram of the configuration process for power-up and reprogram.

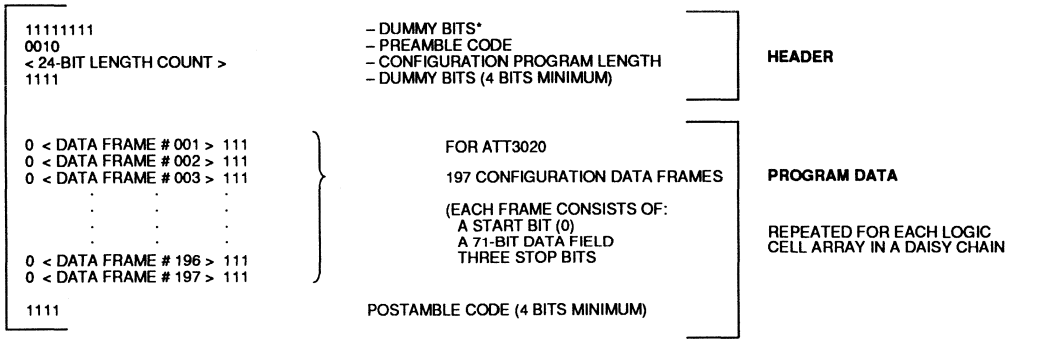
parallel Master configuration modes these signals provide PROM enable control and allow the data pins to be shared with user logic signals.

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

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

Configuration Data

Configuration data to define the function and interconnection within a FPGA 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



*The FPGA Devices Require 4 Dummy Bits Min., *XACT* 3.0 Generates 8 Dummy Bits

2271 11

Device	ATT3020	ATT3030	ATT3042	ATT3064	ATT3090
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

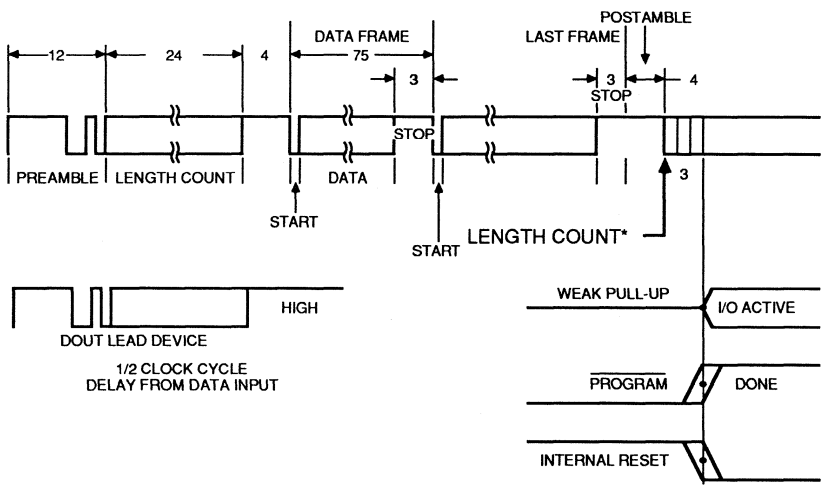
Figure 19. The internal Configuration Data Structure for an FPGA shows the preamble, length count and data frames which are generated by the *XACT* Development System.

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] – ($2 \leq K \leq 4$) where K is a function of DONE and RESET timing selected. An additional 8 is added if roundup increment is less than K. K additional clocks are needed to complete start-up after length count is reached.

mode selection pins at the start of configuration time determine the method to be used. See Table 1. The data may be either bit-serial or byte-parallel, depending on the configuration mode. Various AT&T Programmable Gate Arrays have different sizes and numbers of data frames. For the ATT3020, 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. 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 EDITFPGA 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 FPGA 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 FPGA, 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.

Two user programmable pins are defined in the unconfigured FPGA. High During Configuration (HDC) and Low During Configuration (LDC) as well as DONE/PROG may be used as external control signals during configuration. In Master mode configurations it is convenient to use LDC as an active-low EPROM Chip Enable. After the last configuration data-bit is loaded and the length count compares, the user I/O pins become active. Options 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/PROG out-



* 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.

2271 12

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

put can be AND-tied with multiple FPGAs 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 FPGA 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 serial configuration PROM shown in Figure 21. Parallel Master Low and Master High modes automatically use parallel data supplied to the D0–D7 pins in response to the 16-bit address generated by the FPGA. Figure 22 shows an example of the parallel Master mode connections required. The FPGA 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

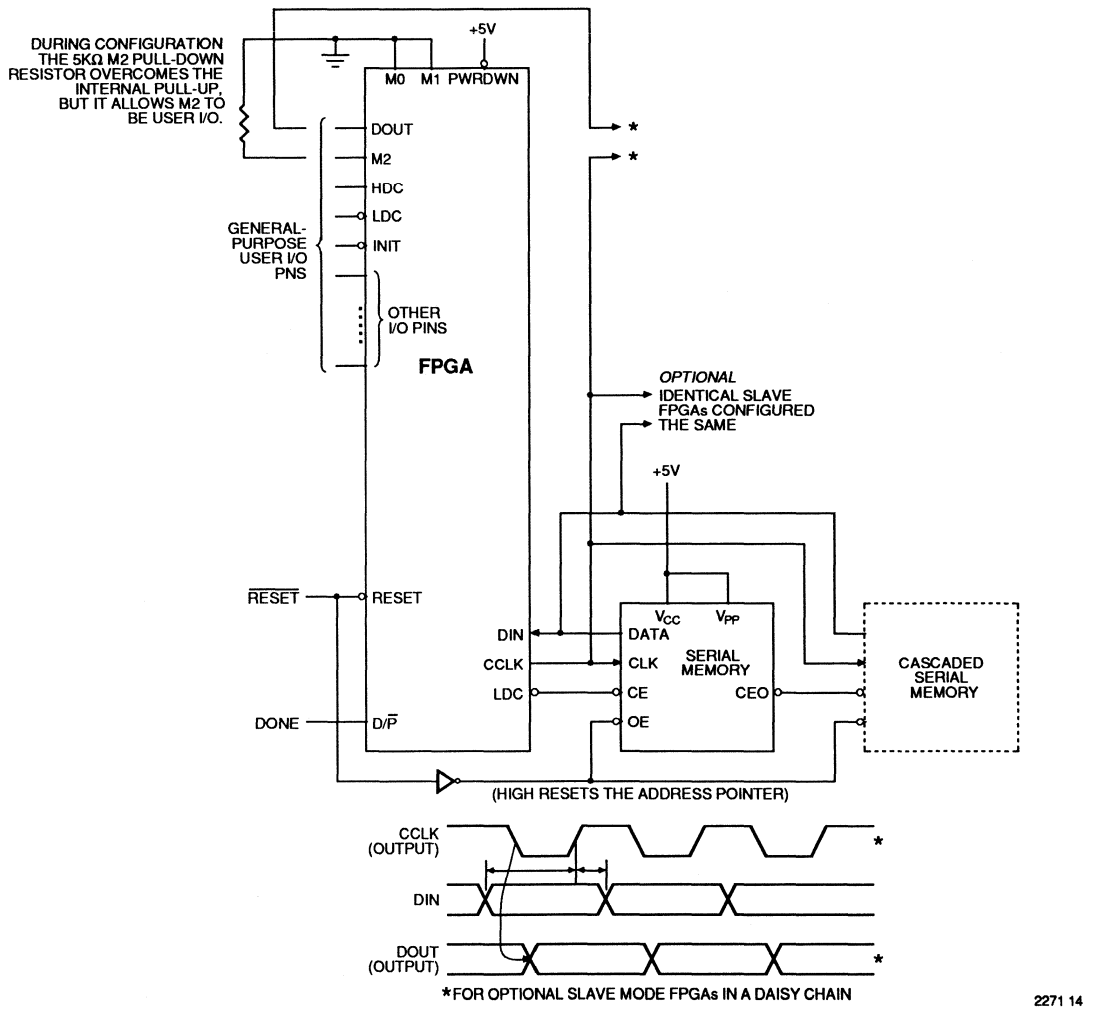


Figure 21. Master Serial Mode. The serial configuration PROM supports automatic loading of configuration programs up to 36K/64K bits. Multiple devices can be cascaded to support additional FPGAs. An early DONE inhibits the data output a $\overline{\text{CCLK}}$ cycle before the FPGA I/O become active.

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 FPGA can be used to interface the configuration program-store and pass additional concatenated configuration data to additional FPGAs in a serial daisy-chain fashion. CCLK is provided for the slaved devices and their serialized data is supplied from DOUT to DIN - DOUT to DIN etc.

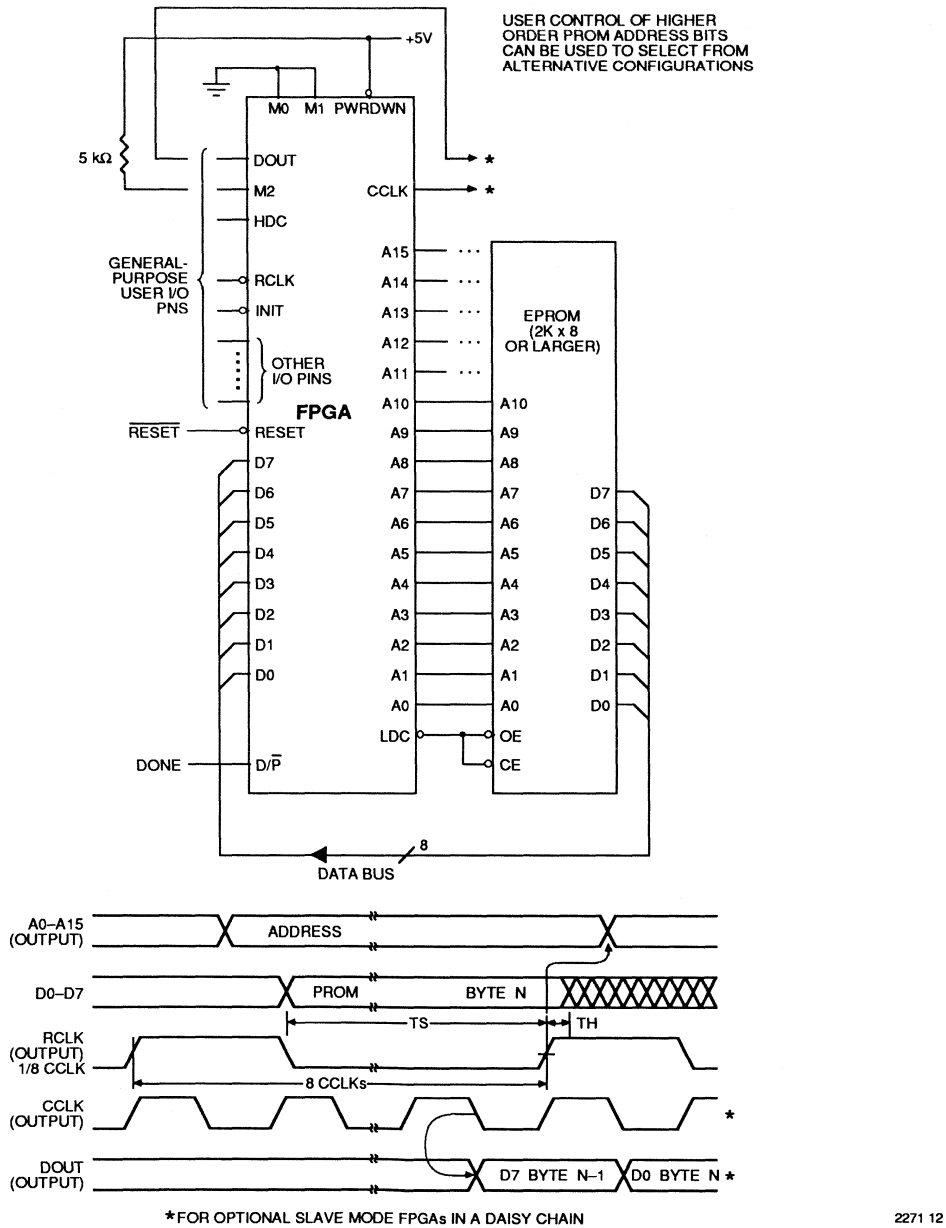


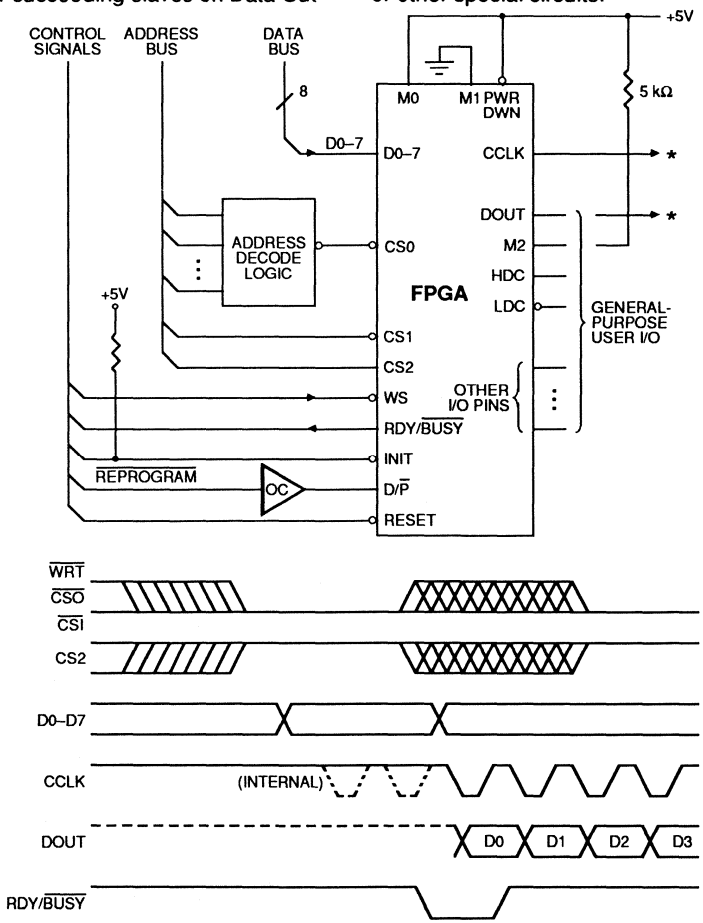
Figure 22. Master Parallel Mode. Configuration data are loaded automatically from an external byte wide PROM. An early DONE inhibits the PROM outputs a CCLK before the FPGA I/O become active.

Peripheral Mode

Peripheral mode provides a simplified interface through which the device may be loaded byte-wide, as a processor peripheral. Figure 23 shows the peripheral mode connections. Processor write cycles are decoded from the common assertion of the active low Write Strobe ($\overline{WS}$), and two active low and one active high Chip Selects ($\overline{CS0}$, $\overline{CS1}$, CS2). If all these signals are not available, the unused inputs should be driven to their respective active levels. The FPGA 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 FPGA generates a configuration clock from the internal timing generator and serializes the parallel input data for internal framing or for succeeding slaves on Data Out (DOUT). A output HIGH on READY/$\overline{BUSY}$ pin indicates the completion of loading for each byte when the input register is ready for a new byte. As with Master modes, Peripheral mode may also be used as a lead device for a daisy-chain of slave devices.

Slave Mode

Slave mode provides a simple interface for loading the FPGA 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 FPGA's data out, while the clock is supplied by a lead device in Master or Peripheral mode. Data may also be supplied by a processor or other special circuits.



*FOR OPTIONAL SLAVE MODE FPGAs IN A DAISY CHAIN

2271 15

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

Daisy-Chain

The *XACT* development system is used to create a composite configuration bit stream for selected FPGAs 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 pre-amble 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 FPGAs. 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.

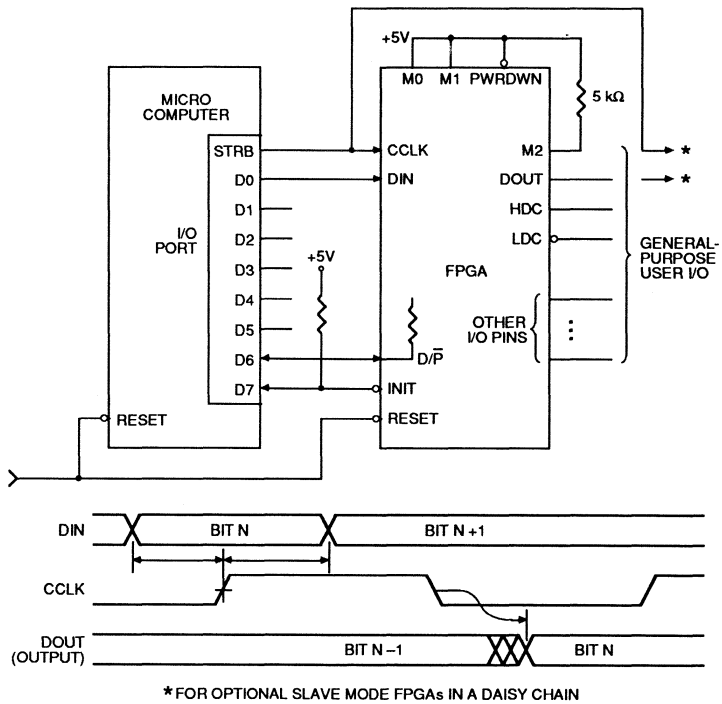


Figure 24. Slave Mode. Bit-serial configuration data are read at rising edge of the CCLK. Data on DOUT are provided on the falling edge of CCLK.

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.

Input Thresholds

Prior to the completion of configuration all FPGA input thresholds are TTL compatible. Upon completion of configuration the input thresholds become either TTL or CMOS compatible as programmed. The use of the TTL threshold option requires some additional supply current for threshold shifting. The exception is the threshold of the PWRDWN input and direct clocks which always have a CMOS input. Prior to the completion of configuration the user I/O pins each have a high impedance pull-up. The configuration program can be used to enable the 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.

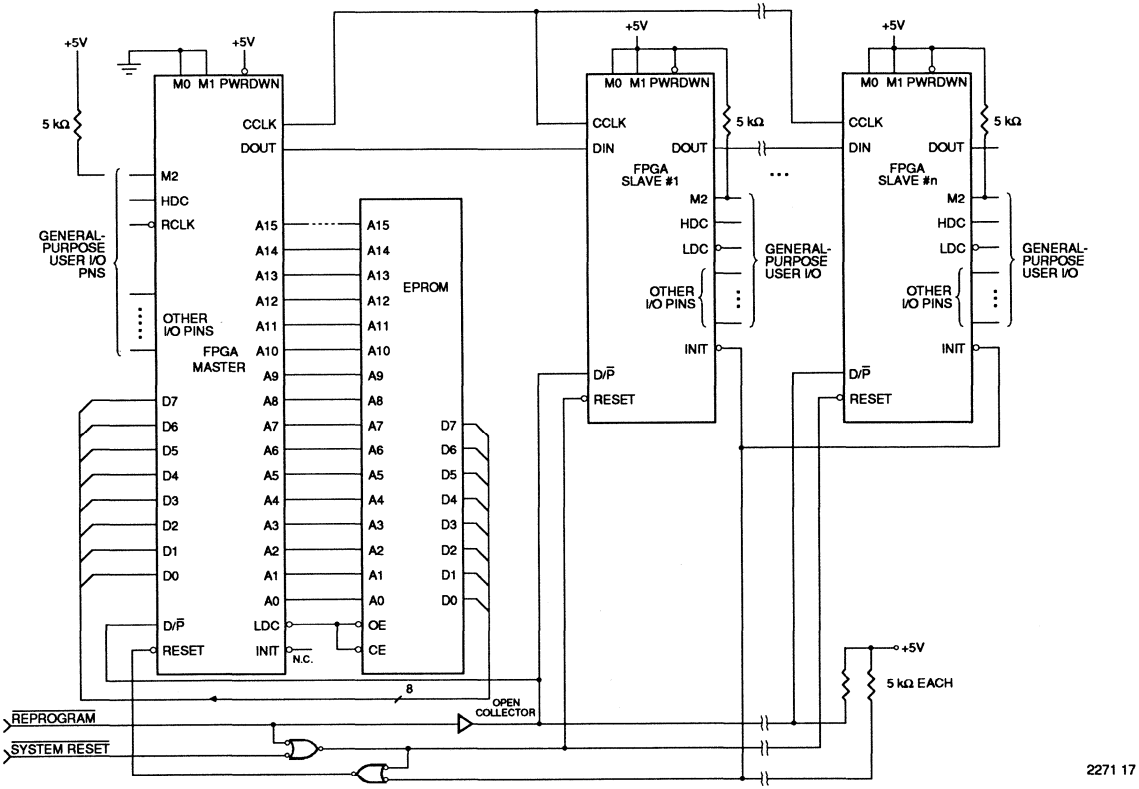


Figure 25. Master Mode configuration with daisy chained slave mode devices. All are configured from the common EPROM source. The Slave mode device INIT signals delay the Master device configuration until they are initialized. A well defined termination of SYSTEM RESET is needed when controlling multiple FPGAs.

Readback

The contents of a FPGA 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 Data I/O Mesa-I In-Circuit Verifier. 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.

Readback is accomplished without the use of any of the user I/O pins; only M0, M1 and CCLK are used. The initiation of readback is produced by a LOW to HIGH transition of the M0/RTRIG (Read Trigger) pin. Once the readback command has been given, the input CCLK is driven by external logic to read back each data bit in a format similar to loading. After two dummy bits, the first data frame is shifted out, in inverted sense, on the M1/$\overline{\text{RDATA}}$ (Read Data) pin. All data frames must be read back to complete the process and return the mode select and CCLK pins to their normal functions.

The readback data includes the current state of each internal logic block storage element, and the state of the [*.i* and *.ri*] connection pins on each I/O Block. These data are imbedded into unused configuration bit positions during readback. This state information is used by the FPGA 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 FPGA 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/$\overline{\text{PROG}}$ must be given a HIGH to LOW transition. To reduce sensitivity to noise, the input signal is filtered for 2 cycles of the FPGA'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 $\overline{\text{INIT}}$ 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/$\overline{\text{PROG}}$ LOW. Once it recognizes a stable request, the FPGA 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 FPGA will begin operation upon completion of configuration.

DONE Pull-up

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

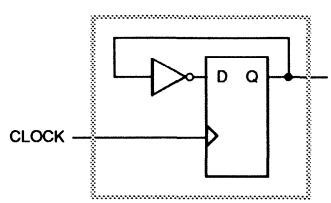
Logic Block Performance

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

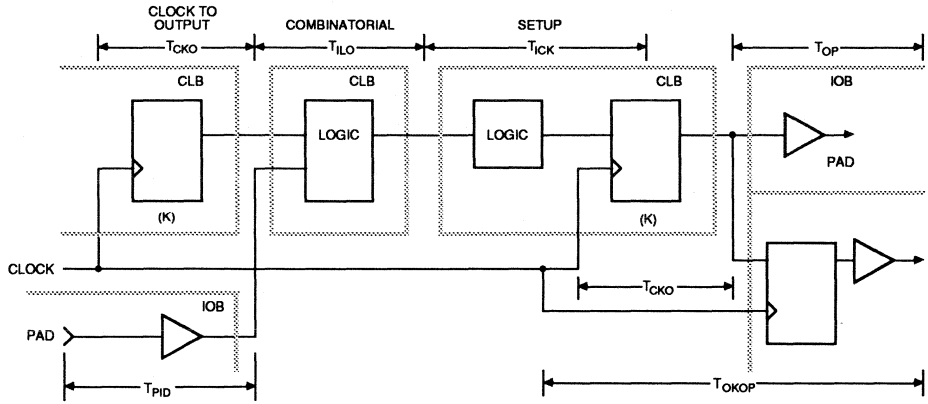
Interconnect performance depends on the routing resource used to implement the signal path. As discussed earlier, direct interconnect from block to block provides a fast path for a signal. The single metal segment used for Long lines exhibits low resistance from end to end, but relatively high capacitance. 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 are functions 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



2271 18

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



		Speed (4)	-70		-100		-125		150		Unit
Signal	Description	Symbol	Min	Max	Min	Max	Min	Max	Min	Max	
Logic input to Output	Combinatorial	TILO		9		7		5.5		4.6	ns
K Clock	To output	TCKO		6		5		4.5		4	ns
	Logic-input setup	TICK	8		7		5.5		4.6		ns
	Logic-input hold	TCKI	0		0		0		0		ns
Input/Output	Pad to input (direct)	TPID		6		4		3		2.8	ns
	Output to pad (fast)	TOP		9		6		5		4.5	ns
	I/O clock to pad (fast)	TOKPO		13		10		9		7	ns
FF toggle frequency		FCLK		70		100		125		150	MHz

▨ = Preliminary.

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.

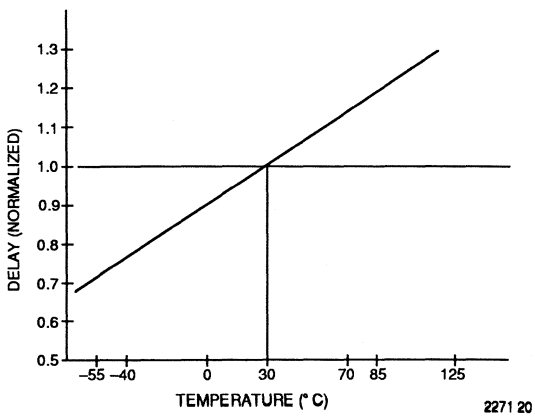


Figure 28. Change in speed performance as a function of temperature, normalized for 30°C.

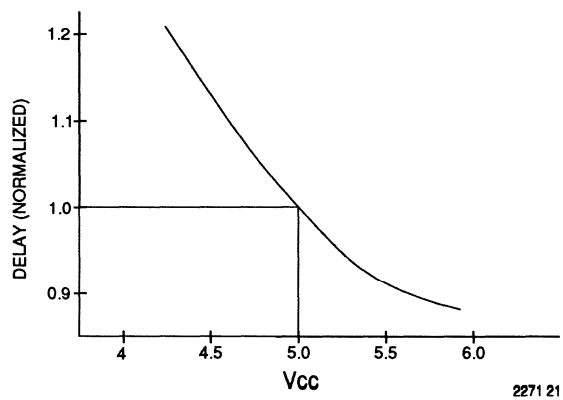


Figure 29. The speed performance of a CMOS device increases with Vcc within the operating range.

the actual block inputs is not significant; the capacitance is in the interconnect metal and switches. Figure 30 illustrates this.

POWER

Power Distribution

Power for the FPGA is distributed through a grid to achieve high noise immunity and isolation between logic and I/O. Inside the FPGA, 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. 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 FPGA 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 FPGA.

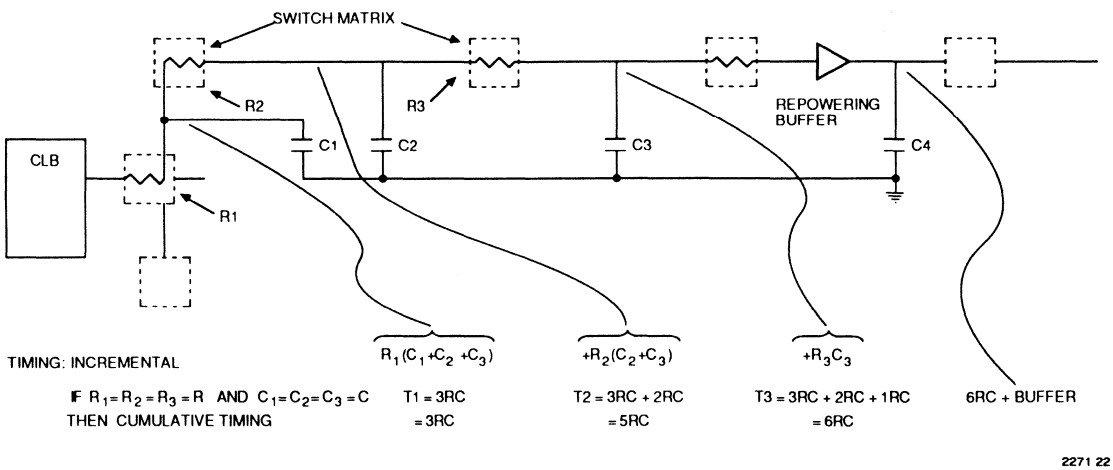


Figure 30. Interconection timing example. Use of the *XACT* timing calculator or *XACT*-generated simulation model provide actual worstcase performance information.

Internal power dissipation is a function of the number and size of the nodes, and the frequency at which they change. In an FPGA 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 ATT3020 and 3.6 mW/MHz for the ATT3090. 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.

Total Power = Vcc • Icco + external (DC + capacitive) + internal (CLB + IOB + Long Line + pull-up)

Because the control storage of the FPGA 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 FPGA 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 FPGA 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 FPGA 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.

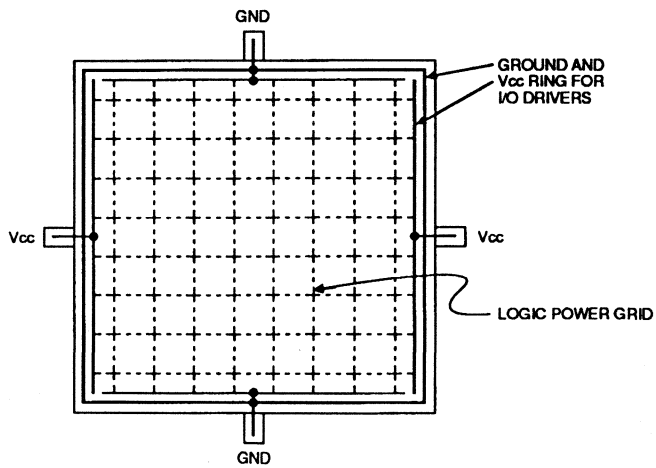


Figure 31. FPGA Power Distribution.

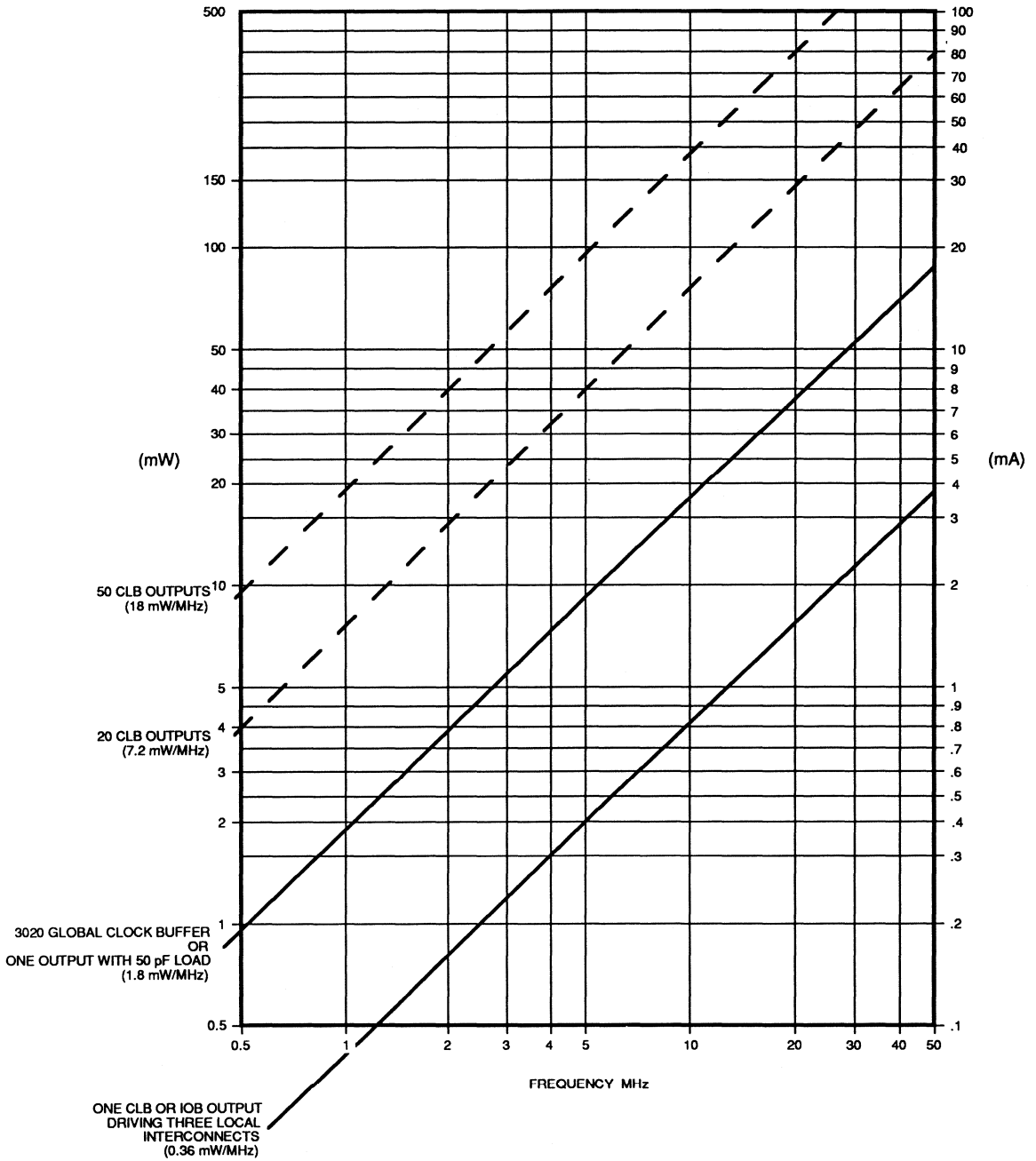


Figure 32. FPGA Power Consumption by Element. Total chip power is the sum of Vcc•Icco plus effective internal and external values of frequency dependent capacitive charging currents and duty factor dependent resistive loads.

PIN DESCRIPTIONS

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/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 functions.

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

If $\overline{\text{RESET}}$ is asserted during a configuration, the FPGA 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 FPGA.

CCLK
During configuration, Configuration Clock is an output of an FPGA in Master mode or Peripheral mode. FPGAs 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 FPGA becomes active in a synchronous order, and DONE may be programmed to occur one cycle before or after that.

$\overline{\text{PROG}}$
Once configuration is done, a HIGH to LOW transition of this pin will cause an initialization of the FPGA 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 FPGA 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 FPGA 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).

Table 2a. ATT3000 Family Configuration Pin Assignments

Configuration Mode: <M2:M1:M0>					***		**				User
Slave <1:1:1>	Master-SER <0:0:0>	Peripheral <1:0:1>	Master-High <1:1:0>	Master-Low <1:0:0>	44 PLCC	68 PLCC	84 PLCC	84 PGA	100 PQFP	100 CQFP	Operation
$\overline{\text{PWR DWN}}$ (I)	$\overline{\text{PWR DWN}}$ (I)	$\overline{\text{PWR DWN}}$ (I)	$\overline{\text{PWR DWN}}$ (I)	$\overline{\text{PWR DWN}}$ (I)	7	10	12	B2	29	14	$\overline{\text{PWR DWN}}$ (I)
VCC	VCC	VCC	VCC	VCC	12	18	22	F3	41	26	VCC
M1 (HIGH) (I)	M1 (LOW) (I)	M1 (LOW) (I)	M1 (HIGH) (I)	M1 (LOW) (I)	16	25	31	J2	52	37	$\overline{\text{RDATA}}$
M0 (HIGH) (I)	M0 (LOW) (I)	M0 (HIGH) (I)	M0 (LOW) (I)	M0 (LOW) (I)	17	26	32	L1	54	39	RTRIG (I)
M2 (HIGH) (I)	M2 (LOW) (I)	M2 (HIGH) (I)	M2 (HIGH) (I)	M2 (HIGH) (I)	18	27	33	K2	56	41	I/O
HDC(HIGH)	HDC(HIGH)	HDC(HIGH)	HDC(HIGH)	HDC(HIGH)	19	28	34	K3	57	42	I/O
$\overline{\text{LDC}}$ (LOW)	$\overline{\text{LDC}}$ (LOW)	$\overline{\text{LDC}}$ (LOW)	$\overline{\text{LDC}}$ (LOW)	$\overline{\text{LDC}}$ (LOW)	20	30	36	L3	59	44	I/O
$\overline{\text{INIT}}$ *	$\overline{\text{INIT}}$ *	$\overline{\text{INIT}}$ *	$\overline{\text{INIT}}$ *	$\overline{\text{INIT}}$ *	22	34	42	K6	65	50	I/O
GND	GND	GND	GND	GND	23	35	43	J6	66	51	GND
					26	43	53	L11	76	61	XTL2 - I/O
$\overline{\text{RESET}}$ (I)	$\overline{\text{RESET}}$ (I)	$\overline{\text{RESET}}$ (I)	$\overline{\text{RESET}}$ (I)	$\overline{\text{RESET}}$ (I)	27	44	54	K10	78	63	$\overline{\text{RESET}}$ (I)
DONE	DONE	DONE	DONE	DONE	28	45	55	J10	80	65	$\overline{\text{PROG}}$ (I)
		DATA 7 (I)	DATA 7 (I)	DATA 7 (I)		46	56	K11	81	66	I/O
					30	47	57	J11	82	67	XTL1 -I/O
		DATA 6 (I)	DATA 6 (I)	DATA 6 (I)		48	58	H10	83	68	I/O
		DATA 5 (I)	DATA 5 (I)	DATA 5 (I)		49	60	F10	87	72	I/O
		$\overline{\text{CS0}}$ (I)				50	61	G10	88	73	I/O
		DATA 4 (I)	DATA 4 (I)	DATA 4 (I)		51	62	G11	89	74	I/O
VCC	VCC	VCC	VCC	VCC	34	52	64	F9	91	76	VCC
		DATA 3 (I)	DATA 3 (I)	DATA 3 (I)		53	65	F11	92	77	I/O
		$\overline{\text{CS1}}$ (I)				54	66	E11	93	78	I/O
		DATA 2 (I)	DATA 2 (I)	DATA 2 (I)		55	67	E10	94	79	I/O
		DATA 1 (I)	DATA 1 (I)	DATA 1 (I)		56	70	D10	98	83	I/O
		RDY/ $\overline{\text{BUSY}}$	$\overline{\text{RCLK}}$	$\overline{\text{RCLK}}$		57	71	C11	99	84	I/O
DIN (I)	DIN (I)	DATA 0 (I)	DATA 0 (I)	DATA 0 (I)	38	58	72	B11	100	85	I/O
DOUT (I)	DOUT (I)	DOUT (I)	DOUT (I)	DOUT (I)	39	59	73	C10	1	86	I/O
CCLK (I)	CCLK	CCLK	CCLK	CCLK	40	60	74	A11	2	87	CCLK (I)
		$\overline{\text{WS}}$ (I)	A0	A0		61	75	B10	5	90	I/O
		CS2 (I)	A1	A1		62	76	B9	6	91	I/O
			A2	A2		63	77	A10	8	93	I/O
			A3	A3		64	78	A9	9	94	I/O
			A15	A15		65	81	B6	12	97	I/O
			A4	A4		66	82	B7	13	98	I/O
			A14	A14		67	83	A7	14	99	I/O
			A5	A5		68	84	C7	15	100	I/O
GND	GND	GND	GND	GND	1	1	1	C6	16	1	GND
			A13	A13		2	2	A6	17	2	I/O
			A6	A6		3	3	A5	18	3	I/O
			A12	A12		4	4	B5	19	4	I/O
			A7	A7		5	5	C5	20	5	I/O
			A11	A11		6	8	A3	23	8	I/O
			A8	A8		7	9	A2	24	9	I/O
			A10	A10		8	10	B3	25	10	I/O
			A9	A9		9	11	A1	26	11	I/O
						X	X	X	X	X	ATT3020
					X	X	X	X	X		ATT3030
							X	X	X	X	ATT3042
							X**				ATT3064
							X**				ATT3090

▒ REPRESENTS A 50 kΩ TO 100 kΩ PULL-UP.

* INIT IS AN OPEN DRAIN OUTPUT DURING CONFIGURATION.

(I) REPRESENTS AN INPUT

** PIN ASSIGNMENTS FOR THE ATT3064/ATT3090 DIFFER FROM THOSE SHOWN, SEE PAGE 35.

*** PERIPHERAL MODE AND MASTER PARALLEL MODE ARE NOT SUPPORTED IN THE 44 PLCC PACKAGE, SEE PAGE 33.

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

Table 2a. ATT3000 Family Configuration Pin Assignments (continued)

Configuration Mode: <M2:M1:M0>					100 SQFP	132 PGA	160 PQFP	164 CQFP	175 PGA	208 QFP	User Operation
Slave <1:1:1>	Master-SER <0:0:0>	Peripheral <1:0:1>	Master-High <1:1:0>	Master-Low <1:0:0>							
PWR DWN (I)	PWR DWN (I)	PWR DWN (I)	PWR DWN (I)	PWR DWN (I)	26	A1	159	20	B2	3	PWR DWN (I)
VCC	VCC	VCC	VCC	VCC	38	C8	20	42	D9	26	VCC
M1 (HIGH) (I)	M1 (LOW) (I)	M1 (LOW) (I)	M1 (HIGH) (I)	M1 (LOW) (I)	49	B13	40	62	B14	48	RDATA
M0 (HIGH) (I)	M0 (LOW) (I)	M0 (HIGH) (I)	M0 (LOW) (I)	M0 (LOW) (I)	51	A14	42	64	B15	50	RTRIG (I)
M2 (HIGH) (I)	M2 (LOW) (I)	M2 (HIGH) (I)	M2 (HIGH) (I)	M2 (HIGH) (I)	53	C13	44	66	C15	56	I/O
HDC(HIGH)	HDC(HIGH)	HDC(HIGH)	HDC(HIGH)	HDC(HIGH)	54	B14	45	67	E14	57	I/O
LDC (LOW)	LDC (LOW)	LDC (LOW)	LDC (LOW)	LDC (LOW)	56	D14	49	71	D16	61	I/O
INIT *	INIT *	INIT *	INIT *	INIT *	62	G14	59	81	H15	77	I/O
GND	GND	GND	GND	GND	63	H12	19	83	J14	25	GND
					73	M13	76	99	P15	100	XTL2 - I/O
RESET (I)	RESET (I)	RESET (I)	RESET (I)	RESET (I)	75	P14	78	101	R15	102	RESET (I)
DONE	DONE	DONE	DONE	DONE	77	N13	80	103	R14	107	PROG (I)
		DATA 7 (I)	DATA 7 (I)	DATA 7 (I)	78	M12	81	104	N13	109	I/O
					79	P13	82	105	T14	110	XTL1 - I/O
		DATA 6 (I)	DATA 6 (I)	DATA 6 (I)	80	N11	86	109	P12	115	I/O
		DATA 5 (I)	DATA 5 (I)	DATA 5 (I)	84	M9	92	115	T11	122	I/O
		CS0 (I)			85	N9	93	116	R10	123	I/O
		DATA 4 (I)	DATA 4 (I)	DATA 4 (I)	86	N8	98	121	R9	128	I/O
VCC	VCC	VCC	VCC	VCC	88	M8	100	123	N9	130	VCC
		DATA 3 (I)	DATA 3 (I)	DATA 3 (I)	89	N7	102	125	P8	132	I/O
		CS1 (I)			90	P6	103	126	R8	133	I/O
		DATA 2 (I)	DATA 2 (I)	DATA 2 (I)	91	M6	108	131	R7	138	I/O
		DATA 1 (I)	DATA 1 (I)	DATA 1 (I)	95	M5	114	137	R5	145	I/O
		RDY/ BUSY	RCLK	RCLK	96	N4	115	138	P5	146	I/O
DIN (I)	DIN (I)	DATA 0 (I)	DATA 0 (I)	DATA 0 (I)	97	N2	119	143	R3	151	I/O
DOUT (I)	DOUT (I)	DOUT (I)	DOUT (I)	DOUT (I)	98	M3	120	144	N4	152	I/O
CCLK (I)	CCLK	CCLK	CCLK	CCLK	99	P1	121	145	R2	153	CCLK (I)
		WS (I)	A0	A0	2	M2	124	148	P2	161	I/O
		CS2 (I)	A1	A1	3	N1	125	149	M3	162	I/O
			A2	A2	5	L2	128	152	P1	165	I/O
			A3	A3	6	L1	129	153	N1	166	I/O
			A15	A15	9	K1	132	156	M1	172	I/O
			A4	A4	10	J2	133	157	L2	173	I/O
			A14	A14	11	H1	136	160	K2	178	I/O
			A5	A5	12	H2	137	161	K1	179	I/O
GND	GND	GND	GND	GND	13	H3	139	164	J3	182	GND
			A13	A13	14	G2	141	2	H2	184	I/O
			A6	A6	15	G1	142	3	H1	185	I/O
			A12	A12	16	F2	147	8	F2	192	I/O
			A7	A7	17	E1	148	9	E1	193	I/O
			A11	A11	20	D1	151	12	D1	199	I/O
			A8	A8	21	D2	152	13	C1	200	I/O
			A10	A10	22	B1	155	16	E3	203	I/O
			A9	A9	23	C2	156	17	C2	204	I/O
											ATT3020
					X						ATT3030
					X	X					ATT3042
						X	X				ATT3064
							X	X	X	X	ATT3090

▒ REPRESENTS A 50 kΩ TO 100 kΩ PULL-UP.
* INIT IS AN OPEN DRAIN OUTPUT DURING CONFIGURATION.
(I) REPRESENTS AN INPUT

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

Table 2b. ATT3000 44-Pin PLCC Pinout

44 PLCC	ATT3030
1	GND
2	I/O
3	I/O
4	I/O
5	I/O
6	I/O
7	$\overline{\text{PWRDWN}}$
8	TCLKIN-I/O
9	I/O
10	I/O
11	I/O
12	Vcc
13	I/O
14	I/O
15	I/O
16	M1- $\overline{\text{RDATA}}$
17	M0-RTRIG
18	M2-I/O
19	HDC-I/O
20	$\overline{\text{LDC}}$ -I/O
21	I/O
22	$\overline{\text{INIT}}$ -I/O
23	GND
24	I/O
25	I/O
26	XTL2(IN)-I/O
27	$\overline{\text{RESET}}$
28	DONE- $\overline{\text{PROG}}$
29	I/O
30	XTL1(OUT)-BCLKIN
31	I/O
32	I/O
33	I/O
34	Vcc
35	I/O
36	I/O
37	I/O
38	DIN-I/O
39	DOUT-I/O
40	CCLK
41	I/O
42	I/O
43	I/O
44	I/O

Notes:

Peripheral Mode and Master Parallel Mode are not supported in the M44 Package.
Parallel address and data pins are not assigned.

Table 2c. ATT3000 Family 68-Pin PLCC, 84-Pin PLCC Pinouts, and 84-Pin PGA Pinouts

68 PLCC	ATT3020 ATT3030, ATT3042	84 PLCC	84 PGA
10	$\overline{\text{PWRDN}}$	12	B2
11	TCLKIN-I/O	13	C2
—	**I/O***	14	B1
12	I/O	15	C1
13	I/O	16	D2
—	I/O	17	D1
14	I/O	18	E3
15	I/O	19	E2
16	I/O	20	E1
17	I/O	21	F2
18	**VCC**	22	F3
19	I/O	23	G3
—	I/O	24	G1
20	I/O	25	G2
21	I/O	26	F1
22	I/O	27	H1
—	I/O	28	H2
23	I/O	29	J1
24	I/O	30	K1
25	M1- $\overline{\text{RDATA}}$	31	J2
26	M0-RTRIG	32	L1
27	M2-I/O	33	K2
28	HDC-I/O	34	K3
29	I/O	35	L2
30	$\overline{\text{LDC}}$ -I/O	36	L3
31	I/O	37	K4
—	**I/O***	38	L4
32	I/O	39	J5
33	I/O	40	K5
—	**I/O***	41	L5
34	$\overline{\text{INIT}}$ -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
—	**I/O***	50	L9
—	**I/O***	51	L10
42	I/O	52	K9
43	XTL2(IN)-I/O	53	L11

68 PLCC	ATT3020 ATT3030, ATT3042	84 PLCC	84 PGA
44	$\overline{\text{RESET}}$	54	K10
45	DONE- $\overline{\text{PROG}}$	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
—	**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
—	**I/O***	79	B8
—	**I/O***	80	A8
65	A15-I/O	81	B6
66	A4-I/O	82	B7
67	A14-I/O	83	A7
68	A5-I/O	84	C7
1	**GND**	1	C6
2	A13-I/O	2	A6
3	A6-I/O	3	A5
4	A12-I/O	4	B5
5	A7-I/O	5	C5
—	**I/O***	6	A4
—	**I/O***	7	B4
6	A11-I/O	8	A3
7	A8-I/O	9	A2
8	A10-I/O	10	B3
9	A9-I/O	11	A1

* Indicates unconnected package pins for the ATT3020.

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

Table 2d. ATT3064 /ATT3090 84-Pin PLCC Pinouts

PLCC Pin Number	ATT3064 ATT3090
12	PWRDN
13	TCLKIN-I/O
14	I/O
15	I/O
16	I/O
17	I/O
18	I/O
19	I/O
20	I/O
21	**GND***
22	**Vcc**
23	I/O
24	I/O
25	I/O
26	I/O
27	I/O
28	I/O
29	I/O
30	I/O
31	M1- RDATA
32	M0-RTRIG
33	M2-I/O
34	HDC-I/O
35	I/O
36	LDC -I/O
37	I/O
38	I/O
39	I/O
40	I/O
41	**INIT/I/O***
42	**Vcc***
43	**GND**
44	I/O
45	I/O
46	I/O
47	I/O
48	I/O
49	I/O
50	I/O
51	I/O
52	I/O
53	XTL2(IN)-I/O

PLCC Pin Number	ATT3064 ATT3090
54	RESET
55	DONE- PROG
56	D7-I/O
57	XTL1(OUT)-BCLKIN-I/O
58	D6-I/O
59	I/O
60	D5-I/O
61	CS0 -I/O
62	D4-I/O
63	I/O
64	**Vcc**
65	**GND***
66	D3 -I/O*
67	CS1-I/O*
68	D2-I/O*
69	I/O
70	D1-I/O
71	RDY/ BUSY - RCLK -I/O
72	D0-DIN-I/O
73	DOUT-I/O
74	CCLK
75	A0- WS -I/O
76	A1-CS2-I/O
77	A2-I/O
78	A3-I/O
76	I/O
80	I/O
81	A15-I/O
82	A4-I/O
83	A14-I/O
84	A5-I/O
1	**GND**
2	**Vcc***
3	A13-I/O*
4	A6-I/O*
5	A12-I/O*
6	A7-I/O*
7	I/O
8	A11-I/O
9	A8-I/O
10	A10-I/O
11	A9-I/O

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

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

Table 2e. ATT3000 Family 100-Pin QFP Pinouts

Pin Number		ATT3020 ATT3030
CQFP	PQFP	ATT3042
1	16	**GND**
2	17	A13-I/O
3	18	A6-I/O
4	19	A12-I/O
5	20	A7-I/O
6*	21*	**I/O**
7*	22*	**I/O**
8	23	A11-I/O
9	24	A8-I/O
10	25	A10-I/O
11	26	A9-I/O
12	27*	**Vcc**
13	28*	**GND**
14	29	$\overline{\text{PWRDN}}$
15	30	TCLKIN-I/O
16*	31*	**I/O**
17*	32*	**I/O**
18*	33*	**I/O**
19	34	I/O
20	35	I/O
21	36	I/O
22	37	I/O
23	38	I/O
24	39	I/O
25	40	I/O
26	41	**Vcc**
27	42	I/O
28	43	I/O
29	44	I/O
30	45	I/O
31	46	I/O
32	47	I/O
33	48	I/O
34	49	I/O

Pin Number		ATT3020 ATT3030
CQFP	PQFP	ATT3042
35*	50*	**I/O**
36*	51*	**I/O**
37	52	M1- $\overline{\text{RD}}$
38	53*	**GND**
39	54	M0-RT
40	55*	**Vcc**
41	56	M2-I/O
42	57	HDC-I/O
43	58	I/O
44	59	$\overline{\text{LDC}}$ -I/O
45*	60*	**I/O**
46*	61*	**I/O**
47	62	I/O
48	63	I/O
49	64	I/O
50	65	$\overline{\text{INIT}}$ -I/O
51	66	**GND**
52	67	I/O
53	68	I/O
54	69	I/O
55	70	I/O
56	71	I/O
57	72	I/O
58	73	I/O
59*	74*	**I/O**
60*	75*	**I/O**
61	76	XTAL2-I/O
62	77*	**GND**
63	78	$\overline{\text{RESET}}$
64	79*	**Vcc**
65	80	DONE- $\overline{\text{PG}}$
66	81	D7-I/O
67	82	XTAL1-BCLKIN-I/O
68	83	D6-I/O

Pin Number		ATT3020 ATT3030
CQFP	PQFP	ATT3042
69*	84*	**I/O**
70*	85*	**I/O**
71	86	I/O
72	87	D5-I/O
73	88	$\overline{\text{CSO}}$ -I/O
74	89	D4-I/O
75	90	I/O
76	91	**Vcc**
77	92	D3-I/O
78	93	$\overline{\text{CS1}}$ -I/O
79	94	D2-I/O
80	95	I/O
81*	96*	**I/O**
82*	97*	**I/O**
83	98	D1-I/O
84	99	$\overline{\text{RCLK}}$ - $\overline{\text{BUSY}}$ /RDY-I/O
85	100	D0-DIN-I/O
86	1	DOUT-I/O
87	2	CCLK
88	3*	**Vcc**
89	4*	**GND**
90	5	A0- $\overline{\text{WS}}$ -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

* This table describes the pinouts of three different chips in two different packages. The third column lists 100 of the 118 pads on the ATT3042 that are connected to the 100 package pins. Two pads, indicated by double asterisks, do not exist on the ATT3030, which has 98 pads; therefore the corresponding pins have no connections. Twenty-six pads, indicated by single or double asterisks, do not exist on the ATT3020, which has 74 pads; therefore, the corresponding pins have no connections. (See Table 2c.)

Note: Unprogrammed IOBs have a default pull-up. This prevents an undefined pad level for unbonded or unused IOBs. Programmed outputs are default slew-limited.
The 3030 is not available in a 100-pin CQFP.

Table 2f. ATT3000 Family 100-Pin SQFP Pinouts

Pin Number SQFP	ATT3020 ATT3030 ATT3042
16	GND
17	A13-I/O
18	A6-I/O
19	A12-I/O
20	A7-I/O
21	I/O
22	I/O
23	A11-I/O
24	A8-I/O
25	A10-I/O
26	A9-I/O
27	Vcc
28	GND
29	PWRDN
30	TCLKIN-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	Vcc
42	I/O
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	M1- RD
53	GND
54	M0-RT
55	Vcc
56	M2-I/O
57	HDC-I/O
58	I/O
59	LDC -I/O
60	I/O
61	I/O
62	I/O
63	I/O
64	I/O
65	INIT -I/O
66	GND
67	I/O
68	I/O
69	I/O
70	I/O
71	I/O
72	I/O
73	I/O
74	I/O
75	I/O
76	XTAL2-I/O
77	GND
78	RESET
79	Vcc
80	DONE- PG
81	D7-I/O
82	XTAL1-BCLKIN-I/O
83	D6-I/O
84	I/O
85	I/O
86	I/O
87	D5-I/O
88	CSO -I/O
89	D4-I/O
90	I/O
91	Vcc
92	D3-I/O
93	CS1 -I/O
94	D2-I/O
95	I/O
96*	I/O
97	I/O
98	D1-I/O
99	RCLK - BUSY /RDY-I/O
100	D0-DIN-I/O
1	DOUT-I/O
2	CCLK
3	Vcc
4	GND
5	A0- WS -I/O
6	A1-CS2-I/O
7	I/O
8	A2-I/O
9	A3-I/O
10	I/O
11	I/O
12	A15-I/O
13	A4-I/O
14	A14-I/O
15	A5-I/O

☐ = Preliminary.

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

Table 2g. ATT3000 Family 132-Pin PGA Pinouts

PGA Pin Number	ATT3042 ATT3064
C4	**GND**
A1	$\overline{\text{PWRDN}}$
C3	TCLKIN-I/O
B2	I/O
B3	I/O
A2	**I/O***
B4	I/O
C5	I/O
A3	**I/O***
A4	I/O
B5	I/O
C6	I/O
A5	I/O
B6	I/O
A6	I/O
B7	I/O
C7	**GND**
C8	**Vcc**
A7	I/O
B8	I/O
A8	I/O
A9	I/O
B9	I/O
C9	I/O
A10	I/O
B10	I/O
A11	**I/O***
C10	I/O
B11	I/O
A12	**I/O***
B12	I/O
A13	**I/O***
C12	I/O

PGA Pin Number	ATT3042 ATT3064
B13	M1- $\overline{\text{RD}}$
C11	**GND**
A14	M0-RT
D12	**Vcc**
C13	M2-I/O
B14	HDC-I/O
C14	I/O
E12	I/O
D13	I/O
D14	$\overline{\text{LDC}}$ -I/O
E13	**I/O***
F12	I/O
E14	I/O
F13	I/O
F14	I/O
G13	I/O
G14	$\overline{\text{INIT}}$ -I/O
G12	**Vcc**
H12	**GND**
H14	I/O
H13	I/O
J14	I/O
J13	I/O
K14	I/O
J12	I/O
K13	I/O
L14	**I/O***
L13	I/O
K12	I/O
M14	I/O
N14	I/O
M13	XTAL2(IN)-I/O
L12	**GND**

PGA Pin Number	ATT3042 ATT3064
P14	$\overline{\text{RESET}}$
M11	**Vcc**
N13	DONE- $\overline{\text{PG}}$
M12	D7-I/O
P13	XTAL1-BCLKIN-I/O
N12	I/O
P12	I/O
N11	D6-I/O
M10	I/O
P11	**I/O***
N10	I/O
P10	I/O
M9	D5-I/O
N9	$\overline{\text{CSO}}$ -I/O
P9	**I/O***
P8	**I/O***
N8	D4-I/O
P7	I/O
M8	**Vcc**
M7	**GND**
N7	D3-I/O
P6	$\overline{\text{CS1}}$ -I/O
N6	**I/O***
P5	**I/O***
M6	D2-I/O
N5	I/O
P4	I/O
P3	I/O
M5	D1-I/O
N4	$\overline{\text{RCLK}}$ - $\overline{\text{BUSY}}$ /RDY-I/O
P2	I/O
N3	I/O
N2	D0-DIN-I/O

PGA Pin Number	ATT3042 ATT3064
M3	DOUT-I/O
P1	CCLK
M4	**Vcc**
L3	**GND**
M2	A0- $\overline{\text{WS}}$ -I/O
N1	A1-CS2-I/O
M1	I/O
K3	I/O
L2	A2-I/O
L1	A3-I/O
K2	I/O
J3	I/O
K1	A15-I/O
J2	A4-I/O
J1	**I/O***
H1	A14-I/O
H2	A5-I/O
H3	**GND**
G3	**Vcc**
G2	A13-I/O
G1	A6-I/O
F1	**I/O***
F2	A12-I/O
E1	A7-I/O
F3	I/O
E2	I/O
D1	A11-I/O
D2	A8-I/O
E3	I/O
C1	I/O
B1	A10-I/O
C2	A9-I/O
D3	**Vcc**

* Indicates unconnected package pins for the ATT3042.

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

Table 2h. ATT3000 Family 160-Pin PQFP Pinout

PQFP Pin No.	ATT3064 ATT3090	PQFP Pin No.	ATT3064 ATT3090	PQFP Pin No.	ATT3064 ATT3090	PQFP Pin No.	ATT3064 ATT3090
1	I/O*	41	GND	81	D7-I/O	121	CCLK
2	I/O*	42	M0-RTRIG	82	XTAL1-BCLKIN-I/O	122	Vcc
3	I/O*	43	VCC	83	I/O*	123	GND
4	I/O	44	M2-I/O	84	I/O	124	A0- $\overline{WS}$ -I/O
5	I/O	45	HDC-I/O	85	I/O	125	A1-CS2-I/O
6	I/O	46	I/O	86	D6-I/O	126	I/O
7	I/O	47	I/O	87	I/O	127	I/O
8	I/O	48	I/O	88	I/O	128	A2-I/O
9	I/O	49	$\overline{LDC}$ -I/O	89	I/O	129	A3-I/O
10	I/O	50	I/O*	90	I/O	130	I/O
11	I/O	51	I/O*	91	I/O	131	I/O
12	I/O	52	I/O	92	D5-I/O	132	A15-I/O
13	I/O	53	I/O	93	$\overline{CS0}$ -I/O	133	A4-I/O
14	I/O	54	I/O	94	I/O*	134	I/O
15	I/O	55	I/O	95	I/O*	135	I/O
16	I/O	56	I/O	96	I/O	136	A14-I/O
17	I/O	57	I/O	97	I/O	137	A5-I/O
18	I/O	58	I/O	98	D4-I/O	138	I/O*
19	GND	59	$\overline{INIT}$ -I/O	99	I/O	139	GND
20	Vcc	60	Vcc	100	Vcc	140	Vcc
21	I/O*	61	GND	101	GND	141	A13-I/O
22	I/O	62	I/O	102	D3-I/O	142	A6-I/O
23	I/O	63	I/O	103	$\overline{CS1}$ -I/O	143	I/O*
24	I/O	64	I/O	104	I/O	144	I/O*
25	I/O	65	I/O	105	I/O	145	I/O
26	I/O	66	I/O	106	I/O*	146	I/O
27	I/O	67	I/O	107	I/O*	147	A12-I/O
28	I/O	68	I/O	108	D2-I/O	148	A7-I/O
29	I/O	69	I/O	109	I/O	149	I/O
30	I/O	70	I/O	110	I/O	150	I/O
31	I/O	71	I/O	111	I/O	151	A11-I/O
32	I/O	72	I/O	112	I/O	152	A8-I/O
33	I/O	73	I/O	113	I/O	153	I/O
34	I/O	74	I/O	114	D1-I/O	154	I/O
35	I/O	75	I/O*	115	RDY- $\overline{BSY}$ / $\overline{RCLK}$ -I/O	155	A10-I/O
36	I/O	76	XTAL2-I/O	116	I/O	156	A9-I/O
37	I/O	77	GND	117	I/O	157	Vcc
38	I/O*	78	$\overline{RESET}$	118	I/O*	158	GND
39	I/O*	79	Vcc	119	D0-DIN-I/O	159	$\overline{PWRDWN}$
40	M1- $\overline{RDATA}$	80	$\overline{DONE}$ -PROG	120	DOUT-I/O	160	TCLKIN-I/O

* Not connected on ATT3064.

Note: Unprogrammed IOBs have a default pull-up. This prevents an undefined pad level for unbonded or unused IOBs. Programmed outputs are default slew-limited. DEVICE POWER MUST BE LESS THAN 1 WATT FOR COMMERCIAL TEMPERATURE RANGE, AND LESS THAN 0.8 WATTS FOR INDUSTRIAL TEMPERATURE RANGE.

Table 2i. ATT3000 Family 164-Pin CQFP Pinouts

CQFP Pin No.	ATT3090
20	PWRDWN
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
61	I/O
62	M1- RDATA
63	GND
64	M0-RTRIG
65	Vcc
66	M2-I/O
67	HDC-I/O
68	I/O
69	I/O
70	I/O
71	LDC -I/O
72	I/O
73	I/O
74	I/O
75	I/O
76	I/O
77	I/O
78	I/O
79	I/O
80	I/O
81	INIT -I/O
82	Vcc
83	GND
84	I/O
85	I/O
86	I/O
87	I/O
88	I/O
89	I/O
90	I/O
91	I/O
92	I/O
93	I/O
94	I/O
95	I/O
96	I/O
97	I/O
98	I/O
99	XTAL2(IN)-I/O
100	GND
101	RESET
102	Vcc
103	DONE- PROG
104	D7-I/O
105	XTAL1(OUT)-BCLKIN-I/O
106	I/O
107	I/O
108	I/O
109	D6-I/O
110	I/O
111	I/O
112	I/O
113	I/O
114	I/O
115	D5-I/O
116	CS0 -I/O
117	I/O
118	I/O
119	I/O
120	I/O
121	D4-I/O
122	I/O
123	Vcc
124	GND
125	D3-I/O
126	CS1 -I/O
127	I/O
128	I/O
129	I/O
130	I/O
131	D2-I/O
132	I/O
133	I/O
134	I/O
135	I/O
136	I/O
137	D1-I/O
138	RDY- BUSY - RCLK -I/O
139	I/O
140	I/O
141	I/O
142	I/O
143	D0-DIN-I/O
144	DOUT-I/O
145	CCLK
146	Vcc
147	GND
148	A0- WS -I/O
149	A1-CS2-I/O
150	I/O
151	I/O
152	A2-I/O
153	A3-I/O
154	I/O
155	I/O
156	A15-I/O
157	A4-I/O
158	I/O
159	I/O
160	A14-I/O
161	A5-I/O
162	I/O
163	I/O
164	GND
1	Vcc
2	A13-I/O
3	A6-I/O
4	I/O
5	I/O
6	I/O
7	I/O
8	A12-I/O
9	A7-I/O
10	I/O
11	I/O
12	A11-I/O
13	A8-I/O
14	I/O
15	I/O
16	A10-I/O
17	A9-I/O
18	Vcc
19	GND

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

Table 2j. ATT3000 Family 175-Pin PGA Pinouts

PGA Pin No.	ATT3090
B2	PWRDWN
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 No.	ATT3090
D13	I/O
B14	M1- RDATA
C14	GND
B15	M0-RTRIG
D14	Vcc
C15	M2-I/O
E14	HDC-I/O
B16	I/O
D15	I/O
C16	I/O
D16	LDC -I/O
F14	I/O
E15	I/O
E16	I/O
F15	I/O
F16	I/O
G14	I/O
G15	I/O
G16	I/O
H16	I/O
H15	INIT -I/O
H14	Vcc
J14	GND
J15	I/O
J16	I/O
K16	I/O
K15	I/O
K14	I/O
L16	I/O
L15	I/O
M16	I/O
M15	I/O
L14	I/O
N16	I/O
P16	I/O
N15	I/O
R16	I/O
M14	I/O
P15	XTAL2(IN)-I/O
N14	GND
R15	RESET
P14	Vcc

PGA Pin No.	ATT3090
R14	DONE- PROG
N13	D7-I/O
T14	XTAL1(OUT)-BCLKIN-I/O
P13	I/O
R13	I/O
T13	I/O
N12	I/O
P12	D6-I/O
R12	I/O
T12	I/O
P11	I/O
N11	I/O
R11	I/O
T11	D5-I/O
R10	CS0 -I/O
P10	I/O
N10	I/O
T10	I/O
T9	I/O
R9	D4-I/O
P9	I/O
N9	Vcc
N8	GND
P8	D3-I/O
R8	CS1 -I/O
T8	I/O
T7	I/O
N7	I/O
P7	I/O
R7	D2-I/O
T6	I/O
R6	I/O
N6	I/O
P6	I/O
T5	I/O
R5	D1-I/O
P5	RDY/ BUSY - RCLK -I/O
N5	I/O
T4	I/O
R4	I/O
P4	I/O

PGA Pin No.	ATT3090
R3	D0-DIN-I/O
N4	DOUT-I/O
R2	CCLK
P3	Vcc
N3	GND
P2	A0- WS -I/O
M3	A1-CS2-I/O
R1	I/O
N2	I/O
P1	A2-I/O
N1	A3-I/O
L3	I/O
M2	I/O
M1	A15-I/O
L2	A4-I/O
L1	I/O
K3	I/O
K2	A14-I/O
K1	A5-I/O
J1	I/O
J2	I/O
J3	GND
H3	Vcc
H2	A13-I/O
H1	A6-I/O
G1	I/O
G2	I/O
G3	I/O
F1	I/O
F2	A12-I/O
E1	A7-I/O
E2	I/O
F3	I/O
D1	A11-I/O
C1	A8-I/O
D2	I/O
B1	I/O
E3	A10-I/O
C2	A9-I/O
D3	Vcc
C3	GND

Notes: 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.

Table 2k. ATT3000 Family 208-Pin QFP Pinouts

QFP Pin No.	ATT3090	QFP Pin No.	ATT3090	QFP Pin No.	ATT3090	QFP Pin No.	ATT3090
1		53		105		157	
2	GND	54		106	Vcc	158	
3	POWERDOWN	55	Vcc	107	DONE- PG	159	
4	TCLKIN-I/O	56	M2-I/O	108		160	GND
5	I/O	57	HDC-I/O	109	I/O-D7	161	I/O- WS -A0
6	I/O	58	I/O	110	I/O-XTAL-BCLKIN	162	I/O-CS2-A1
7	I/O	59	I/O	111	I/O	163	I/O
8	I/O	60	I/O	112	I/O	164	I/O
9	I/O	61	LDC -I/O	113	I/O	165	I/O-A2
10	I/O	62	I/O	114	I/O	166	I/O-A3
11	I/O	63	I/O	115	I/O-D6	167	I/O
12	I/O	64		116	I/O	168	I/O
13	I/O	65		117	I/O	169	
14	I/O	66		118	I/O	170	
15		67		119		171	
16	I/O	68	I/O	120	I/O	172	I/O-A15
17	I/O	69	I/O	121	I/O	173	I/O-A4
18	I/O	70	I/O	122	I/O-D5	174	I/O
19	I/O	71	I/O	123	I/O- CS0	175	I/O
20	I/O	72		124	I/O	176	
21	I/O	73		125	I/O	177	
22	I/O	74	I/O	126	I/O	178	I/O-A14
23	I/O	75	I/O	127	I/O	179	I/O-A5
24	I/O	76	I/O	128	I/O-D4	180	I/O
25	GND	77	INIT -I/O	129	I/O	181	I/O
26	Vcc	78	Vcc	130	Vcc	182	GND
27	I/O	79	GND	131	GND	183	Vcc
28	I/O	80	I/O	132	I/O-D3	184	I/O-A13
29	I/O	81	I/O	133	I/O- CS1	185	I/O-A6
30	I/O	82	I/O	134	I/O	186	I/O
31	I/O	83		135	I/O	187	I/O
32	I/O	84		136	I/O	188	
33	I/O	85	I/O	137	I/O	189	
34	I/O	86	I/O	138	I/O-D2	190	I/O
35	I/O	87	I/O	139	I/O	191	I/O
36	I/O	88	I/O	140	I/O	192	I/O-A12
37		89	I/O	141	I/O	193	I/O-A7
38	I/O	90		142		194	
39	I/O	91		143	I/O	195	
40	I/O	92		144	I/O	196	
41	I/O	93	I/O	145	I/O-D1	197	I/O
42	I/O	94	I/O	146	I/O- BUSY /RDY- RCLK	198	I/O
43	I/O	95	I/O	147	I/O	199	I/O-A11
44	I/O	96	I/O	148	I/O	200	I/O-A8
45	I/O	97	I/O	149	I/O	201	I/O
46	I/O	98	I/O	150	I/O	202	I/O
47	I/O	99	I/O	151	I/O-DIN-D0	203	I/O-A10
48	M1- RDATA	100	XTAL2-I/O	152	I/O-DOUT	204	I/O-A9
49	GND	101	GND	153	CCLK	205	Vcc
50	M0-RTRIG	102	RESET	154	Vcc	206	
51		103		155		207	
52		104		156		208	

☐ = Preliminary.

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

PARAMETRICS

Absolute Maximum Ratings

Stresses in excess of the Absolute Maximum Ratings can cause permanent damage to the device. These are absolute stress ratings only. Functional operation of the device is not implied at these or any other conditions in excess of those given in the operational sections of the data sheet. Exposure to Absolute Maximum Ratings for extended periods can adversely affect device reliability.

Symbol	Parameter	Limit	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

Recommended Operating Conditions

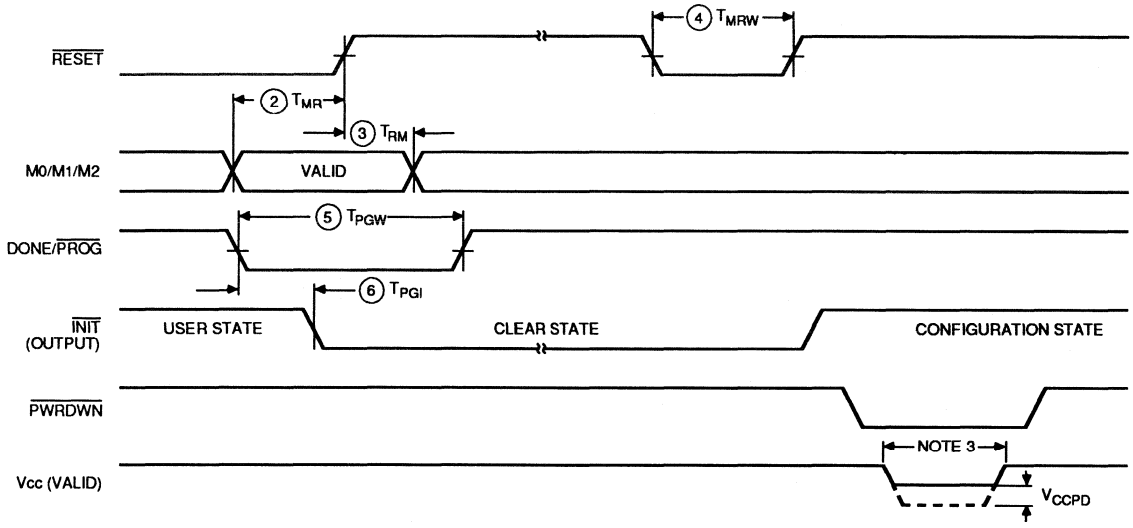
Symbol	Parameter		Min	Max	Units
V_{CC}	Supply voltage relative to GND	Commercial	4.75	5.25	V
		Industrial	4.5	5.5	V
V_{IHT}	High-level input voltage — TTL configuration		2.0	V_{CC}	V
V_{ILT}	Low-level input voltage — TTL configuration		0	0.8	V
V_{IHC}	High-level input voltage — CMOS configuration		70%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

Electrical Characteristics Over Operating Conditions

Symbol	Parameter/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_{OL} = –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_{CCPD}	Power-down supply voltage ($\overline{\text{PWRDWN}}$ must be Low)		2.3		V
I_{CCPD}	Power-down supply current (V_{CC} = 5.0 V @ 70 °C)	ATT3020		50	µA
		ATT3030		80	µA
		ATT3042		120	µA
		ATT3064		170	µA
		ATT3090		250	µA
I_{CCO}	Quiescent FPGA supply current in addition to I_{CCPD}				
	Chip thresholds programmed as CMOS levels			500	µA
	Chip thresholds programmed as TTL levels			10	mA
I_{IL}	Leakage Current		–10	+10	µA
C_{IN}	Input capacitance, all packages except PGA 175 (sample tested)				
	All Pins except XTL1 and XTL2			10	pf
	XTL1 and XTL2			15	pf
	Input capacitance, PGA 175 (sample tested)				
	All Pins except XTL1 and XTL2			15	pf
	XTL1 and XTL2			20	pf
I_{RIN}	Pad pull-up (when selected) @ V_{IN} = 0 V (sample tested)		0.02	0.17	mA
I_{RLL}	Horizontal long line pull-up (when selected) @ logic LOW		0.2	2.5	mA

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

GENERAL FPGA SWITCHING CHARACTERISTICS



Signal	Description	Speed (4) Symbol		–70 Min	–70 Max	–100 Min	–100 Max	–125 Min	–125 Max	150 Min	150 Max	Unit
RESET (2)	M0, M1, M2 setup time required	2	TMR	1		1		0		0		µs
	M0, M1, M2 hold time required	3	TRM	1		1		1		1		µs
	RESET Width (Low) req. for Abort	4	TMRW	6		6		6		6		µs
DONE/PROG	Width (Low0 required for Re-config.	5	TPGW	6		6		6		6		µs
	INIT response after D/P is pulled Low	6	TPGI		7		7		7		7	µs
PWRDWN (3)	Powerdown VCC		VCCPD	2.3		2.3		2.3		2.3		V

▒ = Preliminary.

Notes:
1. At power-up, Vcc must rise from 2.0 V to Vcc minimum in less than 25 ms. If this is not possible, configuration can be delayed by holding RESET Low until Vcc has reached 4.0 V. A very long Vcc rise time of >100 ms, or a non-monotonically rising Vcc may require a >1 µs High level on RESET, followed by a >6 µs Low level on RESET and D/P after Vcc has reached 4.0 V.
2. RESET timing relative to valid mode lines (M0, M1, M2) is relevant when RESET is used to delay configuration.
3. PWRDWN transitions must occur while Vcc > 4.0 V.

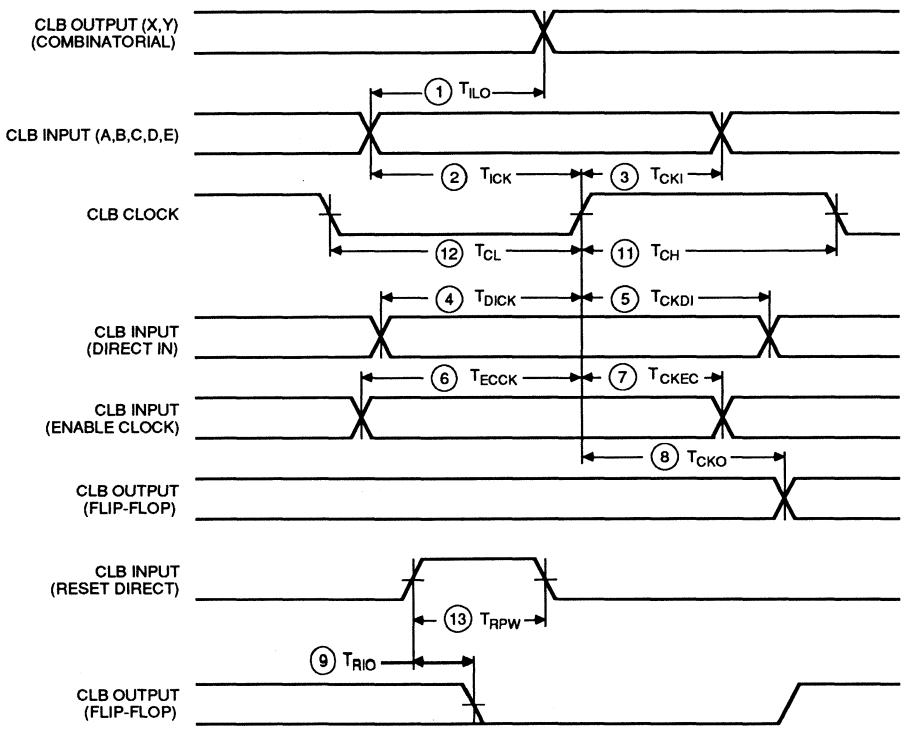
BUFFER (Internal) SWITCHING CHARACTERISTIC GUIDELINES

	Speed Grade	–70	–100	–125	150	Units
Description	Symbol	Max	Max	Max	Max	
Global and Alternate Clock Distribution*						
Either: **Normal** IOB input pad to clock buffer input	TPID	8	7.5	7	6.7	ns
Or: **Fast** (CMOS only) input pad to clock buffer input	TPIDC	6.5	6	5.7	5.5	ns
TBUF driving a Horizontal Long line (L.L.)*						
I to L.L. while T is Low (buffer active)	TIO	5	4.7	4.5	4.3	ns
T↓ to L.L. active and valid with single pull-up resistor	TON	11	10	9	8	ns
T↓ to L.L. active and valid with pair of pull-up resistors	TON	12	11	10	9	ns
T↑ to L.L. High with single pull-up resistor	TPUS	24	22	17	16	ns
T↑ to L.L. High with pair of pull-up resistors	TPUF	17	15	12	11	ns
BIDI						
Bidirectional buffer delay	TBIDI	2	1.8	1.7	1.6	ns

* Timing is based on the ATT3042; for other devices, see *XACT* timing calculator.

▒ = Preliminary.

CLB SWITCHING CHARACTERISTIC GUIDELINES



2271 27

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.

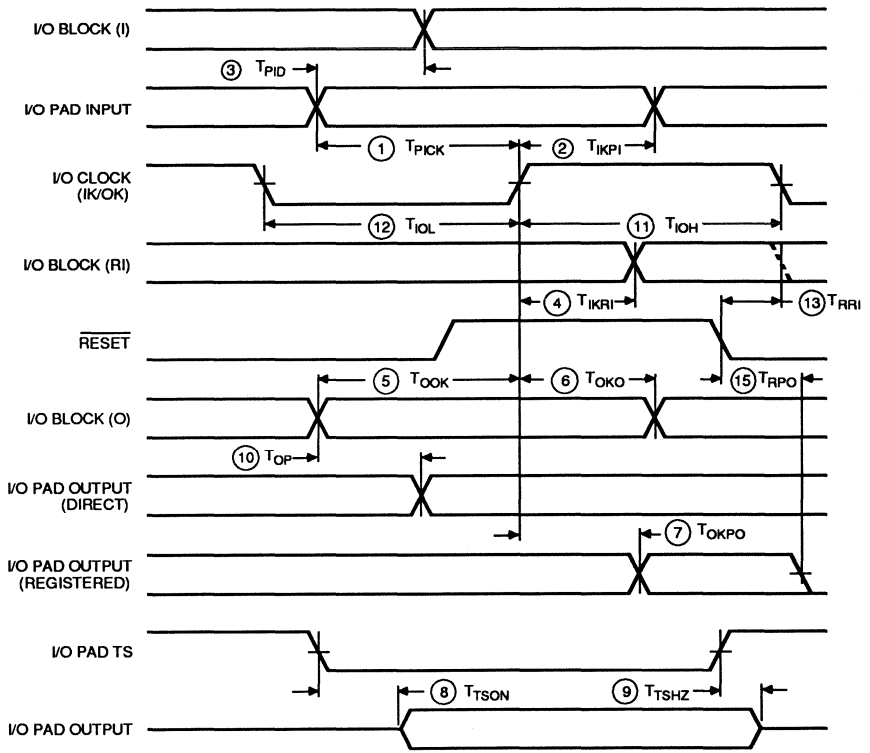
	Speed Grade		-70		-100		-125		150		Units
Description	Symbol		Min	Max	Min	Max	Min	Max	Min	Max	
Combinatorial Delay											
Logic Variables a, b, c, d, e, to outputs x, y	1	TILO		9		7		5.5		4.6	ns
Sequential delay											
clock k to outputs x, y	8	TCKO		6		5		4.5		4	ns
clock k outputs x, y when Q is returned through function generators F or G to drive x, y		TCKOQ		13		10		8		6.7	ns
Set-up time before clock K											
Logic Variables a, b, c, d, e	2	TICK	8		7		5.5		4.6		ns
Data In di	4	TDICK	5		4		3		2		ns
Enable Clock ec	6	TECCK	7		5		4.5		4		ns
Reset Direct inactive rd		TRDCK	1		1		1		1		ns
Hold Time after clock k											
Logic Variables a, b, c, d, e	3	TCKI	0		0		0		0		ns
Data In di	5	TCKDI	4		2		1.5		1.2		ns
Enable Clock ec	7	TCKEC	0		0		0		0		ns
Clock											
Clock High time*	11	TCH	5		4		3		2.5		ns
Clock Low time*	12	TCL	5		4		3		2.5		ns
Max. flip-flop toggle rate*		FCLK	70		100		125		150		MHz
Reset Direct (rd)											
rd width	13	TRPW	8		7		6		5		ns
delay from rd to outputs x, y	9	TRIO		8		7		6		5	ns
Master Reset (MR)											
MR width		TMRW	25		21		20		19		ns
delay from MR to outputs x, y		TMRQ		23		19		17		17	ns

* Timing is based on the ATT3042, for other devices see the *XACT* timing calculation.

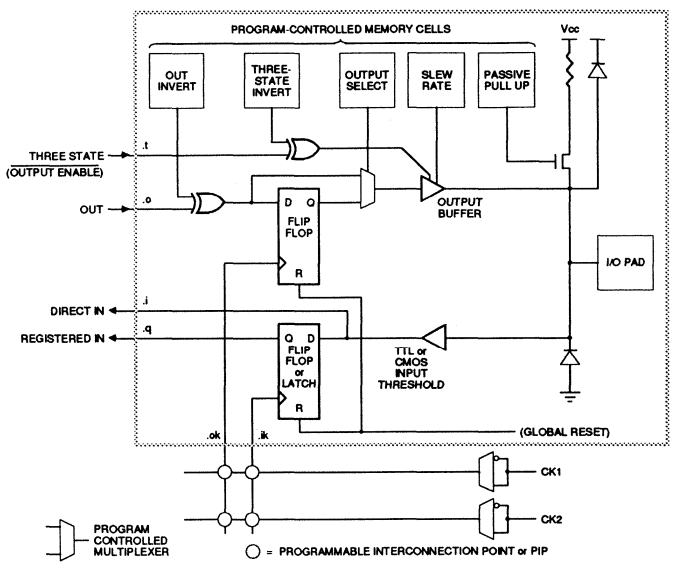
▨ = Preliminary.

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



2271 28



2271 29

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 characteristic guideline values. Actual worst-case timing is provided by the *XACT* Timing calculator or Simulation modeling.

	Speed Grade		–70		–100		–125		150		Units
Description	Symbol		Min	Max	Min	Max	Min	Max	Min	Max	
Propagation Delays (Input)											
Pad to Direct In (i)	3	TPID		6		4		3		2.8	ns
Pad to Registered In (q) with latch transparent		TPTG		21		17		16		15	ns
Clock (ik) to Registered In (q)	4	TIKRI		5.5		4		3		2.8	ns
Set-up Time (Input)											
Pad to Clock (ik) set-up time	1	TPICK	20		17		16		14.5		ns
Propagation Delays (Output)											
Clock (ok) to Pad (fast)	7	TOKPO		13		10		9		7	ns
same (slew-rate limited)	7	TOKPO		33		27		24		22	ns
Output (o) to Pad (fast)	10	TOPF		9		6		5		4.5	ns
same (slew-rate limited)	10	TOPS		29		23		20		15	ns
3-state to Pad begin hi-Z (fast)	9	TTSHZ		8		8		7		7	ns
same (slew-rate limited)	9	TTSHZ		28		25		24		22	ns
3-state to Pad and valid (fast)	8	TTSON		14		12		11		11	ns
same (slew-rate limited)	8	TTSON		34		29		27		26	ns
Set-up and Hold Times (Output)											
Output (o) to clock (ok) set-up time	5	TOCK	10		9		8		7		ns
Output (o) to clock (ok) hold time	6	TOKO	0		0		0		0		ns
Clock											
Clock High time*	11	TCH	5		4		3		2.5		ns
Clock Low time*	12	TCL	5		4		3		2.5		ns
Max. flip-flop toggle rate		FCLK	0		100		125		150		MHz
Master Reset Delays											
$\overline{\text{RESET}}$ Pad to Registered In (q)	13	TRRI		25		24		23		20	ns
$\overline{\text{RESET}}$ Pad to output pad (fast)	15	TRPO		35		33		29		25	ns
(slew-rate limited)	15	TRPO		53		45		42		40	ns

▨ = Preliminary.

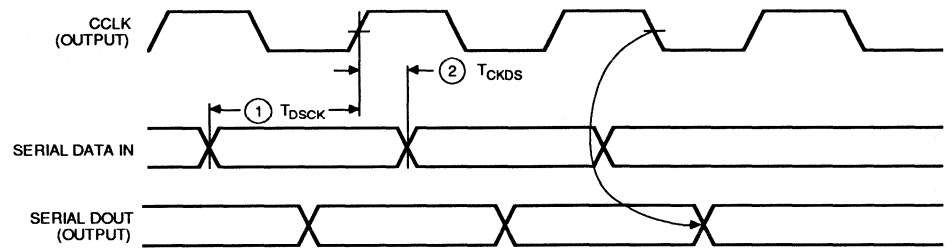
Notes:

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 200 pF per power/ground pin pair. For slew-rate limited outputs this total is 4 times larger.

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.

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.

MASTER SERIAL MODE SWITCHING CHARACTERISTICS GUIDELINES



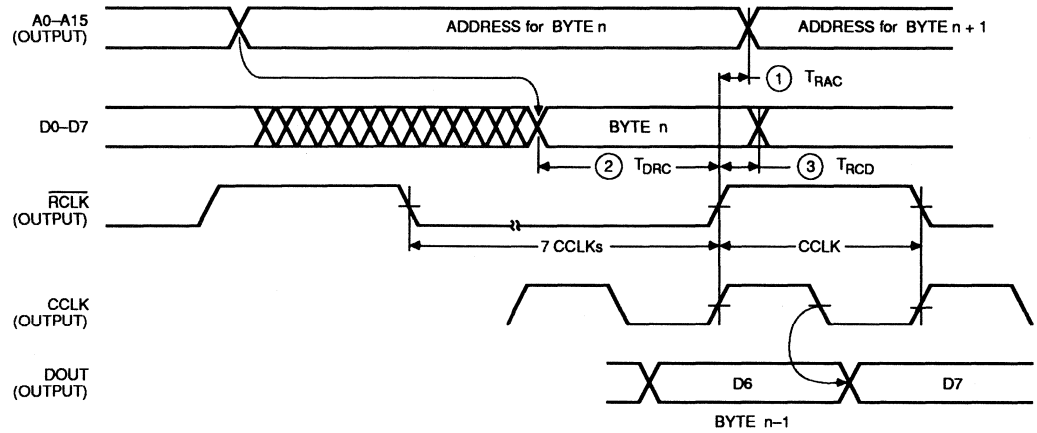
2271 31

		Speed		-70		-100		-125		150		Unit
Signal	Description	Symbol		Min	Max	Min	Max	Min	Max	Min	Max	
CCLK	Data In setup	1	TDSCK	60		60		60		60		ns
	Data in hold	2	TCKDS	0		0		0		0		ns

Notes:
1. At power-up, Vcc must rise from 2.0 V to Vcc minimum in less than 25 ms. If this is not possible, configuration can be delayed by holding RESET Low until Vcc has reached 4.0 V. A very long Vcc rise time of >100 ms, or a non-monotonically rising Vcc may require a >1 µs High level on RESET, followed by a >6 µs Low level on RESET and D/P after Vcc has reached 4.0 V.
2. Configuration can be controlled by holding RESET Low with or until after the 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 characteristic guideline values. Actual worst-case timing is provided by the *XACT* Timing calculator or Simulation modeling.



2271 32

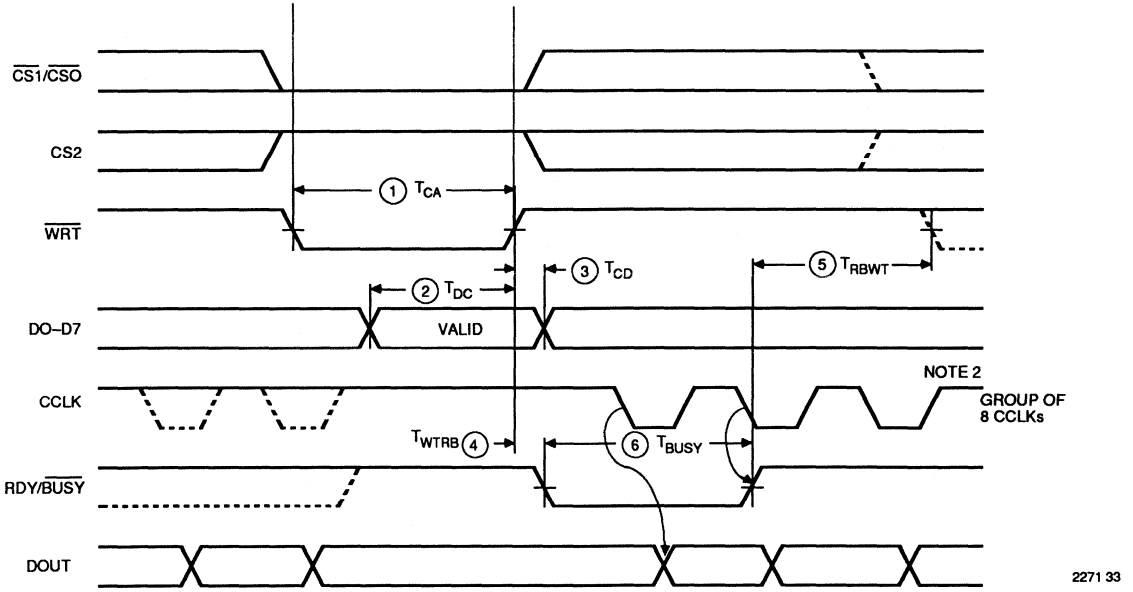
Note: This timing diagram shows that the EPROM requirements are extremely relaxed: EPROM access time can be longer than 4000 ns. EPROM data output has no hold time requirements.

		Speed		–70		–100		–125		150		Unit
Signal	Description	Symbol		Min	Max	Min	Max	Min	Max	Min	Max	
RCLK	To address valid	1	TRAC	0	200	0	200	0	200	0	200	ns
	To data setup	2	TDRC	60		60		60		60		ns
	To data hold	3	TRCD	0		0		0		0		ns
	RCLK high		TRCH	600		600		600		600		ns
	RCLK low		TRCL	4.0		4.0		4.0		4.0		µs

Notes:

1. At power-up, Vcc must rise from 2.0 V to Vcc minimum in less than 25 ms. If this is not possible, configuration can be delayed by holding $\overline{\text{RESET}}$ Low until Vcc has reached 4.0 V. A very long Vcc rise time of >100 ms, or a non-monotonically rising Vcc may require a >1 µs High level on RESET, followed by a >6 µs Low level on $\overline{\text{RESET}}$ and D/$\overline{\text{P}}$ after Vcc has reached 4.0 V.
2. Configuration can be controlled by holding RESET Low with or until after the $\overline{\text{INIT}}$ of all daisy-chain slave-mode devices is High.

PERIPHERAL MODE PROGRAMMING SWITCHING CHARACTERISTICS

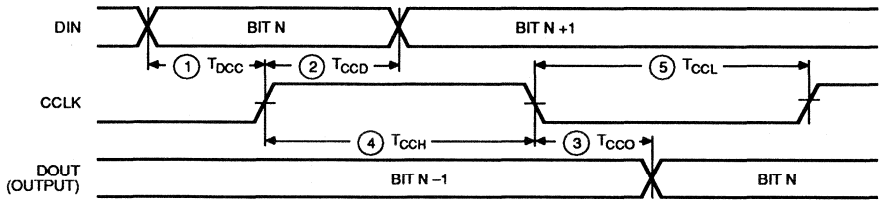


Note: This timing diagram shows very relaxed requirements: Data need not be held beyond the rising edge of $\overline{WS}$. $\overline{BUSY}$ will go active within 60 ns after the end of $\overline{WS}$. $\overline{BUSY}$ will stay active for several microseconds. $\overline{WS}$ may be asserted immediately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

		Speed		–70		–100		–125		150		Unit
Signal	Description	Symbol		Min	Max	Min	Max	Min	Max	Min	Max	
Write	Effective write time required ($\overline{CS0}$ x CS1 x CS2 x $\overline{WS}$	1	TCA	100		100		100		100		ns
	DIN setup time required DIN hold time required	2 3	TDC TCD	60 0		60 0		60 0		60 0		ns ns
	RDY/ $\overline{BUSY}$ delay after end of $\overline{WS}$	4	TWTRB		60		60		60		60	ns
RDY	Earliest next $\overline{WS}$ after end of $\overline{BUSY}$	5	TRBWT	0		0		0		0		ns
	$\overline{BUSY}$ Low time generated	6	TBUSY	2	9	2	9	2	9	2	9	CCLK Periods

Notes:
1. At power-up, Vcc must rise from 2.0 V to Vcc minimum in less than 25 ms. If this is not possible, configuration can be delayed by holding $\overline{RESET}$ Low until Vcc has reached 4.0 V. A very long Vcc rise time of >100 ms, or a non-monotonically rising Vcc may require a >1 µs High level on $\overline{RESET}$, followed by a >6 µs Low level on $\overline{RESET}$ and D/$\overline{P}$ after Vcc has reached 4.0 V.
2. Configuration must be delayed until the $\overline{INIT}$ of all LCAs is High.
3. Time from end of $\overline{WS}$ to CCLK cycle for the new byte of data depends on completion of previous byte processing and the phase of the internal timing generator for CCLK.
4. CCLK and DOUT timing is tested in slave mode.

SLAVE MODE PROGRAMMING SWITCHING CHARACTERISTICS

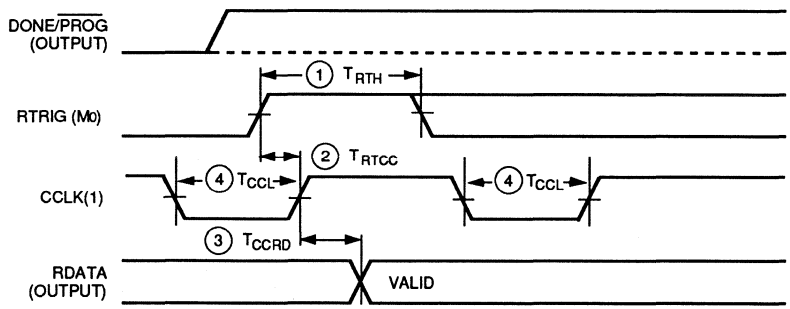


2271 34

		Speed		–70		–100		–125		150		Unit
Signal	Description	Symbol		Min	Max	Min	Max	Min	Max	Min	Max	
CCLK	To DOUT	3	TCCO		100		100		100		100	ns
	DIN setup	1	TDCC	60		60		60		60		ns
	DIN hold	2	TCCD	0		0		0		0		ns
	High time	4	TCCH	0.05		0.05		0.05		0.05		µs
	Low time	5	TCCL	0.05	5.0	0.05	5.0	0.05	5.0	0.05	5.0	µs
	Frequency		FCC		10		10		10		10	MHz

Notes:
1. The max limit of CCLK Low time is caused by dynamic circuitry inside the LCA device.
2. Confiugration must be delayed until the $\overline{INIT}$ of all LCAs is High.
3. At power-up, Vcc must rise from 2.0 V to Vcc minimum in less than 25 ms. If this is not possible, configuration can be delayed by holding $\overline{RESET}$ Low until Vcc has reached 4.0 V. A very long Vcc rise time of >100 ms, or a non-monotonically rising Vcc may require a >1 µs High level on $\overline{RESET}$, followed by a >6 µs Low level on $\overline{RESET}$ and D/$\overline{P}$ after Vcc has reached 4.0 V.

PROGRAM READBACK SWITCHING CHARACTERISTICS



2271 35

		Speed		–70		–100		–125		150		Unit
Signal	Description	Symbol		Min	Max	Min	Max	Min	Max	Min	Max	
RTRIG	RTRIG high	1	TRTH	250		250		250		250		ns
CCLK	$\overline{RTRIG}$ setup	2	TRTCC	200		200		200		200		ns
	RDATA delay	3	TCCRD		100		100		100		100	ns
	High Time	5	TCCH	0.5		0.5		0.5		0.5		µs
	Low time	4	TCCL	0.5	5	0.5	5	0.5	5	0.5	5	µs

Notes:
1. During Readback, CCLK frequency may not exceed 1 MHz.
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.

Advance Data Sheet
June 1992

AT&T Microelectronics

ATT3000 Series Military Field-Programmable Gate Arrays

Features

- User-programmable gate array
- Low-power, CMOS, static memory technology
- Standard product; 100% factory tested
- Total user-controlled design cycle
- Secure design process
- Complete PC or workstation development system:
  - — Schematic capture
  - — Automatic place/route
  - — Logic and timing simulation
  - — Design editor
  - — Logic and timing simulator
  - — *XACTOR* in-circuit verifier

Description

The military version of the CMOS ATT3000 series Field-Programmable Gate Array (FPGA) provides a group of high-performance, high-density, digital, integrated circuits. Their regular, extendable, flexible, user-programmable array architecture consists of three types of configurable elements: I/O blocks, configurable logic blocks, and interconnect. Designers can define individual I/O blocks to interface with external circuitry, select logic blocks to implement logic functions, and set up interconnection networks to compose larger-scale logic functions.

The FPGA's user logic functions and interconnections are determined by the configuration program data stored in internal static memory cells.

The use of on-chip logic allows the configuration data to be automatically loaded at powerup or on command. The program data resides externally in an EEPROM, EPROM, or ROM on the application circuit board or on a floppy disk or hard disk.

The FPGA features several built-in methods to automatically load configuration data and they are determined by logic levels applied to mode-selection pins at configuration times. The form of the data may be either serial or parallel, depending on the configuration mode. The programming data is also dependent on the configuration mode.

The *XACT* development system provides schematic capture and allows designers to define the FPGA logic functions. Logic and timing simulation and in-circuit emulation are available as design verification alternatives. In addition, the *XACT* system is used to compile the data pattern that represents the configuration program. The data can then be converted to a PROM programmer format file to create the configuration program storage. For a full description, refer to the ATT3000 commercial data sheet listed previously in this section.

Table 1. AT&T Military FPGAs

Basic Array	Logic Capacity (usable gates)	Configurable Logic Blocks	User I/Os	Program Data (bits)
ATT3020	2000	64	64	14779
ATT3042	4200	144	96	30784
ATT3090	9000	320	144	64160

Pin Assignments

Configuration Mode: <M2: M1: M0>					84 PGA	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>						
PWRDWN (I)	PWRDWN (I)	PWRDWN (I)	PWRDWN (I)	PWRDWN (I)	B2	14	A1	20	B2	PWRDWN (I)
Vcc	Vcc	Vcc	Vcc	Vcc	F3	26	C8	42	D9	Vcc
M1 (HIGH) (I)	M1 (LOW) (I)	M1 (LOW) (I)	M1 (HIGH) (I)	M1 (LOW) (I)	J2	37	B13	62	B14	RDATA
M0 (HIGH) (I)	M0 (LOW) (I)	M0 (HIGH) (I)	M0 (LOW) (I)	M0 (LOW) (I)	L1	39	A14	64	B15	RTRIG (I)
M2 (HIGH) (I)	M2 (LOW) (I)	M2 (HIGH) (I)	M2 (HIGH) (I)	M2 (HIGH) (I)	K2	41	C13	66	C15	I/O
HDC(HIGH)	HDC(HIGH)	HDC(HIGH)	HDC(HIGH)	HDC(HIGH)	K3	42	B14	67	E14	I/O
LDC (LOW)	LDC (LOW)	LDC (LOW)	LDC (LOW)	LDC (LOW)	L3	44	D14	71	D16	I/O
INIT *	INIT *	INIT *	INIT *	INIT *	K6	50	G14	81	H15	I/O
GND	GND	GND	GND	GND	J6	51	H12	83	J14	GND
					L11	61	M13	99	P15	XTL2 - I/O
RESET (I)	RESET (I)	RESET (I)	RESET (I)	RESET (I)	K10	63	P14	101	R15	RESET (I)
DONE	DONE	DONE	DONE	DONE	J10	65	N13	103	R14	PROG (I)
		DATA 7 (I)	DATA 7 (I)	DATA 7 (I)	K11	66	M12	104	N13	I/O
					J11	67	P13	105	T14	XTL1 -I/O
		DATA 6 (I)	DATA 6 (I)	DATA 6 (I)	H10	68	N11	109	P12	I/O
		DATA 5 (I)	DATA 5 (I)	DATA 5 (I)	F10	72	M9	115	T11	I/O
		CS0 (I)			G10	73	N9	116	R10	I/O
		DATA 4 (I)	DATA 4 (I)	DATA 4 (I)	G11	74	N8	121	R9	I/O
Vcc	Vcc	Vcc	Vcc	Vcc	F9	76	M8	123	N9	Vcc
		DATA 3 (I)	DATA 3 (I)	DATA 3 (I)	F11	77	N7	125	P8	I/O
		CS1 (I)			E11	78	P6	126	R8	I/O
		DATA 2 (I)	DATA 2 (I)	DATA 2 (I)	E10	79	M6	131	R7	I/O
		DATA 1 (I)	DATA 1 (I)	DATA 1 (I)	D10	83	M5	137	R5	I/O
		RDY/ BUSY	RCLK	RCLK	C11	84	N4	138	P5	I/O
DIN (I)	DIN (I)	DATA 0 (I)	DATA 0 (I)	DATA 0 (I)	B11	85	N2	143	R3	I/O
DOUT (I)	DOUT (I)	DOUT (I)	DOUT (I)	DOUT (I)	C10	86	M3	144	N4	I/O
CCLK (I)	CCLK	CCLK	CCLK	CCLK	A11	87	P1	145	R2	CCLK (I)
		WS (I)	A0	A0	B10	90	M2	148	P2	I/O
		CS2 (I)	A1	A1	B9	91	N1	149	M3	I/O
			A2	A2	A10	93	L2	152	P1	I/O
			A3	A3	A9	94	L1	153	N1	I/O
			A15	A15	B6	97	K1	156	M1	I/O
			A4	A4	B7	98	J2	157	L2	I/O
			A14	A14	A7	99	H1	160	K2	I/O
			A5	A5	C7	100	H2	161	K1	I/O
GND	GND	GND	GND	GND	C6	1	H3	164	J3	GND
			A13	A13	A6	2	G2	2	H2	I/O
			A6	A6	A5	3	G1	3	H1	I/O
			A12	A12	B5	4	F2	8	F2	I/O
			A7	A7	C5	5	E1	9	E1	I/O
			A11	A11	A3	8	D1	12	D1	I/O
			A8	A8	A2	9	D2	13	C1	I/O
			A10	A10	B3	10	B1	16	E3	I/O
			A9	A9	A1	11	C2	17	C2	I/O

Notes:

Represents a 50 kΩ to 100 kΩ pull-up.

* INIT is an open drain output during configuration.

(I) Represents an input.

Pin Assignments (continued)

Table 2. ATT3020/3042 Military FPGA, 84-Pin PGA

84 PGA	ATT3020 ATT3042
B2	$\overline{\text{PWRDWN}}$
C2	TCLKIN-I/O
B1	**I/O***
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-RTRIG
K2	M2-I/O
K3	HDC-I/O
L2	I/O
L3	$\overline{\text{LDC}}$ -I/O
K4	I/O
L4	**I/O***
J5	I/O
K5	I/O
L5	**I/O***
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	**I/O***
L10	**I/O***
K9	I/O
L11	XTL2(IN)-I/O

84 PGA	ATT3020 ATT3042
K10	$\overline{\text{RESET}}$
J10	DONE- $\overline{\text{PROG}}$
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	**I/O***
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	**I/O***
A8	**I/O***
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	**I/O***
B4	**I/O***
A3	A11-I/O
A2	A8-I/O
B3	A10-I/O
A1	A9-I/O

* Indicates unconnected package pins for the ATT3020.

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

Pin Assignments (continued)

Table 3. ATT3020 Military FPGA, 100-Pin CQFP

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

Pin No.	Function
35	NC
36	NC
37	M1- $\overline{\text{RD}}$
38	NC
39	M0-RT
40	NC
41	M2-I/O
42	HDC-I/O
43	I/O
44	$\overline{\text{LDC}}$ -I/O
45	NC
46	NC
47	I/O
48	I/O
49	I/O
50	$\overline{\text{INIT}}$ -I/O
51	GND
52	I/O
53	I/O
54	I/O
55	I/O
56	I/O
57	I/O
58	I/O
59	NC
60	NC
61	XTAL2-I/O
62	NC
63	$\overline{\text{RESET}}$
64	NC
65	DONE- $\overline{\text{PG}}$
66	D7-I/O
67	XTAL1-I/O
68	D6-I/O

Pin No.	Function
69	NC
70	NC
71	I/O
72	D5-I/O
73	$\overline{\text{CSO}}$ -I/O
74	D4-I/O
75	I/O
76	Vcc
77	D3-I/O
78	$\overline{\text{CS1}}$ -I/O
79	D2-I/O
80	I/O
81	NC
82	NC
83	D1-I/O
84	$\overline{\text{RCLK}}$ - $\overline{\text{BUSY}}$ /RDY-I/O
85	D0-DIN-I/O
86	DOUT-I/O
87	CCLK
88	NC
89	NC
90	A0- $\overline{\text{WS}}$ -I/O
91	A1-CS2-I/O
92	NC
93	A2-I/O
94	A3-I/O
95	NC
96	NC
97	A15-I/O
98	A4-I/O
99	A14-I/O
100	A5-I/O

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

Pin Assignments (continued)

Table 4. ATT3042 Military FPGA, 100-Pin CQFP Pinouts

CQFP Pin No.	Function
13	**GND**
14	PWRDWN
15	I/O
16	I/O
17	I/O
18	I/O
19	I/O
20	I/O
21	I/O
22	I/O
23	I/O
24	I/O
25	I/O
26	**Vcc**
27	I/O
28	I/O
29	I/O
30	I/O
31	I/O
32	I/O
33	I/O
34	I/O
35	I/O
36	I/O
37	M1- RD
38	**I/O***
39	M1- RT
40	**Vcc**
41	M2-I/O
42	HDC-I/O
43	I/O
44	LDC -I/O
45	I/O
46	I/O

CQFP Pin No.	Function
47	I/O
48	I/O
49	I/O
50	INIT-I/O
51	**GND**
52	I/O
53	I/O
54	I/O
55	I/O
56	I/O
57	I/O
58	I/O
59	I/O
60	I/O
61	XTAL2-I/O
62	**GND**
63	RESET
64	**Vcc**
65	DONE- PG
66	D7-I/O
67	XTAL1-I/O
68	D6-I/O
69	I/O
70	I/O
71	I/O
72	D5-I/O
73	CSO -I/O
74	D4-I/O
75	I/O
76	**Vcc**
77	D3-I/O
78	CS1 -I/O
79	D2-I/O
80	I/O

CQFP Pin No.	Function
81	I/O
82	I/O
83	D1-I/O
84	RCLK - BUSY /RDY-I/O
85	D0-DIN-I/O
86	DOUT-I/O
87	CCLK
88	**Vcc**
89	**GND**
90	A0- WS -I/O
91	A1-CS2-I/O
92	I/O
93	A2-I/O
94	A3-I/O
95	I/O
96	I/O
97	A15-I/O
98	A4-I/O
99	A14-I/O
100	A5-I/O
1	**GND**
2	A13-I/O
3	A6-I/O
4	A12-I/O
5	A7-I/O
6	I/O
7	I/O
8	A11-I/O
9	A8-I/O
10	A10-I/O
11	A9-I/O
12	**Vcc**

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

Pin Assignments (continued)

Table 5. ATT3090 Military FPGA, 164-Pin CQFP Pinouts

CQFP Pin No.	Function
20	PWRDWN
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 No.	Function
61	I/O
62	M1- RDATA
63	GND
64	M0-RTRIG
65	Vcc
66	M2-I/O
67	HDC-I/O
68	I/O
69	I/O
70	I/O
71	LDC -I/O
72	I/O
73	I/O
74	I/O
75	I/O
76	I/O
77	I/O
78	I/O
79	I/O
80	I/O
81	INIT -I/O
82	Vcc
83	GND
84	I/O
85	I/O
86	I/O
87	I/O
88	I/O
89	I/O
90	I/O
91	I/O
92	I/O
93	I/O
94	I/O
95	I/O
96	I/O
97	I/O
98	I/O
99	XTAL2(IN)-I/O
100	GND
101	RESET
102	Vcc

CQFP Pin No.	Function
103	DONE- PROG
104	D7-I/O
105	XTAL1(OUT)-BCLKIN-I/O
106	I/O
107	I/O
108	I/O
109	D6-I/O
110	I/O
111	I/O
112	I/O
113	I/O
114	I/O
115	D5-I/O
116	CS0 -I/O
117	I/O
118	I/O
119	I/O
120	I/O
121	D4-I/O
122	I/O
123	Vcc
124	GND
125	D3-I/O
126	CS1 -I/O
127	I/O
128	I/O
129	I/O
130	I/O
131	D2-I/O
132	I/O
133	I/O
134	I/O
135	I/O
136	I/O
137	D1-I/O
138	RDY/ BUSY - RCLK -I/O
139	I/O
140	I/O
141	I/O
142	I/O

CQFP Pin No.	Function
143	D0-DIN-I/O
144	DOUT-I/O
145	CCLK
146	Vcc
147	GND
148	A0- WS -I/O
149	A1-CS2-I/O
150	I/O
151	I/O
152	A2-I/O
153	A3-I/O
154	I/O
155	I/O
156	A15-I/O
157	A4-I/O
158	I/O
159	I/O
160	A14-I/O
161	A5-I/O
162	I/O
163	I/O
164	GND
1	Vcc
2	A13-I/O
3	A6-I/O
4	I/O
5	I/O
6	I/O
7	I/O
8	A12-I/O
9	A7-I/O
10	I/O
11	I/O
12	A11-I/O
13	A8-I/O
14	I/O
15	I/O
16	A10-I/O
17	A9-I/O
18	Vcc
19	GND

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

Pin Assignments (continued)

Table 6. ATT3090 Military FPGA, 175-Pin CPGA Pinouts

PGA Pin No.	Function
B2	PWRDWN
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 No.	Function
D13	I/O
B14	M1- RDATA
C14	GND
B15	M0-RTRIG
D14	Vcc
C15	M2-I/O
E14	HDC-I/O
B16	I/O
D15	I/O
C16	I/O
D16	LDC -I/O
F14	I/O
E15	I/O
E16	I/O
F15	I/O
F16	I/O
G14	I/O
G15	I/O
G16	I/O
H16	I/O
H15	INIT -I/O
H14	Vcc
J14	GND
J15	I/O
J16	I/O
K16	I/O
K15	I/O
K14	I/O
L16	I/O
L15	I/O
M16	I/O
M15	I/O
L14	I/O
N16	I/O
P16	I/O
N15	I/O
R16	I/O
M14	I/O
P15	XTAL2(IN)-I/O
N14	GND
R15	RESET
P14	Vcc

PGA Pin No.	Function
R14	DONE- PROG
N13	D7-I/O
T14	XTAL1(OUT)-BCLKIN-I/O
P13	I/O
R13	I/O
T13	I/O
N12	I/O
P12	D6-I/O
R12	I/O
T12	I/O
P11	I/O
N11	I/O
R11	I/O
T11	D5-I/O
R10	CS0 -I/O
P10	I/O
N10	I/O
T10	I/O
T9	I/O
R9	D4-I/O
P9	I/O
N9	Vcc
N8	GND
P8	D3-I/O
R8	CS1 -I/O
T8	I/O
T7	I/O
N7	I/O
P7	I/O
R7	D2-I/O
T6	I/O
R6	I/O
N6	I/O
P6	I/O
T5	I/O
R5	D1-I/O
P5	RDY/ BUSY - RCLK -I/O
N5	I/O
T4	I/O
R4	I/O
P4	I/O

PGA Pin No.	Function
R3	D0-DIN-I/O
N4	DOUT-I/O
R2	CCLK
P3	Vcc
N3	GND
P2	A0- WS -I/O
M3	A1-CS2-I/O
R1	I/O
N2	I/O
P1	A2-I/O
N1	A3-I/O
L3	I/O
M2	I/O
M1	A15-I/O
L2	A4-I/O
L1	I/O
K3	I/O
K2	A14-I/O
K1	A5-I/O
J1	I/O
J2	I/O
J3	GND
H3	Vcc
H2	A13-I/O
H1	A6-I/O
G1	I/O
G2	I/O
G3	I/O
F1	I/O
F2	A12-I/O
E1	A7-I/O
E2	I/O
F3	I/O
D1	A11-I/O
C1	A8-I/O
D2	I/O
B1	I/O
E3	A10-I/O
C2	A9-I/O
D3	Vcc
C3	GND

Notes: 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.

Absolute Maximum Ratings

Stresses in excess of the Absolute Maximum Ratings can cause permanent damage to the device. These are absolute stress ratings only. Functional operation of the device is not implied at these or any other conditions in excess of those given in the operational sections of the data sheet. Exposure to Absolute Maximum Ratings for extended periods can adversely affect device reliability.

Parameter	Symbol	Min	Max	Unit
Supply Voltage Relative to GND	V_{CC}	–0.5	7.0	V
Input Voltage with Respect to GND	V_{IN}	–0.5	V_{CC} + 0.5	V
Voltage Applied to 3-state Output	V_{TS}	–0.5	V_{CC} + 0.5	V
Ambient Storage Temperature	T_{stg}	–65	150	°C
Maximum Soldering Temperature (10 s @ 1/16 in.)	T_{SOL}	—	260	°C
Maximum Junction Temperature	T_J	—	150	°C

Electrical Characteristics

Table 7. Electrical Charcteristics Over Operating Conditions

Parameter	Symbol	Conditions (–55 °C ≤ Tc ≤ +125 °C, V_{CC} = 5.0 V ± 10%)	ATT3020		ATT3042		ATT3090		Unit
			Min	Max	Min	Max	Min	Max	
High-Level Output Voltage	V_{OH}	V_{CC} = 4.5 V, I_{OH} = –4.0 mA	3.7	—	3.7	—	3.7	—	V
Low-Level Output Voltage	V_{OL}	V_{CC} = 5.5 V, I_{OH} = 4.0 mA	—	0.4	—	0.4	—	0.4	V
Quiescent Operating Power Supply Current:	I_{CCO}	V_{IN} = V_{CC} = 5.5 V							
CMOS Mode			—	1	—	1.65	—	3.0	mA
TTL Mode			—	15	—	11.15	—	15	mA
Powerdown Supply Current	I_{CCPD}	V_{IN} = V_{CC} = 5.5 V; $\overline{PWRDWN}$ = 0 V	—	0.5	—	1.15	—	2.5	mA
Leakage Current	I_{IL}	V_{CC} = 5.5 V, V_{IN} = V_{CC} and 0 V	–20	20	–20	20	–20	20	µA
Horizontal Long Line Pull-up Current	I_{RLL}	Measured as an Average at V_{CC} = 5.5 V	—	2.4	—	2.4	—	2.4	mA
Input High-level TTL	V_{IHT}	Guaranteed Input High	2.0	—	2.0	—	2.0	—	V
Input Low-level TTL	V_{ILT}	Guaranteed Input Low	—	0.8	—	0.8	—	0.8	V
Input High-level CMOS	V_{IHC}	Guaranteed Input High	0.7 V_{CC}	—	0.7 V_{CC}	—	0.7 V_{CC}	—	V
Input Low-level CMOS	V_{ILC}	Guaranteed Input Low	—	0.2 V_{CC}	—	0.2 V_{CC}	—	0.2 V_{CC}	V

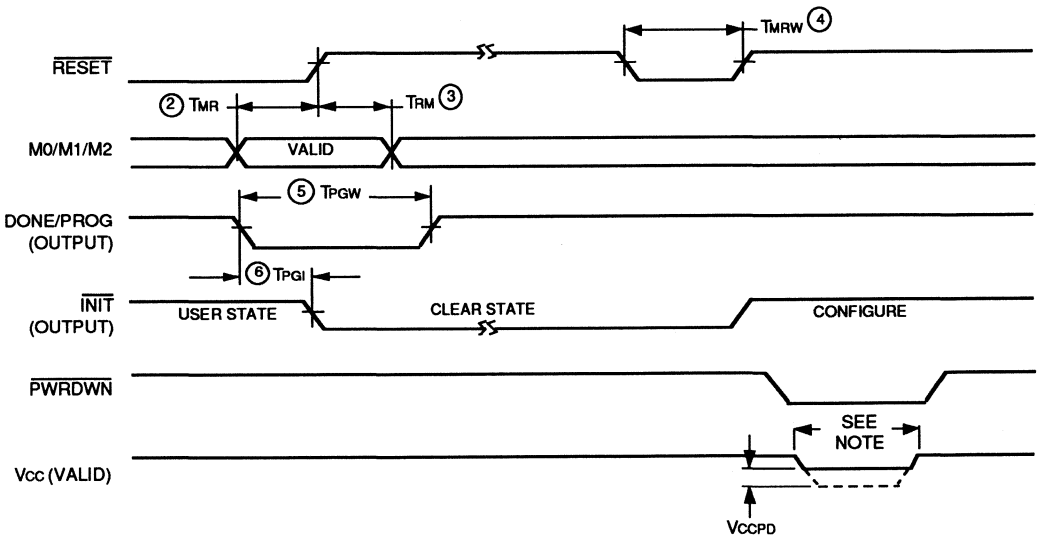
* Measured as an average @ V_{CC} = 5.5 V.

Switching Characteristics

Table 8. General FPGA Switching Characteristics

Parameter	Symbol	Conditions (–55 °C ≤ Tc ≤ +125 °C, Vcc = 5.0 V ± 10%)	ATT3020 (70 MHz)		ATT3042 (70 MHz)		ATT3090 (70 MHz)		Unit
			Min	Max	Min	Max	Min	Max	
DONE/PROG		(See Figure 1.)							
Program Width (low)	TPGW	5	6	—	6	—	6	—	µs
Initialization	TPGI	6	—	7	—	7	—	7	
$\overline{\text{PWRDWN}}$ *									
Powerdown Supply	VCCPD	—	3.5	—	3.5	—	3.5	—	V
Reset †		(See Figure 1.)							
M2, M1, M0 Setup	TMR	2	1	—	1	—	1	—	
M2, M1, M0 Hold	TRM	3	1	—	1	—	1	—	µs
Width (low) Abort	TMRW	4	6	—	6	—	6	—	

* $\overline{\text{PWRDWN}}$ transitions must occur during operational Vcc levels.
† $\overline{\text{RESET}}$ timing relative to valid modelines (M0, M1, M2) is relevant only when $\overline{\text{RESET}}$ is used to delay configuration



Note: $\overline{\text{PWRDWN}}$ transitions must occur during operational Vcc levels.

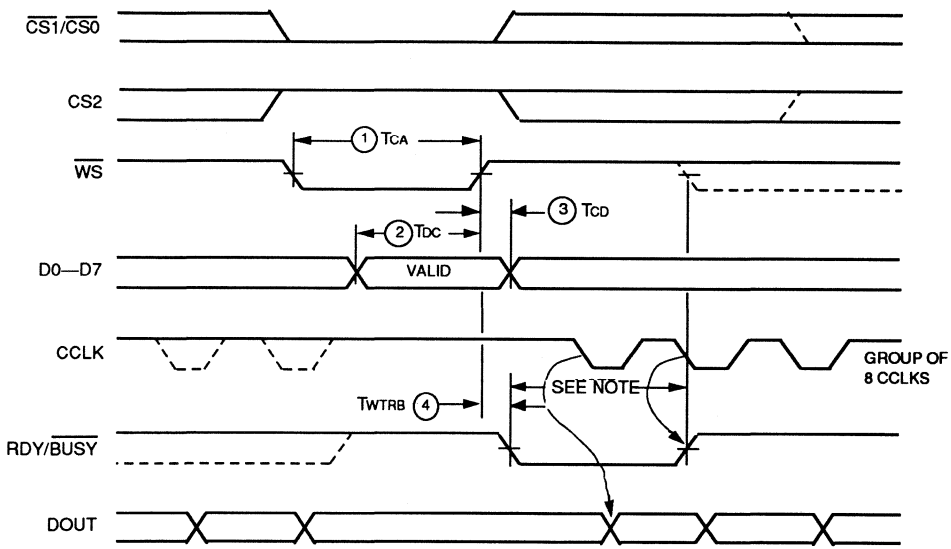
Figure 1. General FPGA Waveforms

Switching Characteristics (continued)

Table 9. Peripheral Mode Programming*

Parameter	Symbol	Conditions (–55 °C ≤ Tc ≤ +125 °C, Vcc = 5.0 V ± 10%)	ATT3020 (70 MHz)		ATT3042 (70 MHz)		ATT3090 (70 MHz)		Unit
			Min	Max	Min	Max	Min	Max	
$\overline{WS}$ Low	TCA	1 (See Figure 2.)	0.5	—	0.5	—	0.5	—	µs
DIN Setup	TDC	2	60	—	60	—	60	—	ns
DIN Hold	TCD	3	0	—	0	—	0	—	ns
Ready/Busy	TWTRB	4	—	60	—	60	—	60	ns

* Configuration must be delayed until the INIT of all FPGAs is high. WS cannot go active until Ready/Busy goes high.



Note: Configuration must be delayed until the INIT of all FPGAs is high. $\overline{WS}$ cannot go active until Ready/$\overline{Busy}$ goes high.

Figure 2. Peripheral Mode Waveforms

Switching Characteristics (continued)

Table 10. Slave Mode Programming*

Parameter	Symbol	Conditions (–55 °C ≤ Tc ≤ +125 °C, Vcc = 5.0 V ± 10%)	ATT3020 (70 MHz)		ATT3042 (70 MHz)		ATT3090 (70 MHz)		Unit
			Min	Max	Min	Max	Min	Max	
CCLK		(See Figure 3.)							
To DOUT	TCCO	3	—	100	—	100	—	100	ns
DIN Setup	TDCC	1	60	—	60	—	60	—	ns
DIN Hold	TCCD	2	0	—	0	—	0	—	ns
High Time	TCCH	4	0.5	—	0.5	—	0.5	—	µs
Low Time	TCCL	5	0.5	1.0	0.5	1.0	0.5	1.0	µs
Frequency	FCC	—	—	1	—	1	—	1	MHz

* Configuration must be delayed until the INIT of all FPGAs is high. $\overline{WS}$ cannot go active until Ready/$\overline{Busy}$ goes high.

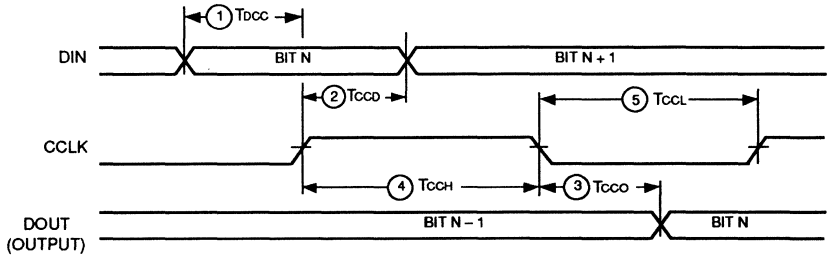


Figure 3. Slave Mode Waveforms

Table 11. Program Readback*†

Parameter	Symbol	Conditions (–55 °C ≤ Tc ≤ +125 °C, Vcc = 5.0 V ± 10%)	ATT3020 (70 MHz)		ATT3042 (70 MHz)		ATT3090 (70 MHz)		Unit
			Min	Max	Min	Max	Min	Max	
RTRIG Setup	TRTH	1 (See Figure 4.)	250	—	250	—	250	—	ns
CCLK		(See Figure 4.)							
RTRIG Setup	TRTCC	2	200	—	200	—	200	—	ns
RDATA Delay	TCCRD	3	—	100	—	100	—	100	ns
Clock Low	TCCLR	4	0.5	1.0	1.2	2.0	1.2	2.0	ns
Clock High	TCCHR	5	0.5	—	0.5	—	0.5	—	ns

* Readback should not be initiated until configuration is complete.

† DOUT timing is the same for slave mode.

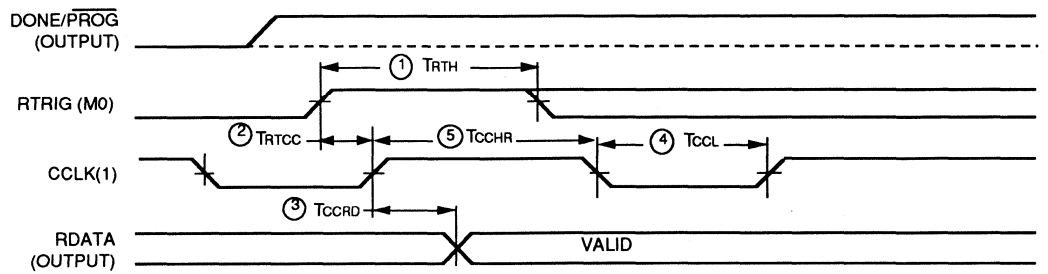


Figure 4. Program Readback Waveforms

Switching Application Guidelines

Testing of 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 is taken at initial device qualification, prior to introduction of signicant changes, and at least twice yearly to monitor correlation between benchmark patterns, device performance, *XACT* software timings, and the data sheet.

Table 12. Benchmark Patterns

Parameter	Symbol	Conditions (–55 °C ≤ Tc ≤ +125 °C, Vcc = 5.0 V ± 10%)	ATT3020 (70 MHz)		ATT3042 (70 MHz)		ATT3090 (70 MHz)		Unit
			Min	Max	Min	Max	Min	Max	
TPID + Interconnect + 8 (TILO)* + TOP Measured on 8 Cols.	TB1	—	—	86	—	122	—	194	ns
TCKO + TICK + TCKI + Interconnect	TB2	Tested on all CLBs	—	21	—	21	—	21	ns
TCKO + TQLO + TILO + TDICK + Interconnect	TB3	Tested on all CLBs	—	34	—	34	—	34	ns
TILO + TECCK + Interconnect	TB4	Tested on all CLBs	—	23	—	23	—	23	ns
TOKPO + TOPS – TOPF + TPICK	TB5	Tested on all CLBs	—	53	—	53	—	53	ns
TCKO + TOLO + TPUS + TICK + Interconnect	TB6	One Long Line Pull-up	—	48	—	48	—	48	ns
TCKO + TQLO + TPUS + TICK + Interconnect	TB7	The Other Long Line Pull-up	—	55	—	55	—	55	ns
TCKO + TQLO + TIO + TICK + Interconnect	TB8	No Pull-up, Lower Long Lines	—	31	—	31	—	31	ns
TCKO + TQLO + TIO + TICK + Interconnect	TB9	No Pull-up, Upper Long Lines	—	38	—	38	—	38	ns

* This figure is 12 (TILO) for the ATT3042 and 20 (TILO) for the ATT3090.

Table 13. Internal Buffer Switching Application Guidelines

Parameter	Symbol	Conditions (–55 °C ≤ Tc ≤ +125 °C, Vcc = 5.0 V ± 10%)	ATT3020 (70 MHz)		ATT3042 (70 MHz)		ATT3090 (70 MHz)		Unit
			Min	Max	Min	Max	Min	Max	
Clock Buffer	TGCK	—	—	6	—	6	—	6	ns
TBUF									
Data to Output	TIO	—	—	5	—	5	—	5	ns
3-State to Output	—	—	—	—	—	—	—	—	—
Single Pull-up	TPUS	—	—	22	—	36	—	36	ns
Pair of Pull-ups	TPUF	—	—	11	—	16	—	16	ns
Bidirectional	TBIDI	—	—	4	—	4	—	4	ns

Switching Application Guidelines (continued)

Table 14. CLB Switching Application Guidelines

Parameter	Symbol	Conditions (–55 °C ≤ Tc ≤ +125 °C, Vcc = 5.0 V ± 10%)	ATT3020 (70 MHz)		ATT3042 (70 MHz)		ATT3090 (70 MHz)		Unit
			Min	Max	Min	Max	Min	Max	
Combinatorial	TILO	1 (See Figure 5.)	—	9	—	9	—	9	ns
Reset to CLB Output	TRIO	9	—	8	—	8	—	8	ns
Reset Direct Width	TRPW	13	8	—	8	—	8	—	ns
Master Reset Pin to CLB Out	TMRQ	—	—	20	—	34	—	34	ns
K Clock*		(See Figure 5.)							
To CLB Output	TCKO	8	—	8	—	8	—	8	ns
Additional for Q Returning Through F or G to CLB Out	TQLO	—	—	7	—	7	—	7	ns
Logic-Input Setup	TICK	2	8	—	8	—	8	—	ns
Logic-Input Hold	TCKI	3	1	—	1	—	1	—	ns
Data In Setup	TDICK	4	5	—	5	—	5	—	ns
Data In Hold (1)	TCKDI	5	4	—	4	—	4	—	ns
Enable Clock Setup	TECCK	6	7	—	7	—	7	—	ns
Enable Clock Hold	TCKEC	7	0	—	0	—	0	—	ns
Clock (high)†	TCH	11	7	—	7	—	7	—	ns
Clock (low)†	TCL	12	7	—	7	—	7	—	ns

* 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.

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

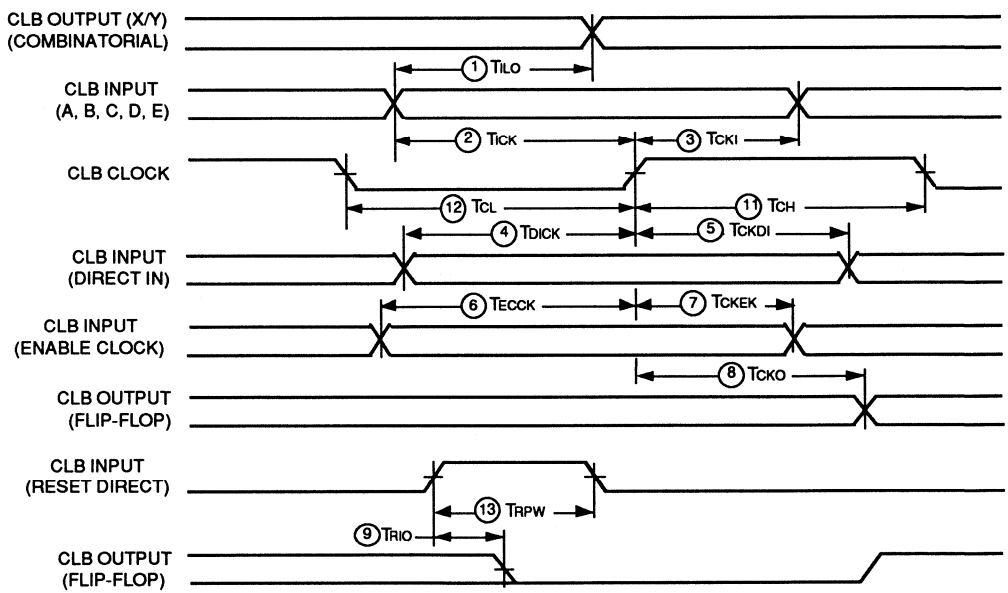


Figure 5. CLB Waveforms

Switching Application Guidelines (continued)

Table 15. IOB Switching Application Guidelines*

Parameter	Symbol	Conditions (–55 °C ≤ Tc ≤ +125 °C, Vcc = 5.0 V ± 10%)	ATT3020 (70 MHz)		ATT3042 (70 MHz)		ATT3090 (70 MHz)		Unit
			Min	Max	Min	Max	Min	Max	
Pad (package pin)		(See Figure 7.)							
To Inputs TCKLIN, BCKLIN	TPIDC	—	—	5	—	5	—	5	ns
To Inputs DIRECT IN	TPID	3	—	6	—	6	—	6	ns
I/O Clock		(See Figure 7.)							
To I/O RI Input (FF)	TIKRI	4	—	7	—	7	—	7	ns
I/O Pad-Input Setup	TPICK	1	20	—	20	—	20	—	ns
I/O Pad-Input Hold	TIKPI	2	0	—	0	—	0	—	ns
I/O Pad (fast)	TOKPO	7	—	13	—	13	—	13	ns
I/O Pad-Output Setup	TOKO	5	10	—	10	—	10	—	ns
I/O Pad-Output Hold	TOOK	6	0	—	0	—	0	—	ns
Clock (high)†	TIOH	11	8	—	8	—	8	—	ns
Clock (low)†	TIOL	12	8	—	8	—	8	—	ns
Output		(See Figure 7.)							
To Pad (enabled fast)	TOPF	10	—	9	—	9	—	9	ns
To Pad (enabled slow)	TOPS	10	—	29	—	29	—	29	ns
3-state		(See Figure 7.)							
To Pad Begin High-Z (fast)	TTSHZ	8	—	12	—	12	—	12	ns
To Pad Valid (fast)	TTSON	9	—	14	—	14	—	14	ns
Master Reset		(See Figure 7.)							
To Input RI	TRRI	13	—	23	—	33	—	33	ns
To Output (FF)	TRPO	14	—	33	—	43	—	47	ns

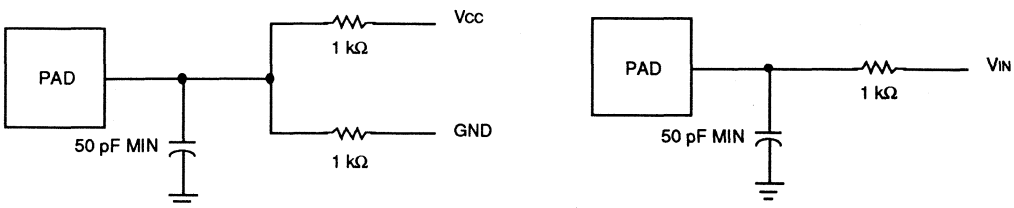
* Voltage levels of unused pads must be valid logic levels. Each can be configured with the internal pull-up resistor, configured as a driven output, or driven from an external source.

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

Note: 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 = 0 V and high = 3 V.

TTSHZ is determined when the output shifts 10% (of the voltage swing) from VOL level or VOH level. Load circuit A is used. (See Figure 6.)

TTSON is measured at 0.5 Vcc level with VI = 0 for 3-state to active-low. Load circuit B is used. (See Figure 6.)



Load Circuit A **Load Circuit B**

Figure 6. Load Circuits

Switching Application Guidelines (continued)

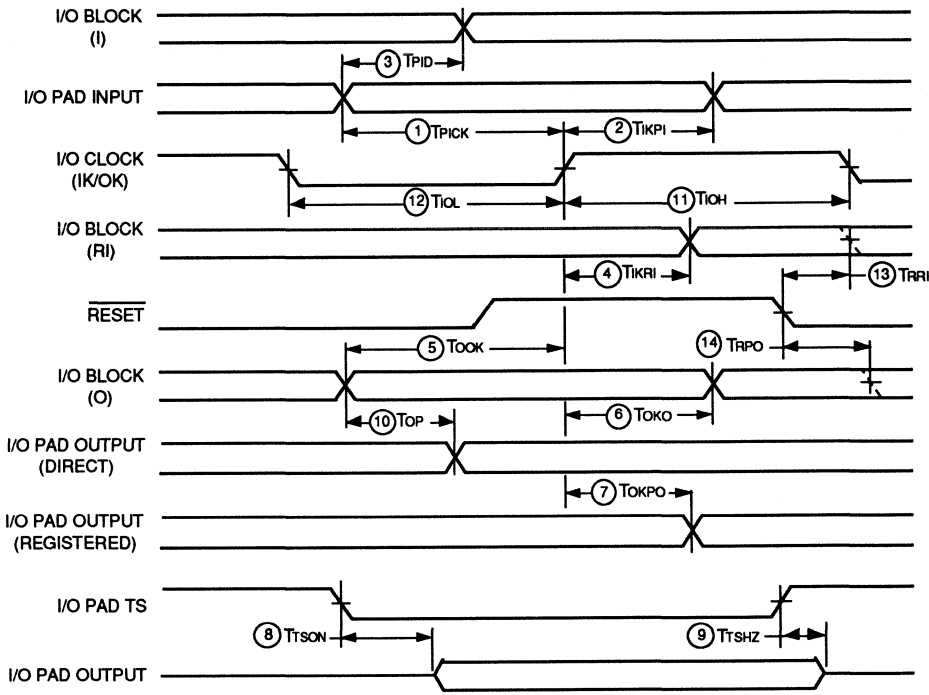


Figure 7. IOB Waveforms

Switching Application Guidelines (continued)

Table 16. Master Parallel Mode Programming Switching Application Guidelines*

Parameter	Symbol	Conditions (–55 °C ≤ Tc ≤ +125 °C, Vcc = 5.0 V ± 10%)	ATT3020 (70 MHz)		ATT3042 (70 MHz)		ATT3090 (70 MHz)		Unit
			Min	Max	Min	Max	Min	Max	
$\overline{\text{RCLK}}$		(See Figure 8.)							
To Address Valid	TRAC	1	0	200	0	200	0	200	ns
To Data Setup	TDRC	2	60	—	60	—	60	—	ns
To Data Hold	TRCD	3	0	—	0	—	0	—	ns
$\overline{\text{RCLK}}$ High	TRCH	4	600	—	600	—	600	—	ns
$\overline{\text{RCLK}}$ Low	TRCL	5	4	—	4	—	4	—	µs

* At powerup, Vcc must rise from 2.0 V to Vcc minimum in less than 10 ms. Otherwise, delay configuration using $\overline{\text{RESET}}$.

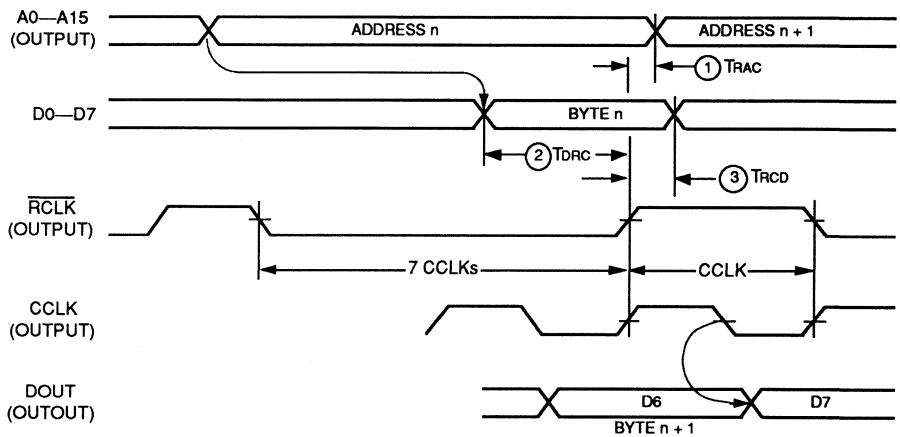
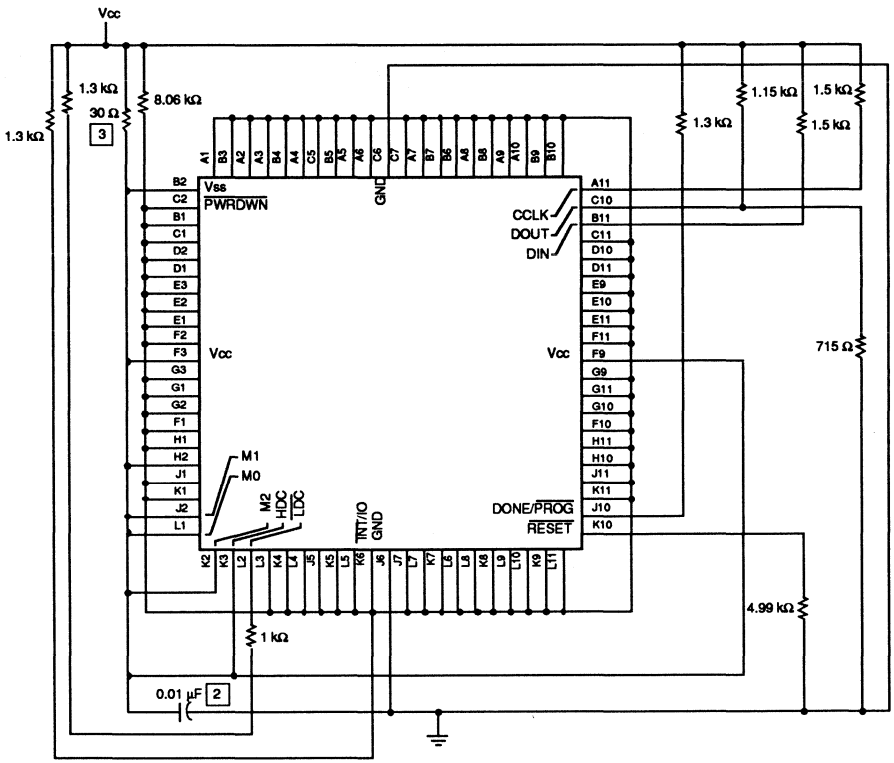


Figure 8. Master Parallel Mode Waveforms

Package Information

Refer to Section 7 for details on package types and outline drawings.

Static Burn-In Circuits



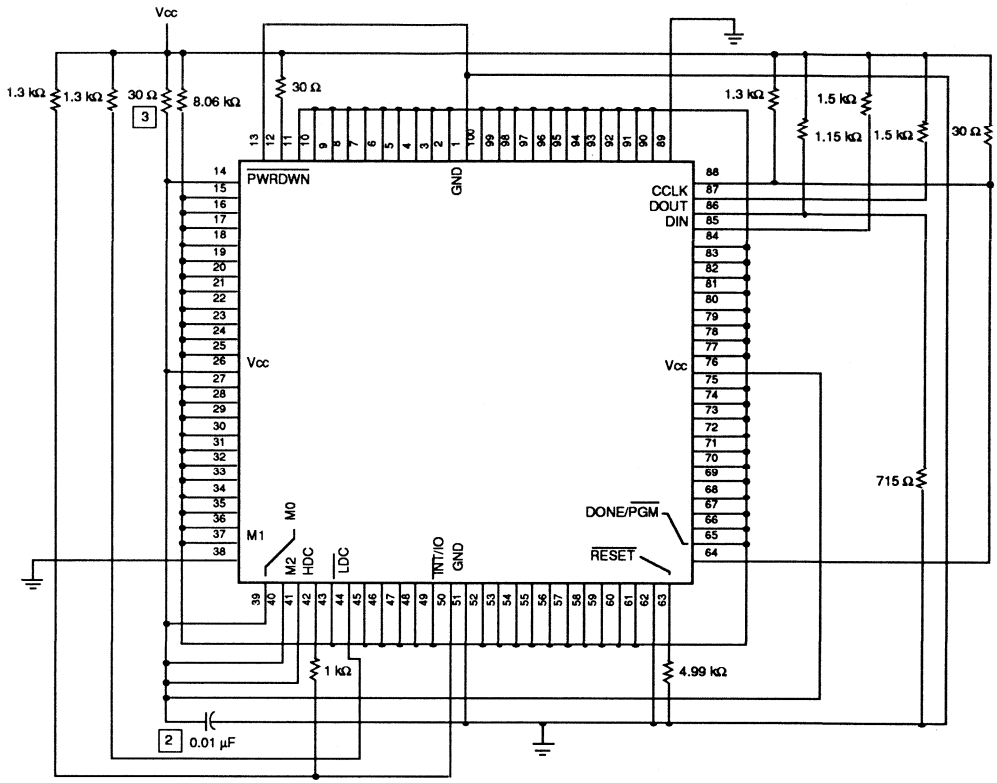
Notes:
Unless otherwise specified, all resistors are metal film and are rated for 1/8 W at 150 °C with a build tolerance of 1% and 5% tolerance over life.

Capacitor has a 10%, 50 V rating with an X7R temperature characteristic.

The 30 Ω resistor is metal oxide and rated for 1 W at 150 °C with a tolerance of 5%.

Figure 9. ATT3020/3042 84-Pin PGA

Static Burn-In Circuits (continued)



Notes:
Unless otherwise specified, all resistors are metal film and are rated for 1/8 W at 150 °C with a build tolerance of 1% and 5% tolerance over life.

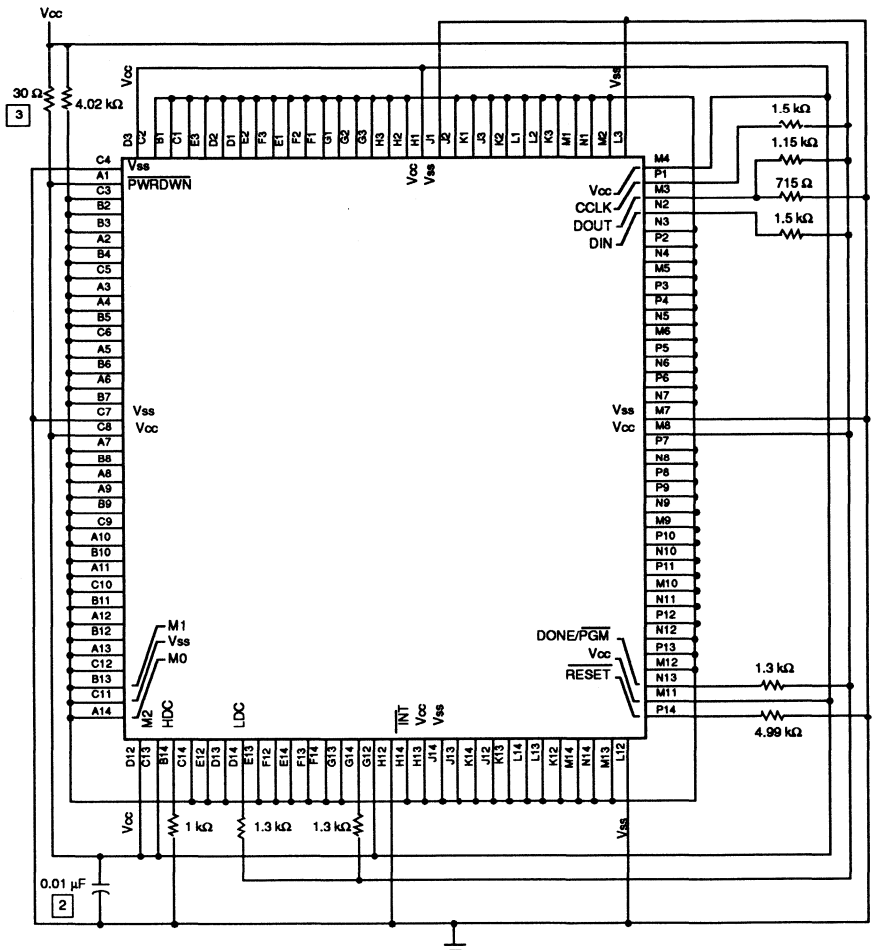
Capacitor has a 10%, 50 V rating with an X7R temperature characteristic.

The 30 Ω resistor is metal oxide and rated for 1 W at 150 °C with a tolerance of 5%.

Unless otherwise specified, socket will be ENPLAS, part number FPQ-1320.635-01, or Wells, part number CP-10582.

Figure 10. ATT3020/3042 100-Pin CQFP

Static Burn-In Circuits (continued)



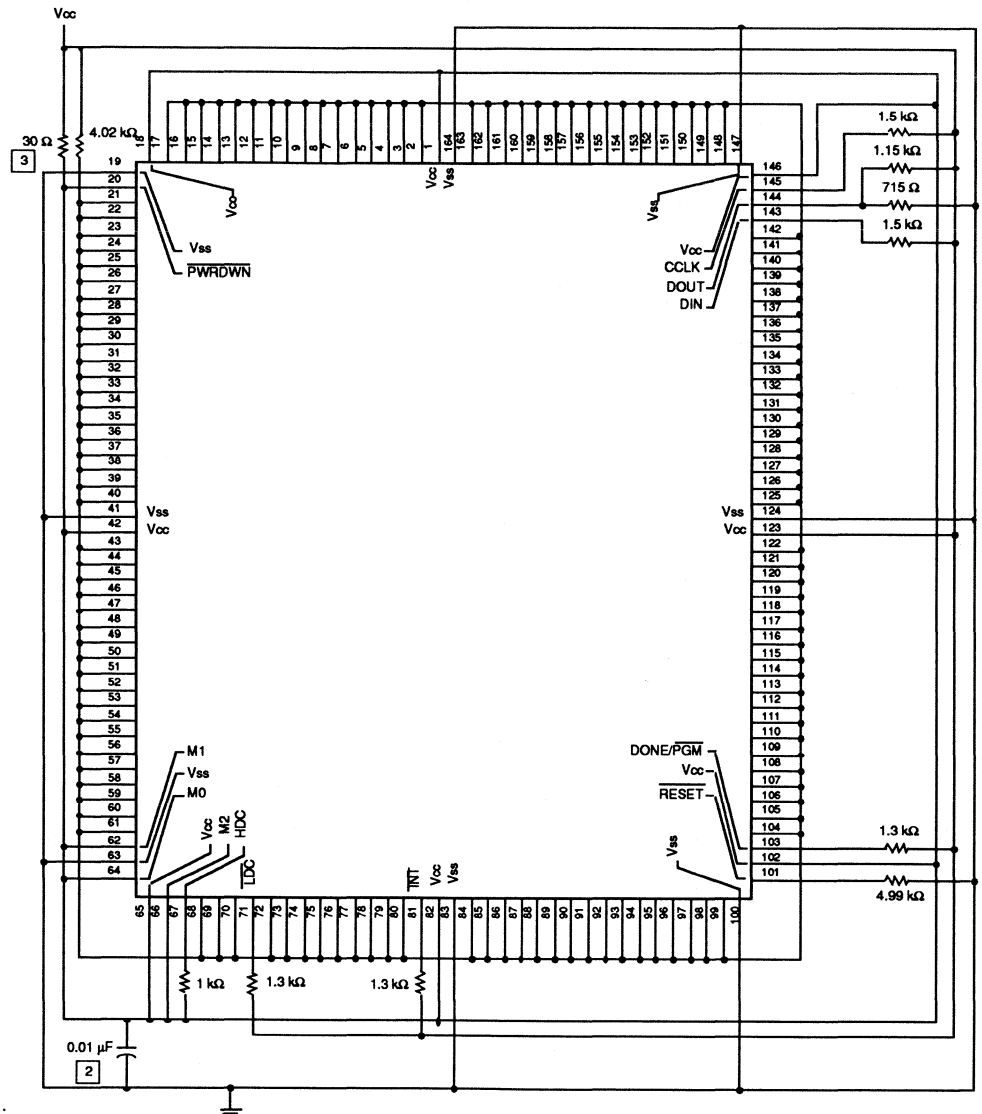
Notes:
Unless otherwise specified, all resistors are metal film and are rated for 1/8 W at 150 °C with a build tolerance of 1% and 5% tolerance over life.

Capacitor has a 10%, 50 V rating with an X7R temperature characteristic.

The 30 Ω resistor is metal oxide and rated for 1 W at 150 °C with a tolerance of 5%.

Figure 11. ATT3042 132-Pin PGA

Static Burn-In Circuits (continued)



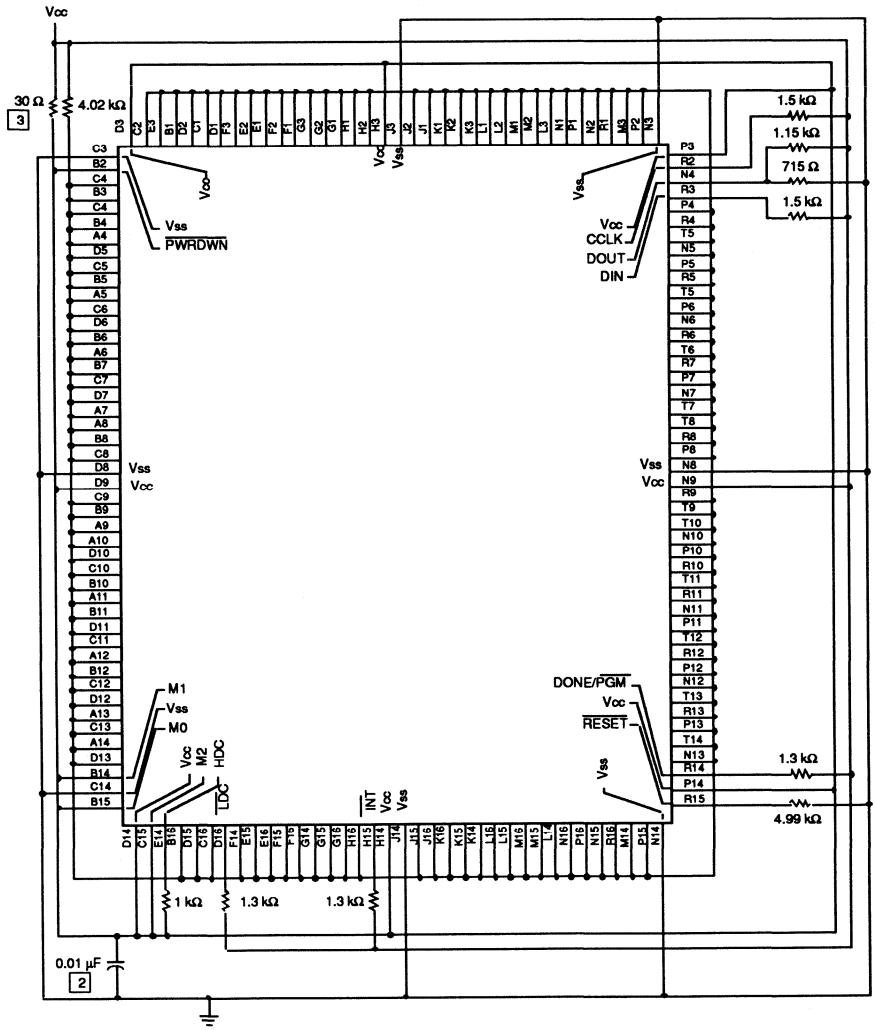
Notes:
Unless otherwise specified, all resistors are metal film and are rated for 1/8 W at 150 °C with a build tolerance of 1% and 5% tolerance over life.

Capacitor has a 10%, 50 V rating with an X7R temperature characteristic.

The 30 Ω resistor is metal oxide and rated for 1 W at 150 °C with a tolerance of 5%.

Figure 12. ATT3090 164-Pin CQFP

Static Burn-In Circuits (continued)



Notes:
Unless otherwise specified, all resistors are metal film and are rated for 1/8 W at 150 °C with a build tolerance of 1% and 5% tolerance over life.

Capacitor has a 10%, 50 V rating with an X7R temperature characteristic.

The 30 Ω resistor is metal oxide and rated for 1 W at 150 °C with a tolerance of 5%.

Figure 13. ATT3090 175-Pin PGA

AT&T Microelectronics

ATT3000 Series FPGA Die Information

Die Information

Detailed die information for the ATT3000 Series FPGA is presented in Table 1 to Table 6.

Table 1. Mechanical Dimensions

Item	Description
Die Dimensions*	
ATT3020	X = 183 mils Y = 219 mils
ATT3030	X = 215 mils Y = 259 mils
ATT3042	X = 242 mils Y = 295 mils
ATT3064	X = 270 mils Y = 366 mils
ATT3090	X = 299 mils Y = 437 mils
Die Thickness	21 mils
Wafer Diameter	5 inches
Wafer Thickness	25 mils

* Die dimensions are step and repeat.

Table 2. Metalization

Item	Description
Number of Layers	2
Composition of Each Layer	M1—Ti/AlCuSi M2—AlCuSi
Minimum Metal Stripe Width(s)	M1—0.9 µm M2—1.2 µm
Minimum Metal Thickness(es)	M1—0.5 µm M2—1.0 µm
Minimum Metal Spacings(s)	M1—1.0 µm M2—1.3 µm
Dielectric Layer Thickness(es)	D1—3,500 A D2—9,000 A
Dielectric Materials	D1—Boron/Phosphorous Doped Glass D2—Undoped Glass
Minimum Contact/ Via Size	1.0 µm/1.2 µm

Table 3. Passivation

Item	Description
Number of Layers	2
Composition(s)	Undoped Glass and SiN
Thickness(es)	1.3 µm

Table 4. Nominal Oxide Thicknesses and Composition (as Applicable)

Item	Description
Isolation Oxide	3,500 Å
Gate Oxide	150 Å

Table 5. Input Protection

An ESD protection network is included on each input.

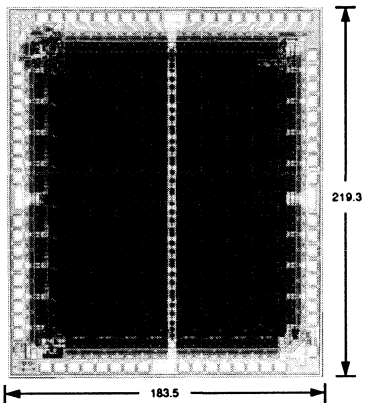
Item	Description
Process Technique	LDD
Barrier Metal (under M1 + Contacts)	TiN
Barrier Metal Thickness	600 A—M1
Junction Depth	N ch—0.35 µm P ch—0.45 µm
Contact to Diffusion Edge Spacing	5 µm
Input Pins Not Protected	None

Table 6. Output Protection

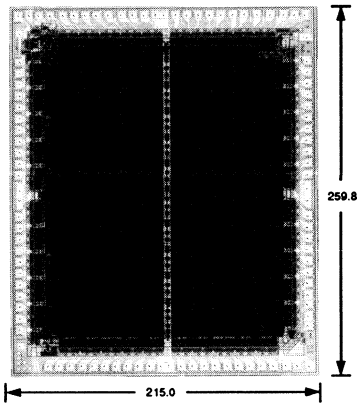
Item	Description
Process Technique	TiSi2
Channel Length of Diffused Structure	0.75 µm
Diffusion to Diffusion Spacing	2 µm
Output Pins Not Protected	None

Die Photos

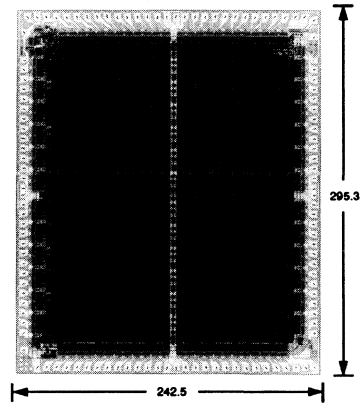
Photographs of the ATT3000 Series FPGA dies are shown below.



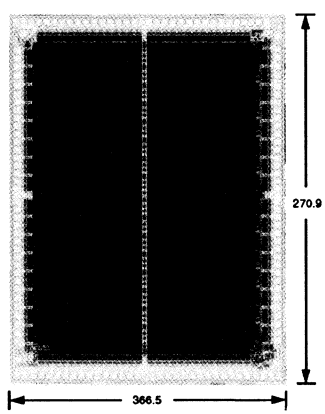
ATT3020 FPGA



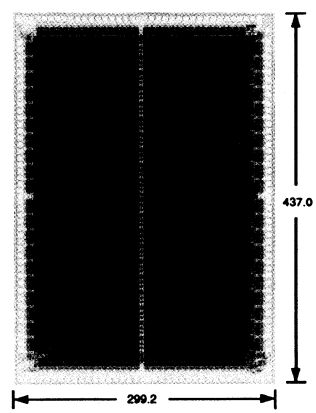
ATT3030 FPGA



ATT3042 FPGA



ATT3064 FPGA



ATT3090 FPGA

Data Sheet
June 1992

AT&T Microelectronics

ATT1736F/1765F/17128F Serial E² ROM

Features

- Reprogrammable Flash-EECMOS
- 36,288 x 1 bit, 65,536 x 1 bit, and 131072 x 1 bit serial E² ROMs for FPGA configuration
- Designed to store configuration programs for programmable gate arrays
- Daisy-chain configuration support for multiple FPGAs
- Cascadable to provide more memory for additional configurations
- Cascadable to support future higher-density arrays
- Stores configurations for single or multiple FPGAs
- Low-power EECMOS process
- 8-pin, plastic DIP
- Programming support from leading programmer manufacturers
- Pin-for-pin functional replacements for *Xilinx XC1700* devices

Description

The ATT1736F/1765F/17128F Serial E² Configuration ROM (SeROM) devices serve as easy-to-use, cost-effective, reprogrammable, and nonvolatile configuration memories for the AT&T 3000 family and ORCA Series of field-programmable gate arrays (FPGAs). SeROM devices of all sizes are currently available in the industry-standard, 8-pin, plastic, skinny dual-in-line package (DIP).

The ATT1700 SeROM devices are pin-for-pin functional replacements for normal read operation for the *Xilinx XC1700* family of serial ROM devices. The AT&T 1700 SeROM family can be programmed by a variety of commercially available programming units fielded by qualified third-party programmable device support of vendors such as BP Microsystems, Data I/O, and Stag Microsystems. A listing of qualified (AT&T tested and verified) third-party programming platforms is scheduled to be available by the fourth quarter of 1992.

For multiple FPGA devices connected in a daisy-chain configuration, or for extremely large FPGA devices requiring large configuration memories, cascaded SeROM devices offer an almost unlimited memory capacity.

Since the AT&T SeROM family is manufactured by using proprietary AT&T (Flash-EECMOS) non-destructive programming technology, an individual SeROM can be reliably reprogrammed a minimum of 10 times†. A SeROM that has been reprogrammed 10 or fewer times† is designed to meet all published data book specifications.

In a design environment where long-term data retention and tolerance to harsh environments are not at issue, SeROMs can typically be reprogrammed **hundreds** of times. Programmed information is not lost upon the removal, or momentary interruption, of operating voltage at the SeROM's power supply connections. The SeROM is truly nonvolatile and is projected to have a data-retention period in excess of 10 years*.

Unlike other ROM/PROM technologies, Flash-EECMOS technology offers the benefits of 100% testability, high-speed operation, low power, and split-second erasure for high-reliability reprogramming. The AT&T SeROM device offers a combination of flexibility, reliability, and bit capacity not available with other FPGA-specific data storage mechanisms.

* These figures are based upon preproduction analysis. Actual production figures may vary.

Description (continued)

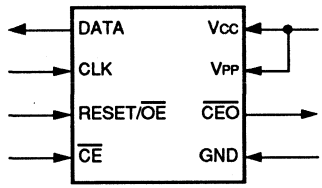


Figure 1. ATT1736F/1765F/17128F Functionality During READ

Pin Information

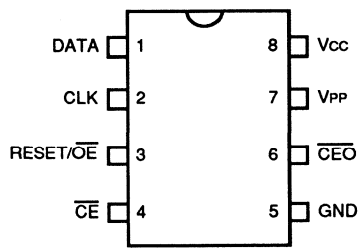


Figure 2. 8-Pin DIP Diagram

Table 1. Pin Descriptions

Pin #	Symbol	Type	Name/Function
1	DATA	O	**3-State DATA Out for Reading.** Output pin for normal read operation.
2	CLK	I	**Clock Input.** Used to increment the internal address and bit counters for normal read operation.
3	RESET/ OE	I	**RESET/Output Enable.** A low level on both the CE and RESET/ OE inputs enables the data output driver. A high level on RESET/ OE resets both the address and bit counters. The logic polarity of this input is programmable as either RESET/ OE or RESET /OE. This document describes the pin as RESET/ OE .
4	CE	I	**Chip Enable.** A low level on both CE and RESET/ OE enables the data output driver. A high level on 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	CEO	O	**Chip Enable Out.** This signal is asserted low on the clock cycle following the last bit read from the memory. It will stay low as long as CE and OE are both low. It will then follow CE , but if RESET/ OE goes high, CEO will stay high until the entire ROM is read again.
7	VPP	I	**Programming Voltage Supply.** Must be connected directly to Vcc for normal read operation.
8	Vcc	I	**+5 V Power Supply.**

Functional Description

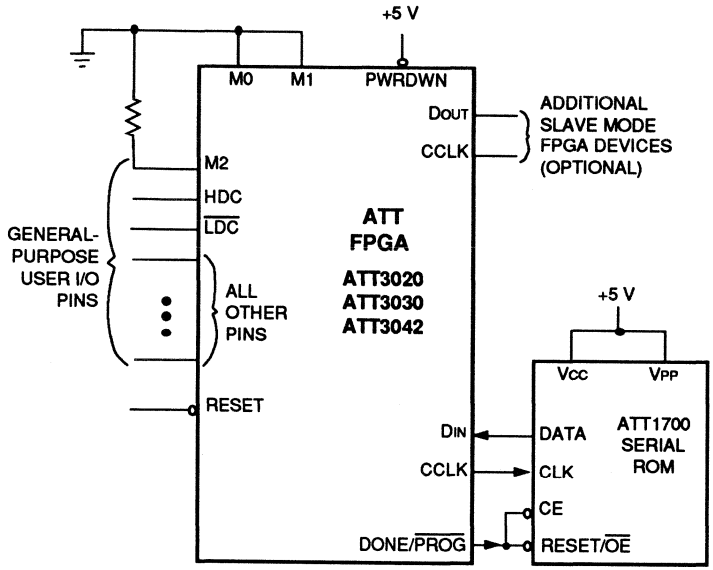


Figure 3. Master Serial Mode Configuration

FPGA Master Serial Mode Summary

The I/O and logic functions of the AT&T programmable gate array, and their associated interconnections, are established by a configuration program. The program is loaded either automatically upon powerup, or on command, depending on the state of the three FPGA mode pins. In master mode, the field-programmable gate array automatically loads the configuration program from an external memory. The serial configuration ROM has been designed for compatibility with the master serial mode.

Upon powerup or upon reconfiguration, an FPGA will enter master serial mode whenever all three of the FPGA's mode select pins are low (M0 = 0, M1 = 0, M2 = 0). Data is read from the serial E² configuration ROM 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 the FPGA. Data from the serial configuration ROM is read sequentially, accessed via the internal address and bit counters, which are incremented on every valid rising edge of CCLK.

Programming the FPGA with Counters Reset upon Completion

Figure 3 illustrates the connections between an FPGA and its SeROM. The DATA line from the SeROM is connected to the DIN input of the FPGA. The CCLK output from the FPGA is connected to the CLK input of the SeROM. At powerup or upon reconfiguration, the DONE/ PROG signal goes low (pulled low by the FPGA at reset or by external circuitry for reconfiguration), enabling the SeROM and its DATA output.

Functional Description (continued)

Programming the FPGA with Counters Reset upon Completion (continued)

During the configuration process, CCLK will clock data out of the SeROM on every rising clock edge. At the completion of configuration, the DONE/ PROG signal will go high and reset the internal address counters of the SeROM.

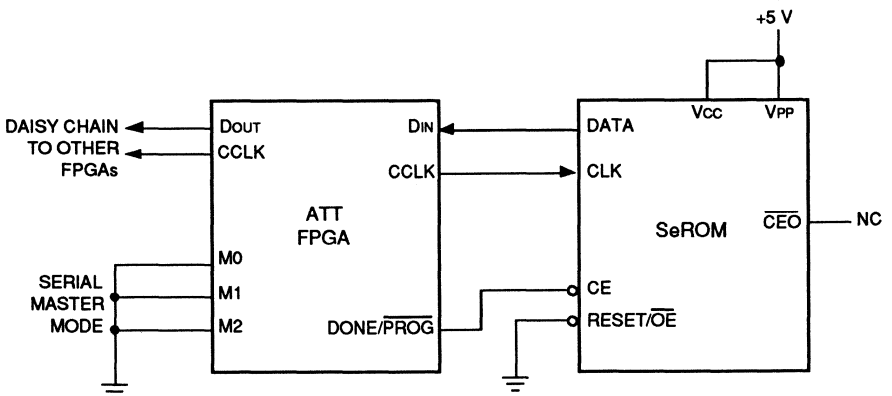
If the user-programmable, dual-function DIN and CCLK pins are used only for the configuration process, they should be programmed on the FPGA so that no nodes are floating or in contention. For example, both DIN and CCLK can be programmed as output highs during normal operation. An alternate method is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

If DIN and CCLK are used for another function after configuration, the user must avoid contention. The low during configuration (LDC) pin can be used to control the SeROM's CE and OE inputs to disable the SeROM's DATA pin, one clock cycle before DONE/ PROG is active.

If the FPGA is reprogrammed after initial powerup, note that the FPGA requires several microseconds to respond after the DONE/ PROG pin is pulled low. In this case, the LDC pin can be used instead of the DONE/ PROG pin to control the SeROM.

Programming the FPGA with Counters Unchanged upon Completion

When multiple FPGA configurations for a single FPGA are stored in a serial configuration ROM, the OE pin of the SeROM should be tied low as illustrated in Figure 4. Upon powerup, the internal address counters will be reset and configuration will begin with the first program stored in memory. Since the OE pin is held low, the address counters are left unchanged after configuration is complete. Therefore, to reprogram the FPGA with another program, the DONE/ PROG line is pulled low and configuration begins at the last value of the address counters.



Notes:
If M2 is tied directly to ground, it should be programmed as an input during operation.

If the FPGA is reset during configuration, it will abort back to initialization state. DONE/ PROG will not go high, so an external signal is required to reset the ATT1700 counters.

Figure 4. Address Counters Not Reset

Functional Description (continued)

Cascading SeROMs

For multiple FPGAs configured in a daisy chain, or for future FPGAs requiring larger configuration memories, cascaded SeROMs provide additional memory.

After the last bit from the first SeROM is read, the SeROM asserts its CEO output low and disables its own DATA line. The next SeROM recognizes the low level on its CE input and enables its own DATA output. (See Figure 5.)

After configuration is complete, the address counters of all of the cascaded SeROMs will be reset when DONE/ PROG output from the FPGA goes high, forcing the RESET/ OE on each SeROM to go high.

If the address counters are not reset upon completion, then the OE inputs can be tied to ground, as illustrated in Figure 4.

To reprogram the FPGA with another program, the DONE/ 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.

The DONE/ PROG signal is an open collector type of output and may be bused. Extremely large, cascaded serial memories in some systems may require additional logic if the rippled chip enable is too slow to activate successive SeROMs.

Standby Mode

The ATT1700 enters a low-power standby mode whenever CE is asserted high. In this mode, the SeROM consumes less than 0.5 mA of current. The output remains in a high-impedance state regardless of the state of the RESET/ OE input.

RESET/ OE Polarity

The ATT1700 enables the user to choose the reset/output enable polarity as either RESET/ OE or RESET /OE. The PROM programmer software prompts the user for the desired polarity.

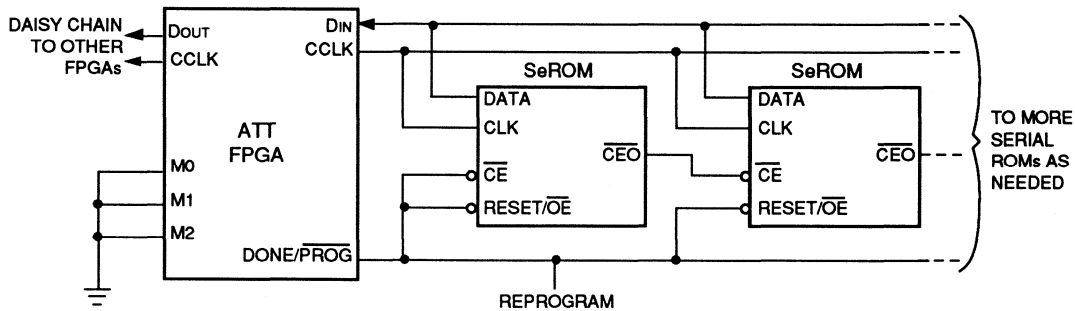


Figure 5. Cascading SeROMs

Absolute Maximum Ratings

Stresses in excess of the Absolute Maximum Ratings can cause permanent damage to the device. These are absolute stress ratings only. Functional operation of the device is not implied at these or any other conditions in excess of those given in the operational sections of the data sheet. Exposure to Absolute Maximum Ratings for extended periods can adversely affect device reliability.

Parameter	Symbol	Min	Max	Unit
Storage Temperature	T_{stg}	–65	125	°C
Soldering Temperature (10 s)	T_{SOL}	—	260	°C
Supply Voltage	V_{CC}	–0.5	7.0	V
Input Voltage	V_{IN}	–0.5	V_{CC} + 0.5	V
Voltage Applied to 3-state Output	V_{TS}	–0.5	V_{CC} + 0.5	V

Electrical Characteristics

Table 2. dc Characteristics

Parameter	Symbol	Min	Max	Unit
Supply Voltage Relative to GND:				
Commercial/Industrial	V_{CC}	4.5	5.5	V
(–40 °C to +85 °C)	V_{PP}*	4.5	5.5	V
Input Voltage:				
Low	V_{IL}	0	0.8	V
High	V_{IH}	2.0	V_{CC}	V
Output Voltage:				
Low	V_{OL} (I_{OL} = 4 mA)	—	0.32	V
High	V_{OH} (I_{OH} = –4 mA)	3.86	—	V
Supply Current:				
Active Mode	I_{CCA} (I_{CC} + I_{PP})	—	10	mA
Standby Mode	I_{CCS} (I_{CC} + I_{PP})	—	0.5	mA
Input or Output Leakage Current	I_L	–10	10	µA

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

Timing Characteristics

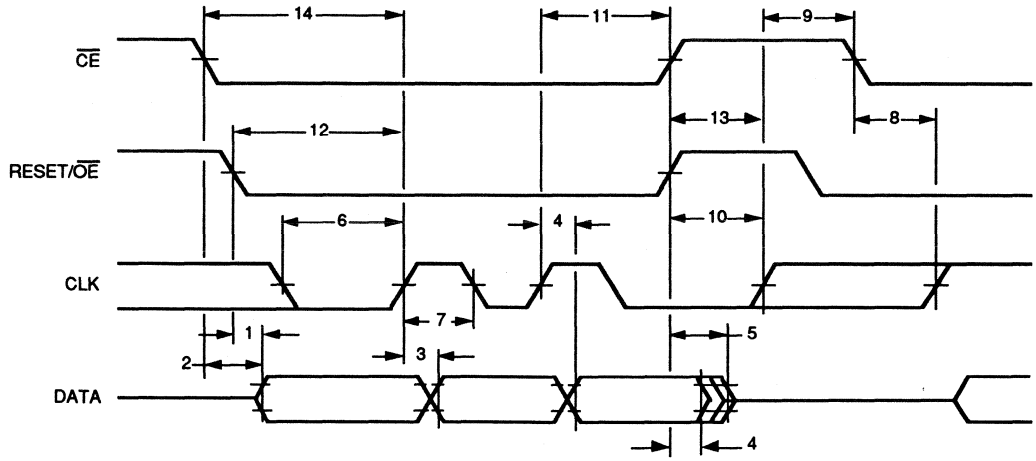


Figure 6. ac Characteristics 1

Table 3. ac Characteristics

Reference Number	Symbol	Parameter	Min	Max	Unit
1	toe	$\overline{OE}$ to Data Delay	—	100	ns
2	tce	$\overline{CE}$ to Data Delay	—	250	ns
3	tcac	CLK to Data Delay	—	400	ns
4	toh	Data Hold from $\overline{CE}$, $\overline{OE}$, or CLK	0	—	ns
5	tdf	$\overline{CE}$ or $\overline{OE}$ to Data Float Delay	—	50	ns
6	tlc	CLK Low Time	200	—	ns
7	thc	CLK High Time	200	—	ns
8	tscel	$\overline{CE}$ Low Setup Time to CLK*	100	—	ns
9	thcel	$\overline{CE}$ High Hold Time to CLK†	0	—	ns
10	thoe	$\overline{OE}$ High Time ($\overline{CE}$ can be high or low)‡	100	—	ns
11	thcbh	$\overline{CE}$ Low Hold Time to CLK*	100	—	ns
12	tsre	$\overline{OE}$ Setup Time to CLK§	100	—	ns
13	tsceh	$\overline{CE}$ High Setup Time to CLK†	100	—	ns
14	tscel1	$\overline{CE}$ Low Setup Time to First CLK§	250	—	ns

* Guarantees counters will change.
† Guarantees counters will not change.
‡ Guarantees counters are reset.
§ Guarantees first bit access.

Timing Characteristics (continued)

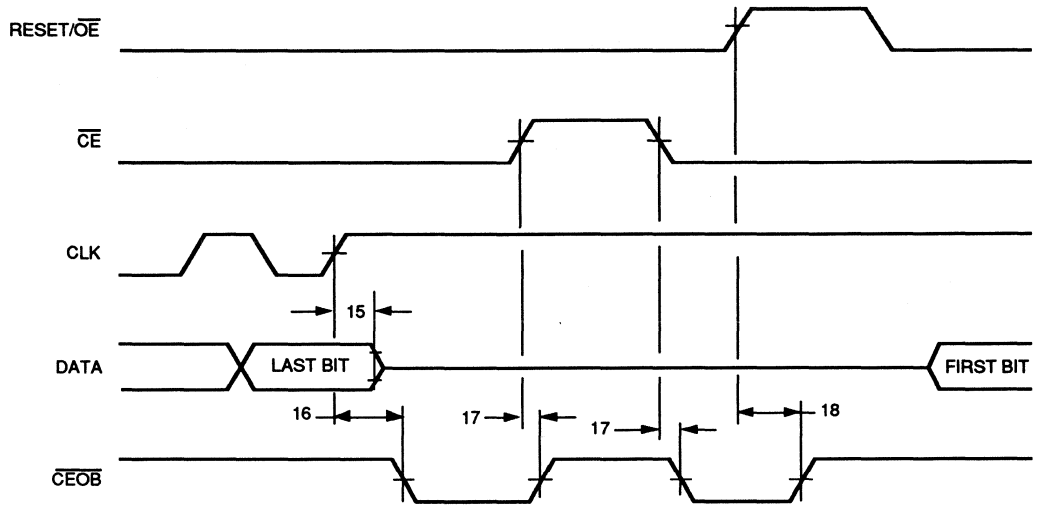
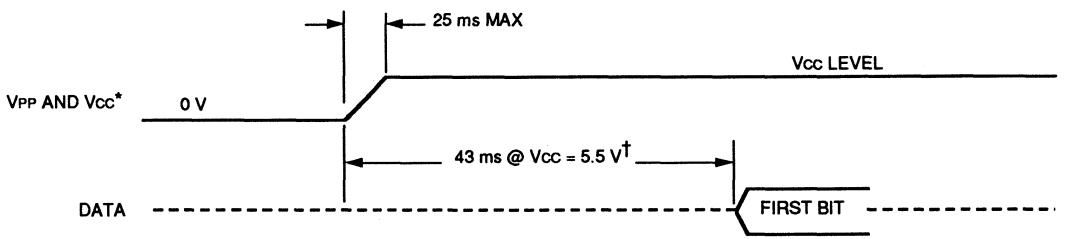


Figure 7. ac Characteristics 2

Table 4. ac Characteristics

Reference Number	Symbol	Parameter	Min	Max	Unit
15	tcdf	CLK to Data Disable Delay	—	40	ns
16	tock	CLK to $\overline{CEO}$ Delay	—	100	ns
17	toce	$\overline{CE}$ to $\overline{CEO}$ Delay	—	100	ns
18	tooe	$\overline{OE}$ to $\overline{CEO}$ Delay	—	100	ns



* VCC and VPP are tied together during a normal read operation.
† First bit data is not valid before 43 ms after powerup.

Figure 8. Powerup

Data Sheet
June 1992

AT&T Microelectronics

ATT1736/1765/17128 Serial PROM

Features

- 36,288 x 1 bit, 65,536 x 1 bit, and 131072 x 1 bit serial PROMs for FPGA configuration
- Designed to store configuration programs for programmable gate arrays
- Daisy-chain configuration support for multiple FPGAs
- Cascadable to provide more memory for additional configurations
- Cascadable to support future higher-density arrays
- Stores configurations for single or multiple FPGAs
- Low-power CMOS process
- 8-pin, plastic DIP
- Programming support from leading programmer manufacturers
- Pin-for-pin functional replacements for *Xilinx XC1700* devices

Description

The ATT1736/1765/17128 Serial Configuration PROM devices serve as easy-to-use, cost-effective non-volatile configuration memories for the AT&T 3000 family and ORCA Series of field-programmable gate arrays (FPGAs). Devices of all sizes are currently available in the industry-standard 8-pin, plastic, skinny dual-in-line package (DIP).

The ATT1700 Serial PROM devices are pin-for-pin functional replacements for normal read operation for the *Xilinx XC1700* family of serial PROM devices. The AT&T 1700 Serial PROM family can be programmed by a variety of commercially available programming units fielded by qualified third-party programmable device support of vendors such as BP Microsystems, Data I/O, and Stag Microsystems. A listing of qualified (AT&T tested and verified) third-party programming platforms is scheduled to be available by the fourth quarter of 1992.

For multiple FPGA devices connected in a daisy-chain configuration, or for extremely large FPGA devices requiring large configuration memories, cascaded serial PROM devices offer an almost unlimited memory capacity.

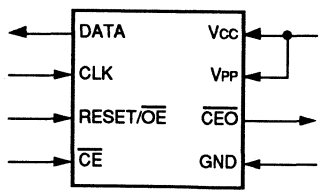


Figure 1. ATT1736/1765/17128 Functionality During READ

Pin Information

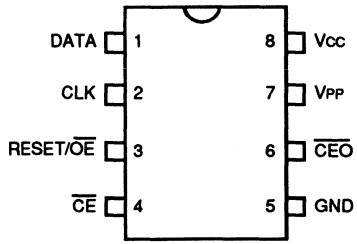


Figure 2. 8-Pin DIP Diagram

Table 1. Pin Descriptions

Pin #	Symbol	Type	Name/Function
1	DATA	O	**3-State DATA Out for Reading.** Output pin for normal read operation.
2	CLK	I	**Clock Input.** Used to increment the internal address and bit counters for normal read operation.
3	RESET/ OE	I	**RESET/Output Enable.** A low level on both the CE and RESET/ OE inputs enables the data output driver. A high level on RESET/ OE resets both the address and bit counters. The logic polarity of this input is programmable as either RESET/ OE or RESET /OE. This document describes the pin as RESET/ OE .
4	CE	I	**Chip Enable.** A low level on both CE and RESET/ OE enables the data output driver. A high level on 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	CEO	O	**Chip Enable Out.** This signal is asserted low on the clock cycle following the last bit read from the memory. It will stay low as long as CE and OE are both low. It will then follow CE , but if RESET/ OE goes high, CEO will stay high until the entire PROM is read again.
7	VPP	I	**Programming Voltage Supply.** Must be connected directly to Vcc for normal read operation.
8	VCC	I	**+5 V Power Supply.**

Functional Description

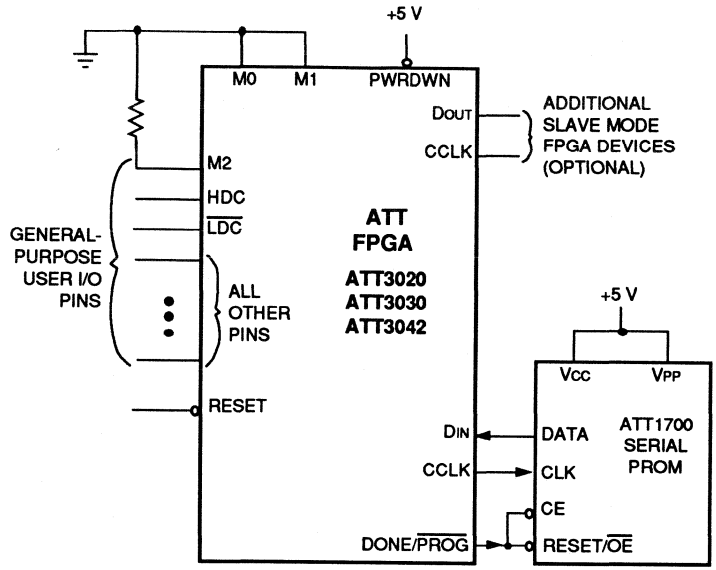


Figure 3. Master Serial Mode Configuration

FPGA Master Serial Mode Summary

The I/O and logic functions of the AT&T programmable gate array, and their associated interconnections, are established by a configuration program. The program is loaded either automatically upon powerup, or on command, depending on the state of the three FPGA mode pins. In master mode, the field-programmable gate 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 powerup or upon reconfiguration, an FPGA will enter master serial mode whenever all three of the FPGA's mode select pins are low (M0 = 0, M1 = 0, M2 = 0). Data is 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 the FPGA. 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 FPGA with Counters Reset upon Completion

Figure 3 illustrates the connections between an FPGA and its serial PROM. The DATA line from the serial PROM is connected to the DIN input of the FPGA. The CCLK output from the FPGA is connected to the CLK input of the serial PROM. At powerup or upon reconfiguration, the DONE/ $\overline{\text{PROG}}$ signal goes low (pulled low by the FPGA at reset or by external circuitry for reconfiguration), enabling the serial PROM and its DATA output.

Functional Description (continued)

Programming the FPGA with Counters Reset upon Completion (continued)

During the configuration process, CCLK will clock data out of the serial PROM on every rising clock edge. At the completion of configuration, the DONE/ PROG signal will go high and reset the internal address counters of the serial PROM.

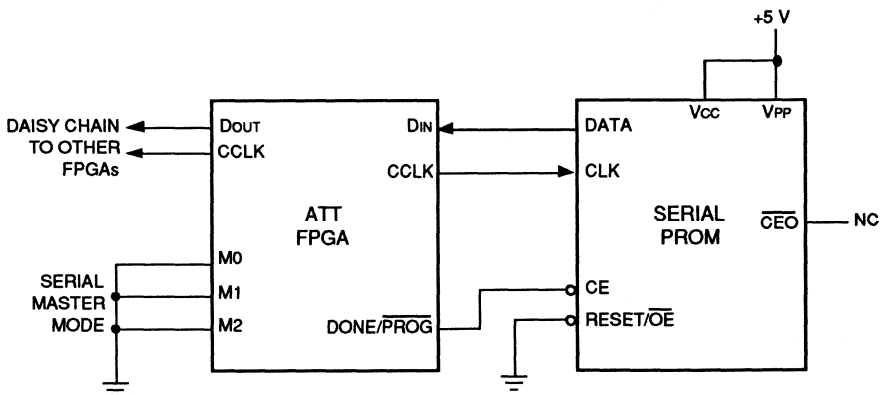
If the user-programmable, dual-function DIN and CCLK pins are used only for the configuration process, they should be programmed on the FPGA so that no nodes are floating or in contention. For example, both DIN and CCLK can be programmed as output highs during normal operation. An alternate method is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

If DIN and CCLK are used for another function after configuration, the user must avoid contention. The low during configuration (LDC) pin can be used to control the serial PROM's CE and OE inputs to disable the serial PROM's DATA pin, one clock cycle before DONE/ PROG is active.

If the FPGA is reprogrammed after initial powerup, note that the FPGA requires several microseconds to respond after the DONE/ PROG pin is pulled low. In this case, the LDC pin can be used instead of the DONE/ PROG pin to control the serial PROM.

Programming the FPGA with Counters Unchanged upon Completion

When multiple FPGA configurations for a single FPGA are stored in a serial configuration PROM, the OE pin of the serial PROM should be tied low as illustrated in Figure 4. Upon powerup, the internal address counters will be reset and configuration will begin with the first program stored in memory. Since the OE pin is held low, the address counters are left unchanged after configuration is complete. Therefore, to reprogram the FPGA with another program, the DONE/ PROG line is pulled low and configuration begins at the last value of the address counters.



Notes:
If M2 is tied directly to ground, it should be programmed as an input during operation.

If the FPGA is reset during configuration, it will abort back to initialization state. DONE/ PROG will not go high, so an external signal is required to reset the ATT1700 counters.

Figure 4. Address Counters Not Reset

Functional Description (continued)

Cascading Serial PROMs

For multiple FPGAs configured in a daisy chain, or for future FPGAs requiring larger configuration memories, cascaded serial PROMs provide additional memory.

After the last bit from the first serial PROM is read, the serial PROM asserts its CEO output low and disables its own DATA line. The next serial PROM recognizes the low level on its CE input and enables its own DATA output. (See Figure 5.)

After configuration is complete, the address counters of all of the cascaded serial PROMs will be reset when DONE/ PROG output from the FPGA goes high, forcing the RESET/ OE on each serial PROM to go high.

If the address counters are not reset upon completion, then the OE inputs can be tied to ground, as illustrated in Figure 4.

The DONE/ PROG signal is an open collector type of output and may be bused. Extremely large, cascaded serial memories in some systems may require additional logic if the rippled chip enable is too slow to activate successive serial PROMs.

Standby Mode

The ATT1700 enters a low-power standby mode whenever CE is asserted high. In this mode, the serial PROM consumes less than 0.5 mA of current. The output remains in a high-impedance state regardless of the state of the RESET/ OE input.

RESET/ OE Polarity

The ATT1700 enables the user to choose the reset/output enable polarity as either RESET/ OE or RESET /OE. The PROM programmer software prompts the user for the desired polarity.

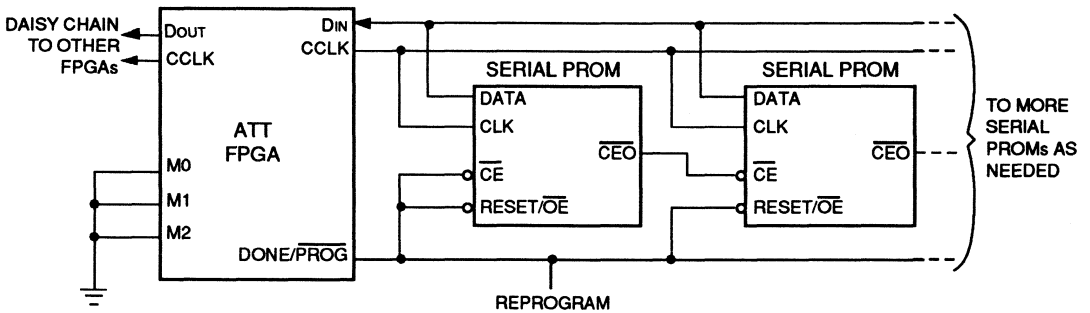


Figure 5. Cascading Serial PROMs

Absolute Maximum Ratings

Stresses in excess of the Absolute Maximum Ratings can cause permanent damage to the device. These are absolute stress ratings only. Functional operation of the device is not implied at these or any other conditions in excess of those given in the operational sections of the data sheet. Exposure to Absolute Maximum Ratings for extended periods can adversely affect device reliability.

Parameter	Symbol	Min	Max	Unit
Storage Temperature	T_{stg}	–65	125	°C
Soldering Temperature (10 s)	T_{SOL}	—	260	°C
Supply Voltage	V_{CC}	–0.5	7.0	V
Input Voltage	V_{IN}	–0.5	V_{CC} + 0.5	V
Voltage Applied to 3-state Output	V_{TS}	–0.5	V_{CC} + 0.5	V

Electrical Characteristics

Table 2. dc Characteristics

Parameter	Symbol	Min	Max	Unit
Supply Voltage Relative to GND:				
Commercial/Industrial	V_{CC}	4.5	5.5	V
(–40 °C to +85 °C)	V_{PP}*	4.5	5.5	V
Input Voltage:				
Low	V_{IL}	0	0.8	V
High	V_{IH}	2.0	V_{CC}	V
Output Voltage:				
Low	V_{OL} (I_{OL} = 4 mA)	—	0.32	V
High	V_{OH} (I_{OH} = –4 mA)	3.86	—	V
Supply Current:				
Active Mode	I_{CCA} (I_{CC} + I_{PP})	—	10	mA
Standby Mode	I_{CCS} (I_{CC} + I_{PP})	—	0.5	mA
Input or Output Leakage Current	I_L	–10	10	µA

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

Timing Characteristics

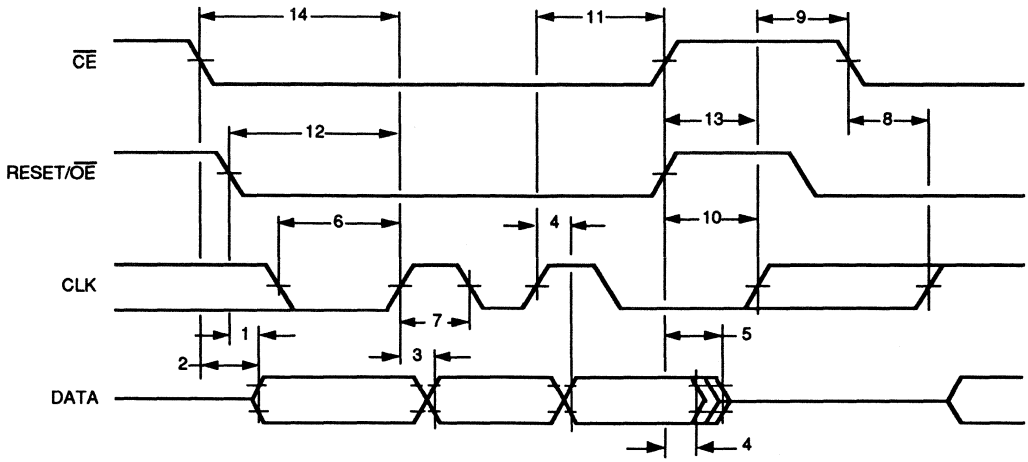


Figure 6. ac Characteristics 1

Table 3. ac Characteristics

Reference Number	Symbol	Parameter	Min	Max	Unit
1	toe	$\overline{OE}$ to Data Delay	—	100	ns
2	tce	$\overline{CE}$ to Data Delay	—	250	ns
3	tcac	CLK to Data Delay	—	400	ns
4	toh	Data Hold from $\overline{CE}$, $\overline{OE}$, or CLK	0	—	ns
5	tdf	$\overline{CE}$ or $\overline{OE}$ to Data Float Delay	—	50	ns
6	tlc	CLK Low Time	200	—	ns
7	thc	CLK High Time	200	—	ns
8	tscel	$\overline{CE}$ Low Setup Time to CLK*	100	—	ns
9	thcel	$\overline{CE}$ High Hold Time to CLK†	0	—	ns
10	thoe	$\overline{OE}$ High Time ($\overline{CE}$ can be high or low)‡	100	—	ns
11	thcbh	$\overline{CE}$ Low Hold Time to CLK*	100	—	ns
12	tsre	$\overline{OE}$ Setup Time to CLK§	100	—	ns
13	tsceh	$\overline{CE}$ High Setup Time to CLK†	100	—	ns
14	tscel1	$\overline{CE}$ Low Setup Time to First CLK§	250	—	ns

* Guarantees counters will change.
† Guarantees counters will not change.
‡ Guarantees counters are reset.
§ Guarantees first bit access.

Timing Characteristics (continued)

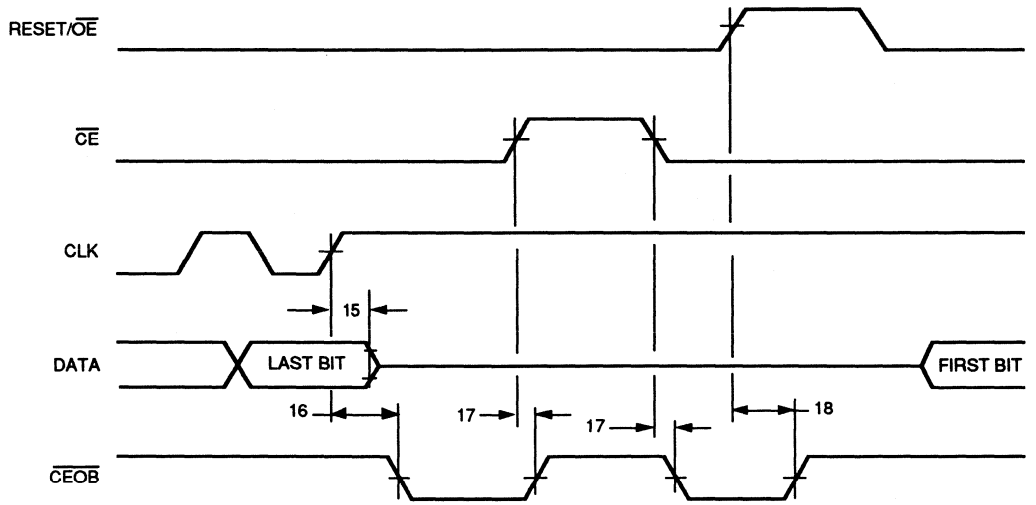
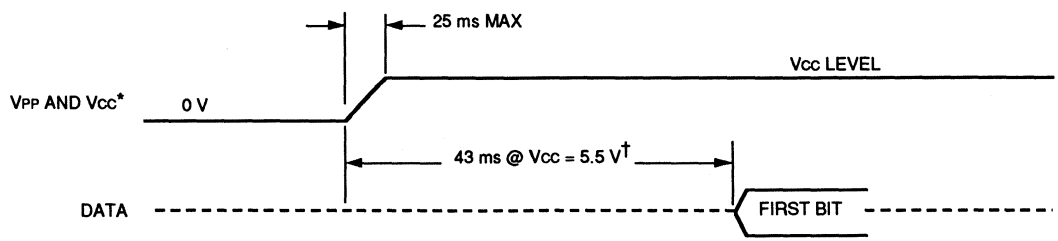


Figure 7. ac Characteristics 2

Table 4. ac Characteristics

Reference Number	Symbol	Parameter	Min	Max	Unit
15	tcdf	CLK to Data Disable Delay	—	40	ns
16	tock	CLK to $\overline{CEO}$ Delay	—	100	ns
17	toce	$\overline{CE}$ to $\overline{CEO}$ Delay	—	100	ns
18	tooe	$\overline{OE}$ to $\overline{CEO}$ Delay	—	100	ns



* Vcc and VPP are tied together during a normal read operation.
† First bit data is not valid before 43 ms after powerup.

Figure 8. Powerup

Section 4.
Software Development Systems

4. Software Development Systems

Table of Contents

Contents **Page**

Product Brief
June 1992

AT&T Microelectronics

AT&T Field-Programmable Gate Arrays (FPGA) Automatic CAE Tools Product Overview

***XACT* Design Implementation System shown with a compatible PC.**

Description

Designing with AT&T FPGAs can be accomplished by utilizing a three-step process:

- Step 1: Design entry
- Step 2: Design implementation
- Step 3: Design verification

The AT&T Automatic CAE Tools are built around the *XACT* Design Implementation System, with logic libraries and interface products for popular schematic logic drawing systems and simulation tools available to support design entry and verification. The drawing systems, supplied by multiple vendors, provide easy entry to the *XACT* Design Implementation System. Logic entry from Boolean equations or a variety of state machine language systems are also supported in the design implementation phase.

Logic synthesis, partitioning, and optimization programs translate the design specifications into configurable logic blocks (CLBs) and input/output blocks (IOBs) unique to the FPGA architecture. Subsequent programs perform automatic placement and routing (APR) to complete the FPGA design.

While completely automatic implementation is desirable for both low- and high-complexity designs, the designer may prefer an interactive process, especially in high-performance designs. This interactive editing can range from rerouting a few previously automatically routed nets, to prerouting critical nets or preplacing CLBs prior to design completion using APR, to more extensive control over logic partitioning and placement into CLBs.

The automatic place and route software gives the designer an option for direct control over specific logic mapped into CLBs (partitioning) to provide better distribution of logic signal routing through the FPGA device. The *XACT* Design Editor (XDE) is extremely versatile, ranging from design entry to CLB and signal

Description (continued)

routing manipulations. This combination of automatic and interactive design editing capability is a unique feature provided by AT&T. Logic simulation or actual in-circuit emulation provides for functional verification, while timing analysis permits verification of critical timing paths under worst-case conditions. The system contains a compiler to generate bit stream patterns to configure the FPGA device according to the designer's specification. The overall design flow is illustrated on page 4-3.

An important feature of the *XACT* Design Implementation System is the capability to incorporate design changes frequently encountered during verification. Small changes made to the schematics can be automatically incorporated into the existing design with minimal impact on existing routing and performance. Using this incremental design capability, the designer can develop production quality programmable gate arrays on a PC or engineering workstation.

Platform and Environment Support

The *XACT* Design Implementation System is currently available for the following platforms:

- *IBM* PC/*AT*, *PS/2*, and compatibles
- *Apollo*
- *Sun-3*
- *Sun-4*

AT&T and third-party vendors have developed library and interface products compatible with a variety of design entry and simulation environments. AT&T has provided a standard interface file specification, XNF, to simplify file transfers into and out of the *XACT* Design Implementation System.

AT&T directly supports the following design environments:

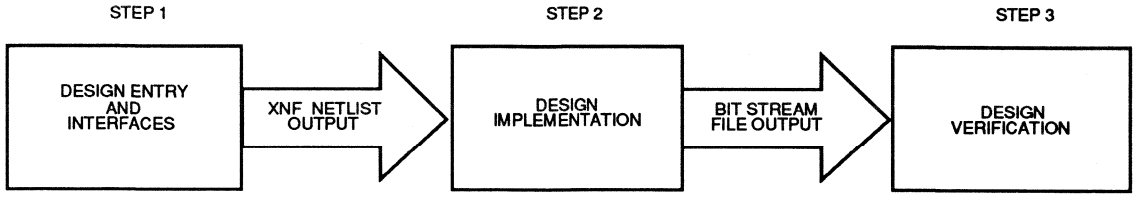
- *FutureNet DASH*
- *Viewlogic Viewdraw* and *Viewsim*
- *Mentor Graphics NETED* and *Quicksim*
- *OrCAD SDT* and *VST*
- *SILOS*

Several other environments are supported by third-party vendors.

A collection of more than 100 TTL logic macrofunctions is available for the schematic editors and is included in the appropriate packages at no charge.

The *XACT* Design Manager (XDM) simplifies the selection of command-line options with pull-down menus and on-line help text. Application programs ranging from schematic capture to APR can be accessed from the XDM, while the sequence of program commands is generated and stored for documentation prior to execution. The XMAKE command in the XDM automates the entire translation, optimization, merging, and mapping process.

AT&T Three-Step Design Flow



DS31 Interface and Library for *FutureNet DASH* Schematic Editor

DS310 *FutureNet Dash-LCA* Schematic Editor, Interface, and Library

DS343 Interfaces and Library for *Mentor Graphics NETED* Schematic Editor and *Quicksim* Simulator

DS35 Interface and Library for *OrCAD SDT* Schematic Editor

DS390 *Viewlogic Viewdraw-LCA* Schematic Editor, Interface, and Library

DS391 Interfaces and Library for *Viewlogic Viewdraw* Schematic Editor and *Viewsim* Simulator

DS501 ATT3000 *XACT* Design Implementation System

DS22 SimuCad *SILOS* Simulator (16K gates) and Interface

DS290 *Viewlogic Viewsim* Simulator and Interface

DS295 *Viewlogic Viewdraw-LCA* Schematic Editor, Interface and Library, and *Viewsim* Simulator

Figure 1. Design Flow

Product Nomenclature

System Type	Product Name	Platform
DS-Design Implementation System	3xx-Design Entry and Interfaces	AP1-*Apollo*
SC-Service Contract	5xx-Design Implementation	SN1-*Sun-3*
DX-Software Upgrade	2xx-Design Simulation	SN2-*Sun-4*
		PC1-PC

Example: ATT-DS501-SN1 = ATT3000 series Design Implementation System for *Sun-3* Workstation.

AT&T FPGA Design Solutions

Table 1. Low-Cost PC Solution

DS310-PC1	***DASH-LCA* Schematic Editor and Interface**
	■ The *DASH-LCA* Schematic Editor is a special version providing FPGA design capability ■ Macro library of more than 100 standard logic family equivalents derived from the *XACT* macro library ■ Library of logic symbols including two-input, three-input, and four-input AND, OR, and XOR gates plus storage, input/output, and clock elements ■ Additional 100 7400 TTL library elements ■ User control for flagging critical paths for the automated placement and routing program ■ Converts schematic drawings to the FPGA netlist format (XNF) output file ■ Output compatibility with the ATT-DS501 *XACT* Design Implementation System
DS501-PC1	**ATT3000 *XACT* Design Implementation System**
	■ Complete system for implementing programmable gate array designs into SRAM-based FPGA architecture ■ Accepts any combination of schematic netlists (XNF format), Boolean equations, or state-machine descriptions ■ Automated logic reduction and partitioning removes unused, disabled logic ■ Automatic placement and routing program reduces design implementation time ■ Point-to-point timing calculations for critical path analysis ■ Demo board for training and working with concept designs ■ Download cable to transfer configuration programs from PC to FPGA in target system ■ Interactive editor for design optimization
DS22-PC1	**PC-*SILOS* Simulator and Interface**
	■ PC-based simulator for FPGA design verification ■ Simulates any FPGA 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 *IBM* PC/*AT* or compatible personal computer ■ Simulates logic complexities up to 16,000 gates
Alternative Low-Cost PC Solutions: ■ If customer already has *FutureNet DASH* Schematic Editor, then purchase license for **DS31-PC1 *DASH*** Schematic Interface and Library. ■ If customer has *OrCAD SDT* Schematic Editor, then purchase license for **DS35-PC1 *OrCAD SDT*** Schematic Interface and Design Library.	

AT&T FPGA Design Solutions (continued)

Table 2. Preferred PC Solution

DS390-PC1	***Viewdraw-LCA* Editor and Interface**
	■ *Viewlogic Viewdraw-LCA* (supports AT&T FPGAs only) Schematic Editor provides easy-to-use hierarchical FPGA design capability ■ Macro library of more than 100 standard logic family equivalents derived from the *XACT* macro library ■ Library of logic symbols including two-input, three-input, and four-input AND, OR, and XOR gates plus storage, input/output, and clock elements ■ Additional 100 7400 TTL library elements ■ User control for flagging critical paths for the automated placement and routing program ■ Converts schematic drawings to the FPGA netlist format (XNF) output file ■ Output compatibility with the ATT-DS501 *XACT* Design Implementation System
DS501-PC1	**ATT3000 *XACT* Design Implementation System**
	■ Complete system for implementing programmable gate array designs into SRAM-based FPGA architecture ■ Accepts any combination of schematic netlists (XNF format), Boolean equations, or state-machine descriptions ■ Automated logic reduction and partitioning removes unused, disabled logic ■ Automatic placement and routing program reduces design implementation time ■ Point-to-point timing calculations for critical path analysis ■ Demo board for training and working with concept designs ■ Download cable to transfer configuration programs from PC to FPGA in target system ■ Interactive editor for design optimization
DS290-PC1	***Viewlogic Viewsim* Simulator and Interface**
	■ PC-based simulator for FPGA design verification ■ Simulates any FPGA design, regardless of design input format (combined logic schematics, Boolean equations, and state-machine descriptions) ■ General-purpose, event-driven logic and timing simulator ■ Input automatically generated from XNF file ■ Control and observation of any physical circuit node ■ Sophisticated waveform display ■ Integrated schematic capture and simulation (when using *Viewdraw*) ■ Simulates logic complexities of the largest AT&T FPGAs ■ Runs on *IBM* PC/*AT* or compatible personal computer
Alternative Preferred PC Solution: ■ If customer already has *Viewlogic Viewdraw* Schematic Editor and/or *Viewsim* device simulator, then purchase license for **DS391-PC1-*Viewdraw* and *Viewsim*** Interfaces and Library.	

AT&T FPGA Design Solutions (continued)

Low-Cost PC Solution Host System Requirements

- *IBM* 386 PC/*AT* or 386 *PS/2* or fully compatible computer with:
  - — *MS-DOS* Version 3
  - — One 3.5 in. or 5.25 in. high-density disk drive
  - — *IBM* compatible VGA display
  - — One parallel interface port
  - — One serial interface port
  - — *Mouse Systems, Microsoft, Logitech,* or compatible mouse
  - — Minimum 40 Mbyte hard disk
  - — 640 Kbytes of base RAM, and additional extended memory of 6 Mbytes for 3000 series

Preferred PC Solution Host System Requirements

- *IBM* 386 PC/*AT* or 386 *PS/2* or fully compatible computer with:
  - — *MS-DOS* Version 3
  - — One 3.5 in. or 5.25 in. high-density disk drive
  - — *IBM* compatible VGA display
  - — One parallel interface port
  - — One serial interface port
  - — *Mouse System* M4 or *Logitech* C7 mouse
  - — Minimum 40 Mbyte hard disk
  - — 640 Kbytes of base RAM, and additional extended memory of 6 Mbytes for 3000 series

AT&T FPGA Design Solutions (continued)

Table 3. *Sun-3* Solution

DS391-SN1	***Viewlogic* Schematic and Simulation Interface (Does not include Schematic Editor; purchase from *Viewlogic*)**
	■ Library and translator for users of the *Viewlogic Viewdraw* Schematic Editor and/or the *Viewsim* simulator ■ Macro library of more than 100 standard logic family equivalents derived from the *XACT* macro library ■ Library of logic symbols including two-input, three-input, and four-input AND, OR, and XOR gates plus storage, input/output, and clock elements ■ Additional 100 7400 TTL library elements ■ User control for flagging critical paths for the automated placement and routing program ■ Converts schematic drawings to the FPGA netlist format (XNF) output file ■ Converts XNF files to format accepted by *Viewsim* simulator for logic and timing simulation ■ Output compatibility with the ATT-DS501 *XACT* Design Implementation System
DS501-SN1	**ATT3000 *XACT* Design Implementation System**
	■ Complete system for implementing programmable gate array designs into SRAM-based FPGA architecture ■ Accepts any combination of schematic netlists (XNF format), Boolean equations, or state-machine descriptions ■ Automated logic reduction and partitioning removes unused, disabled logic ■ Automatic placement and routing program reduces design implementation time ■ Point-to-point timing calculations for critical path analysis ■ Demo board for training and working with concept designs ■ Download cable to transfer configuration programs from workstation to FPGA in target system ■ Interactive editor for design optimization

Sun-3 System Requirements

Sun-3 series/60 and above:

- *Sun* operating system *OS* 4.0
- 60 Mbytes allocated for AT&T designs
- 16 Mbytes of RAM
- Color monitor
- *X-Windows*

AT&T FPGA Design Solutions (continued)

Table 4. *Sun-4* Solution

DS391-SN2	***Viewlogic* Schematic and Simulation Interface (Does not include Schematic Editor; purchase from *Viewlogic*)**
	■ Library and translator for users of the *Viewlogic Viewdraw* Schematic Editor and/or the *Viewsim* simulator ■ Macro library of more than 100 standard logic family equivalents derived from the *XACT* macro library ■ Library of logic symbols including two-input, three-input, and four-input AND, OR, and XOR gates plus storage, input/output, and clock elements ■ Additional 100 7400 TTL library elements ■ User control for flagging critical paths for the automated placement and routing program ■ Converts schematic drawings to the FPGA netlist format (XNF) output file ■ Converts XNF files to format accepted by *Viewsim* simulator for logic and timing simulation ■ Output compatibility with the ATT-DS501 *XACT* Design Implementation System
DS501-SN2	**ATT3000 *XACT* Design Implementation System**
	■ Complete system for implementing programmable gate array designs into SRAM-based FPGA architecture ■ Accepts any combination of schematic netlists (XNF format), Boolean equations, or state-machine descriptions ■ Automated logic reduction and partitioning removes unused, disabled logic ■ Automatic placement and routing program reduces design implementation time ■ Point-to-point timing calculations for critical path analysis ■ Demo board for training and working with concept designs ■ Download cable to transfer configuration programs from workstation to FPGA in target system ■ Interactive editor for design optimization

Sun-4 System Requirements

Sun-4 and *SPARCstation* series

- *Sun* operating system *OS* 4.0
- 60 Mbytes allocated for AT&T designs
- 16 Mbytes of RAM
- Color monitor
- *X-Windows*

AT&T FPGA Design Solutions (continued)

Table 5. Alternate *Sun-4* Solution

DS295-SN2	***Viewlogic ViewDraw-LCA* and *Viewsim***
	■ *Viewlogic Viewdraw-LCA* Schematic Editor provides easy-to-use hierarchical FPGA design capability ■ Macro library of more than 100 standard logic family equivalents derived from the *XACT* macro library ■ Library of logic symbols including two-input, three-input, and four-input AND, OR, and XOR gates plus storage, input/output, and clock elements ■ Additional 100 7400 TTL library elements ■ User control for flagging critical paths for the automated placement and routing program ■ Converts schematic drawings to the FPGA netlist format (XNF) output file ■ Output compatibility with the ATT-DS501 *XACT* Design Implementation System ■ *Sun-4*-based simulator for FPGA design verification ■ Simulates any FPGA design, regardless of design input format (combined logic schematics, Boolean equations, and state-machine descriptions) ■ General-purpose, event-driven logic and timing simulator ■ Input automatically generated from XNF file ■ Control and observation of any physical circuit node ■ Sophisticated waveform display ■ Integrated schematic capture and simulation (when using *Viewdraw*) ■ Simulates logic complexities of the largest AT&T FPGAs
DS501-SN2	**ATT3000 *XACT* Design Implementation System**
	■ Complete system for implementing programmable gate array designs into SRAM-based FPGA architecture ■ Accepts any combination of schematic netlists (XNF format), Boolean equations, or state-machine descriptions ■ Automated logic reduction and partitioning removes unused, disabled logic ■ Automatic placement and routing program reduces design implementation time ■ Point-to-point timing calculations for critical path analysis ■ Demo board for training and working with concept designs ■ Download cable to transfer configuration programs from workstation to FPGA in target system ■ Interactive editor for design optimization

Sun-4 System Requirements

Sun-4 and *SPARCstation* series

- *Sun* operating system *OS* 4.0
- 60 Mbytes allocated for AT&T designs
- 16 Mbytes of RAM
- Color monitor
- *X-Windows*

AT&T FPGA Design Solutions (continued)

Table 6. *Apollo* Solution

DS343-AP1	***Mentor Graphics* Schematic and Simulation Interface**
	■ *Mentor Graphics* interfaces ■ The *IDEA* Interface Station can be used for schematic entry and simulation of programmable gate arrays ■ Full timing with post placement/routing information ■ Primitive library includes flip-flops, latches, AND, OR, XOR, NAND, and NOR gates ■ Macro library includes more than 100 standard logic elements (counters, multiplexers, registers, etc.) ■ Additional 100 7400 TTL library elements ■ User control for flagging critical paths for the automated placement and routing program ■ Output compatibility with the ATT-DS501 *XACT* Design Implementation System ■ Available on *Apollo* SR10.1 and *Mentor Graphics IDEA* V7.0
DS501-AP1	**ATT3000 *XACT* Design Implementation System**
	■ Complete system for implementing programmable gate array designs into SRAM-based FPGA architecture ■ Accepts any combination of schematic netlists (XNF format), Boolean equations, or state-machine descriptions ■ Automated logic reduction and partitioning removes unused, disabled logic ■ Automatic placement and routing program reduces design implementation time ■ Point-to-point timing calculations for critical path analysis ■ Demo board for training and working with concept designs ■ Download cable to transfer configuration programs from workstation to FPGA in target system ■ Interactive editor for design optimization

Apollo System Requirements

Apollo DN4000 series

- *Apollo* operating system SR10.1
- *Mentor Graphics* Version 7.0
- 60 Mbytes allocated for AT&T designs
- 16 Mbytes of RAM
- Color monitor
- *Domain* X11 V1

Annual Software Support Agreements

All AT&T Design Implementation Systems come with one year of free software support. Annual software support agreements can be purchased for the following:

PC Platform

- ATT-SC22-PC1 *P/C-SILOS* Simulator (16,000 gates) with *LCA* Interface
- ATT-SC290-PC1 *Viewsim* Simulator and Interface
- ATT-SC31-PC1 *FutureNet DASH* Schematic Interface and Library
- ATT-SC310-PC1 *DASH-LCA* Schematic Editor and Interface
- ATT-SC35-PC1 *OrCAD SDT* Schematic Editor and Interface
- ATT-SC390-PC1 *Viewdraw-LCA* Schematic Editor and Interface
- ATT-SC391-PC1 *Viewdraw* and *Viewsim* Interfaces
- ATT-SC501-PC1 3000 Series *XACT* Design Implementation System for *IBM* PC/*AT*

Apollo Platform

- ATT-SC343-AP1 *Mentor Graphics* Certified Schematic and Simulation Interface
- ATT-SC501-AP1 3000 Series *XACT* Design Implementation System for *Apollo* Workstation

Sun-3 Platform

- ATT-SC391-SN1 *Viewdraw* and *Viewsim* Interfaces and Library on *Sun-3*
- ATT-SC501-SN1 3000 Series *XACT* Design Implementation System for *Sun-3* Workstation

Sun-4 Platform

- ATT-SC391-SN2 *Viewdraw* and *Viewsim* Interfaces and Library on *Sun-4*
- ATT-SC501-SN2 3000 Series *XACT* Design Implementation System for *Sun-4* workstation
- ATT-DS295-SN2 *Viewdraw-LCA* Schematic Editor, *Viewsim* Simulator, and Interfaces

AT&T Microelectronics

Development System Hardware Requirements

AT&T provides an integrated development system for design and implementation of FPGA devices. The *XACT* Development System operates on a PC/*AT* or *PS/2* model 60 or 80, *Apollo*, *Sun-3*, *Sun-4*, and *DECstation* 3100 and provides a range of support features. This provides the user with an effective, convenient, low-risk method of logic-design entry, simulation, FPGA generation, and verification for single-chip logic designs of up to 9000 gates. In addition, several popular PC and workstation CAE vendors have developed and offer design-entry and simulation programs compatible with the *XACT* Development System.

PC Requirements

The recommended PC-system configuration needed to run the ATT3000 Series with the *XACT* Development System consists of:

- A 20 MHz 386 PC/*AT* or *PS/2* model 60 or better
- 40 Mbyte hard disk drive plus a 1.2 Mbyte high-density floppy disk drive
- Two RS-232-C serial ports
- One parallel port
- EGA or VGA graphics display
- Mouse
- *MS-DOS* version 3.0 or higher
- *IBM*-compatible BIOS and keyboard
- A math co-processor can enhance performance of APR by 10% or 20%
- Extended memory as follows:

FPGA Gates	Total Memory Required for *XACT* 3.0
2000 or less	2.50 Mbytes
3000	3.25 Mbytes
4200	4.00 Mbytes
6400	5.25 Mbytes
9000	6.50 Mbytes

To ensure integrity, all AT&T software is tested on *IBM* systems and several compatible systems. FPGA development software includes some of the first DOS-based programs to make extensive use of the protected mode of the processor. This has exposed protected mode *IBM*-incompatibilities of some clones, usually in the BIOS or keyboard controller. AT&T software includes system exercises called PMTEST and PMINFO to help test *IBM* compatibility and measure relative performance. Note that this amount of memory must be available to *XACT*, i.e., it does not include the memory used for other resident programs.

Note also that the Compaq 386 has only 640 Kbytes of its first megabyte available to any user.

PC I/O Ports

The *FPGA* 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 AT&T software to execute but is virtually transparent, and the port can be used simultaneously for a parallel printer or the AT&T download cable. Several printer types are supported for text or graphic hard copy. Serial COM ports are used for a mouse, the *XACTOR* in-circuit design verifier and the configuration PROM programmer.

PC Mouse

The AT&T Development System programs are compatible with several varieties of mice offered for the PC. These include *Mouse Systems* PC mouse (no device driver required), *Microsoft* (serial or parallel), *LogiTech* C7, and the *FutureNet* mouse. The AT&T software supports any mouse directly that emulates the PC mouse or has a device driver that provides *Microsoft* compatibility and defines the PC COM port.

Please note, however, that only the Mouse System M4 and the *LogiTech* C7 will work with the *Viewogic* software, *Viewdraw-LCA,* and *Viewsim.*

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 follow:

```
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 AT&T 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 AT&T installation instructions and PC manuals for additional information.

Workstation Requirements

The workstation system requirements needed to run AT&T software include the following:

Apollo Requirements

DN4000 Series

- *Apollo* Operating System SR10.1
- *Mentor Graphics* Version 7.0
- 60 Mbytes allocated for AT&T designs
- 16 Mbytes of RAM
- Color monitor
- *Domain* XII VI

Sun-3 Requirements

Series 960 and above

- *Sun* Operating System OS4.0
- 60 Mbytes allocated for AT&T designs
- 16 Mbytes of RAM
- Color monitor
- *X-Windows*

Sun-4 Requirements

Sun-4 & *SPARCstation* Series

- *Sun* Operating System OS4.0
- 60 Mbytes allocated for AT&T designs
- 16 Mbytes of RAM
- Color monitor
- *X-Windows*

Workstation Requirements (continued)

DECStation Requirements

DECstation 3100 Series

- Worksystem V2.2
- 60 Mbytes allocated for AT&T designs
- 16 Mbytes of RAM
- Color monitor

Product Brief
June 1992

AT&T Microelectronics

ATT-DS22 P/C-*SILOS* Simulators

Features

- PC-based simulator for FPGA design verification
- Simulates any FPGA design, regardless of design input format (combined logic schematics, Boolean equations, and state-machine descriptions)
- General-purpose, event-driven logic and timing simulator
- Input automatically generated from XNF file
- Control and observation of any physical circuit node
- Multiple-file input for vectors and commands
- Interactive or batch-mode operation
- Output available in printed or tabular formats
- Simulates logic complexities up to 16,000 gates
- Runs on a PC/*AT* or compatible personal computer

Description

P/C-SILOS is a powerful PC-based simulator that provides event-driven logic and timing simulation of FPGA 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 FPGA 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.

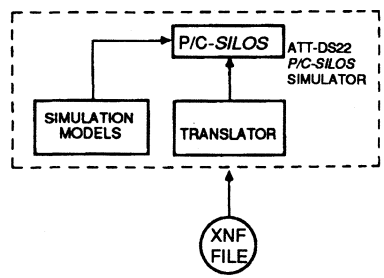


Figure 1. P/C-*SILOS* Interface

Product Brief
June 1992

AT&T Microelectronics

ATT-DS290

Viewlogic Viewsim Simulator

Features

- PC-based simulator for FPGA design verification
- Simulates any FPGA design, regardless of design input format (combined logic schematics, Boolean equations, and state machine descriptions)
- General-purpose event-driven logic and timing simulator
- Input automatically generated from XNF file
- Control and observation of any physical circuit node
- Sophisticated waveform display
- Schematic capture and simulation integrated in one environment (when using the *Viewdraw*)
- Simulates logic complexities of the largest FPGAs
- Runs on a *PC/XT* or compatible personal computer

General

Viewsim is a powerful PC-based simulator that provides event-driven logic and timing simulation of FPGA designs. Simulation is particularly useful for testing designs or design segments as well as for verifying critical timing over worst-case power supply, temperature, and process conditions.

Simulation is useful in several stages of the design cycle. After design entry, simulation may be used to debug logic in an unplaced and unrouted design. This saves design time because logic errors can be detected and corrected prior to final placement and routing.

After a circuit has been placed, routed, and then fully debugged using in-circuit emulation, timing simulation can be performed. This enables the user to select the correct FPGA speed grade for a particular application.

Network inputs for FPGA designs are automatically created by the XNF2WIR 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.

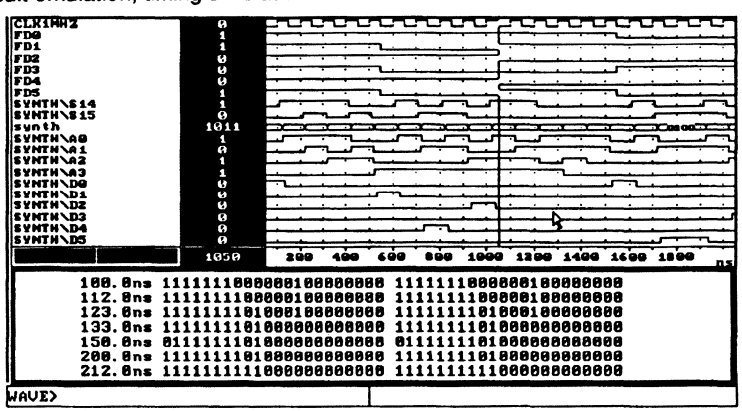


Figure 1. *Viewsim* Simulator

Product Brief
June 1992

AT&T Microelectronics

ATT-DS31

DASH Schematic Interface and Library

Features

- Library and translator for users of the *DASH* Schematic Designer
- Macro library of over 100 standard logic family equivalents derived from the *XACT* macro library
- Library of logic symbols including all two-input, three-input, and four-input AND, OR and XOR gates plus storage, input/output, and clock elements
- Additional one hundred 7400 MSI library elements
- User control for flagging critical paths for the automated placement and routing program
- Converts schematic drawings to a netlist format output file
- Output compatibility with ATT-DS501 *XACT* Design Implementation System
- Runs on PC/*AT* or compatible personal computers, *Sun-3 and Sun-4*

Description

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

The *DASH* Schematic Designer Library provides the symbol library and conversion utility to permit designers to enter FPGA designs with the *DASH* Schematic Designer. The AT&T library provides the logic, I/O, and macro symbols to be used in the schematic. An AT&T conversion utility converts the schematic into a netlist output file.

Once partitioned, the design may be placed and routed with the ATT-DS501 *XACT* Design Implementation System. The AT&T 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.

AT&T provides ongoing support for users of the *DASH* Schematic Designer Library. For the first year, software updates are included. After that, the user may purchase the ATT-SC31 Annual Support Agreement to continue to receive the latest software releases.

Product Brief
June 1992

AT&T Microelectronics

ATT-DS310

DASH-FPGA Schematic Editor, Interface Library

Features

- *FutureNet DASH*-FPGA Schematic Editor provides easy-to-use hierarchical FPGA design capability
- Macro library of over 100 standard logic family equivalents derived from the *XACT* macro library
- Library of logic symbols including all two-input, three-input, and four-input AND, OR, and XOR gates plus storage, input/output, and clock elements
- Additional 100 7400 TTL library elements
- User control for flagging critical paths for the automated placement and routing program
- Converts schematic drawings to a netlist format output file
- Output compatibility with ATT-DS501 *XACT* Design Implementation System
- Runs on PC/*AT* or compatible personal computers

Description

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

DASH-FPGA supports unlimited levels of hierarchy. The *DASH*-FPGA Schematic Library provides the symbol library and conversion utility to permit designers to enter FPGA designs with the *DASH*-FPGA Schematic Editor. The AT&T library provides the logic, I/O, and macro symbols to be used in the schematic. An AT&T conversion utility converts the schematic into an netlist output file.

Once partitioned, the design may be placed and routed with the ATT-DS501 *XACT* Design Implementation System. The AT&T 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.

Product Brief
June 1992

AT&T Microelectronics

ATT-DS343

Mentor Graphics Schematic and Simulation Interfaces and Library

Features

- *Mentor Graphics* certified interfaces
- The *IDEA* interface station can be used for schematic entry and simulation of programmable-gate-array designs
- Full timing simulation with post placement/routing information
- Primitive library includes flip-flops, latches, AND, OR, XOR, NAND, NOR gates
- Macro library includes over 100 standard logic elements (counters, multiplexers, registers, etc.)
- Additional one hundred 7400 MSI library elements included at no charge
- Netlist format (XNF) output is compatible with ATT-DS501 Design Implementation System
- Available on *Apollo* SR10.1 and *Mentor Graphics IDEA* V7.0

Description

Schematic entry and automatic partitioning of FPGA designs shorten logic reduction and product-development times. Complex designs can be specified schematically and quickly implemented for full timing simulation and in-circuit design verification.

The ATT-DS343 package provides the symbol library and conversion utility to permit designers to enter FPGA designs with the *Mentor Graphics NETED* Schematic Editor.

The AT&T library provides the logic, I/O, macro, and TTL symbols to be used in the schematic. An AT&T conversion utility converts the schematic into a netlist output file.

Once partitioned, the design may be placed and routed with the *Apollo*-based ATT-DS501 *XACT* Design Implementation System. The 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.

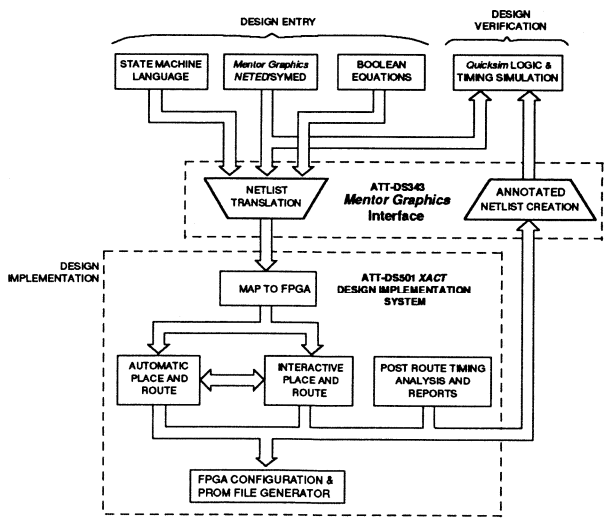


Figure 1. Design Flow

Product Brief
June 1992

AT&T Microelectronics

ATT-DS35

OrCAD SDT Schematic Entry Interface and Design Library

Features

- Library and translator for users of the *OrCAD SDT* Schematic Editor
- Library of over 100 standard logic macros
- Library of logic symbol primitives includes AND, OR, NAND, NOR, and XOR gates plus storage, input/output and clock elements
- Additional one hundred 7400 MSI library elements included at no charge
- User control for flagging critical paths for the automated placement and routing program
- Converts schematic drawings to a netlist format output file
- Output compatibility with *XACT* Design Implementation System
- Runs on a PC/*AT* or compatible personal computer

Description

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

The *OrCAD* Schematic Entry Interface provides the symbol library and conversion utility to permit designers to enter FPGA designs with the *SDT* Schematic Editor. The library provides the logic, I/O, and macro symbols to be used in the schematic. A conversion utility converts the schematic into a netlist output file.

Once partitioned, the design may be placed and routed with the PC-based *XACT* Automated Design Implementation Program. The 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.

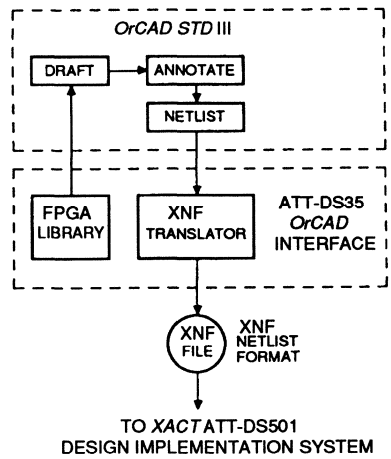


Figure 1. Schematic Entry Interface

Product Brief
June 1992

AT&T Microelectronics

ATT-DS371

FPGA Logic Synthesis Tools

Features

AT&T State-Machine Language:

- Combines the simplicity of popular *PLD* languages with the power and expressiveness of *VHDL*
- Allows complex state-machine implementation using a simplified *VHDL*-like language
- Supports multiple state machines
- External-file-reference capability and syntax-alias capability provide versatile environment for state-machine entry

AT&T State-Machine Compiler:

- Automatically synthesizes the AT&T state-machine language, and generates an optimized netlist format file
- Supports both automatic state-encoding algorithms optimized for the FPGA architectures and user-defined state-bit assignments
- User can optimize for speed or area

AT&T *PLD* Synthesis:

- Automatically synthesizes *PALASM2* format and generates an optimized AT&T netlist

Description

AT&T Logic Synthesis Tools allow designers to describe designs in easily understood and modified forms such as Boolean equations or state-machine description language, in addition to schematic diagrams. AT&T Logic Synthesis Tools efficiently map technology-independent logic descriptions such as state-machine descriptions into the AT&T architecture. The implementation can be optimized for either speed or the amount of logic used.

The figure on page 4-3 illustrates the design flow from a design schematic with some glue logic, a TTL macro, *PLD* symbol, or state-machine description.

The *PLD* design is entered using *ABEL, CUPL, Log/IC, PGADesigner,* or directly through *PALASM2* Boolean Equations. The state-machine design is entered using the AT&T state-machine language, which is an intuitive and powerful language for state-machine entry. An example is given below.

The multiple-mode design is combined into the FPGA architecture in the following steps. The schematic is converted by the netlist translator to the AT&T netlist format file. Then, the *PALASM2* and AT&T state-machine language are translated and optimized, resulting in a second netlist file. The design files are then merged and partitioned into CLBs and IOBs.

PLD synthesis and netlist format optimization will continue to be included in the DS501 package until the DS371 becomes available.

Product Brief
June 1992

AT&T Microelectronics

ATT-DS390

Viewdraw-FPGA Schematic Editor, Interface and Library

Features

- *Viewogic Viewdraw*-FPGA Schematic Editor provides easy-to-use hierarchical FPGA design capability
- Macro library of over 100 standard logic family equivalents derived from the *XACT* macro library
- Library of logic symbols including all two-input, three-input, and four-input AND, OR, and XOR gates plus storage, input/output and clock elements
- Additional one hundred 7400 MSI library elements included at no charge.
- User control for flagging critical paths for the automated placement and routing
- Converts schematic drawings to an AT&T netlist format output file
- Output compatibility with ATT-DS501 *XACT* Design Implementation System
- Runs on PC-compatible personal computers

Description

Schematic entry and automatic partitioning of FPGA designs shorten logic-reduction and product-development times. Complex designs can be specified schematically and then quickly implemented for in-circuit design verification.

Viewdraw-FPGA supports unlimited levels of hierarchy. The *Viewdraw*-FPGA library provides the symbol library and conversion utility to permit designers to enter FPGA designs with the *Viewdraw* Schematic Editor. The AT&T library provides the logic, I/O, and macro symbols to be used in the schematic. An AT&T conversion utility converts the schematic into a netlist output file.

Once partitioned, the design may be placed and routed with the PC- or workstation-based ATT-DS501 *XACT* Design Implementation System. The 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.

Product Brief
June 1992

AT&T Microelectronics

ATT-DS391

Viewlogic Viewdraw and *Viewsim* Interfaces and Library

Features

- Library and translator for users of the *Viewlogic Viewdraw* Schematic Editor and *Viewsim* Simulator
- Macro library of over 100 standard logic family equivalents derived from the *XACT* macro library
- Library of logic symbols including all 2-input, 3-input, and 4-input AND, OR, and XOR gates plus storage, input/output and clock elements
- Additional 100 7400 TTL library elements
- User control for flagging critical paths for the automated placement and routing
- Converts schematic drawings to an AT&T netlist format output file
- Converts XNF files to format accepted by *Viewsim* Simulator for logic and timing simulation
- Output compatibility with ATT-DS501 *XACT* Design Implementation System
- Runs on PC/*AT*-compatible personal computers, *Sun-3, Sun-4* and *DECstation* 3100

Description

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

The *Viewdraw* library provides the symbol library and conversion utility to permit designers to enter FPGA designs with the *Viewdraw* Schematic Designer. The AT&T library provides the logic, I/O, and macro symbols to be used in the schematic. A conversion utility converts the schematic into an AT&T output file.

Once partitioned, the design may be placed and routed with the PC- or workstation-based ATT-DS501 *XACT* Design Implementation System. The AT&T symbol library includes symbols to flag critical data and clock signals which the automatic placement and routing program uses to prioritize those signals for minimum delay.

With the *Viewsim* Simulation Interface, designers can use the *Viewlogic* simulation environment to perform postlayout simulation. All postlayout timing information, including pin-to-pin delays, is back annotated into the *Viewlogic* environment for full timing simulation.

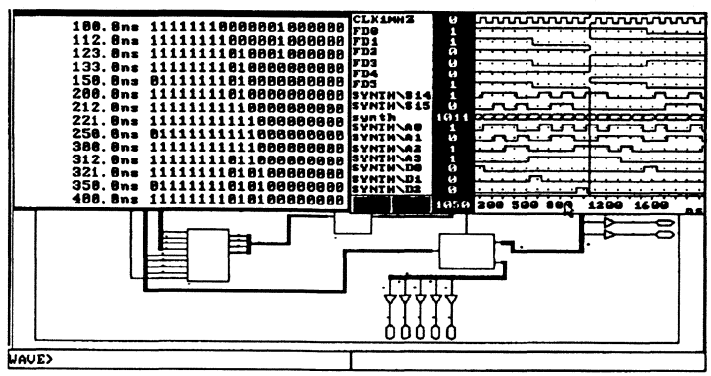


Figure 1. ATT-DS391 Interface

Product Brief
June 1992

AT&T Microelectronics

ATT-DS501

XACT Design Implementation System

Features

- Complete system for implementing programmable gate array designs into FPGA architecture
- Accepts AT&T netlists created from schematic editors, Boolean equations, or state-machine descriptions
- Automated logic reduction and partitioning removes unused, disabled logic
- Automated placement and routing of logic minimizes design cycle time
- Interactive editor for design optimization
- Point-to-point timing calculations for critical-path analysis
- Demo board for training and trying out concept designs
- Download cable to transfer configuration programs from PC or workstation to FPGA in target system
- Available on PC/*AT, Apollo, Sun-3, Sun-4,* and *DECstation* computers

Design Implementation Process

Designers often describe portions of their design, such as counters and glue logic, with schematics, and other portions of the design, such as decoders, with Boolean equations. Automatic placement and routing software permits designers to merge multiple modes of design entry into a single design.

The design flow can begin from a design schematic with some glue logic, a 7400-MSI macro, and a *PLD* symbol. The design files are merged and partitioned into CLBs and IOBs.

The automatic placement and routing software is very flexible. Routing resources can be specified to eliminate clock skews and minimize routing delays for optical paths.

The *XACT* Design Editor (XDE) can then be used to modify design placement and routing, when required to meet critical timing requirements.

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

Interactive point-to-point timing delay calculation is provided for timing analysis and critical-path determination. This ability enables the user to quickly identify and correct timing problems while the design is in progress.

A download cable is included with the DS501. It is useful for transferring configuration programs serially from the PC or workstation to an FPGA device installed in a system or on the demo board. During product development and verification, this capability can be used to save the time required to write a modified configuration program into an EPROM.

Technical Note
June 1992

AT&T Microelectronics

FPGA Development System

Overview

The AT&T FPGA Development System has three major components:

- Design entry:
  - — Describing a logic function in electronic form
- Design implementation:
  - — Converting the design to the FPGA technology architecture
- Hardware circuit verification:
  - — Ensuring that the design functions correctly

The AT&T logic libraries and XNF interface products support design entry with popular schematic logic drawing systems supplied by multiple vendors, allowing easy entry to the AT&T 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 FPGA architecture. Subsequent programs perform automatic placement and routing (APR) to complete the FPGA array design.

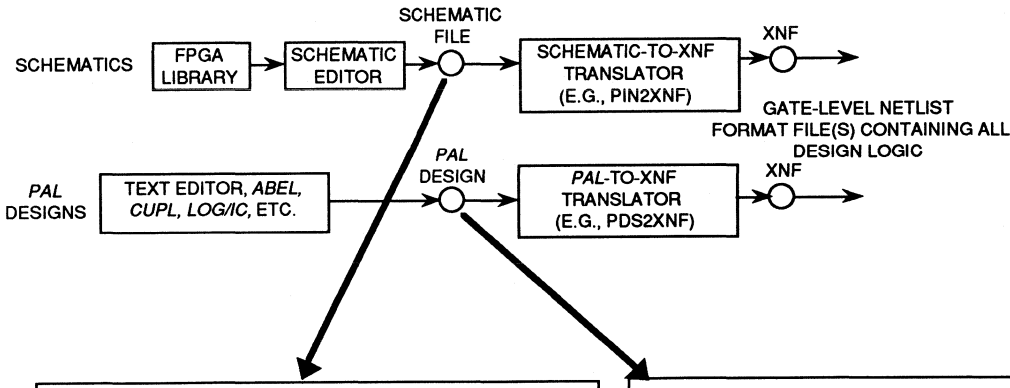
While completely automated implementation is desirable for low-complexity designs, the designer often prefers an interactive process, especially in high-performance, high-complexity designs. This interactive editing can range from rerouting a few previously autorouted nets, to prerouting critical nets or preplacing CLBs prior to design completion using APR, to more extensive control over logic partitioning and placement into CLBs.

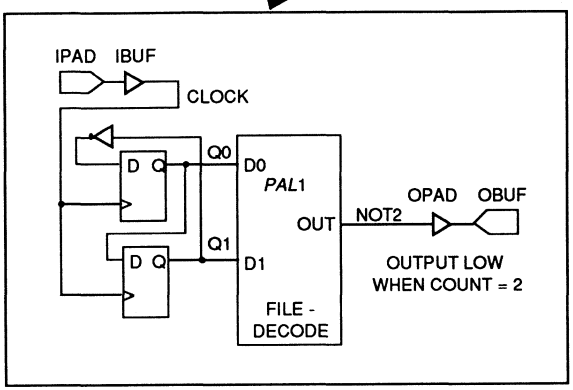
The automated design implementation software gives the designer an option to have direct control over specific logic mapped into CLBs (partitioning) for better distribution of logic signal routing through the FPGA array. The *XACT* Graphic Editor is extremely versatile, ranging from design entry to CLB and signal routing manipulations. This combination of automated and interactive design editing capability is an important feature provided by AT&T.

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 FPGA array according to the designer's specification.

Another important feature of the AT&T 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 FPGAs resident on a PC or engineering workstation.

Design Entry and XNF Translation





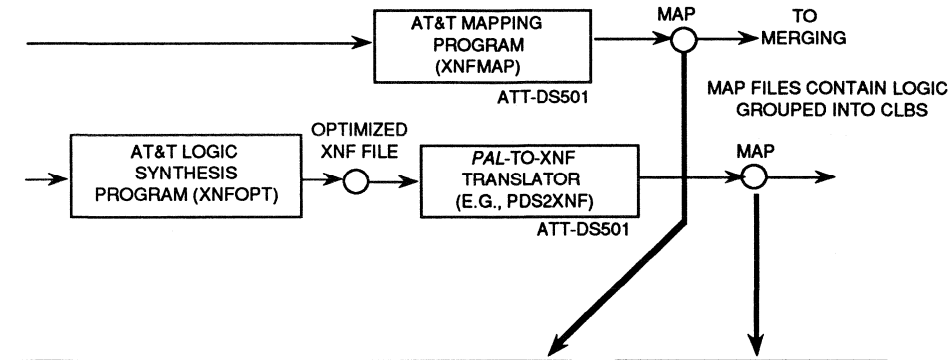
Schematics can include any number of *PAL* devices created with Boolean equations and/or state machines.

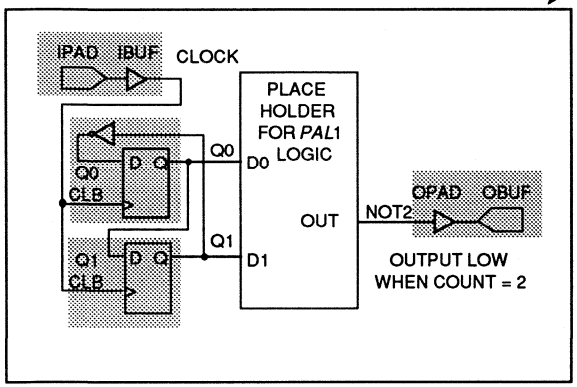
```
Title:          DECODE.PDS
Author
Company         AT&T
Date
Chip            DECODE PAL10HB
; Input Pins
                d0  d1
; Output Pins
                out
;
; Define counter states
;
STRING ZERO             '/D1 * /D0'
STRING ONE              '/D1 * D0'
STRING TWO              'D1 * /D0'
STRING THREE            'D1 * D0'

EQUATIONS
OUT = ZERO + ONE + THREE
```

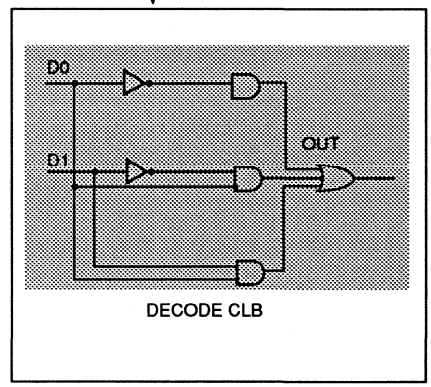
Very simple *PAL* design included in schematic at left with *PAL*1 symbol.

Optimization and Mapping



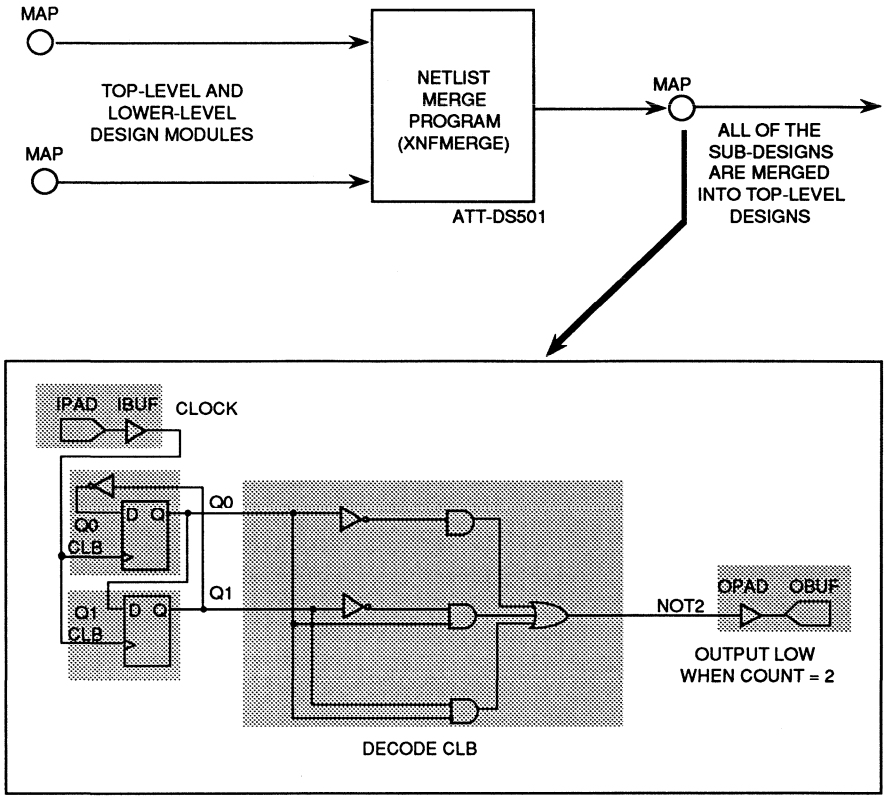


This figure is graphical representation of the top-level MAP file. Unused logic (if any) has been deleted and the remaining logic has been grouped (mapped) into configurable logic blocks and I/O blocks.



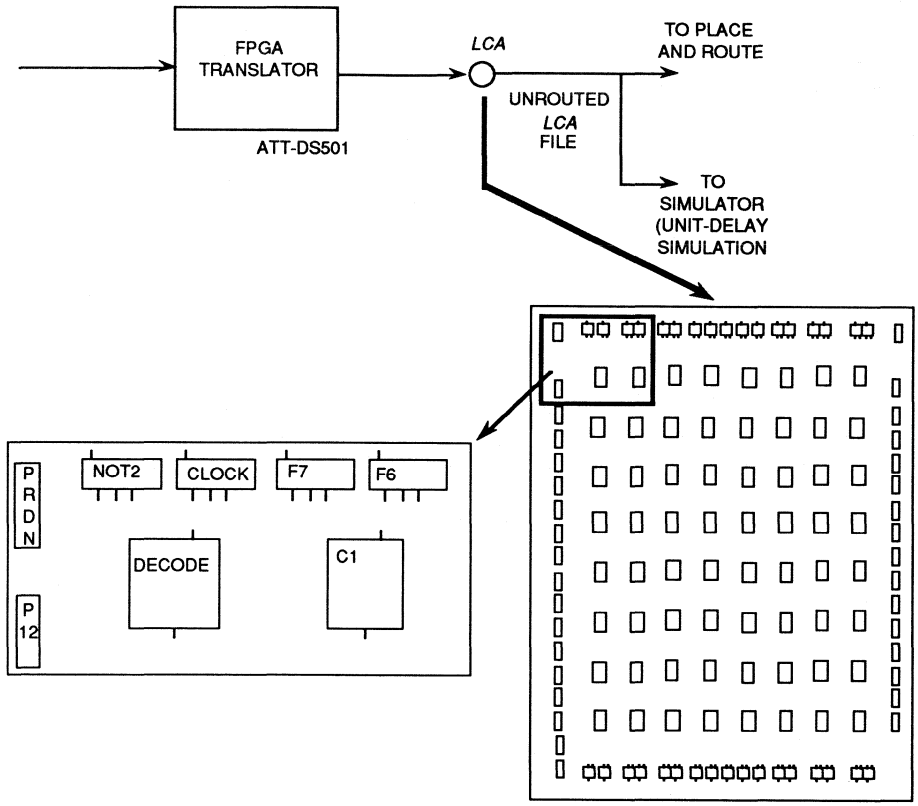
This figure is a graphical representation of the CLB containing the PAL logic. (In a typical *PAL* design, several CLBs would be used.)

Merging



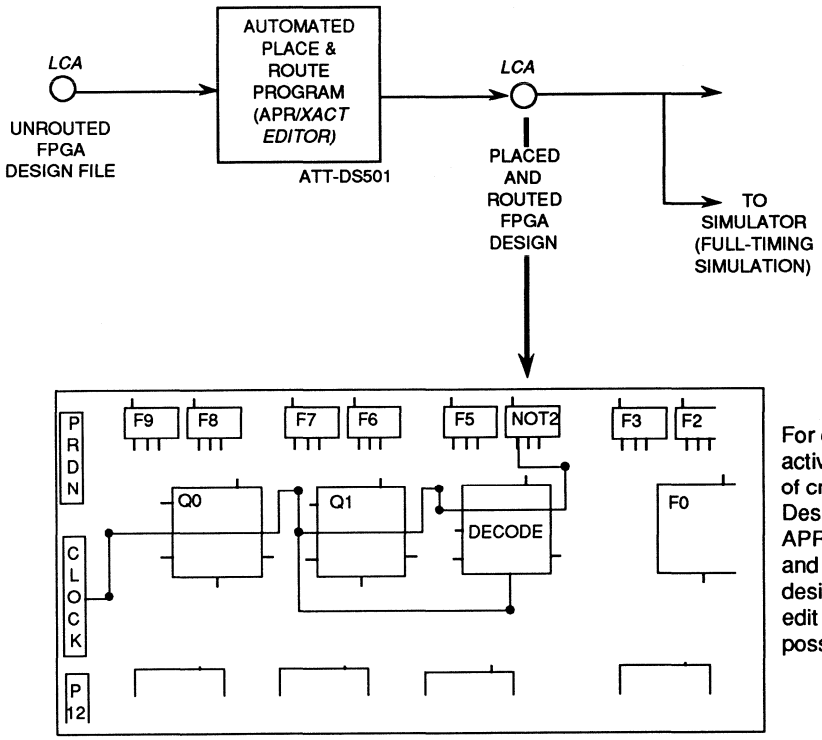
The merged design contains the CLBs and IOBs for the entire design.

Translating to FPGA



Initially, before place and route, the FPGA design is unrouted and the configurable logic and I/O blocks are put in random locations.

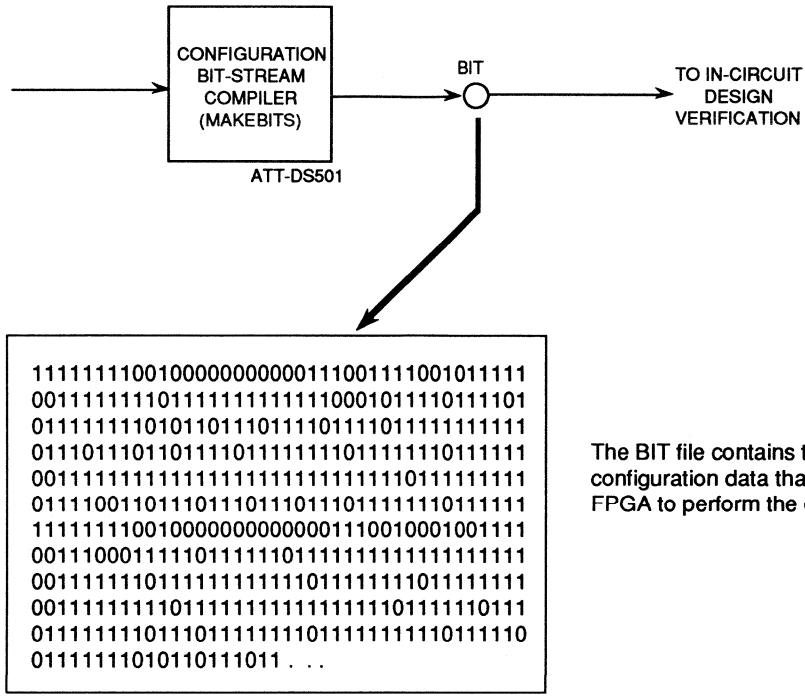
Placing and Routing FPGA



A simple placed-and-routed design (close-up of upper left corner).

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

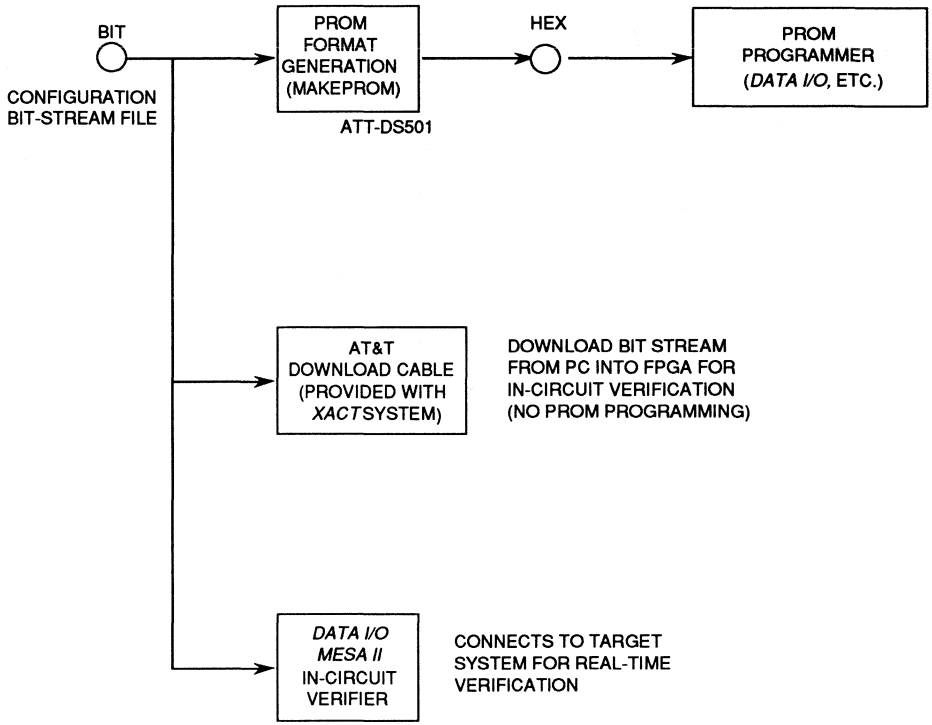
Bit-Stream Generation



The BIT file contains the binary configuration data that programs the FPGA to perform the design function.

FPGA Configuration Bit Stream

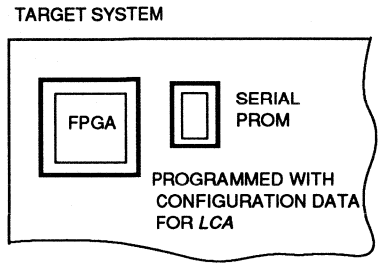
Real-Time, In-Circuit Verification



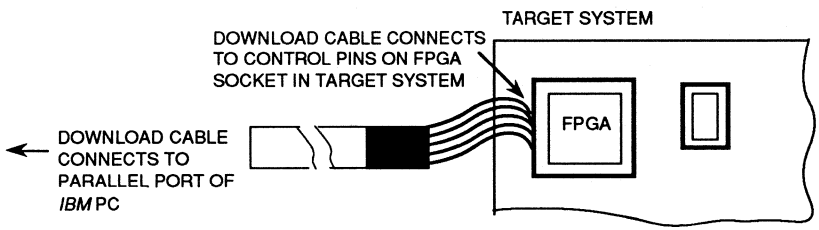
Real-Time, In-Circuit Verification (continued)

In-circuit verification allows designers to immediately see how FPGA designs function.

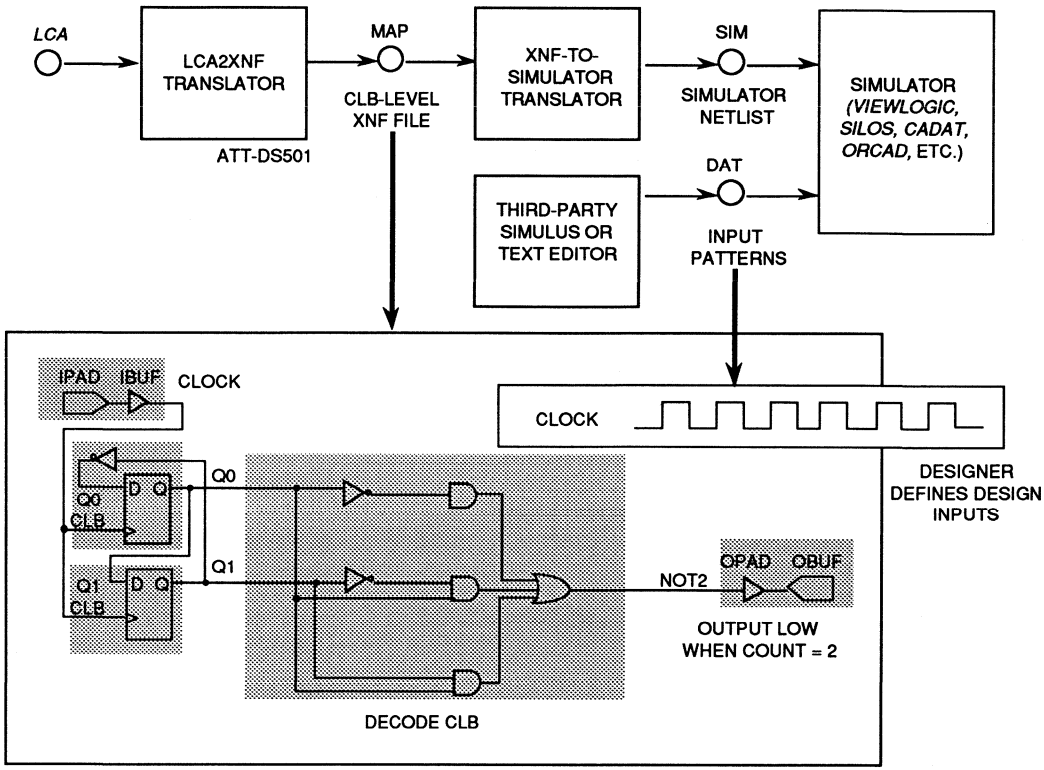
Program a PROM



Use the Download Cable



Simulation



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

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

Design Expectations and User Involvement

First read the user tutorial in the AT&T user manual and install the software according to the installation instructions provided. The design manager simplifies entering command line instructions and documents how various files are combined to create the FPGA design. If you are using *FutureNet DASH* for schematic entry, follow the example illustrated in the *FutureNet DASH Tutorial* Application Note. Then you are ready to start your own design following the basic design flow as illustrated.

- Small designs, less than 80% utilization
  - — Appropriate for 3020 designs
  - — Basic design flow should be used first, then more complex methodology if necessary.
- Larger designs with buses, high I/O pin utilization, and/or incorporating PALs
  - — Use the suggestions in the *Introduction to Hierarchical Design and Merging* Application Note.
- Very large and/or dense designs
  - — Study the FPGA architecture carefully. It may be necessary to read *Advanced Design Methodology* Application Note.

Design Flow

The AT&T Development System uses several common file types (XNF, MAP, and *LCA*) to describe the field-programmable gate array as shown in the basic design flow procedures, next page. The design sequence to generate these files depends on the logic complexity (density) and the type (high-fanout buses versus state-machine logic).

Design entry can be from various methods ranging from schematics to Boolean equations, to state-machine language. There are several interface programs which convert information in these entry systems to a common AT&T format called XNF. From this XNF file (or a merged set of XNF files), a design can be processed into the FPGA utilizing program options/features and various sequences of program execution.

In the standard design implementation flow, designers can utilize the automated translation, partitioning, placement, and routing software to generate a FPGA design that can be loaded into the target system during design verification.

The XNF2LCA sequence of programs map the logic defined in the XNF file into specific CLB and IOB configurations described in the FPGA design file. The translation process is divided into two steps (XNFMAP and MAP2LCA) to provide additional flexibility. The APR program automatically places and routes these CLBs and IOBs in the design file.

Frequently, there are design changes made during system debugging. An important feature of the AT&T development system is the capability to incorporate these changes, utilizing several advanced program options. Small changes can be made to the schematic and then automatically processed into the existing design with minimal impact, using a unique incremental design feature.

In more complex designs, the designer may prefer to partition and process portions of the logic separately for improved routability, but with less densely packed CLBs. This design methodology of independent mapping followed by merging and APR is an important feature, and is discussed in the *Advanced Design Methodology* Application Note. This designer-directed grouping is performed by using a special symbol in the schematic to indicate which logic functions or functional blocks are to be placed into individual CLBs and not combined with other logic functions.

Design verification is performed by simulation or hardware in-circuit evaluation. An XNF file with post route timing is extracted from the design file with the LCA2XNF program. Translators to several simulators allow the designer to simulate the time-dependent behavior of the programmed FPGA in response to various input stimuli. In addition, there is the capability to perform in-circuit testing of the FPGA design in the target system. The MAKEBITS program generates a bit-stream file that describes the unique logic programming configuration that is downloaded to the FPGA using a cable provided with the AT&T product. *DATA I/O* offers the *MESA II* in-circuit verifier that allows the designer to both download the bit stream program and retrieve a snapshot of the I/O and CLB flip-flop logic state internal to the FPGA, with stationary inputs.

Basic Design Flow Procedure

Design Entry

Reference

Step 1. Draw the schematic(s).
Use hierarchy unless schematic is only one drawing sheet.

Schematic Editor Documentation

Step 2. Convert the schematic(s) to XNF format.
XNF is a standard format for logic input into the *Xilinx* development system, regardless of source.

For *FutureNet:*	Run DCM, PINC on each schematic and PIN2XNF on top-level schematics.
For *OrCAD:*	Run ANNOTATE (use /U and /D) and OR2XNF (use /A).
For *Daisy:*	Run DANCE, DRINK, and DRW2XNF on personal logician.
For Mentor Graphics:	Run *LCA*_EXPAND and EREL2XNF

User Manual, *DASH* Interface

Design Implementation

Step 3. Convert XNF file to an *LCA* design file.
(*LCA* design file contains definitions of I/O and logic blocks with interconnection details for routed designs.)

- *LCA* part type (e.g., 3020PC68) must be specified either in the schematic or on the command line when invoking XNF2LCA (See specific schematic document editor documentation).

Schematic Editor Documentation

- XNF2LCA converts the XNF file to the *LCA* format.
  - — Converts logic gates and flip-flops to logic blocks
  - — Eliminates unused or unnecessary logic
  - — Conversion is done in two steps: XNFMAP and MAP2LCA
  - — Creates files (MAP, *LCA, CRF, AKA)*

Reference Manual, Chapter 6

Step 4. Place and Route the *LCA* design file.
Output *LCA* file from XNF2LCA is randomly placed and routed.

- Automatic place and route with APR
  - — Invoked with APR design_name
  - — Recommended for all designs as a first pass
  - — Highest probability of success for designs (100% routing) using 65—70% of resources and limited use of buses

Reference Manual, Chapter 4

- Interactive place and route with *XACT*
  - — Interactive graphics process
  - — Can be used in conjunction with APR
  - — Invoked with *XACT,* EDITLCA program
  - — User operates at device architecture level

Reference Manual, Chapter 3

Design Verification

Step 5. Verify design.

- Make bit stream (MAKEBITS), then download from XAC*T*
- Functional simulation can be done on XNF file(s) before implementation
  - — Create simulator files: LCA2XNF timing annotated XNF file
- *DATA I/O MESA* in-circuit verifier
  - — Readback and display internal flip-flops or program configuration

Reference Manual, Chapter 7

Note: For iterative design, APR can be used with -G option to add logic while preserving the existing placement and routing.

General Guidelines & Recommendations

- Draw schematics hierarchically unless the logic is only one drawing sheet.
- Do not lock I/O pin locations:
  — On the first design
  — On complex designs using a large percentage of I/O and logic resources.
- Study the FPGA architecture to understand I/O and logic blocks.
- Specify the AT&T FPGA device on the schematic. This avoids errors or false results from assumptions made by the system. It can always be changed later in the process.
- Be careful with buses:
  — If there are three or fewer sources internally, use multiplexer instead of TBUF.
  — If TBUFs are required, reduce utilization of the I/O and logic blocks to make routing resources available.

Development System Architecture

Design entry is supported in a variety of schematic capture design systems resident on *IBM* or compatible personal computers ranging from PC/*AT* to *PS/2*. These design systems support the AT&T logic library and interface to the AT&T design implementation system through several netlist transfer programs. The bundling of software programs in the ATT-DS501 accepts design entry from a variety of schematic and state-machine formats as shown.

The logic library and interface programs are supported by AT&T and a range of third-party vendors. The standard interface file specification (XNF format) simplifies file transfer into the AT&T development system. AT&T directly supports the *FutureNet DASH* or *DASH-LCA* and *OrCAD SDT* schematic capture products. An additional collection of TTL logic macrofunctions is available with the *FutureNet DASH* and *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 design manager 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.

File Structure

XNF Files

The AT&T Netlist File (XNF) is the highest-level file type used to communicate between various logic entry systems and the AT&T development system. It is an ASCII text file that can be read using any text editor.

All schematic editors create some type of file (usually unique) that describes the design. AT&T has adopted the XNF format as a common interface to all outside file formats. All schematic editors which support AT&T FPGAs designs have a schematic-to-XNF translator. The XNF file created by this translator has a one-to-one correspondence with the original schematic.

An XNF file is usually created by simple translator programs. This XNF file can have references to other XNF or MAP files which contain additional logic for the design. This logic may have been designed with the same schematic editor, with a different schematic editor, with Boolean equations, state-machine expressions, or from separately compiled macros or submodules.

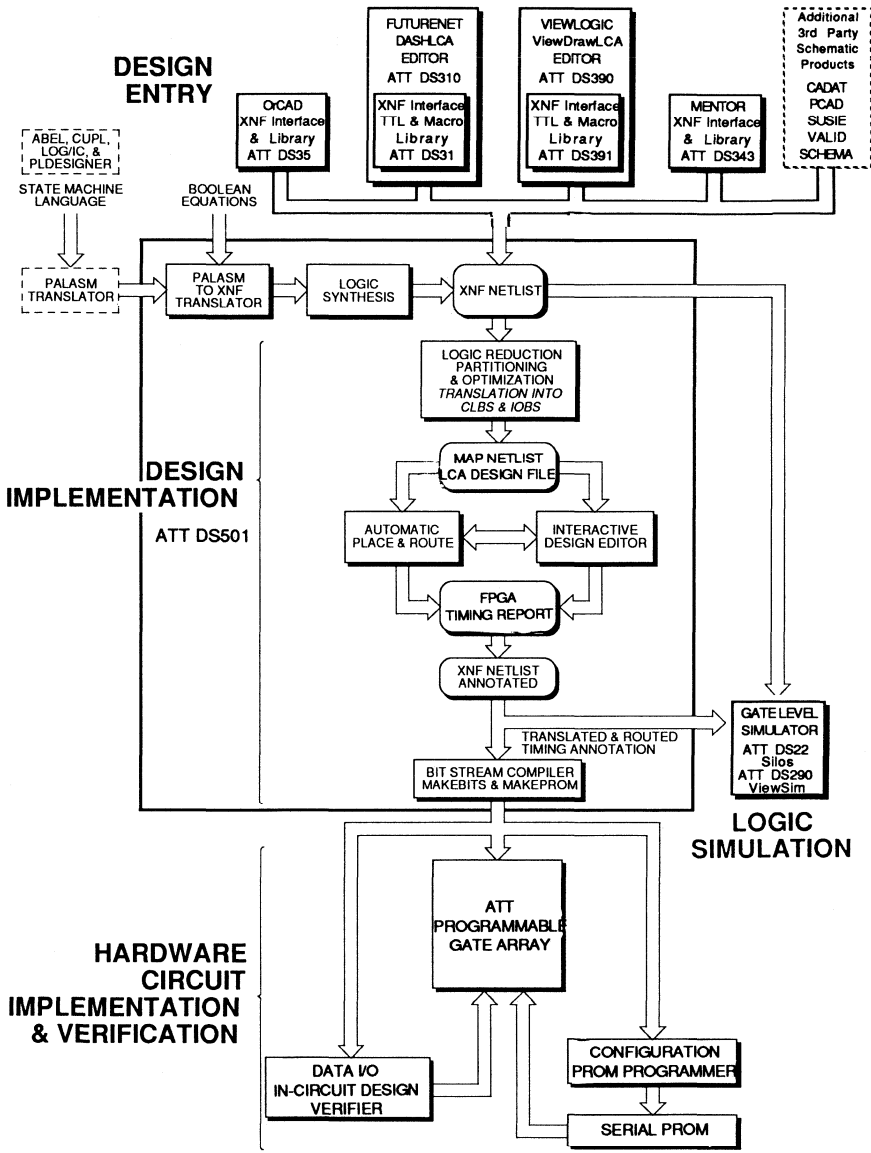
MAP Files

This file is also an ASCII text file and appears very similar to the XNF file. This MAP file is created by running the XNFMAP program, which is actually part of the XNF2LCA process. It is different from a straight XNF file in that the logic already has been partitioned or mapped into the FPGA architecture. There is no longer a one-to-one correspondence with the original schematic or logic entry description.

The MAP files consist of CLB and IOB symbol records with all of the appropriate I/O and logic models describing the function. Once a MAP file exists for a logic entity, it is essentially a stand-alone unit. No additional logic can be mixed into the CLB.

This independent mapping methodology allows isolation of the logic functions for improved signal routing, especially in large designs. The standard design flow can group dissimilar functions into the same CLB for better FPGA density. However, this may increase the localized wiring congestion which can be difficult to resolve.

AT&T Development System Architecture



File Structure

LCA Files

This ASCII text file is the lowest-level file that describes the physical layout details of the FPGA design. It describes exactly what is in each CLB and IOB, as well as the precise routing path that each signal takes. Typically, it is created by the XNF2LCA program.

The *XACT* graphical editor displays the details of the design described in the *LCA* file. The design can be modified in *XACT*, changing the *LCA* file; however, these changes are not reflected in the original schematic. If these changes require documentation in the schematic, then the incremental design methodology should be utilized.

The *LCA* file is an input to and from the APR (automated place and route) program. It is also the source for creating the bit stream for loading into the FPGA.

Product Summary

Typical users require one AT&T product from each category, or optional third-party schematic editor and/or simulator *(Viewlogic, Mentor Graphics, OrCAD,* or *FutureNet DASH* systems).

Design Entry

ATT-DS310	*DASH-LCA* Schematic Editor with Schematic Interface and TTL Library
ATT-DS31	*FutureNet* Schematic Interface and TTL Library
ATT-DS343	*Mentor Graphics* Schematic & Simulation Interface
ATT-DS35	*OrCAD STD* Schematic Interface
ATT-DS390	*Viewdraw* with Schematic Interface
ATT-DS391	*Viewdraw* and *Viewsim* Interface

Design Implementation

ATT-DS501	*XACT* Design Implementation System on PC
ATT-DS501-AP1	*XACT* Design Implementation System on *Apollo* Workstation (Interim PC version of graphic layout editor on PC)
ATT-DS501-SN1	*XACT* Design Implementation System on *Sun* Workstation (Interim PC version of graphic layout editor on PC)

Design Verification

ATT-DS22	*SILOS* Simulator and Interface on PC
ATT-DS290	*Viewsim* Simulator, *ViewGen*, and Interfaces on PC

Note: Previous ATT-DS53 is replaced by combination of ATT-DS501 and ATT-DS310.

AT&T Microelectronics

AT&T Cell Library

Overview

Schematic symbols referencing a specific CLB/IOB configuration for each logic cell function are described in this section. Cases where all the functional logic is contained within a single CLB/IOB are referred to as primitives. When more than one CLB is required, a suggested placement and routing interconnection is provided. With the EDITLCA graphic editor, logic can be entered either at the schematic level or during placement.

All entries presented in the AT&T cell library section are primitives. TLL cells are presented in the following section.

Logic

The logic section includes a schematic diagram for the function performed by the cell. The schematics are drawn with schematic symbols interconnected by signal lines. In addition, a symbol and/or a truth table may appear in this section. For a description of these, refer to their individual sections.

The schematic symbols used include simple logic gates, flip-flops, and latches. A variety of flip-flops and latches are used that can be distinguished by their input labeling. The same labeling is used for these symbols as is used for the flip-flop and latch macros documented in the flip-flop and latch sections.

For example, a **D** input is a data input and an **rd** input is an asynchronous reset direct input. A bold type font number appears within some of the schematic symbols.

This number indicates that the output of this symbol is the output of the block with that parameter number. Since CLBs can have two outputs, an **a** or **b** may be appended to the number to indicate that this symbol produces the output for half of that block.

Some symbols won't have a number denoted on them because a CLB can perform a function equivalent to many gates, so only the gate that contains the actual CLB output is denoted with a number. For schematic capture cells, the function may not be partitioned as shown in the schematic.

All input signals are labeled. The outputs are labeled with the word **name** prepended. What this indicates is that when the cell is invoked, the instance name will be placed where the word **name** appears.

Symbol

The symbol is used by schematic capture programs that support FPGA designs, such as *FutureNet DASH-LCA*. Symbol labeling might differ slightly between schematic capture packages, but this is the standard symbol for each cell.

Sample Placement

The sample placement is for use in choosing the blocks to place a macro, when using the *XACT* Graphic Editor (EDITLCA) to enter a design. The sample placement is just one possible way to place a cell. It was created with resource usage and signal timing in mind. Other placements may fit a particular design better than the sample placement.

Truth Table

The truth table shows the function of the cell in a tabular form. All inputs are shown to the left of the double lines, and the outputs are shown to the right. The notations **H** and **L** are used to denote a logical high or logical low signal value. For clock inputs, the notations ≠ and **L** are used to denote a rising edge or a constant level. The output is denoted as having the same value as one of the inputs by placing the input name as the value of the output. For synchronous cells, a subscript of 0 is used to denote the value of the signal before the clock edge.

Table of Contents

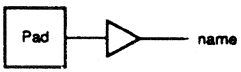
Table of Contents (continued)

Table of Contents (continued)

Table of Contents (continued)

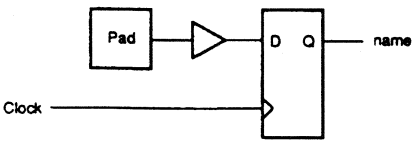
PIN Input Pad

Logic



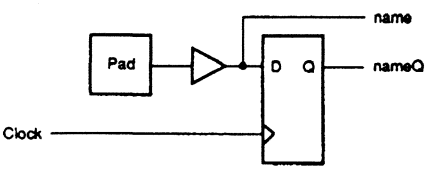
PINQ Input Pad with Registered Input

Logic



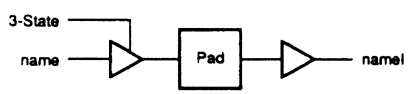
PINR Input Pad with Registered and Direct Inputs

Logic



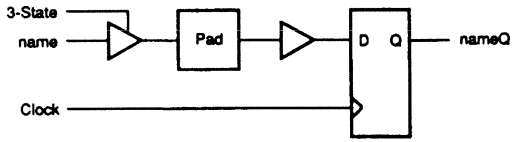
PIO Input/Output Pad

Logic



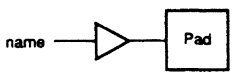
PIOQ Input/Output Pad with Registered Input

Logic



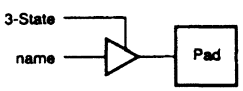
POUT Output Pad

Logic



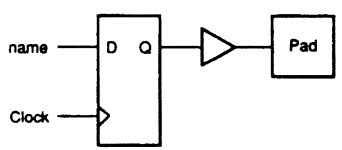
POUTZ Output Pad with 3-State Control

Logic



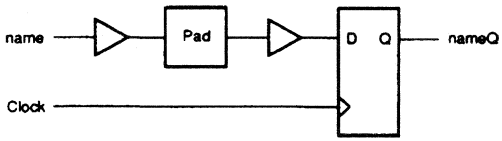
POUTQ Output Pad with Registered Output

Logic



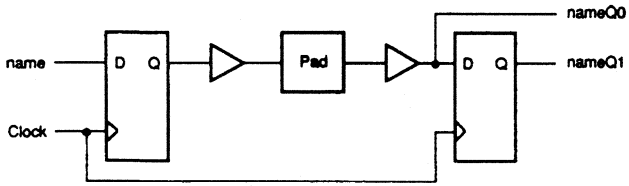
PREG Pad with Data Register

Logic



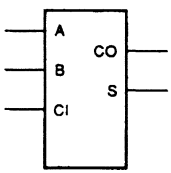
PREG2 Pad with 2-bit Shift Register

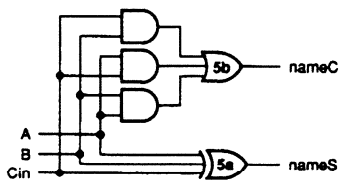
Logic



GADD 1-bit Full Adder

Logic

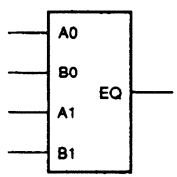


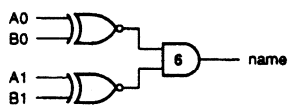


A	B	Cin	Sum	Cout
L	L	L	L	L
L	L	H	H	L
L	H	L	H	L
L	H	H	L	H
H	L	L	H	L
H	L	H	L	H
H	H	L	L	H
H	H	H	H	H

GCOMP 2-bit Comparator

Logic

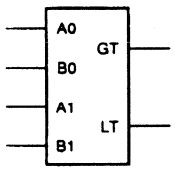


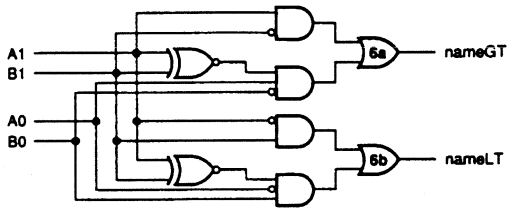


A0	B0	A1	B1	Output
L	L	L	L	H
L	L	H	H	H
H	H	L	L	H
H	H	H	H	L
L	L	X	X	L
H	L	X	X	L
X	X	L	H	L
X	X	H	L	L

GLTGT 2-bit Less-Than/Greater-Than Comparator

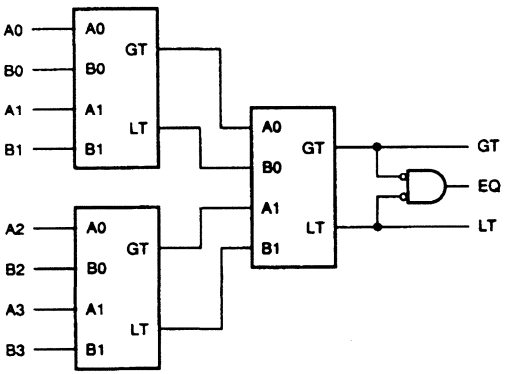
Logic





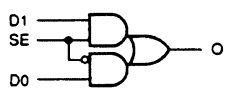
Inputs				Outputs	
A1	B1	A0	B0	GT	L
L	L	L	L	L	L
L	L	L	H	L	H
L	L	H	L	H	L
L	L	H	H	L	L
L	H	X	X	L	H
H	L	X	X	H	L
H	H	L	L	L	L
H	H	L	H	L	H
H	H	H	L	H	L
H	H	H	H	L	L

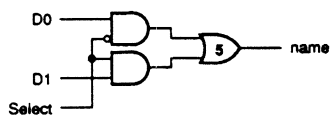
Expansion Example



GMUX Two-to-One Mux

Logic

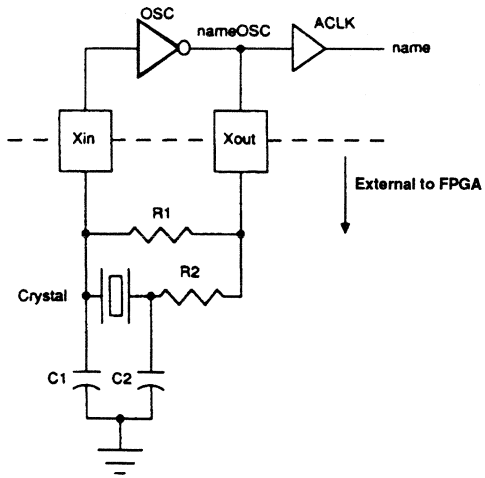




Select	D1	D0	Output
L	X	L	L
L	X	H	H
H	L	X	L
H	H	X	H

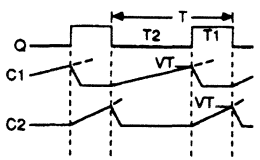
GXTL20 Crystal Oscillator for an ATT-3020

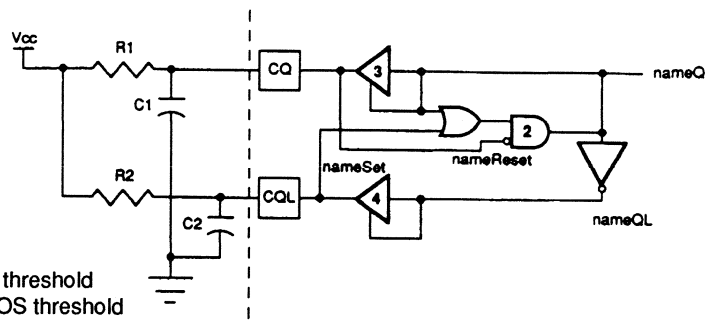
Logic



GOSC Low-Frequency Resistor-Capacitor Oscillator

Logic

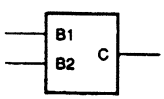




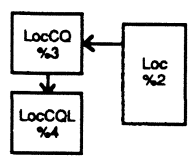
T = T1 + T2 = N (R1 C1) + (R2 C2)
where N = approx. 0.35 for TTL threshold
= approx. 0.75 for CMOS threshold
when each capacitor is allowed to be discharged by the FPGA during the opposite timing phase.

If the interconnect timing of the pad output is shorter than the interconnect timing of the pad 3-state network, the buffer might supply charge to the capacitor and shorten the period. A method to avoid this is to provide another net which is used to drive the output pins with a constant low.

Symbol

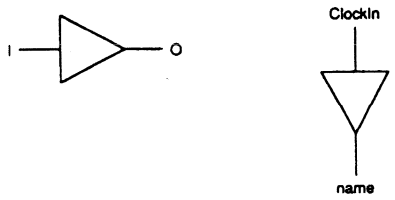


Sample Placement



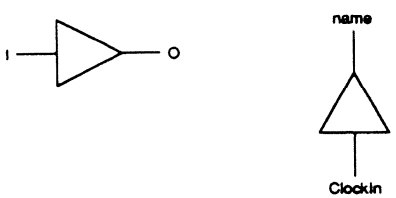
ACLK Auxiliary Buffer

Logic



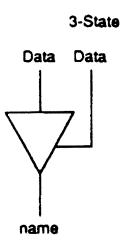
GCLK Global Buffer

Logic

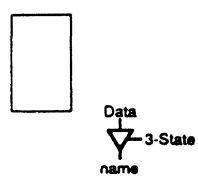


TBUF 3-State Buffer

Logic

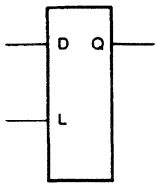


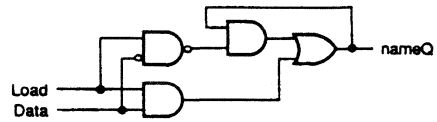
Sample Placement



LD Data Latch

Logic



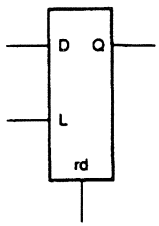


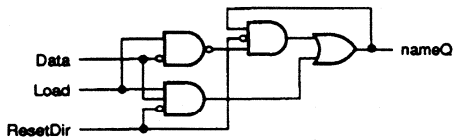
nameQ = L * D + Q * ~(L * ~D)

Load	Data	Q
L	X	Q_0
H	L	L
H	H	H

LDRD Data Latch with Reset Direct

Logic



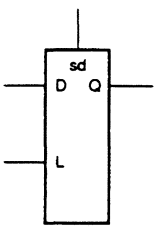


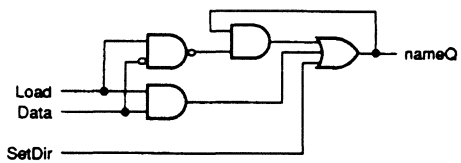
nameQ = L * D * ~RD + Q * ~RD * ~(L * ~D)

ResetDir	Load	Data	Q
H	X	X	L
L	L	X	Q_0
L	H	L	L
L	H	H	H

LDSD Data Latch with Set Direct

Logic



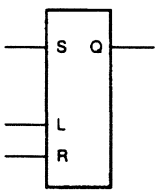


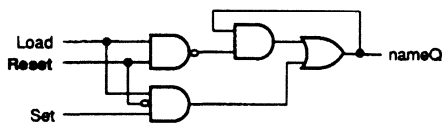
nameQ = L * D + Q * ~(L * ~D) + SD

SetDir	Load	Data	Q
H	X	X	H
L	L	X	Q_0
L	H	L	L
L	H	H	H

LRS Set-Reset Latch with Reset Dominant

Logic



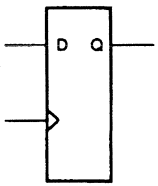


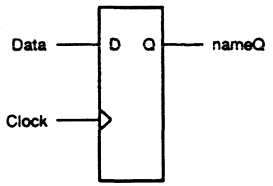
nameQ = L * S * ~R + Q * ~(L * R)

Load	Reset	Set	Q
L	X	X	Q_0
H	H	X	L
H	L	L	Q_0
H	L	H	H

FD D Flip-Flop

Logic



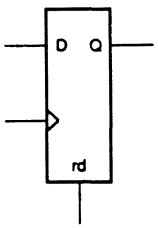


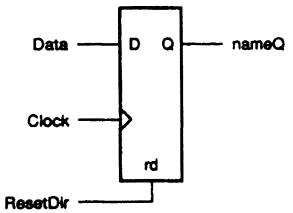
nextQ = D

Clock	Data	Q
L	X	Q_0
↑	L	L
↑	H	H

FDRD D Flip-Flop with Reset Direct

Logic



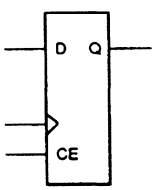


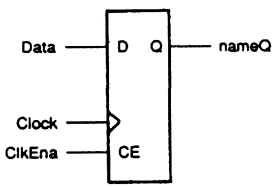
nextQ = D

ResetDir	Clock	Data	Q
H	X	X	L
L	L	X	Q_0
L	↑	L	L
L	↑	H	H

FDC D Flip-Flop with Clock Enable

Logic



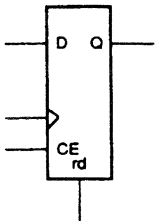


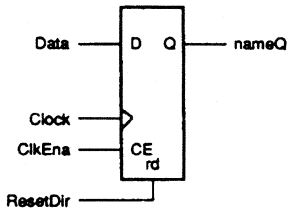
nextQ = D * CE + Q * ~CE

ClkEna	Clock	Data	Q
L	X	X	Q_0
X	L	X	Q_0
H	↑	L	L
H	↑	H	H

FDCRD D Flip-Flop with Clock Enable, Reset Direct

Logic



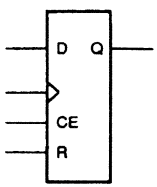


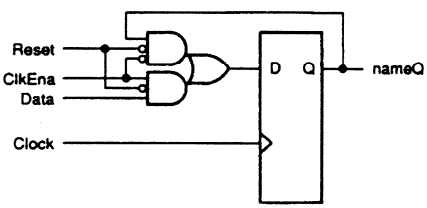
nextQ = D * CE + Q * ~CE

ResetDir	ClkEna	Clock	Data	Q
H	X	X	X	L
L	L	X	X	Q_0
L	X	L	X	Q_0
L	H	↑	L	L
L	H	↑	H	H

FDCR D Flip-Flop with Clock Enable, Reset

Logic



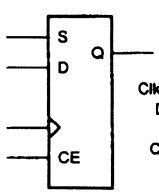


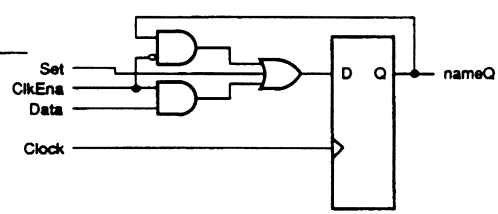
nextQ = D * CE * ~R + Q * ~CE * ~R

Reset	ClkEna	Clock	Data	Q
X	X	L	X	Q_0
H	X	↑	X	L
L	L	↑	X	Q_0
L	H	↑	L	L
L	H	↑	H	H

FDCS D Flip-Flop with Clock Enable, Set

Logic



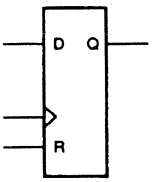


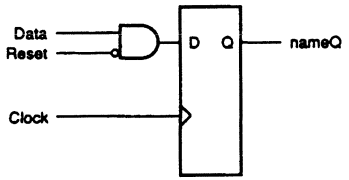
nextQ = D * CE + Q * ~CE + S

Set	ClkEna	Clock	Data	Q
X	X	L	X	Q_0
H	X	↑	X	H
L	L	↑	X	Q_0
L	H	↑	L	L
L	H	↑	H	H

FDR D Flip-Flop with Reset

Logic



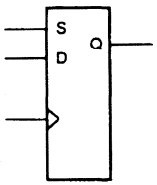


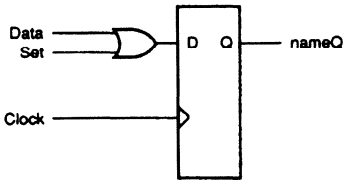
nextQ = D * ~R

Reset	Clock	Data	Q
X	L	X	Q_0
H	↑	X	L
L	↑	L	L
L	↑	H	H

FDS D Flip-Flop with Set

Logic



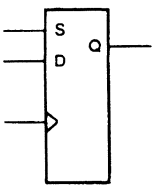


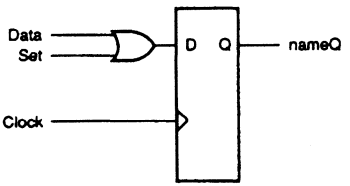
nextQ = D + S

Set	Clock	Data	Q
X	L	X	Q_0
H	↑	X	H
L	↑	L	L
L	↑	H	H

FRS Set-Reset Flip-Flop with Reset Dominant

Logic



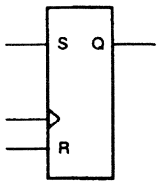


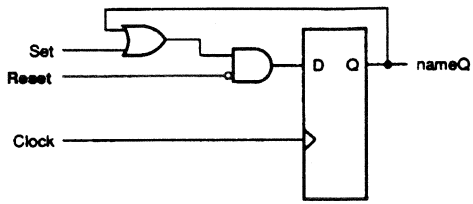
nextQ = D + S

Set	Clock	Data	Q
X	L	X	Q_0
H	↑	X	H
L	↑	L	L
L	↑	H	H

FSR Set-Reset Flip-Flop with Set Dominant

Logic



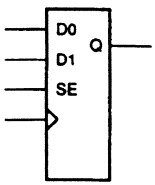


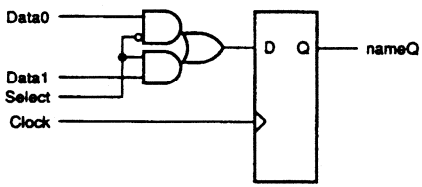
nextQ = (Q + S) * ~R

Reset	Set	Clock	Q
X	X	L	Q_0
H	X	↑	L
L	L	↑	Q_0
L	H	↑	H

FDM D Flip-Flop with 2-Input Data Mux

Logic



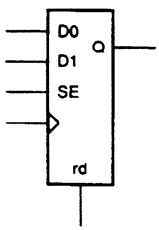


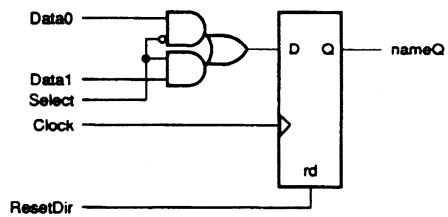
nextQ = D1 * Sel + D0 * ~Sel

Select	Clock	Q
X	L	Q_0
L	↑	D0
H	↑	D1

FDMRD D Flip-Flop with 2-Input Data Mux, Reset Direct

Logic



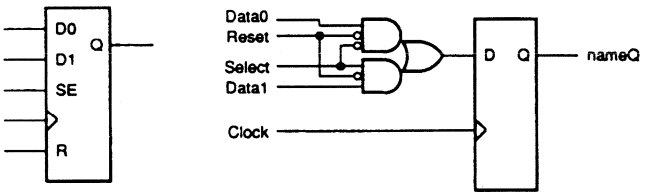


nextQ = D1 * Sel + D0 * ~Sel

ResetDir	Select	Clock	Q
H	X	X	L
L	X	L	Q_0
L	L	↑	D0
L	H	↑	D1

FDMR D Flip-Flop with 2-Input Data Mux, Reset

Logic

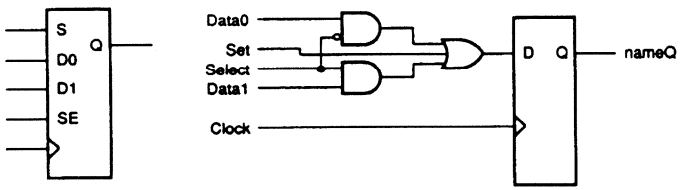


nextQ = D1 * Sel * ~R + D0 * ~Sel * ~R

Reset	Select	Clock	Q
X	X	L	Q_0
H	X	↑	L
L	L	↑	D0
L	H	↑	D1

FDMS D Flip-Flop with 2-Input Data Mux, Set

Logic

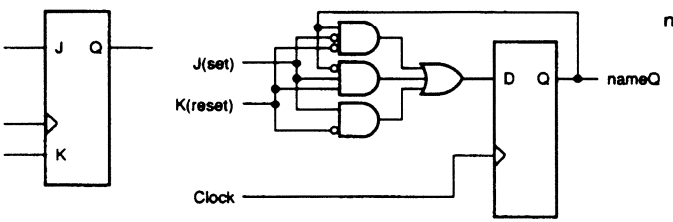


nextQ = D1 * Sel + D0 * ~Sel + S

Set	Select	Clock	Q
X	X	L	Q_0
H	X	↑	H
L	L	↑	D0
L	H	↑	D1

FJK J-K Flip-Flop

Logic

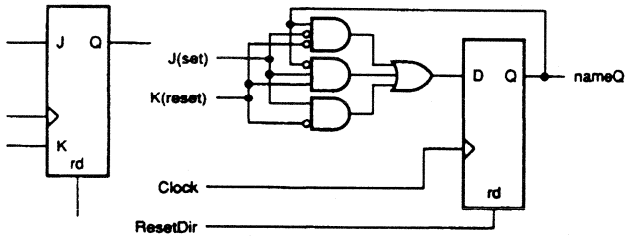


nextQ = J * ~K + J * K * ~Q + ~J * ~K * Q

Clock	J(set)	K(reset)	Q
L	X	X	Q_0
↑	L	L	Q_0
↑	L	H	L
↑	H	L	H
↑	H	H	$\overline{Q_0}$

FJKRD J-K Flip-Flop with Reset Direct

Logic

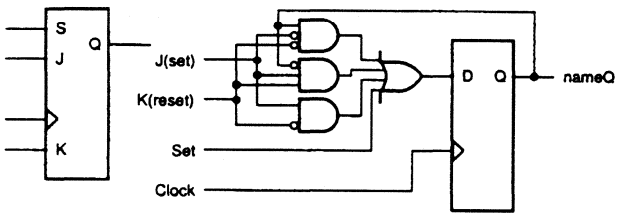


nextQ = J * ~K + J * K * ~Q + ~J * ~K * Q

ResetDir	Clock	J(set)	K(reset)	Q
H	X	X	X	L
L	L	X	X	Q_0
L	↑	L	L	Q_0
L	↑	L	H	L
L	↑	H	L	H
L	↑	H	H	$\overline{Q_0}$

FJKS J-K Flip-Flop with Set

Logic

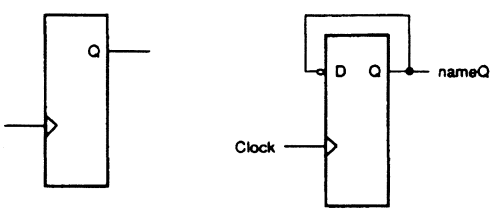


nextQ = J * ~K + J * K * ~Q + ~J * ~K * Q + S

Set	Clock	J(set)	K(reset)	Q
X	L	X	X	Q_0
H	↑	X	X	H
L	↑	L	L	Q_0
L	↑	L	H	L
L	↑	H	L	H
L	↑	H	H	$\overline{Q_0}$

FT0 Self-Toggle Flip-Flop

Logic

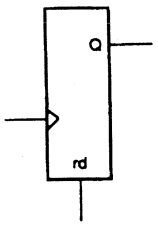


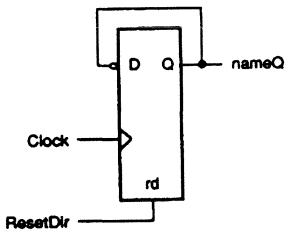
nextQ = ~Q

Clock	Q
L	Q_0
↑	$\overline{Q_0}$

FT0RD Self-Toggle Flip-Flop with Reset Direct

Logic



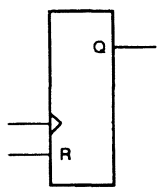


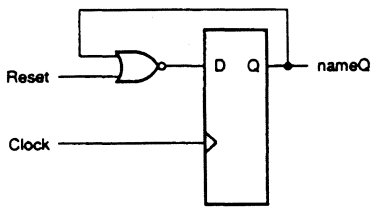
nextQ = ~Q

ResetDir	Clock	Q
H	X	L
L	L	Qp
L	↑	Qp

FT0R Self-Toggle Flip-Flop with Reset

Logic



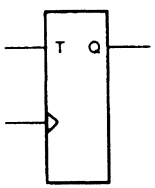


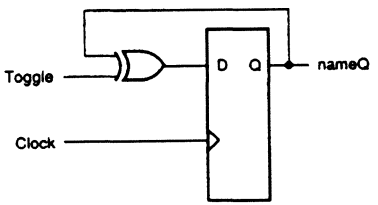
nextQ = ~Q * ~R

Reset	Clock	Q
X	L	Qp
L	↑	Qp
H	↑	L

FT Toggle Flip-Flop

Logic



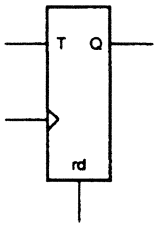


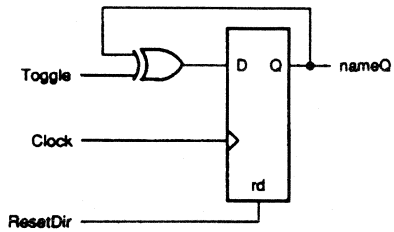
nextQ = T @ Q

Clock	Toggle	Q
L	X	Qp
↑	L	Qp
↑	H	Qp

FTRD Toggle Flip-Flop with Reset Direct

Logic



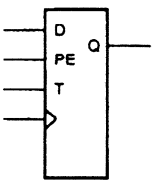


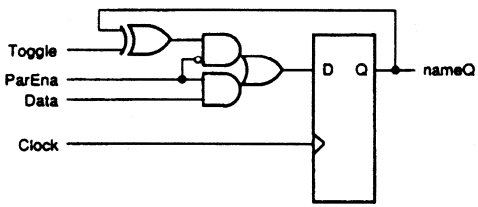
nextQ = T @ Q

ResetDir	Clock	Toggle	Q
H	X	X	L
L	L	X	Q
L	↑	L	Q
L	↑	H	Q

FTP Toggle Flip-Flop with Parallel Enable

Logic



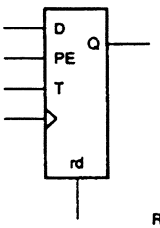


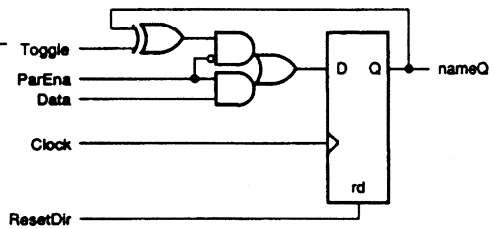
nextQ = PE * D + (Q @ Toggle) * ~PE

ParEna	Clock	Toggle	Q
X	L	X	Q
H	↑	X	D
L	↑	L	Q
L	↑	H	Q

FTPRD Toggle Flip-Flop with Parallel Enable, Reset Direct

Logic



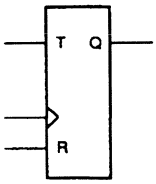


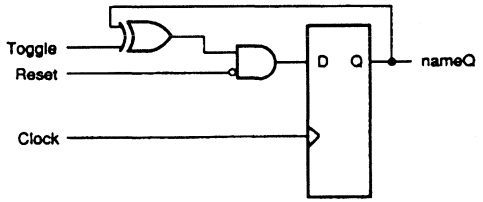
nextQ = PE * D + (Q @ Toggle) * ~PE

ResetDir	ParEna	Clock	Toggle	Q
H	X	X	X	L
L	X	L	X	Q
L	H	↑	X	D
L	L	↑	L	Q
L	L	↑	H	Q

FTR Toggle Flip-Flop with Reset

Logic



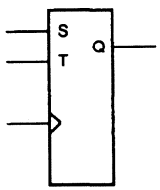


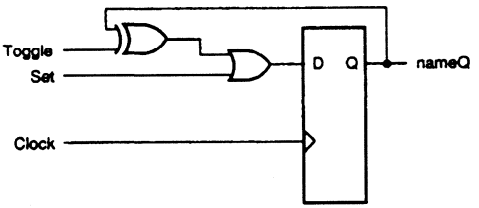
nextQ = (Q @ T) * ~R

Reset	Clock	Toggle	Q
X	L	X	Q_0
H	↑	X	L
L	↑	L	Q_0
L	↑	H	$\overline{Q_0}$

FTS Toggle Flip-Flop with Set

Logic



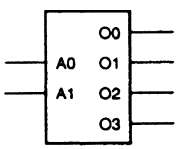


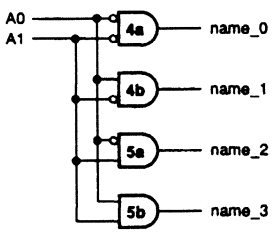
nextQ = (Q @ T) + S

Set	Clock	Toggle	Q
X	L	X	Q_0
H	↑	X	H
L	↑	L	Q_0
L	↑	H	$\overline{Q_0}$

D2-4 1-of 4 Decoder

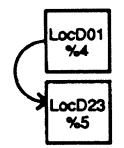
Logic





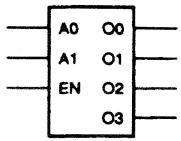
Select		Outputs			
A1	A0	O0	O1	O2	O3
L	L	H	L	L	L
L	H	L	H	L	L
H	L	L	L	H	L
H	H	L	L	L	H

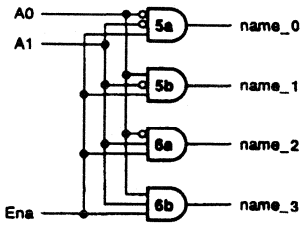
Sample Placement



D2-4E 1-of 4 Decoder with Enable

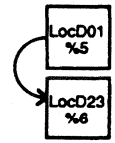
Logic





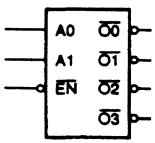
Enable	Select		Outputs			
	A1	A0	O0	O1	O2	O3
L	X	X	L	L	L	L
H	L	L	H	L	L	L
H	L	H	L	H	L	L
H	H	L	L	L	H	L
H	H	H	L	L	L	H

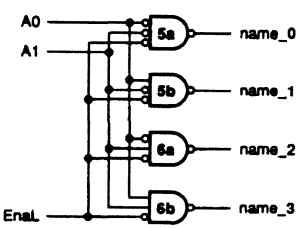
Sample Placement



74-139 1-of 4 Single Decoder with Enable, Low Output

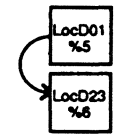
Logic





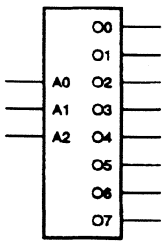
EnaL	Select		Outputs			
	A1	A0	$\overline{O0}$	$\overline{O1}$	$\overline{O2}$	$\overline{O3}$
H	X	X	H	H	H	H
L	L	L	L	H	H	H
L	L	H	H	L	H	H
L	H	L	H	H	L	H
L	H	H	H	H	H	L

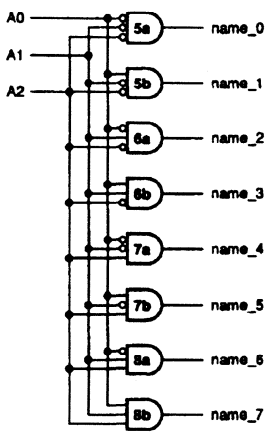
Sample Placement



D3-8 1-of 8 Decoder

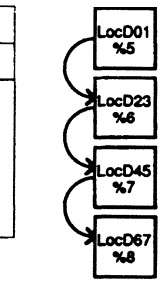
Logic





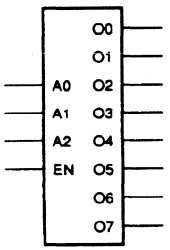
Select			Outputs							
A2	A1	A0	O0	O1	O2	O3	O4	O5	O6	O7
L	L	L	H	L	L	L	L	L	L	L
L	L	H	L	H	L	L	L	L	L	L
L	H	L	L	L	H	L	L	L	L	L
L	H	H	L	L	L	H	L	L	L	L
H	L	L	L	L	L	L	H	L	L	L
H	L	H	L	L	L	L	L	H	L	L
H	H	L	L	L	L	L	L	L	H	L
H	H	H	L	L	L	L	L	L	L	H

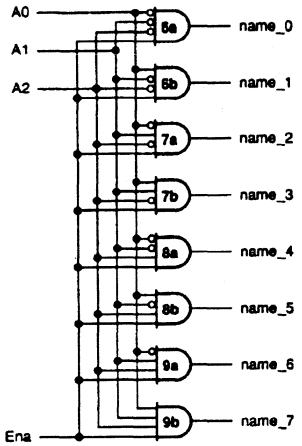
Sample Placement



D3-8E 1-of 8 Decoder with Enable

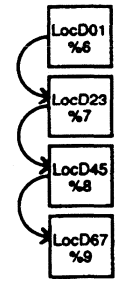
Logic





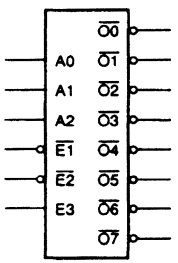
Enable	Select			Outputs							
	A2	A1	A0	O0	O1	O2	O3	O4	O5	O6	O7
L	X	X	X	L	L	L	L	L	L	L	L
H	L	L	L	H	L	L	L	L	L	L	L
H	L	L	H	L	H	L	L	L	L	L	L
H	L	H	L	L	L	H	L	L	L	L	L
H	L	H	H	L	L	L	H	L	L	L	L
H	H	L	L	L	L	L	L	H	L	L	L
H	H	L	H	L	L	L	L	L	H	L	L
H	H	H	L	L	L	L	L	L	L	H	L
H	H	H	H	L	L	L	L	L	L	L	H

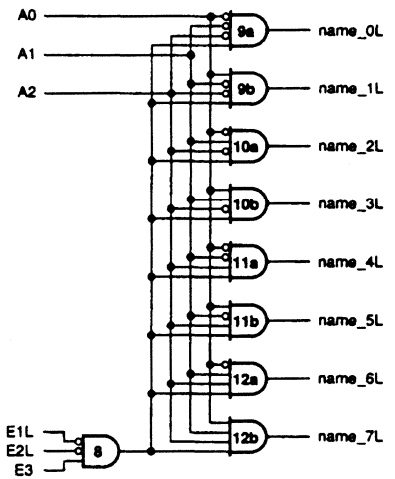
Sample Placement



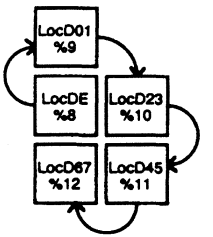
74-138 1-of 8 Decoder with Enables, Low Output

Logic



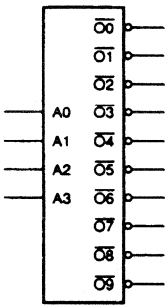


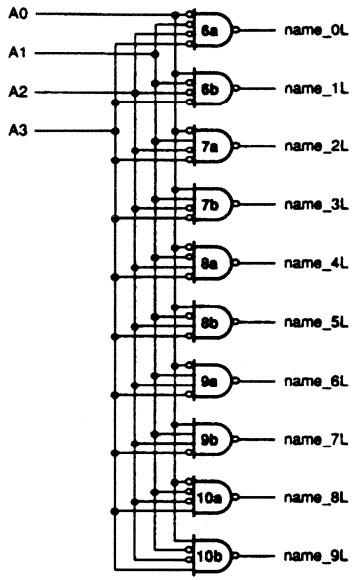
Sample Placement



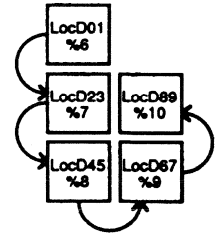
74-42 1-of 10 Decoder with Low Output

Logic



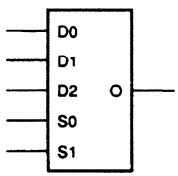


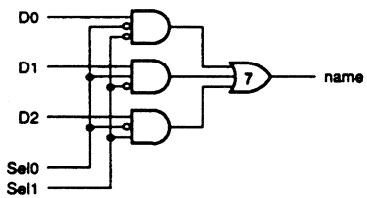
Sample Placement



M3-1 3-to-1 Mux

Logic

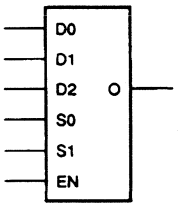


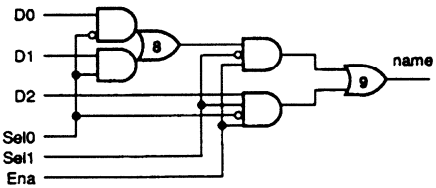


Select		Output
S1	S0	
L	L	D0
L	H	D1
H	L	D2
H	H	L

M3-1E 3-to-1 Mux, Enable

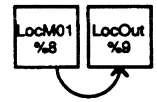
Logic





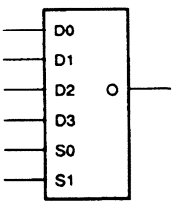
Enable	Select		Output
	S1	S0	
L	X	X	L
H	L	L	D0
H	L	H	D1
H	H	L	D2
H	H	H	L

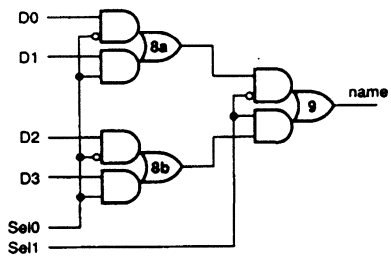
Sample Placement



M4-1 4-to-1 Mux

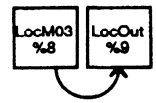
Logic





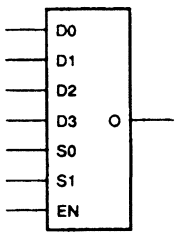
Select		Output
S1	S0	
L	L	D0
L	H	D1
H	L	D2
H	H	D3

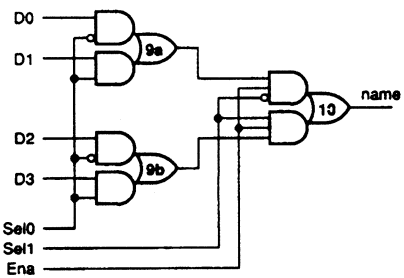
Sample Placement



M4-1E 4-to-1 Mux, Enable

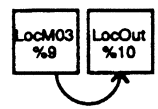
Logic





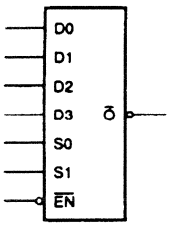
Enable	Select		Output
	S1	S0	
L	X	X	L
H	L	L	D0
H	L	H	D1
H	H	L	D2
H	H	H	D3

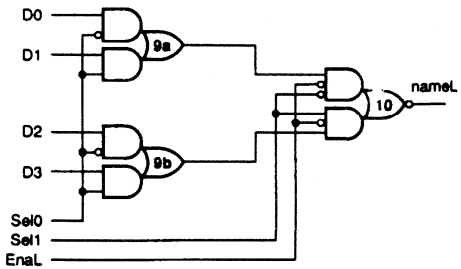
Sample Placement



74-352 4-to-1 Mux, Enable, Low Output

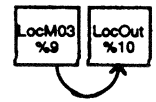
Logic





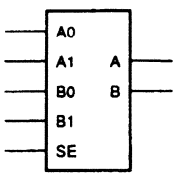
EnaL	Select		Output
	S1	S0	
H	X	X	H
L	L	L	$\overline{D0}$
L	L	H	$\overline{D1}$
L	H	L	$\overline{D2}$
L	H	H	$\overline{D3}$

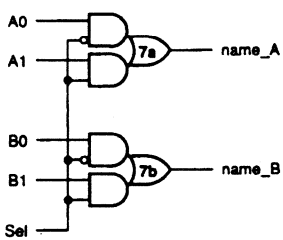
Sample Placement



M4-2 4-to-2 Mux

Logic

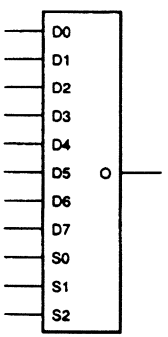


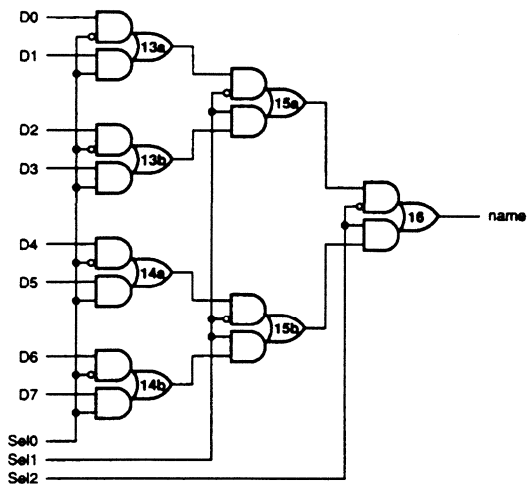


Select	A	B
L	A0	B0
H	A1	B1

M8-1 8-to-1 Mux

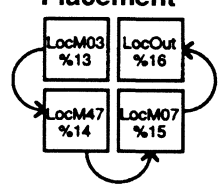
Logic





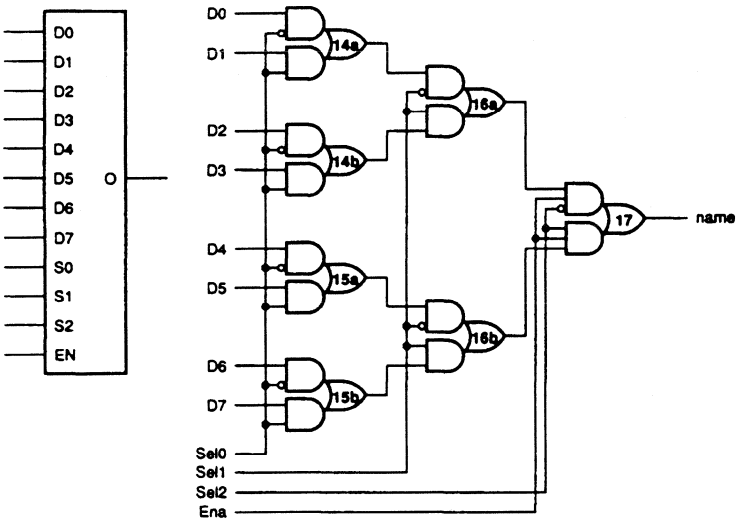
Select			Output
S2	S1	S0	
L	L	L	D0
L	L	H	D1
L	H	L	D2
L	H	H	D3
H	L	L	D4
H	L	H	D5
H	H	L	D6
H	H	H	D7

Sample Placement



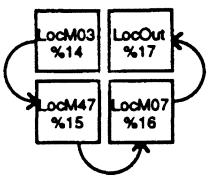
M8-1E 8-to-1 Mux, Enable

Logic



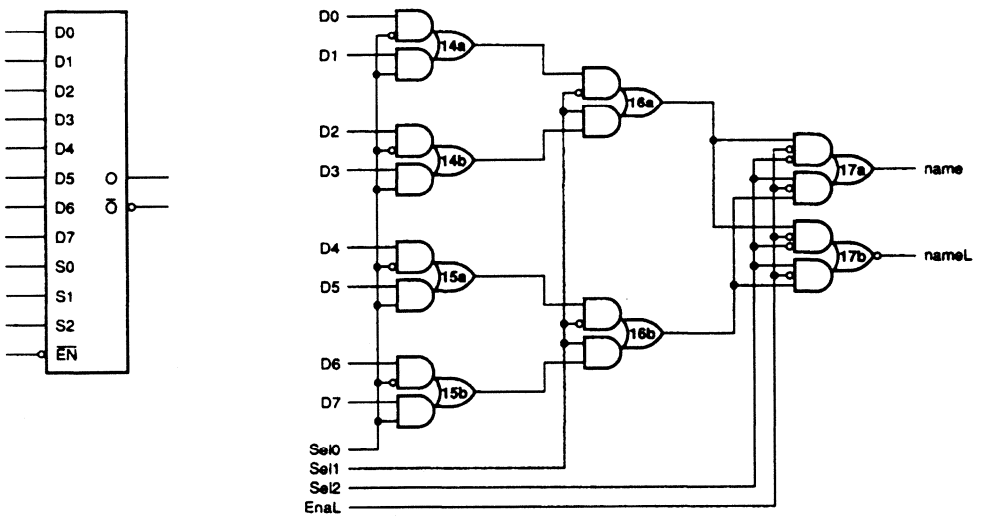
Enable	Select			Output
	S2	S1	S0	
L	X	X	X	L
H	L	L	L	D0
H	L	L	H	D1
H	L	H	L	D2
H	L	H	H	D3
H	H	L	L	D4
H	H	L	H	D5
H	H	H	L	D6
H	H	H	H	D7

Sample Placement



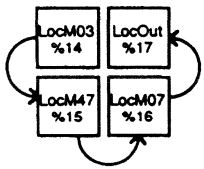
74-151 8-to-1 Mux, Enable, Complementary Outputs

Logic



74-151 8-to-1 Mux, Enable, Complementary Outputs (continued)

Sample Placement

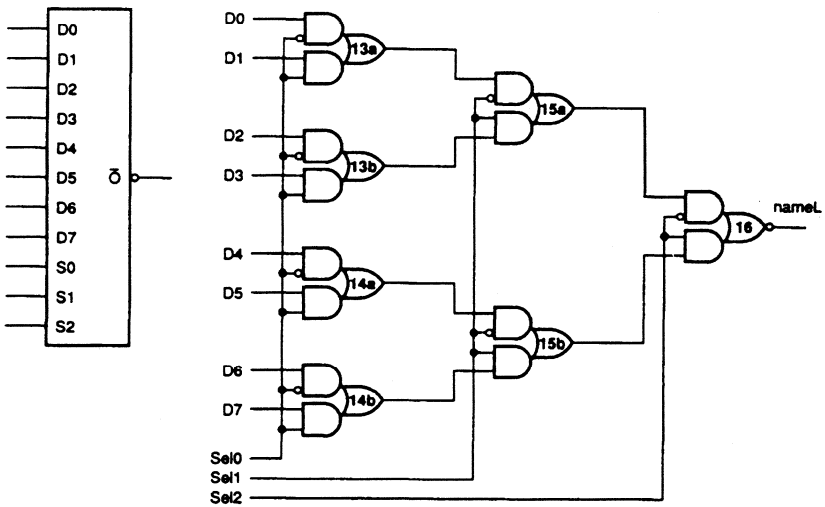


Truth Table

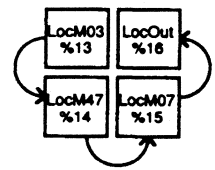
EnaL	Select			Output	$\overline{\text{Output}}$
	S2	S1	S0		
H	X	X	X	L	H
L	L	L	L	D0	$\overline{D0}$
L	L	L	H	D1	$\overline{D1}$
L	L	H	L	D2	$\overline{D2}$
L	L	H	H	D3	$\overline{D3}$
L	H	L	L	D4	$\overline{D4}$
L	H	L	H	D5	$\overline{D5}$
L	H	H	L	D6	$\overline{D6}$
L	H	H	H	D7	$\overline{D7}$

74-152 8-to-1 Mux, Low Output

Logic



Sample Placement

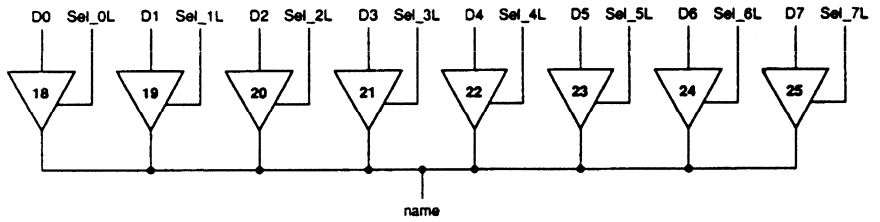


Truth Table

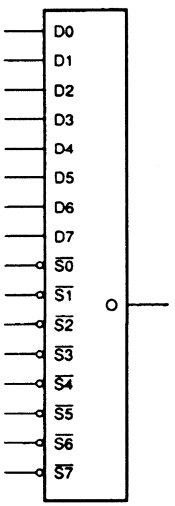
Select			$\overline{\text{Output}}$
S2	S1	S0	
L	L	L	$\overline{D0}$
L	L	H	$\overline{D1}$
L	H	L	$\overline{D2}$
L	H	H	$\overline{D3}$
H	L	L	$\overline{D4}$
H	L	H	$\overline{D5}$
H	H	L	$\overline{D6}$
H	H	H	$\overline{D7}$

MZ8-1 8-to-1 Mux Using 3-State Buffers

Logic



Symbol



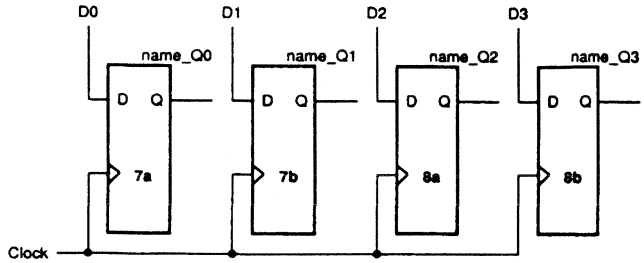
Truth Table

Select								Output
S7L	S6L	S5L	S4L	S3L	S2L	S1L	S0L	
H	H	H	H	H	H	H	H	High Z
H	H	H	H	H	H	H	L	D0
H	H	H	H	H	H	L	H	D1
H	H	H	H	H	L	H	H	D2
H	H	H	H	L	H	H	H	D3
H	H	H	L	H	H	H	H	D4
H	H	L	H	H	H	H	H	D5
H	L	H	H	H	H	H	H	D6
L	H	H	H	H	H	H	H	D7

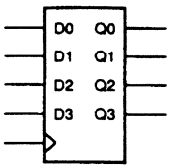
Note: If more than one of the select inputs is low, the output is invalid.

RD4 4-bit Data Register

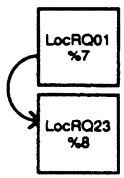
Logic



Symbol



Sample Placement

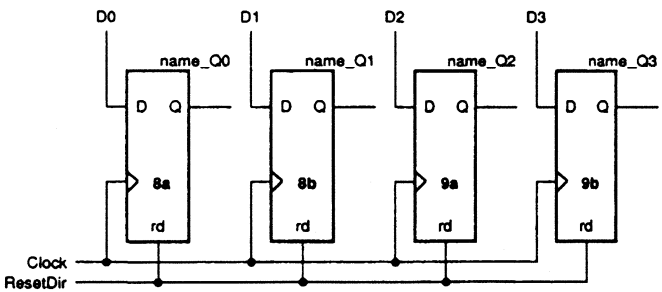


Truth Table

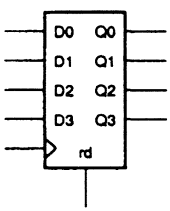
Clock	Di	Qi
L	X	Qi_0
↑	L	L
↑	H	H

RD4RD 4-bit Data Register, Reset Direct

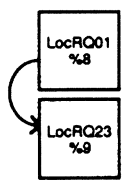
Logic



Symbol



Sample Placement

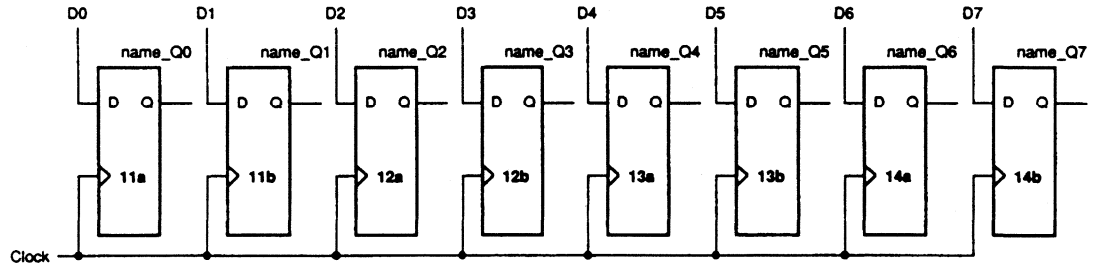


Truth Table

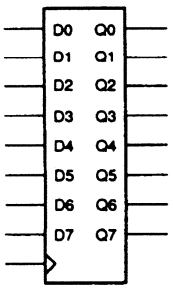
ResetDir	Clock	Di	Qi
H	X	X	L
L	L	X	Qi_0
L	↑	L	L
L	↑	H	H

RD8 8-bit Data Register

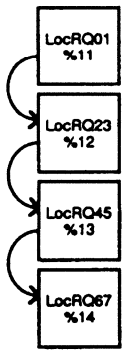
Logic



Symbol



Sample Placement

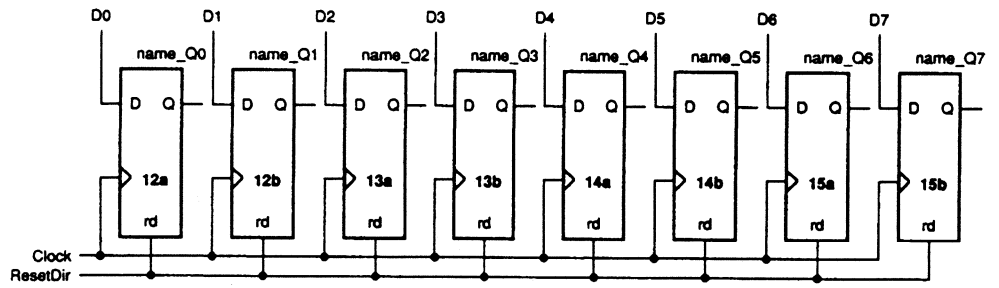


Truth Table

Clock	DI	QI
L	X	QI_0
↑	L	L
↑	H	H

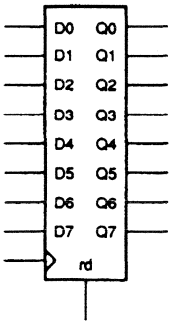
RD8RD 8-bit Data Register, Reset Direct

Logic

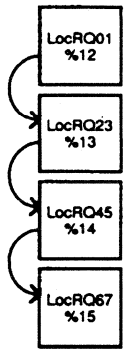


RD8RD 8-bit Data Register, Reset Direct (continued)

Symbol



Sample Placement

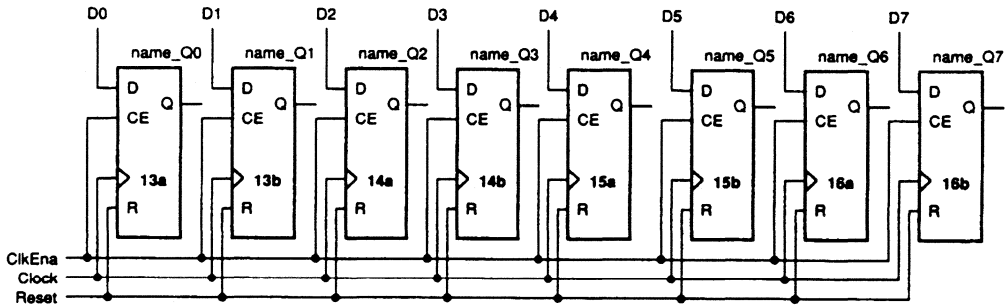


Truth Table

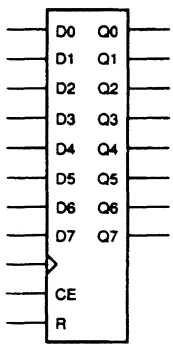
ResetDir	Clock	Di	Qi
H	X	X	L
L	L	X	Qi_0
L	↑	L	L
L	↑	H	H

RD8CR 8-bit Data Register, Clock Enable, Reset Direct

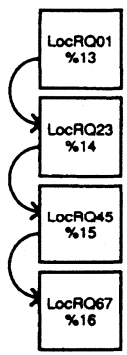
Logic



Symbol



Sample Placement

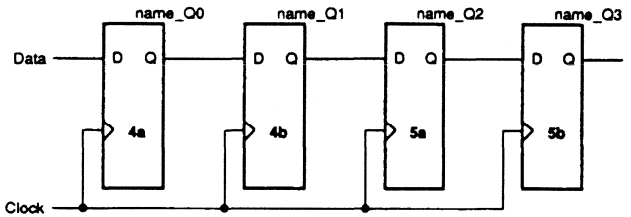


Truth Table

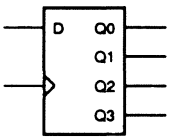
Reset	ClkEna	Clock	Di	Qi
X	X	L	X	Qi_0
H	X	↑	X	L
L	L	↑	X	Qi_0
L	H	↑	L	L
L	H	↑	H	H

RS4 4-bit Shift Register

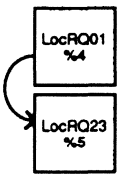
Logic



Symbol



Sample Placement

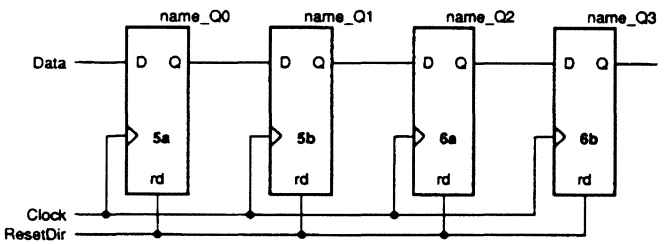


Truth Table

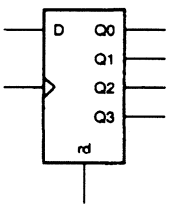
Clock	Outputs			
	Q0	Q1	Q2	Q3
L	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$
↑	Data	$Q0_0$	$Q1_0$	$Q2_0$

RS4RD 4-bit Shift Register, Reset Direct

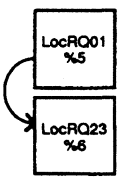
Logic



Symbol



Sample Placement

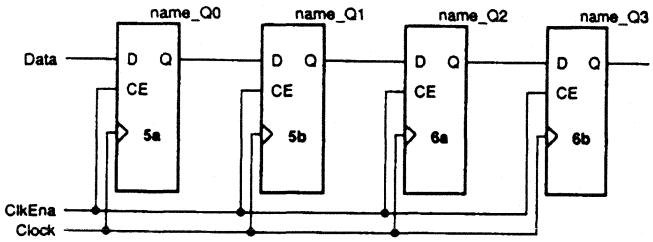


Truth Table

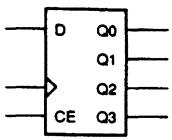
ResetDir	Clock	Outputs			
		Q0	Q1	Q2	Q3
H	X	L	L	L	L
L	L	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$
L	↑	Data	$Q0_0$	$Q1_0$	$Q2_0$

RS4C 4-bit Shift Register, Clock Enable

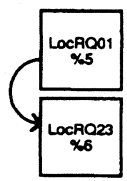
Logic



Symbol



Sample Placement

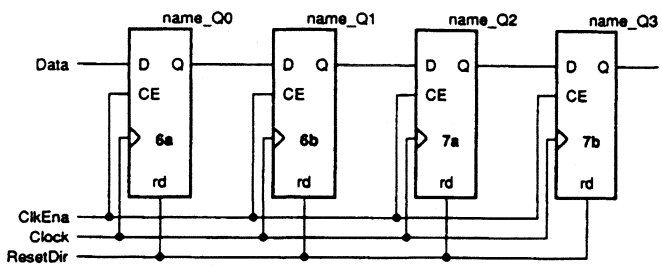


Truth Table

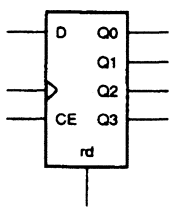
ClkEna	Clock	Outputs			
		Q0	Q1	Q2	Q3
X	L	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$
L	X	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$
H	↑	Data	$Q0_0$	$Q1_0$	$Q2_0$

RS4CRD 4-bit Shift Register, Clock Enable, Reset Direct

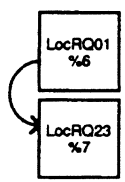
Logic



Symbol



Sample Placement

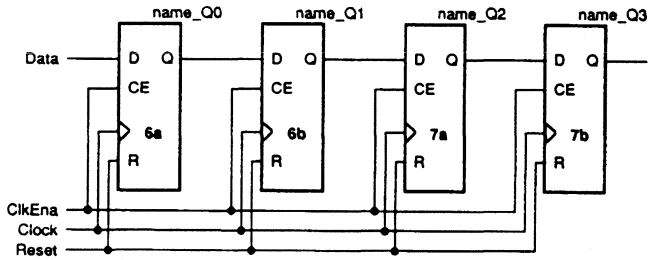


Truth Table

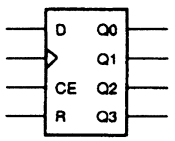
ResetDir	ClkEna	Clock	Outputs			
			Q0	Q1	Q2	Q3
H	X	X	L	L	L	L
L	X	L	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$
L	L	X	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$
L	H	↑	Data	$Q0_0$	$Q1_0$	$Q2_0$

RS4CR 4-bit Shift Register, Clock Enable, Reset

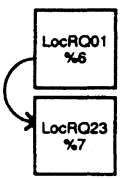
Logic



Symbol



Sample Placement

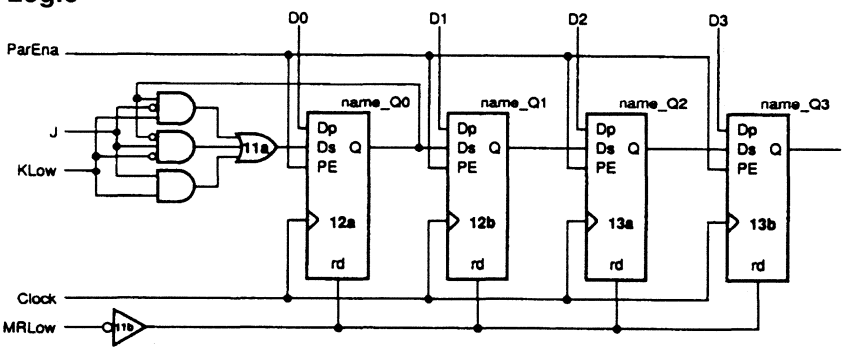


Truth Table

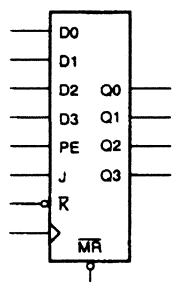
Reset	ClkEna	Clock	Outputs			
			Q0	Q1	Q2	Q3
X	X	L	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$
H	X	↑	L	L	L	L
L	L	↑	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$
L	H	↑	Data	$Q0_0$	$Q1_0$	$Q2_0$

74-195 4-bit Serial-to-Parallel Shift Register, Parallel Enable, Master Reset Low

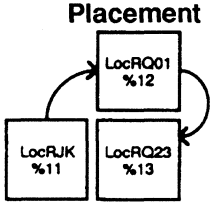
Logic



Symbol



Sample Placement

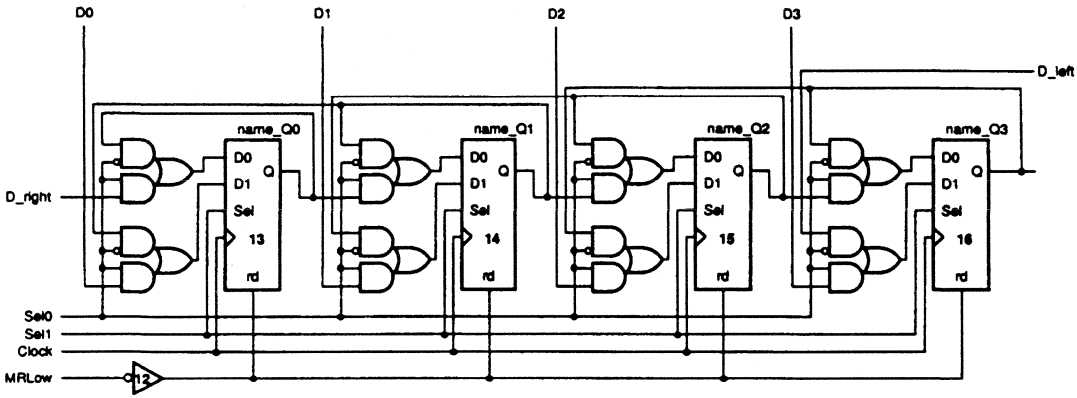


Truth Table

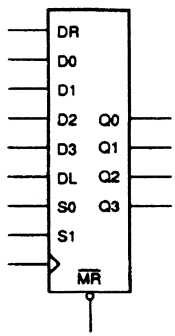
MRLow	ParEna	Clock	J	KLow	Outputs			
					Q0	Q1	Q2	Q3
L	X	X	X	X	L	L	L	L
H	X	L	X	X	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$
H	H	↑	X	X	D0	D1	D2	D3
H	L	↑	L	L	L	$Q0_0$	$Q1_0$	$Q2_0$
H	L	↑	L	H	$\overline{Q0_0}$	$Q0_0$	$Q1_0$	$Q2_0$
H	L	↑	H	L	$\overline{Q0_0}$	$Q0_0$	$Q1_0$	$Q2_0$
H	L	↑	H	H	H	$Q0_0$	$Q1_0$	$Q2_0$

74-194 4-bit Bidirectional Shift Register, Parallel Enable, Master Reset Low

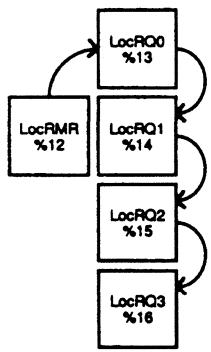
Logic



Symbol



Sample Placement

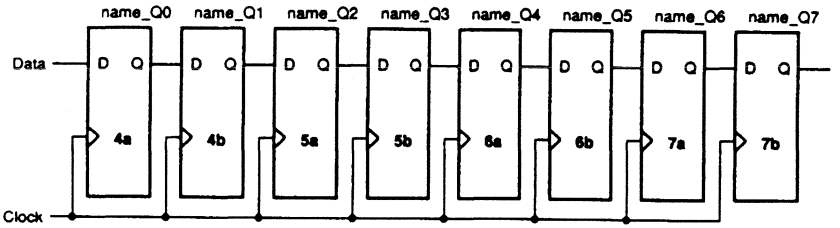


Truth Table

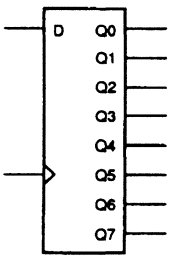
MRLow	Sel1	Sel0	Clock	Outputs			
				Q0	Q1	Q2	Q3
L	X	X	X	L	L	L	L
H	X	X	L	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$
H	L	L	↑	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$
H	L	H	↑	DR	$Q0_0$	$Q1_0$	$Q2_0$
H	H	L	↑	$Q1_0$	$Q2_0$	$Q3_0$	DL
H	H	H	↑	D0	D1	D2	D3

RS8 8-bit Shift Register

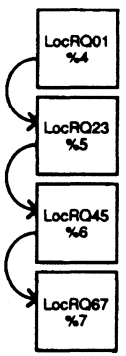
Logic



Symbol



Sample Placement

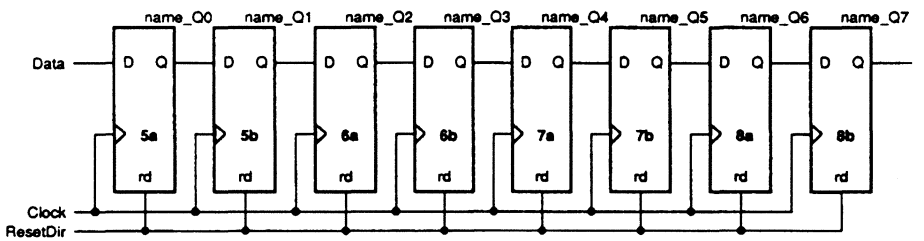


Truth Table

Clock	Di	Qi
L	X	Qi_0
↑	L	L
↑	H	H

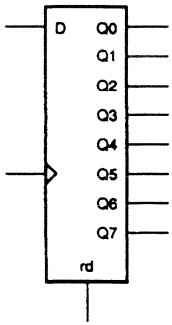
RS8RD 8-bit Shift Register, Reset Direct

Logic

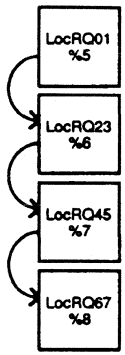


RS8RD 8-bit Shift Register, Reset Direct (continued)

Symbol



Sample Placement

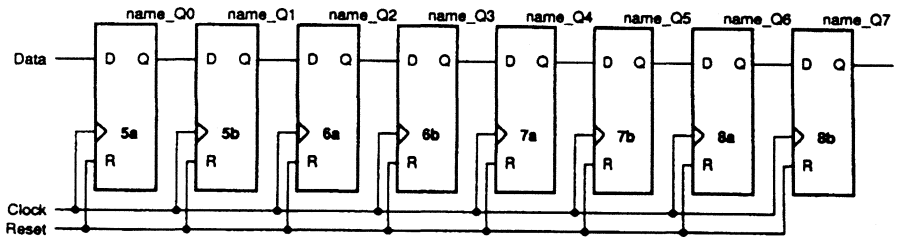


Truth Table

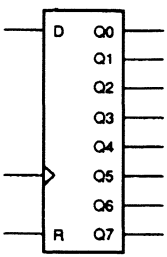
ResetDir	Clock	Outputs							
		Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7
H	X	L	L	L	L	L	L	L	L
L	L	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$	$Q4_0$	$Q5_0$	$Q6_0$	$Q7_0$
L	↑	Data	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$	$Q4_0$	$Q5_0$	$Q6_0$

RS8R 8-bit Shift Register, Reset

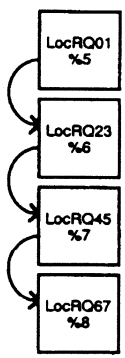
Logic



Symbol



Sample Placement

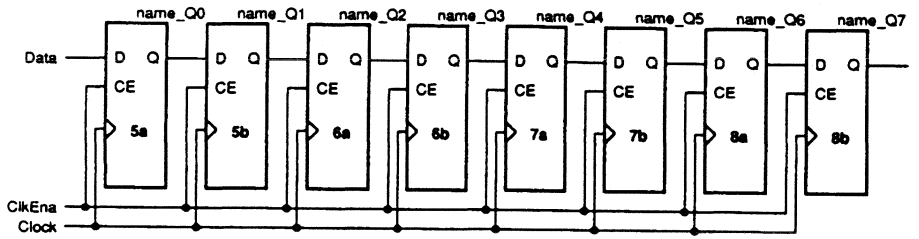


Truth Table

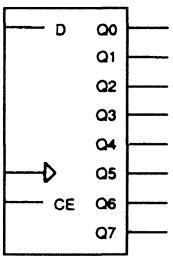
Reset	Clock	Outputs							
		Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7
X	L	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$	$Q4_0$	$Q5_0$	$Q6_0$	$Q7_0$
L	↑	Data	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$	$Q4_0$	$Q5_0$	$Q6_0$
H	↑	L	L	L	L	L	L	L	L

RS8C 8-bit Shift Register, Clock Enable

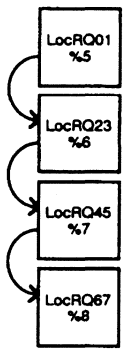
Logic



Symbol



Sample Placement

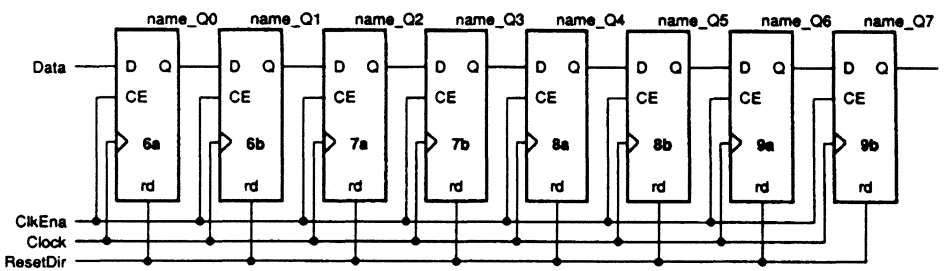


Truth Table

ClkEna	Clock	Outputs							
		Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7
X	L	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$	$Q4_0$	$Q5_0$	$Q6_0$	$Q7_0$
L	X	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$	$Q4_0$	$Q5_0$	$Q6_0$	$Q7_0$
H	↑	Data	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$	$Q4_0$	$Q5_0$	$Q6_0$

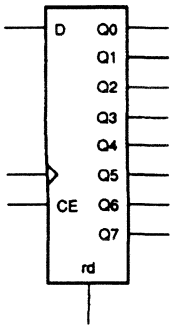
RS8CRD 8-bit Shift Register, Clock Enable, Reset Direct

Logic

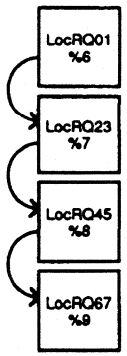


RS8CRD 8-bit Shift Register, Clock Enable, Reset Direct (continued)

Symbol



Sample Placement

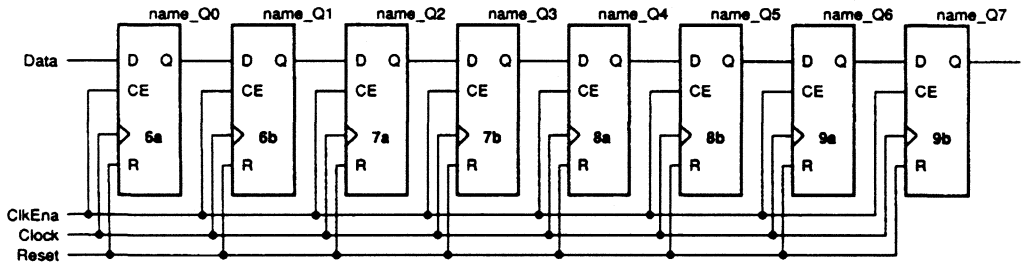


Truth Table

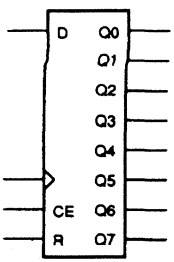
ResetDir	ClkEna	Clock	Outputs							
			Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7
H	X	X	L	L	L	L	L	L	L	L
L	X	L	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$	$Q4_0$	$Q5_0$	$Q6_0$	$Q7_0$
L	L	X	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$	$Q4_0$	$Q5_0$	$Q6_0$	$Q7_0$
L	H	↑	Data	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$	$Q4_0$	$Q5_0$	$Q6_0$

RS8CR 8-bit Shift Register, Clock Enable, Reset

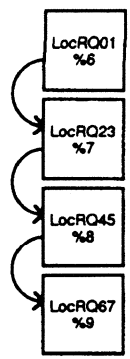
Logic



Symbol



Sample Placement

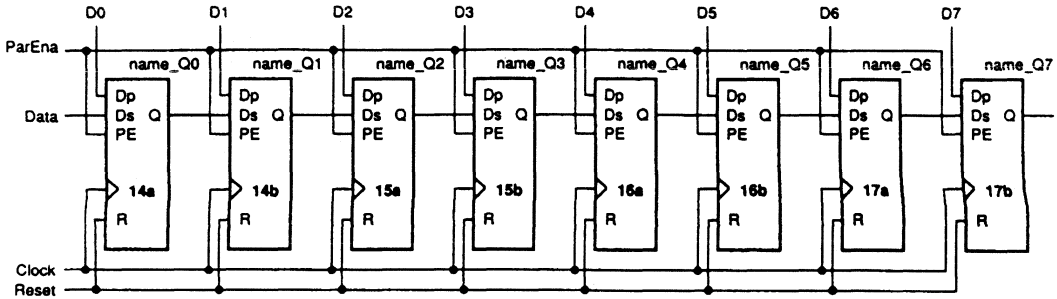


Truth Table

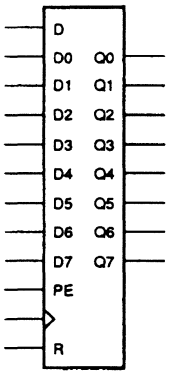
Reset	ClkEna	Clock	Outputs							
			Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7
X	X	L	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$	$Q4_0$	$Q5_0$	$Q6_0$	$Q7_0$
H	X	↑	L	L	L	L	L	L	L	L
L	L	↑	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$	$Q4_0$	$Q5_0$	$Q6_0$	$Q7_0$
L	H	↑	Data	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$	$Q4_0$	$Q5_0$	$Q6_0$

RS8PR 8-bit Shift Register, Parallel Enable, Reset

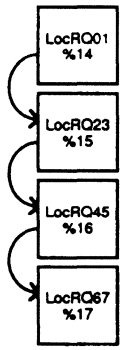
Logic



Symbol



Sample Placement

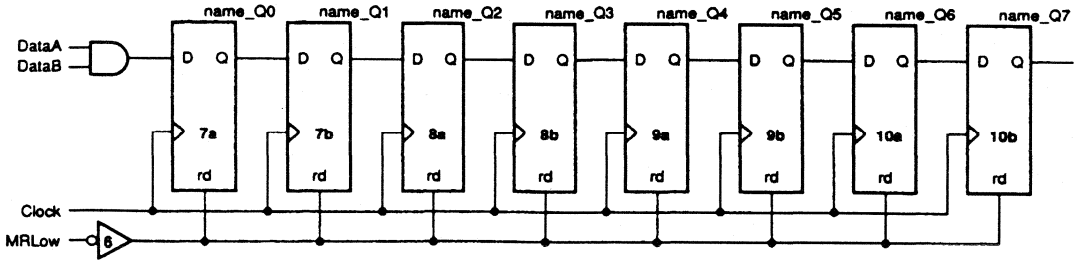


Truth Table

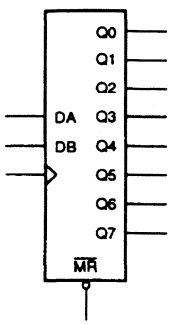
Reset	ParEna	Clock	Outputs							
			Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7
X	X	L	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$	$Q4_0$	$Q5_0$	$Q6_0$	$Q7_0$
H	X	↑	L	L	L	L	L	L	L	L
L	L	↑	Data	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$	$Q4_0$	$Q5_0$	$Q6_0$
L	H	↑	D0	D1	D2	D3	D4	D5	D6	D7

74-164 8-bit Serial-to-Parallel Shift Register, Master Reset Low

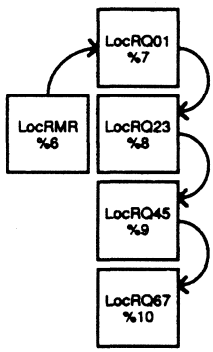
Logic



Symbol



Sample Placement

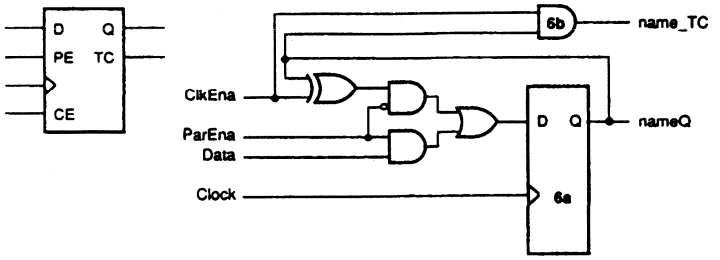


Truth Table

MRLow	Clock	DataA	DataB	Outputs							
				Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7
L	X	X	X	L	L	L	L	L	L	L	L
H	L	X	X	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$	$Q4_0$	$Q5_0$	$Q6_0$	$Q7_0$
H	↑	L	X	L	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$	$Q4_0$	$Q5_0$	$Q6_0$
H	↑	X	L	L	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$	$Q4_0$	$Q5_0$	$Q6_0$
H	↑	H	H	H	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$	$Q4_0$	$Q5_0$	$Q6_0$

C2BCP 1-bit Binary Counter, Clock Enable, Parallel Enable

Logic

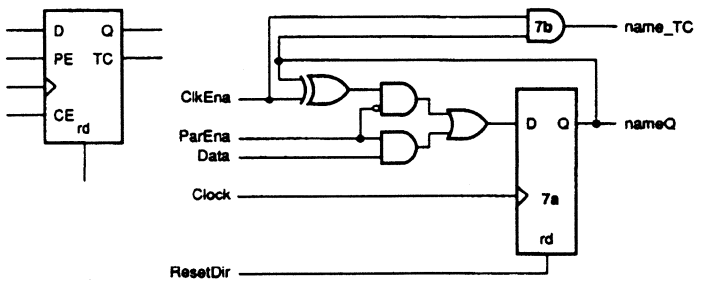


Truth Table

ParEna	ClkEna	Clock	Q
X	X	L	Q_0
H	X	↑	Data
L	L	↑	Q_0
L	H	↑	$\overline{Q_0}$

C2BCPRD 1-bit Binary Counter, Clock Enable, Parallel Enable, Reset Direct

Logic

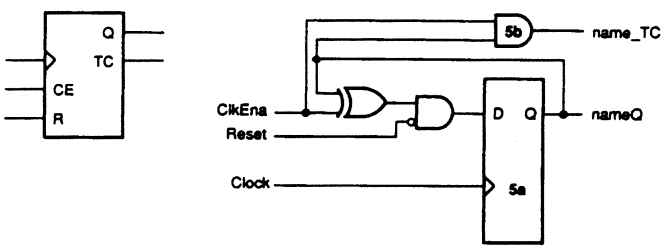


Truth Table

ResetDir	ParEna	ClkEna	Clock	Q
H	X	X	X	L
L	X	X	L	Q_0
L	H	X	↑	Data
L	L	L	↑	Q_0
L	L	H	↑	$\overline{Q_0}$

C2BCR 1-bit Binary Counter, Clock Enable, Reset

Logic

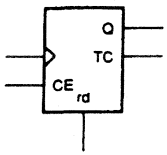


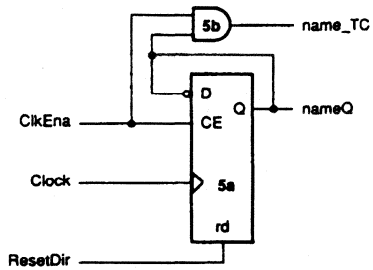
Truth Table

ResetDir	ClkEna	Clock	Q
H	X	X	L
L	X	L	Q_0
L	L	↑	Q_0
L	H	↑	$\overline{Q_0}$

C2BCRD 1-bit Binary Counter, Clock Enable, Reset Direct

Logic



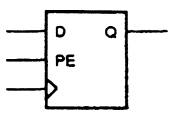


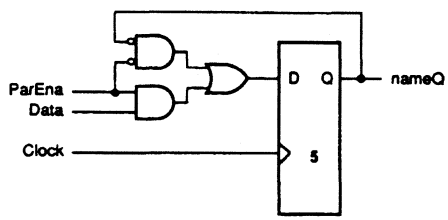
Truth Table

ResetDir	ClkEna	Clock	Q
H	X	X	L
L	X	L	Q_0
L	L	↑	Q_0
L	H	↑	$\overline{Q}_0$

C2BP 1-bit Binary Counter, Parallel Enable

Logic



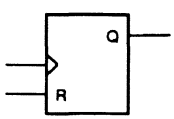


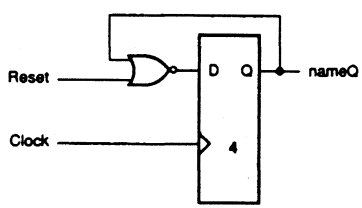
Truth Table

ParEna	Clock	Q
X	L	Q_0
H	↑	Data
L	↑	$\overline{Q}_0$

C2BR 1-bit Binary Counter, Reset

Logic



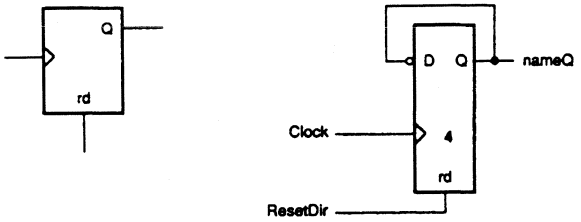


Truth Table

Reset	Clock	Q
X	L	Q_0
H	↑	L
L	↑	$\overline{Q}_0$

C2BRD 1-bit Binary Counter, Reset Direct

Logic

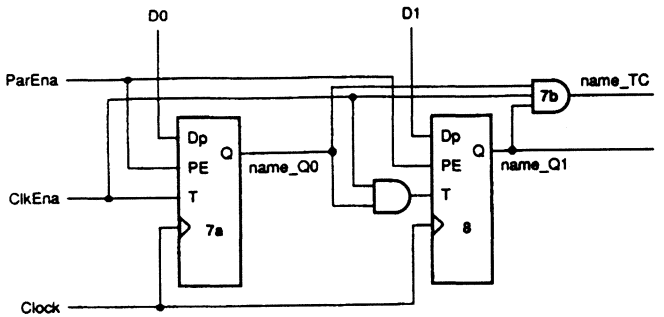


Truth Table

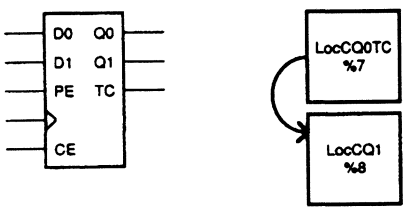
ResetDir	Clock	Q
H	X	L
L	L	$Q0_0$
L	↑	$\overline{Q0_0}$

C4BCP 2-bit Binary Counter, Clock Enable, Parallel Enable

Logic



Symbol

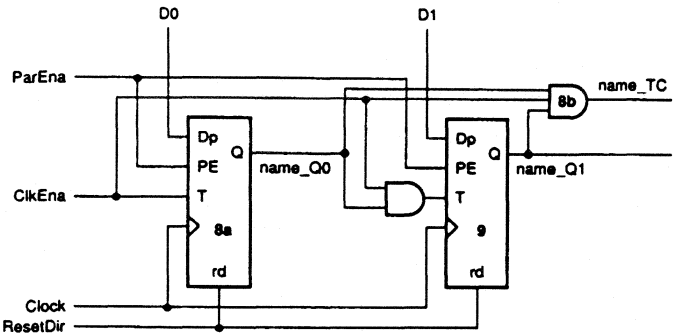


Truth Table

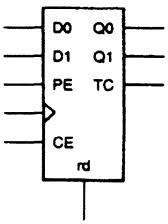
ParEna	ClkEna	Clock	Count Q0 Q1
X	X	L	$Q0_0$ $Q1_0$
H	X	↑	D0 D1
L	L	↑	$Q0_0$ $Q1_0$
L	H	↑	Count + 1

C4BCPRD 2-bit Binary Counter, Clock Enable, Parallel Enable, Reset Direct

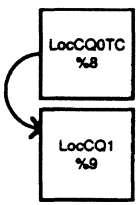
Logic



Symbol



Sample Placement

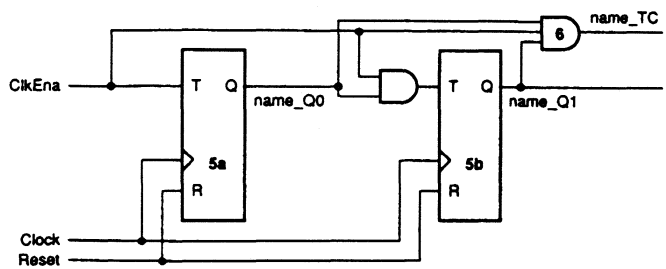


Truth Table

ResetDir	ParEna	ClkEna	Clock	Count	
				Q0	Q1
H	X	X	X	L	L
L	X	X	L	$Q0_0$	$Q1_0$
L	H	X	↑	D0	D1
L	L	L	↑	$Q0_0$	$Q1_0$
L	L	H	↑	Count + 1	

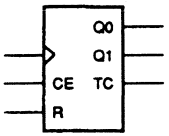
C4BCR 2-bit Binary Counter, Clock Enable, Reset

Logic

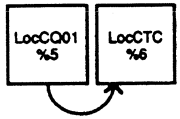


C4BCR 2-bit Binary Counter, Clock Enable, Reset (continued)

Symbol



Sample Placement

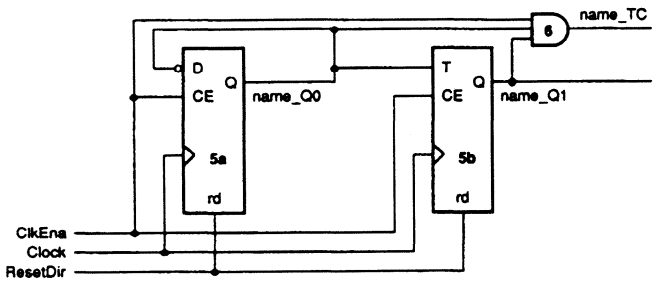


Truth Table

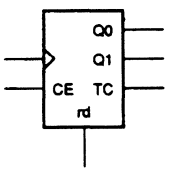
Reset	ClkEna	Clock	Count Q0	Count Q1
X	X	L	$Q0_0$	$Q1_0$
H	X	↑	L	L
L	L	↑	$Q0_0$	$Q1_0$
L	H	↑	Count + 1	

C4BCRD 2-bit Binary Counter, Clock Enable, Reset Direct

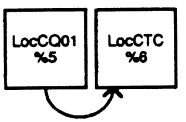
Logic



Symbol



Sample Placement

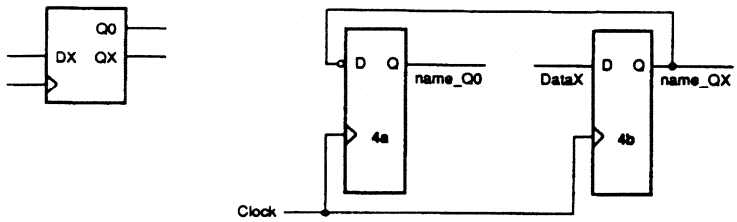


Truth Table

ResetDir	ClkEna	Clock	Count Q0	Count Q1
H	X	X	L	L
L	X	L	$Q0_0$	$Q1_0$
L	L	↑	$Q0_0$	$Q1_0$
L	H	↑	Count + 1	

C4JX 2-bit Expandable Johnson Counter

Logic

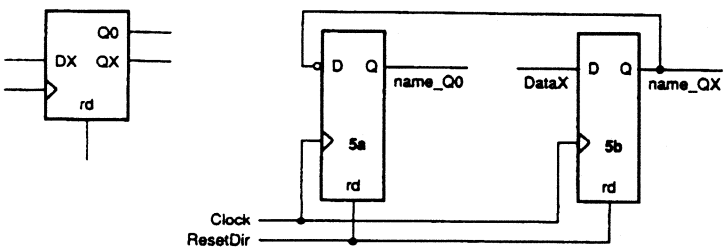


Truth Table

ClkEna	Clock	Count	
		Q0	QX
X	L	$Q0_0$	QX_0
L	X	$Q0_0$	QX_0
H	↑	$\overline{QX}_0$	DX

C4JXRD 2-bit Expandable Johnson Counter, Reset Direct

Logic

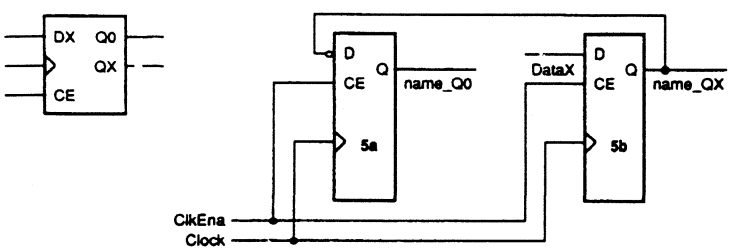


Truth Table

ResetDir	Clock	Count	
		Q0	QX
H	X	L	L
L	L	$Q0_0$	QX_0
L	↑	$\overline{QX}_0$	DX

C4JXC 2-bit Expandable Johnson Counter, Clock Enable

Logic

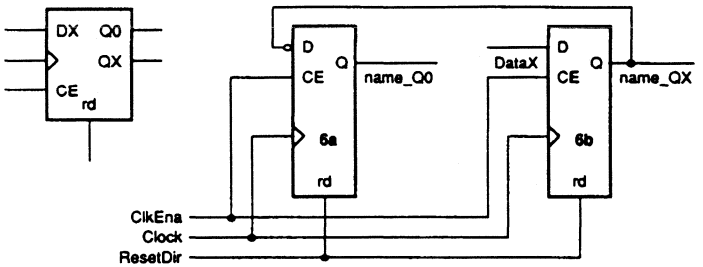


Truth Table

ClkEna	Clock	Count	
		Q0	QX
X	L	$Q0_0$	QX_0
L	X	$Q0_0$	QX_0
H	↑	$\overline{QX}_0$	DX

C4JXCRD 2-bit Expandable Johnson Counter, Clock Enable, Reset Direct

Logic

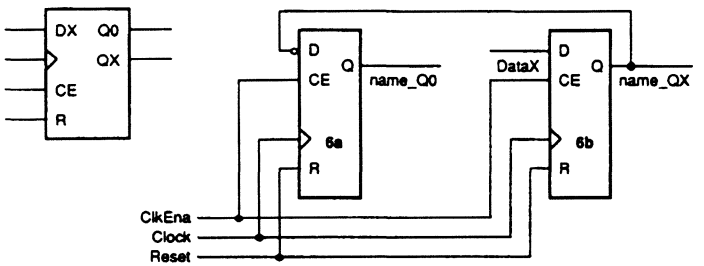


Truth Table

ResetDir	ClkEna	Clock	Count	
			Q0	QX
H	X	X	L	L
L	X	L	$Q0_0$	QX_0
L	L	↑	$Q0_0$	QX_0
L	H	↑	$\overline{QX}_0$	DX

C4JXCR 2-bit Expandable Johnson Counter, Clock Enable, Reset

Logic

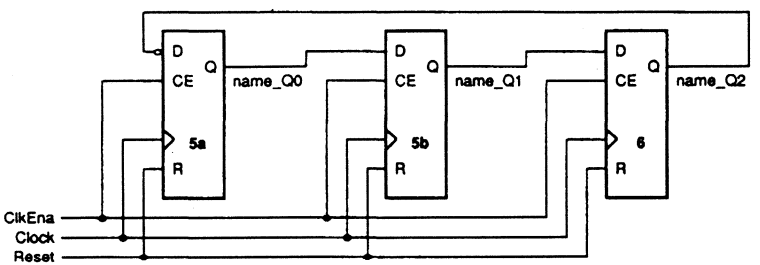


Truth Table

Reset	ClkEna	Clock	Count	
			Q0	QX
X	X	L	$Q0_0$	QX_0
H	X	↑	L	L
L	L	↑	$Q0_0$	QX_0
L	H	↑	$\overline{QX}_0$	DX

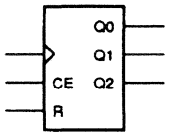
C6JCR 3-bit Johnson Counter, Clock Enable, Reset

Logic

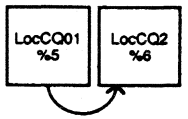


C6JCR 3-bit Johnson Counter, Clock Enable, Reset (continued)

Symbol



Sample Placement

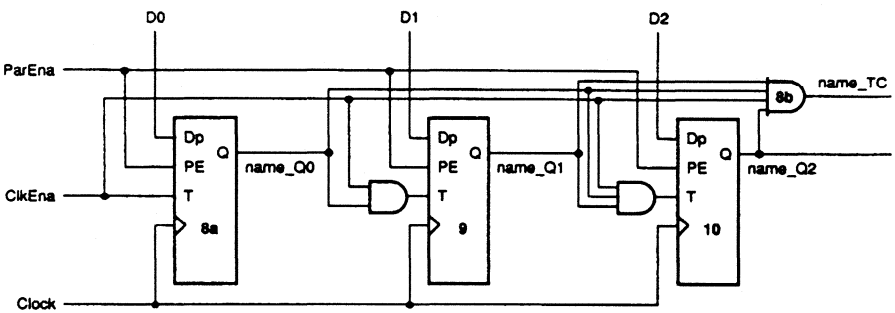


Truth Table

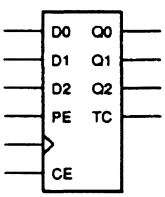
Reset	ClkEna	Clock	Count		
			Q0	Q1	Q2
X	X	L	$Q0_0$	$Q1_0$	$Q2_0$
H	X	↑	L	L	L
L	L	↑	$Q0_0$	$Q1_0$	$Q2_0$
L	H	↑	$\overline{Q2}_0$	$Q0_0$	$Q1_0$

C8BCP 3-bit Binary Counter, Clock Enable, Parallel Enable

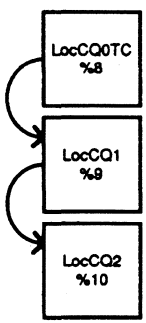
Logic



Symbol



Sample Placement

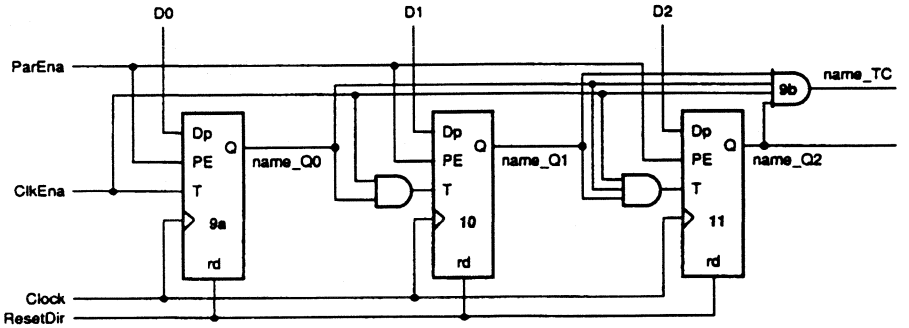


Truth Table

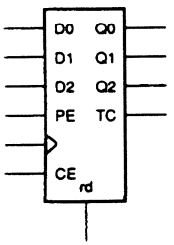
ParEna	ClkEna	Clock	Count		
			Q0	Q1	Q2
X	X	L	$Q0_0$	$Q1_0$	$Q2_0$
H	X	↑	D0	D1	D2
L	L	↑	$Q0_0$	$Q1_0$	$Q2_0$
L	H	↑	Count + 1		

C8BCPRD 3-bit Binary Counter, Clock Enable, Parallel Enable, Reset Direct

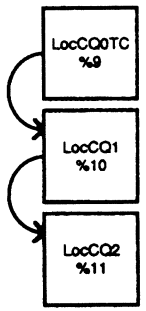
Logic



Symbol



Sample Placement

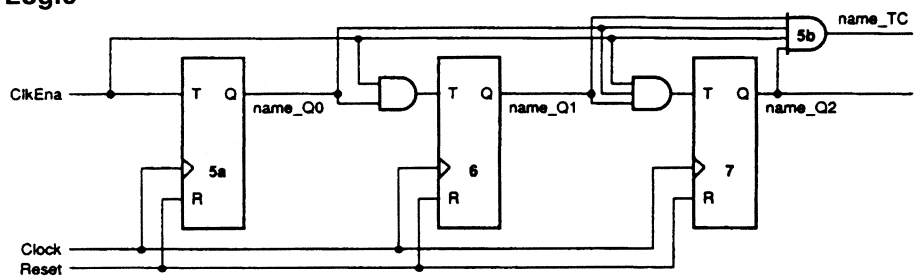


Truth Table

ResetDir	ParEna	ClkEna	Clock	Count Q0	Count Q1	Count Q2
H	X	X	X	L	L	L
L	X	X	L	$Q0_0$	$Q1_0$	$Q2_0$
L	H	X	↑	D0	D1	D2
L	L	L	↑	$Q0_0$	$Q1_0$	$Q2_0$
L	L	H	↑	Count + 1		

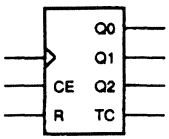
C8BCR 3-bit Binary Counter, Clock Enable, Reset

Logic

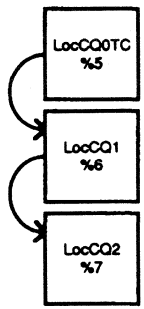


C8BCR 3-bit Binary Counter, Clock Enable, Reset (continued)

Symbol



Sample Placement

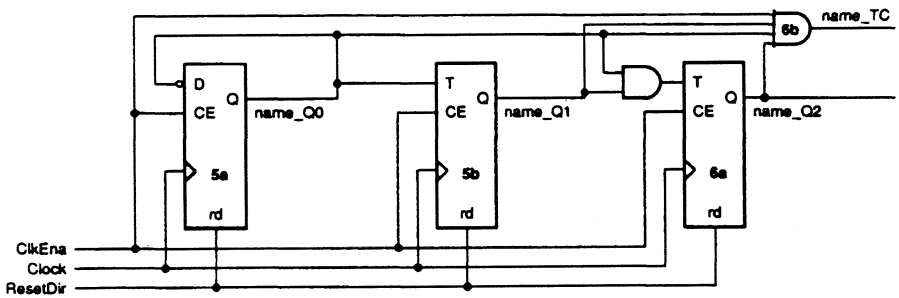


Truth Table

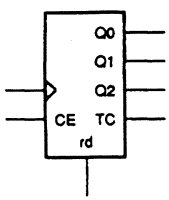
Reset	ClkEna	Clock	Count		
			Q0	Q1	Q2
X	X	L	$Q0_0$	$Q1_0$	$Q2_0$
H	X	↑	L	L	L
L	L	↑	$Q0_0$	$Q1_0$	$Q2_0$
L	H	↑	Count + 1		

C8BCRD 3-bit Binary Counter, Clock Enable, Reset Direct

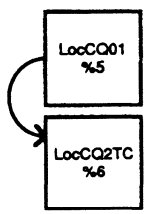
Logic



Symbol



Sample Placement

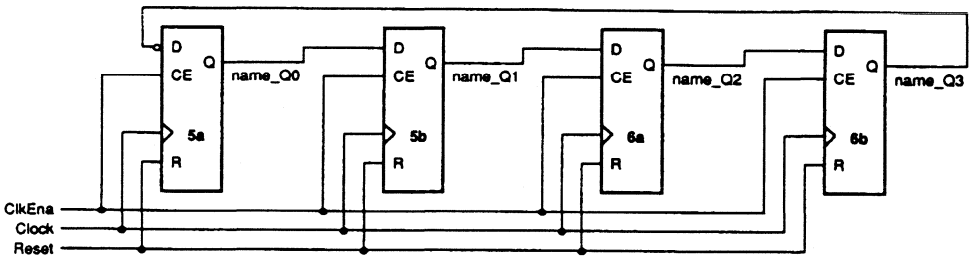


Truth Table

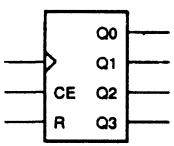
ResetDir	ClkEna	Clock	Count		
			Q0	Q1	Q2
H	X	X	L	L	L
L	X	L	$Q0_0$	$Q1_0$	$Q2_0$
L	L	↑	$Q0_0$	$Q1_0$	$Q2_0$
L	H	↑	Count + 1		

C8JCR 4-bit Johnson Counter, Clock Enable, Reset

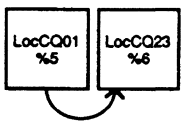
Logic



Symbol



Sample Placement

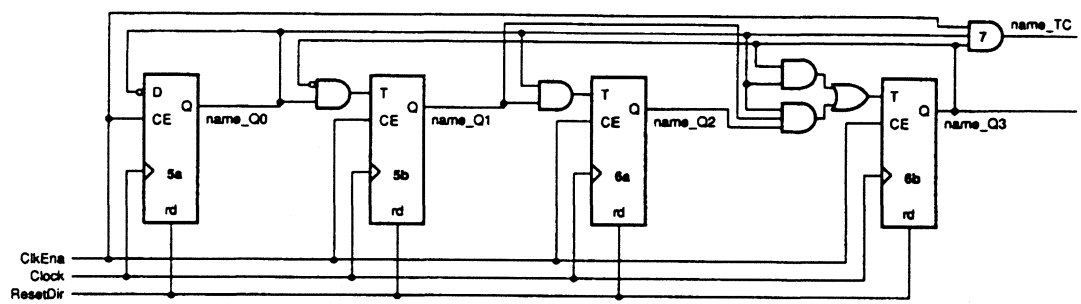


Truth Table

Reset	ClkEna	Clock	Count			
			Q0	Q1	Q2	Q3
X	X	L	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$
H	X	↑	L	L	L	L
L	L	↑	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$
L	H	↑	$\overline{Q3_0}$	$Q0_0$	$Q1_0$	$Q2_0$

C10BCRD 4-bit BCD Counter, Clock Enable, Parallel Enable, Reset Direct

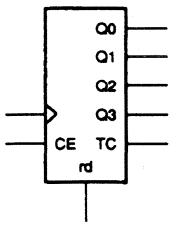
Logic



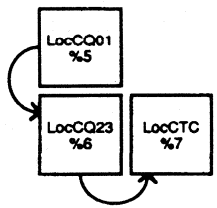
C10BCRD 4-bit BCD Counter, Clock Enable, Parallel Enable, Reset Direct

(continued)

Symbol



Sample Placement

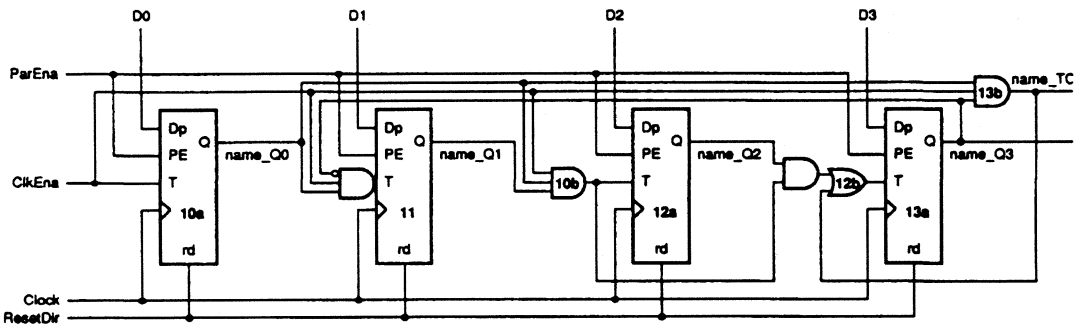


Truth Table

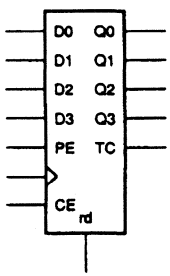
ResetDir	ClkEna	Clock	Count			
			Q0	Q1	Q2	Q3
H	X	X	L	L	L	L
L	X	L	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$
L	L	↑	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$
L	H	↑	Next Count			

C10BCPRD 4-bit BCD Counter, Clock Enable, Parallel Enable, Reset Direct

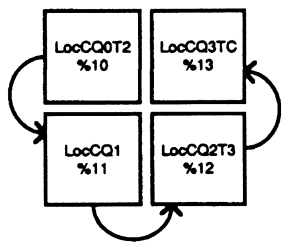
Logic



Symbol



Sample Placement

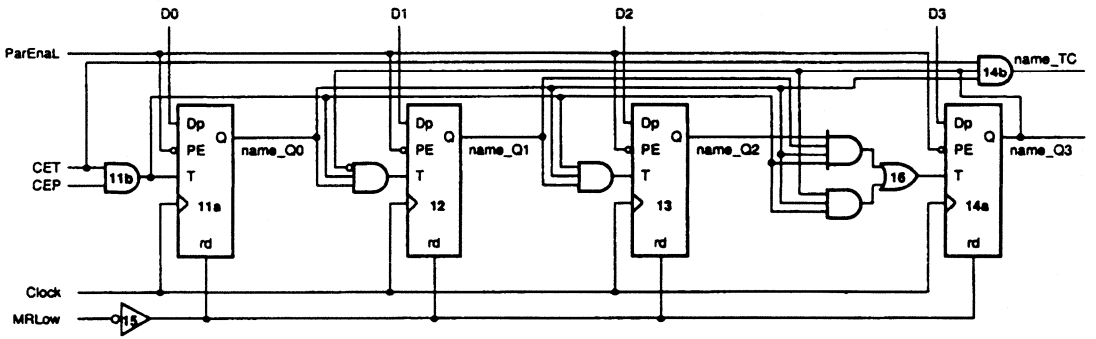


Truth Table

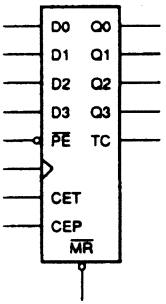
ResetDir	ParEna	ClkEna	Clock	Count			
				Q0	Q1	Q2	Q3
H	X	X	X	L	L	L	L
L	X	X	L	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$
L	H	X	↑	D0	D1	D2	D3
L	L	L	↑	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$
L	L	H	↑	Next Count			

74-160 4-bit BCD Counter, Clock Enable, Parallel Enable Low, Master Reset Low

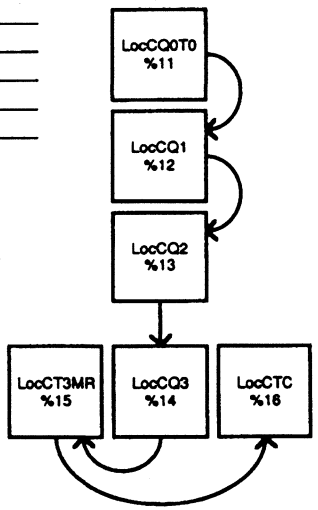
Logic



Symbol



Sample Placement

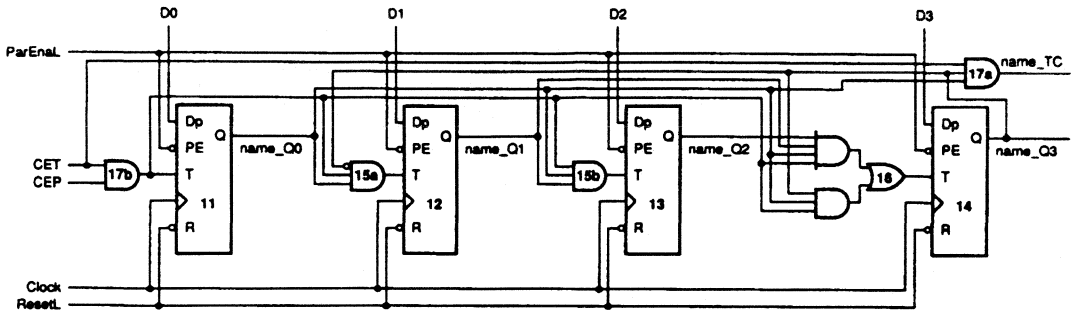


Truth Table

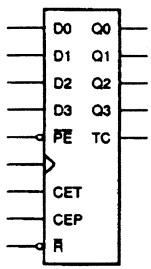
MRLow	ParEnaL	CET	CEP	Clock	Count			
					Q0	Q1	Q2	Q3
L	X	X	X	X	L	L	L	L
H	X	X	X	L	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$
H	L	X	X	↑	D0	D1	D2	D3
H	H	L	X	↑	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$
H	H	X	L	↑	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$
H	H	H	H	↑	Next Count			

74-162 4-bit BCD Counter, Clock Enable, Parallel Enable Low, Reset Low

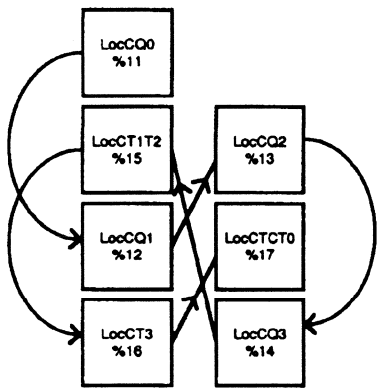
Logic



Symbol



Sample Placement

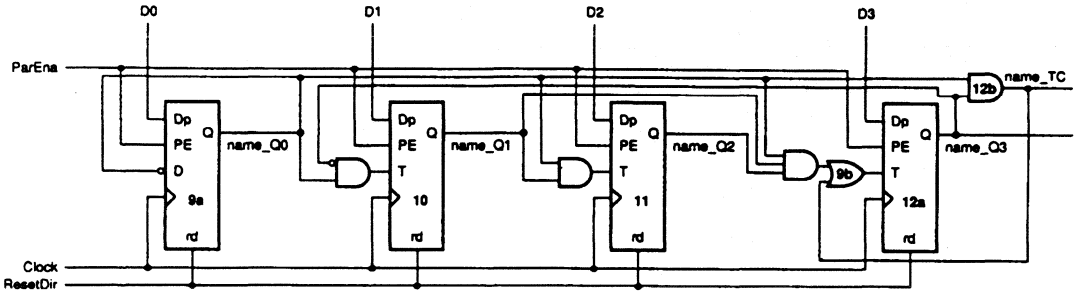


Truth Table

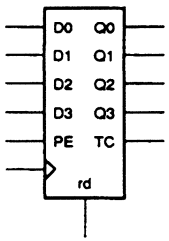
ResetL	ParEnaL	CET	CEP	Clock	Count			
					Q0	Q1	Q2	Q3
X	X	X	X	L	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$
L	X	X	X	↑	L	L	L	L
H	L	X	X	↑	D0	D1	D2	D3
H	H	L	X	↑	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$
H	H	X	L	↑	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$
H	H	H	H	↑	Next Count			

C10BPRD 4-bit BCD Counter, Parallel Enable, Reset Direct

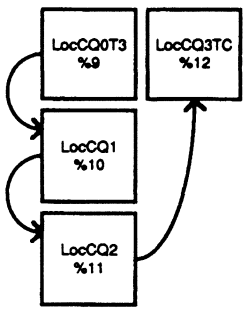
Logic



Symbol



Sample Placement

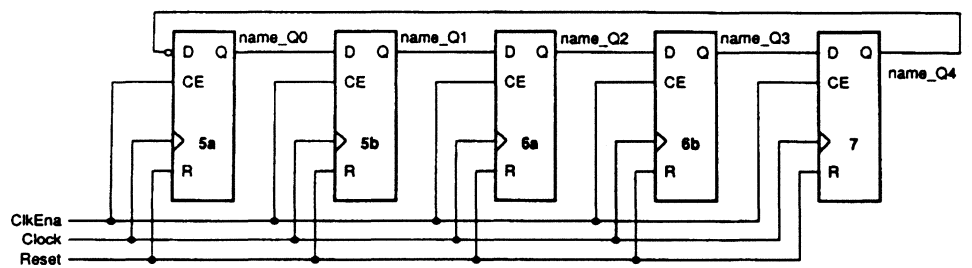


Truth Table

ResetDir	ParEna	Clock	Count			
			Q0	Q1	Q2	Q3
H	X	X	L	L	L	L
L	X	L	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$
L	L	↑	Next Count			
L	H	↑	D0	D1	D2	D3

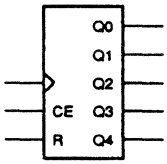
C10JCR 5-bit Johnson Counter, Clock Enable, Reset

Logic

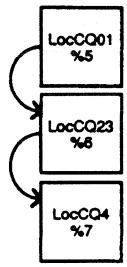


C10JCR 5-bit Johnson Counter, Clock Enable, Reset (continued)

Symbol



Sample Placement

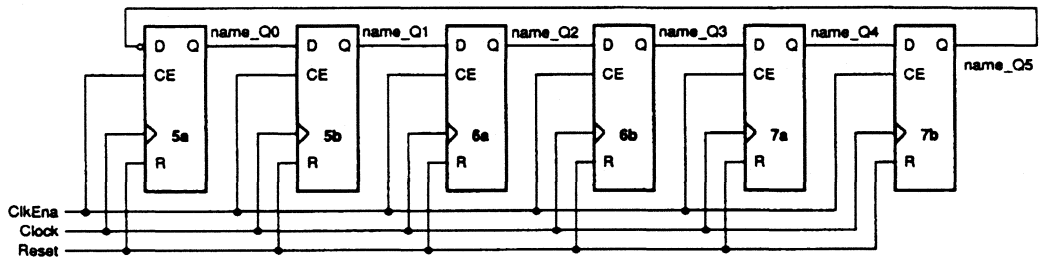


Truth Table

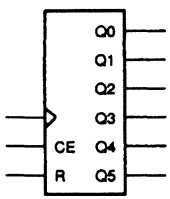
Reset	ClkEna	Clock	Count				
			Q0	Q1	Q2	Q3	Q4
X	X	L	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$	$Q4_0$
H	X	↑	L	L	L	L	L
L	L	↑	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$	$Q4_0$
L	H	↑	$\overline{Q4_0}$	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$

C12JCR 6-bit Johnson Counter, Clock Enable, Reset

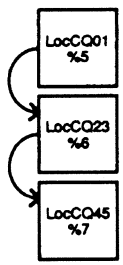
Logic



Symbol



Sample Placement

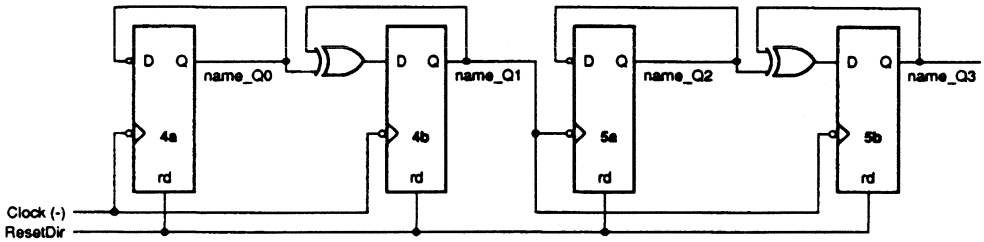


Truth Table

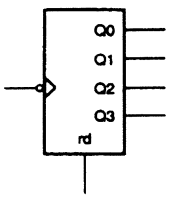
Reset	ClkEna	Clock	Count					
			Q0	Q1	Q2	Q3	Q4	Q5
X	X	L	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$	$Q4_0$	$Q5_0$
H	X	↑	L	L	L	L	L	L
L	L	↑	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$	$Q4_0$	$Q5_0$
L	H	↑	$\overline{Q5_0}$	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$	$Q4_0$

C16BARD 4-bit Binary Ripple Counter, Reset Direct

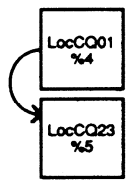
Logic



Symbol



Sample Placement

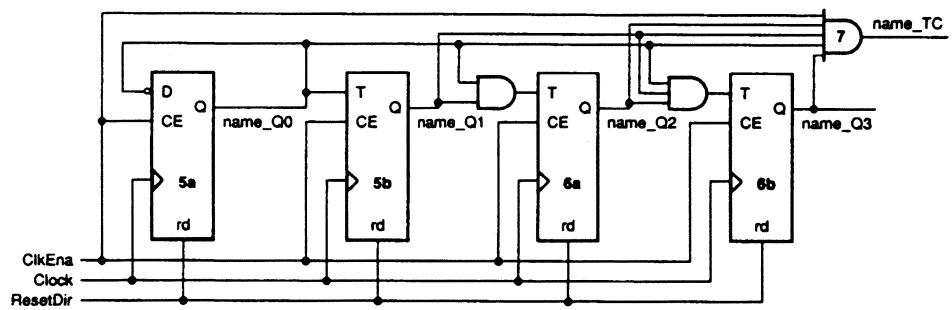


Truth Table

ResetDir	Clock	Count			
		Q0	Q1	Q2	Q3
H	X	L	L	L	L
L	H	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$
L	↓	Count + 1			

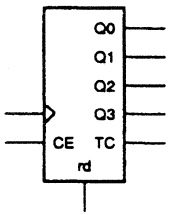
C16BCRD 4-bit Binary Counter, Clock Enable, Reset Direct

Logic

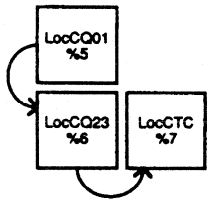


C16BCRD 4-bit Binary Counter, Clock Enable, Reset Direct (continued)

Symbol



Sample Placement

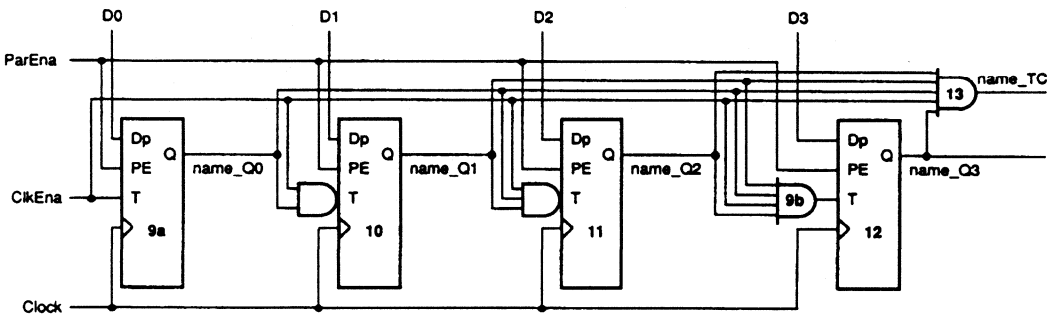


Truth Table

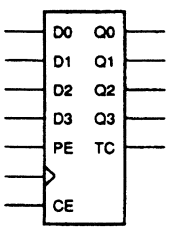
ResetDir	ClkEna	Clock	Count			
			Q0	Q1	Q2	Q3
H	X	X	L	L	L	L
L	X	L	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$
L	L	↑	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$
L	H	↑	Count + 1			

C16BCP 4-bit Binary Counter, Clock Enable, Parallel Enable

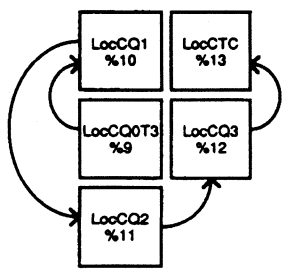
Logic



Symbol



Sample Placement

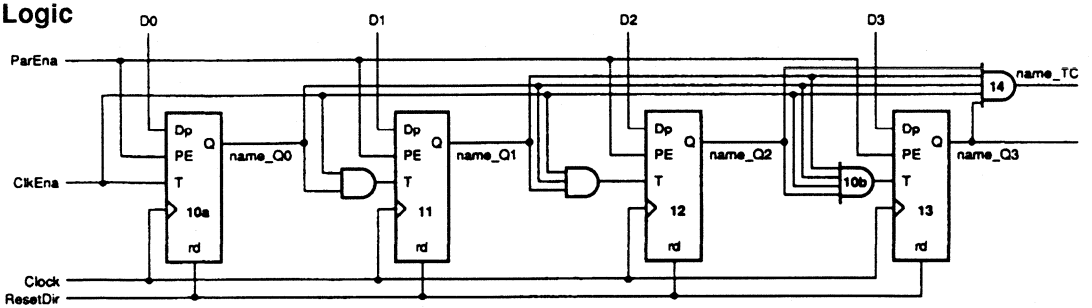


Truth Table

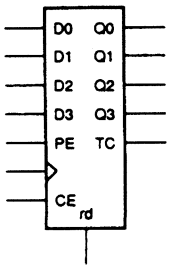
ParEna	ClkEna	Clock	Count			
			Q0	Q1	Q2	Q3
X	X	L	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$
H	X	↑	D0	D1	D2	D3
L	L	↑	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$
L	H	↑	Count + 1			

C16BCPRD 4-bit Binary Counter, Clock Enable, Parallel Enable, Reset Direct

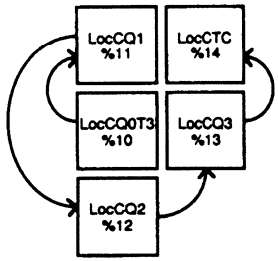
Logic



Symbol



Sample Placement

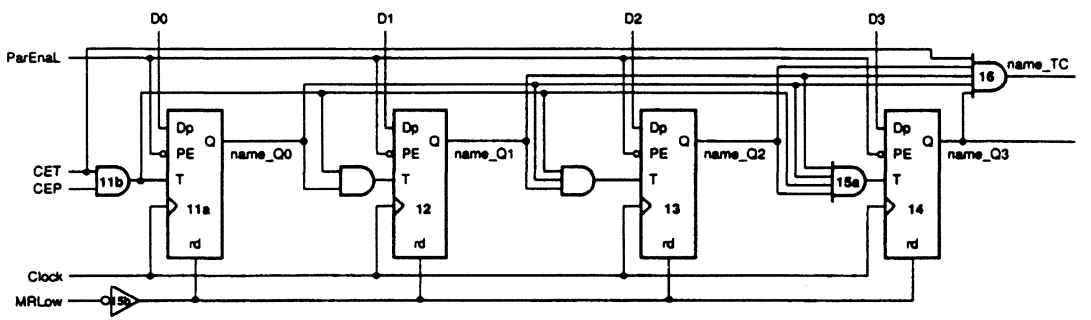


Truth Table

ResetDir	ParEna	ClkEna	Clock	Count			
				Q0	Q1	Q2	Q3
H	X	X	X	L	L	L	L
L	X	X	L	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$
L	H	X	↑	D0	D1	D2	D3
L	L	L	↑	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$
L	L	H	↑	Count + 1			

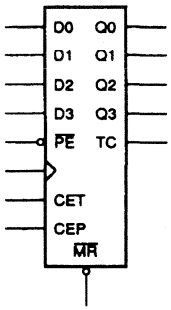
74-161 4-bit Binary Counter, Clock Enable, Parallel Enable Low, Master Reset Low

Logic

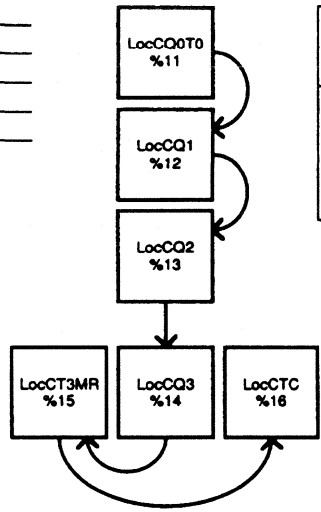


74-161 4-bit Binary Counter, Clock Enable, Parallel Enable Low, Master Reset Low (continued)

Symbol



Sample Placement

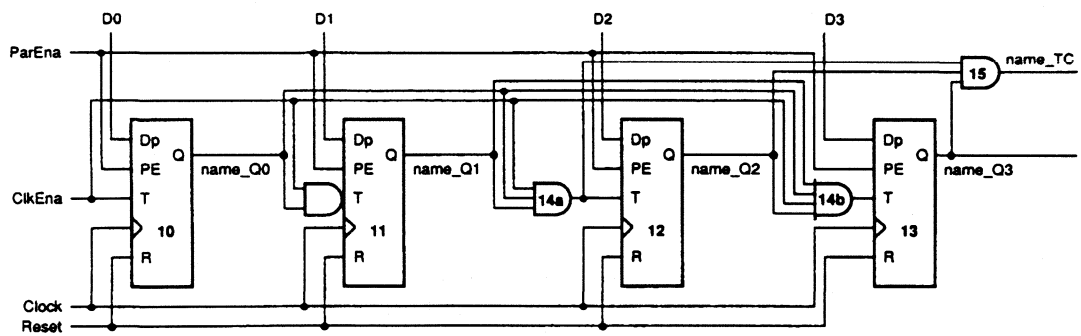


Truth Table

MRLow	ParEnaL	CET	CEP	Clock	Count Q0	Count Q1	Count Q2	Count Q3
L	X	X	X	X	L	L	L	L
H	X	X	X	L	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$
H	L	X	X	↑	D0	D1	D2	D3
H	H	L	X	↑	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$
H	H	H	L	↑	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$
H	H	H	H	↑	Count + 1			

C16BCPR 4-bit Binary Counter, Clock Enable, Parallel Enable, Reset

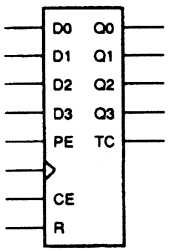
Logic



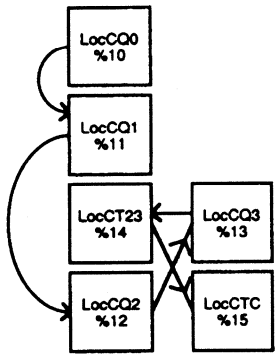
C16BCPR 4-bit Binary Counter, Clock Enable, Parallel Enable, Reset

(continued)

Symbol



Sample Placement

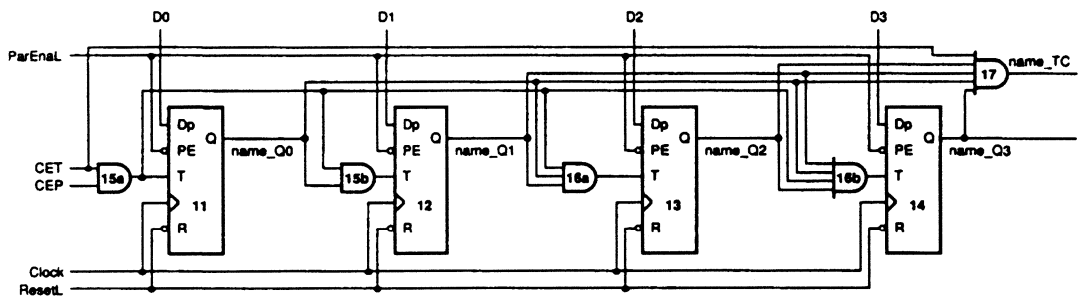


Truth Table

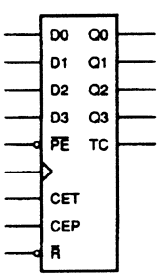
Reset	ParEna	ClkEna	Clock	Count			
				Q0	Q1	Q2	Q3
X	X	X	L	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$
H	X	X	↑	L	L	L	L
L	H	X	↑	D0	D1	D2	D3
L	L	L	↑	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$
L	L	H	↑	Count + 1			

74-163 4-bit Binary Counter, Clock Enable, Parallel Enable Low, Reset Low

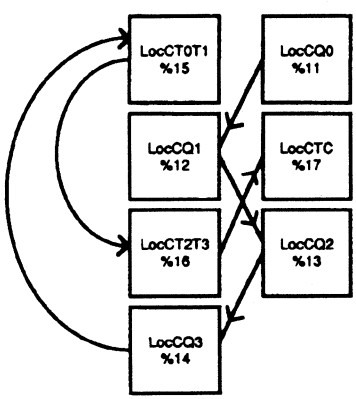
Logic



Symbol



Sample Placement

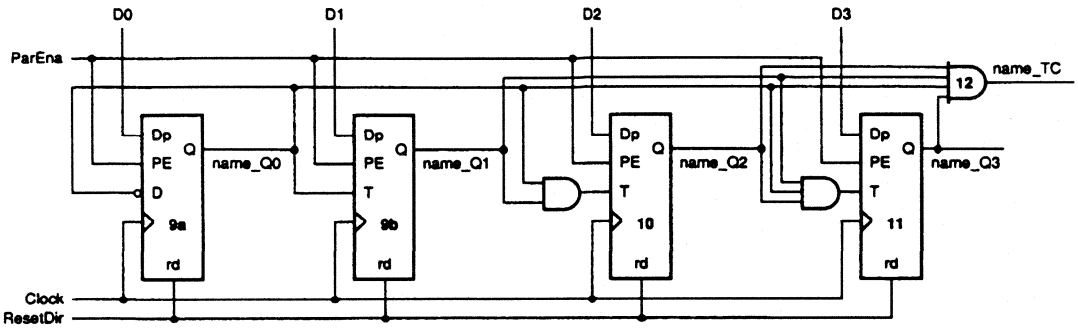


Truth Table

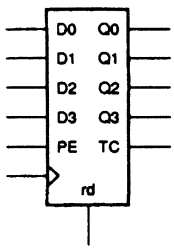
ResetL	ParEnaL	CET	CEP	Clock	Count			
					Q0	Q1	Q2	Q3
X	X	X	X	L	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$
L	X	X	X	↑	L	L	L	L
H	L	X	X	↑	D0	D1	D2	D3
H	H	L	X	↑	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$
H	H	X	L	↑	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$
H	H	H	H	↑	Count + 1			

C16BPRD 4-bit Binary Counter, Parallel Enable, Reset Direct

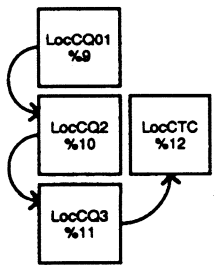
Logic



Symbol



Sample Placement

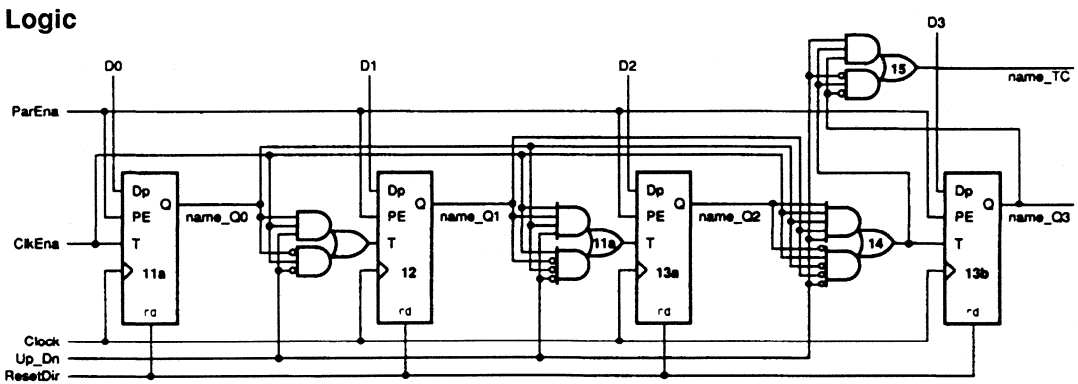


Truth Table

ResetDir	ParEna	Clock	Count			
			Q0	Q1	Q2	Q3
H	X	X	L	L	L	L
L	X	L	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$
L	L	↑	Count + 1			
L	H	↑	D0	D1	D2	D3

C16BUDRD 4-bit Binary Up/Down Counter, Parallel Enable, Reset Direct

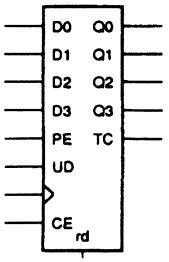
Logic



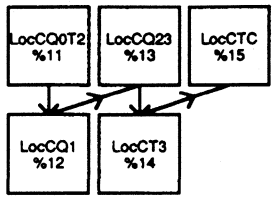
C16BUDRD 4-bit Binary Up/Down Counter, Parallel Enable, Reset Direct

(continued)

Symbol



Sample Placement

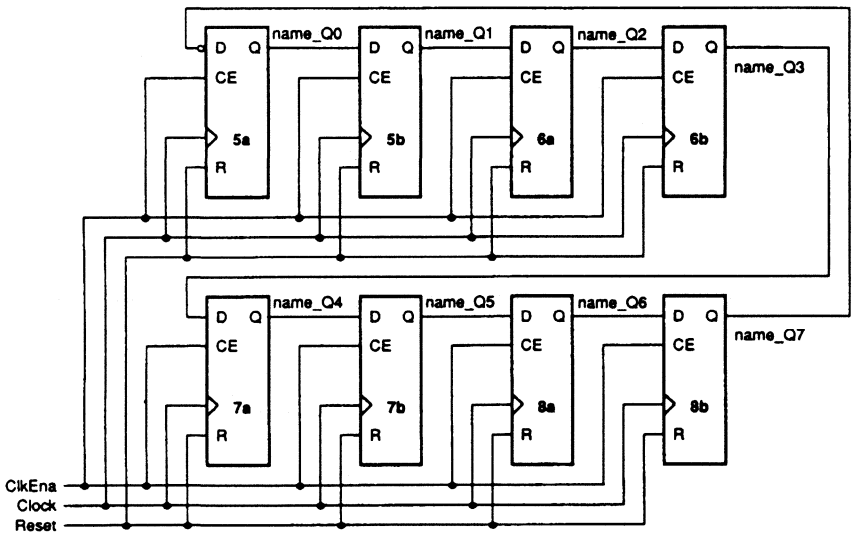


Truth Table

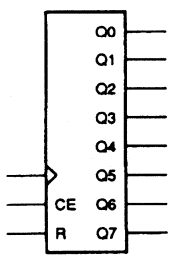
ResetDir	ParEna	ClkEna	Up_Dn	Clock	Count			
					Q0	Q1	Q2	Q3
H	X	X	X	X	L	L	L	L
L	X	X	X	L	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$
L	H	X	X	↑	D0	D1	D2	D3
L	L	L	X	↑	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$
L	L	H	L	↑	Count - 1			
L	L	H	H	↑	Count + 1			

C16JCR 8-bit Johnson Counter, Clock Enable, Reset

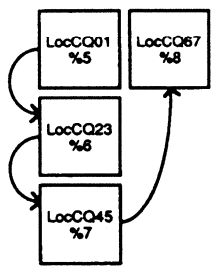
Logic



Symbol



Sample Placement

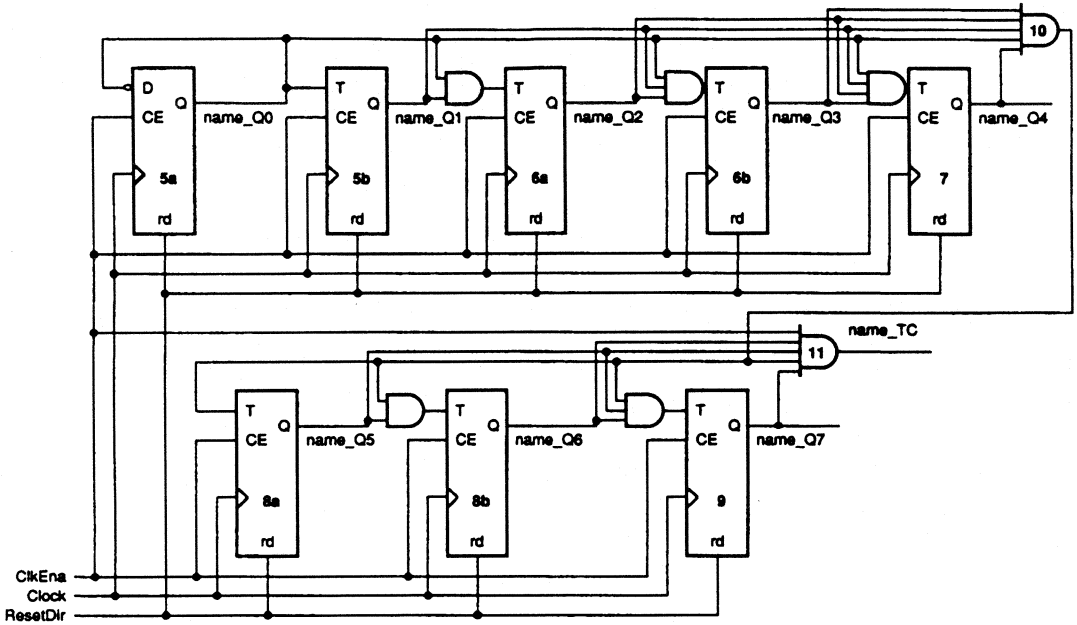


Truth Table

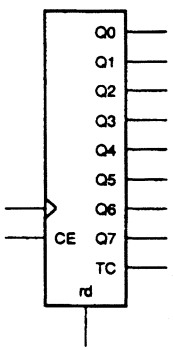
Reset	ClkEna	Clock	Count							
			Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7
X	X	L	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$	$Q4_0$	$Q5_0$	$Q6_0$	$Q7_0$
H	X	↑	L	L	L	L	L	L	L	L
L	L	↑	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$	$Q4_0$	$Q5_0$	$Q6_0$	$Q7_0$
L	H	↑	$Q7_0$	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$	$Q4_0$	$Q5_0$	$Q6_0$

C256BCRD 8-bit Binary Counter, Clock Enable, Reset Direct

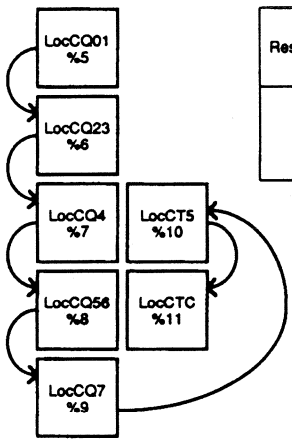
Logic



Symbol



Sample Placement

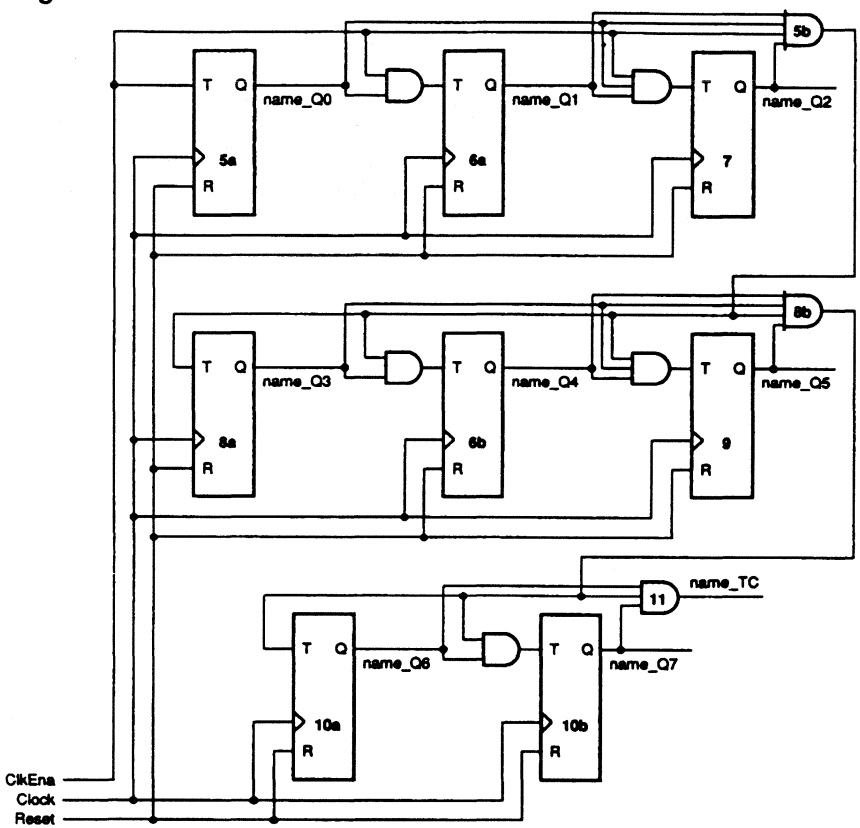


Truth Table

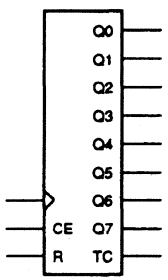
ResetDir	ClkEna	Clock	Count							
			Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7
H	X	X	L	L	L	L	L	L	L	L
L	X	L	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$	$Q4_0$	$Q5_0$	$Q6_0$	$Q7_0$
L	L	↑	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$	$Q4_0$	$Q5_0$	$Q6_0$	$Q7_0$
L	H	↑	Count + 1							

C256BCR 8-bit Binary Counter, Clock Enable, Reset

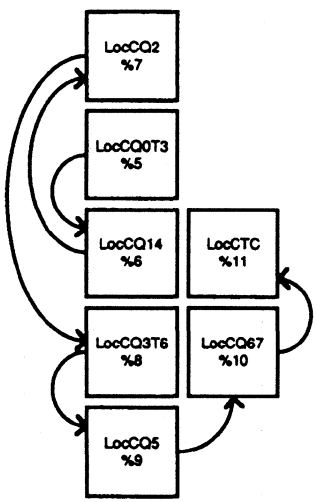
Logic



Symbol



Sample Placement

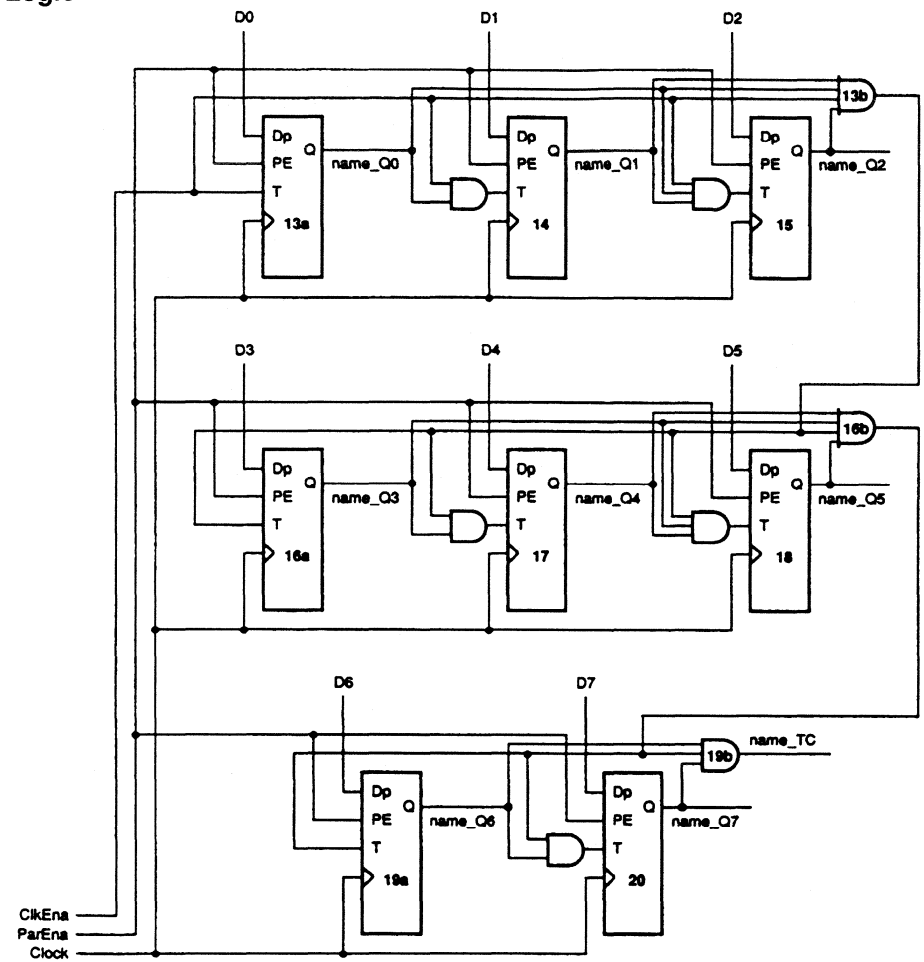


Truth Table

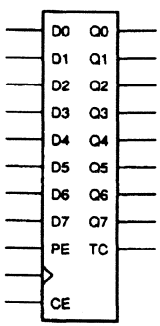
Reset	ClkEna	Clock	Count							
			Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7
X	X	L	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$	$Q4_0$	$Q5_0$	$Q6_0$	$Q7_0$
H	X	↑	L	L	L	L	L	L	L	L
L	L	↑	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$	$Q4_0$	$Q5_0$	$Q6_0$	$Q7_0$
L	H	↑	Count + 1							

C256BCP 8-bit Binary Counter, Clock Enable, Parallel Enable

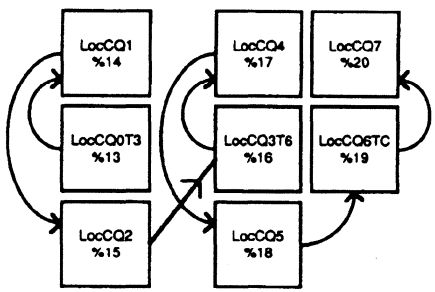
Logic



Symbol



Sample Placement

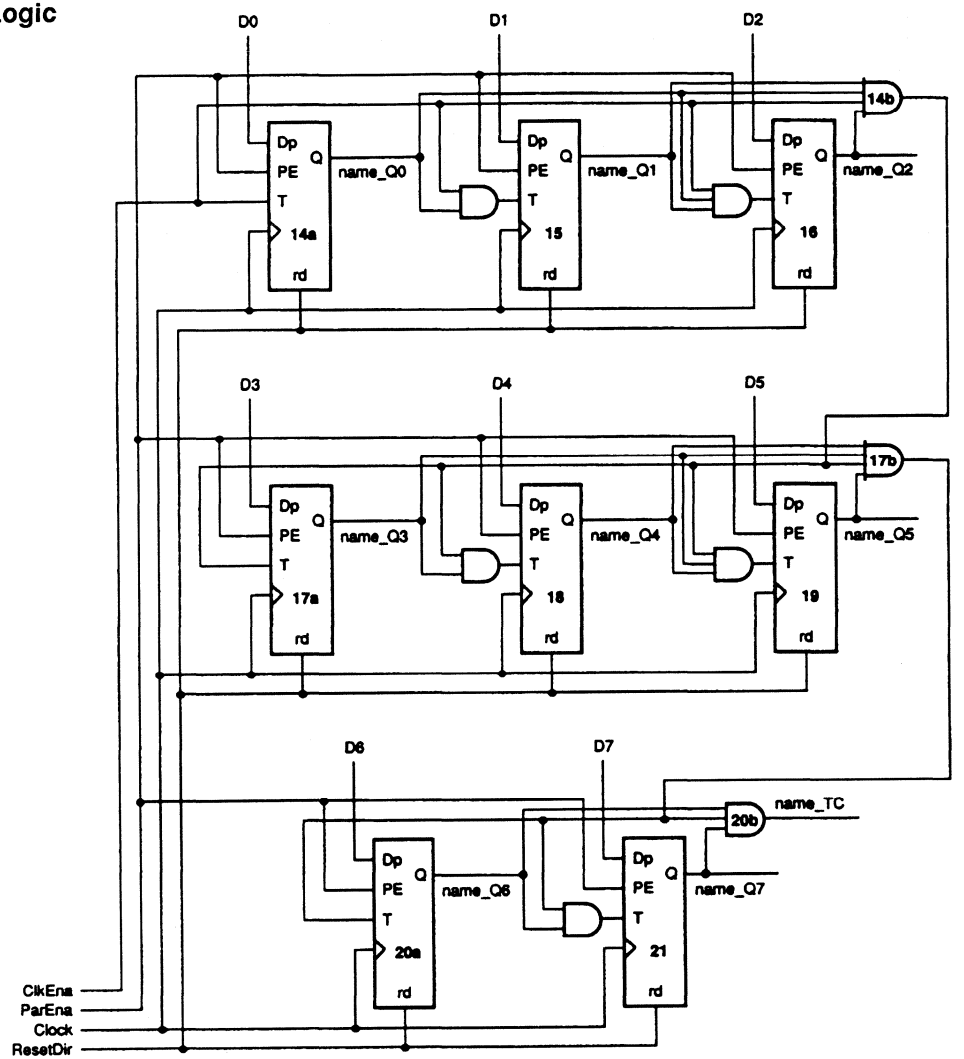


Truth Table

ParEna	ClkEna	Clock	Count							
			Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7
X	X	L	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$	$Q4_0$	$Q5_0$	$Q6_0$	$Q7_0$
H	X	↑	D0	D1	D2	D3	D4	D5	D6	D7
L	L	↑	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$	$Q4_0$	$Q5_0$	$Q6_0$	$Q7_0$
L	H	↑	Count + 1							

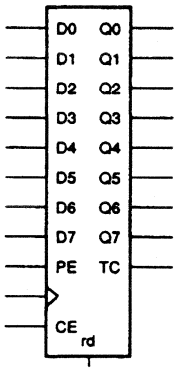
C256BCPRD 8-bit Binary Counter, Clock Enable, Parallel Enable, Reset Direct

Logic

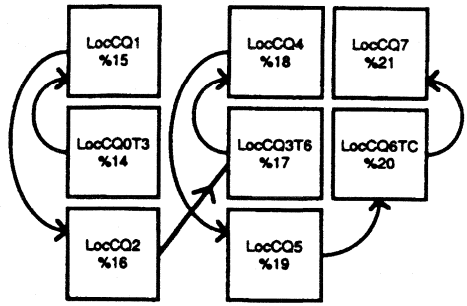


C256BCPRD 8-bit Binary Counter, Clock Enable, Parallel Enable, Reset Direct (continued)

Symbol



Sample Placement

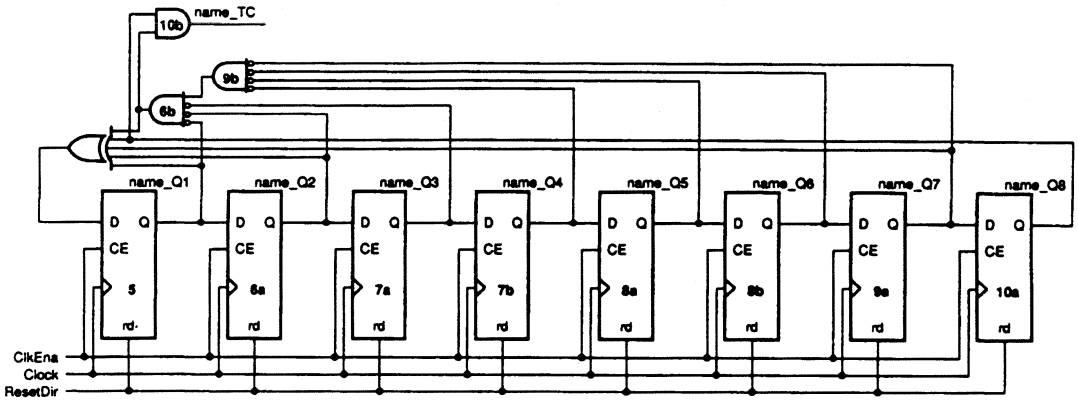


Truth Table

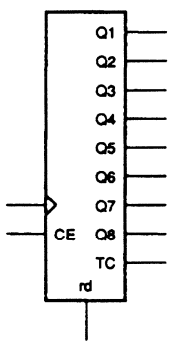
ResetDir	ParEna	ClkEna	Clock	Count							
				Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7
H	X	X	X	L	L	L	L	L	L	L	L
L	X	X	L	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$	$Q4_0$	$Q5_0$	$Q6_0$	$Q7_0$
L	H	X	↑	D0	D1	D2	D3	D4	D5	D6	D7
L	L	L	↑	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$	$Q4_0$	$Q5_0$	$Q6_0$	$Q7_0$
L	L	H	↑	Count + 1							

C256FCRD 8-bit Modulo 256 Feedback Shift Register, Clock Enable, Reset Direct

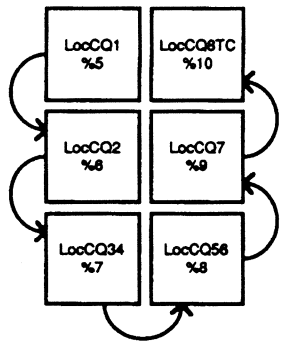
Logic



Symbol



Sample Placement



Truth Table

ResetDir	ClkEna	Clock	Count							
			Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7
H	X	X	L	L	L	L	L	L	L	L
L	X	L	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$	$Q4_0$	$Q5_0$	$Q6_0$	$Q7_0$
L	L	↑	$Q0_0$	$Q1_0$	$Q2_0$	$Q3_0$	$Q4_0$	$Q5_0$	$Q6_0$	$Q7_0$
L	H	↑	Next Count							

June 1992

AT&T Microelectronics

TTL 7400 Series Schematic Macro Library

Introduction

The TTL schematic macro library provides a convenient way for the user to enter logic functions using 55 familiar 7400 series MSI functions. The library functions and logic sense of signals are identical to the popular 7400 LS-TTL devices with two exceptions: 74390 and 74393 functions are implemented as synchronous rather than ripple counters for improved performance.

For TTL functions with totally independent elements, only a single element is implemented. For example, the 7477 function, consisting of two 2-bit latches, is implemented as a single 2-bit latch sharing a common enable signal. Refer to page 4-116 for the schematic drawing.

As shown in the following figures, the 7400 device pin numbers are used as net names which will be replaced with user names when interconnected. Active-low signals are implied by a trailing B in the signal name.

Also note that the convention for 3-state buffers (TBUFs and IOBs) is that a high level on the control signal provides a high impedance and a low level signal enables the buffer drive.

In addition, there are 51 custom MSI functions that are slight variations of standard TTL devices.

To enhance the routability and speed, all of these macros have been partitioned into CLBMAPs to take advantage of the FPGA architecture. Sample placement and route using direct interconnects have been included. Deviating from the suggested partitioning, placement, and routing will most likely decrease the performance.

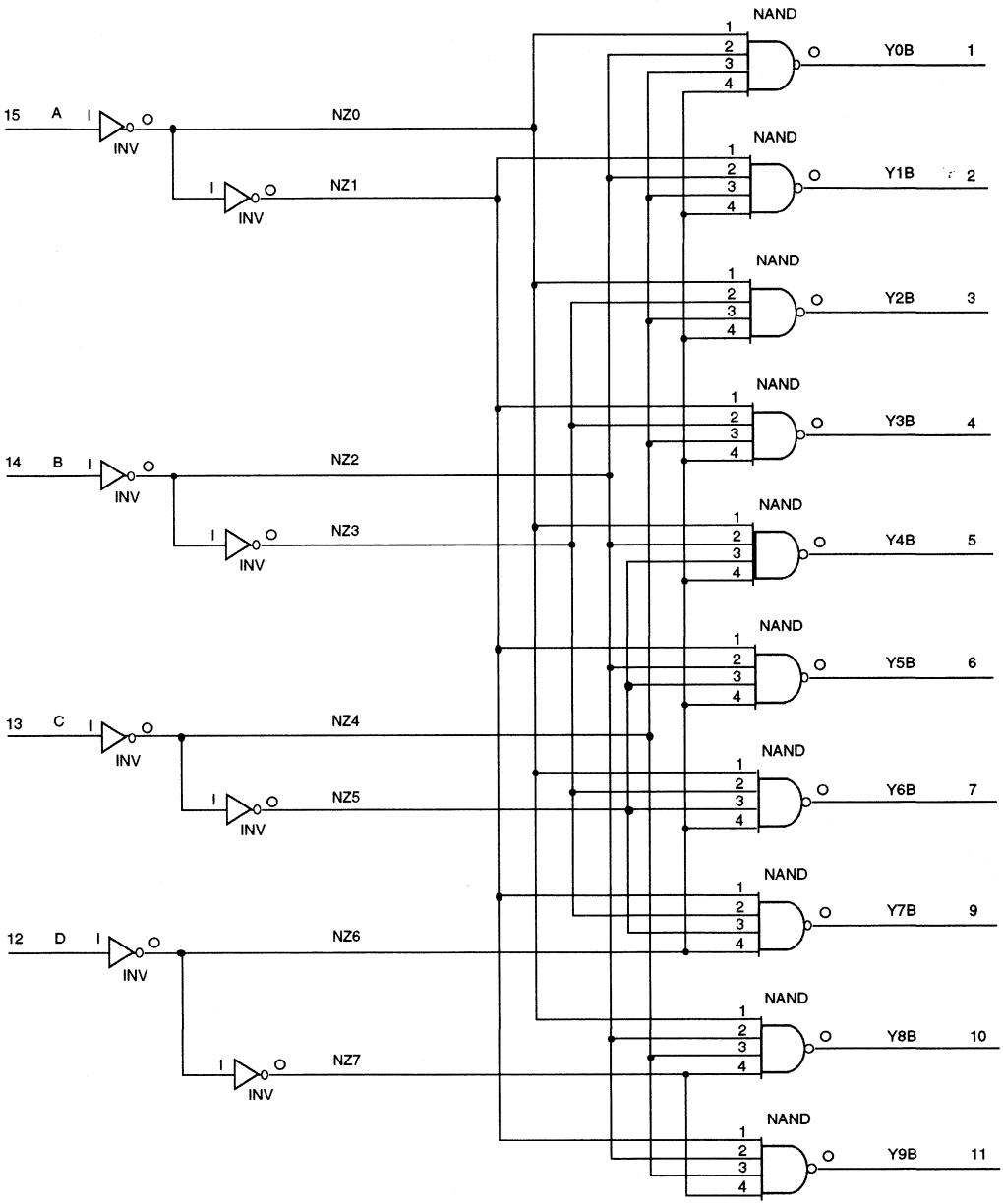
The TTL schematic macro library for *FutureNet-DASH* contains DWG and PIN files that install in the *XACT*\PIN3 and *XACT*\DWG3 directories.

The TTL schematic macro library for *OrCaD* contains SP3 and PIN files which install in the ORC3 directory.

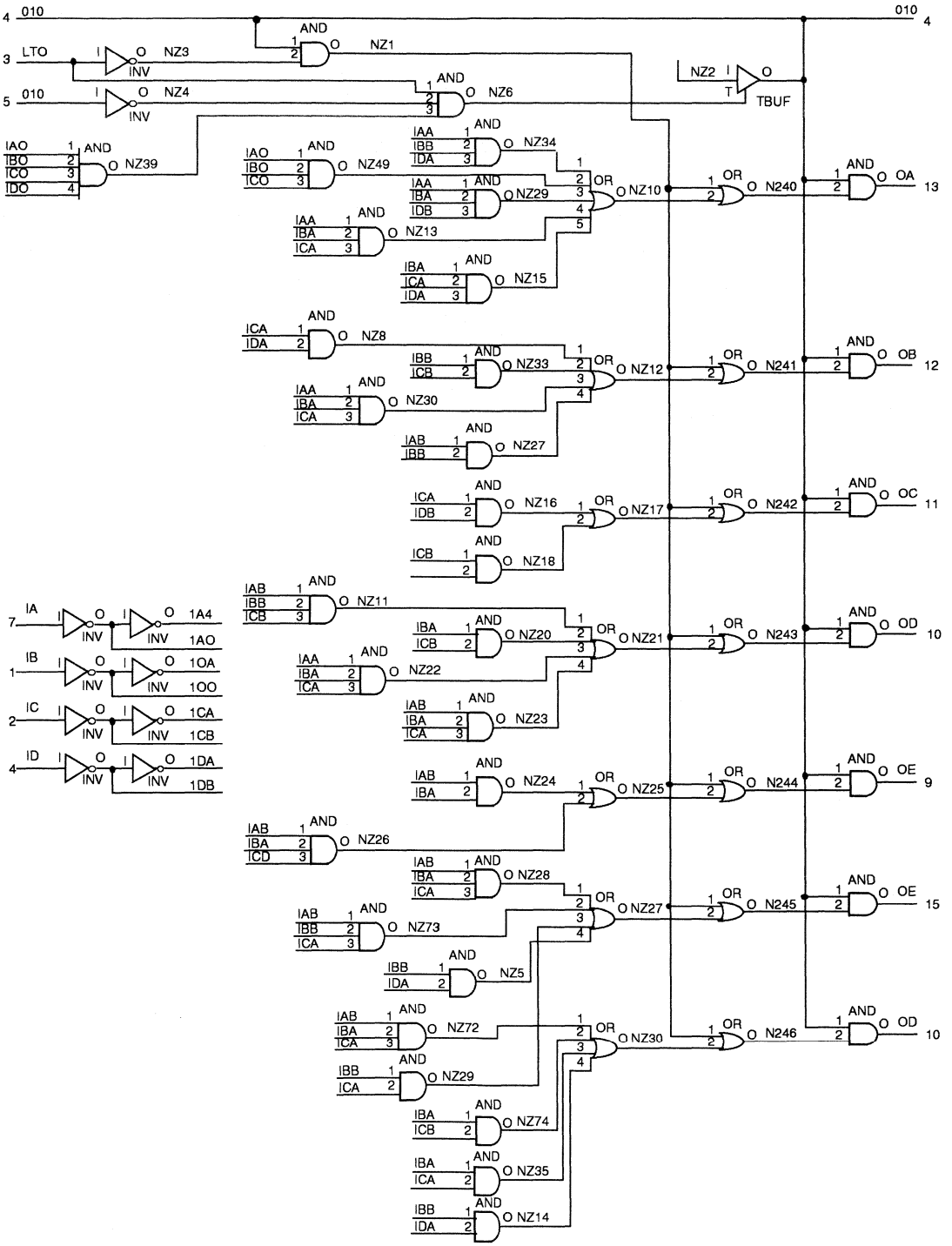
Table Of Contents

Contents **Page**

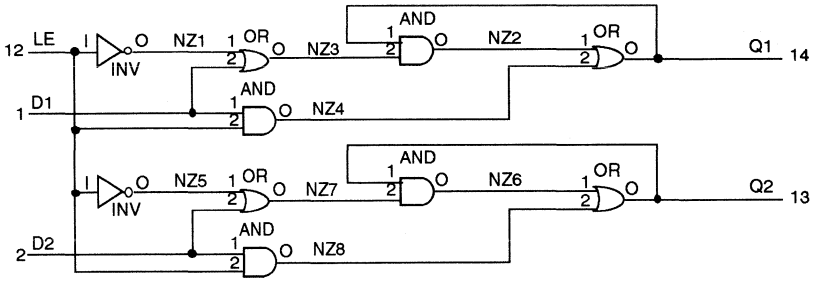
HX42 7442 4-to-10 Line Decoder



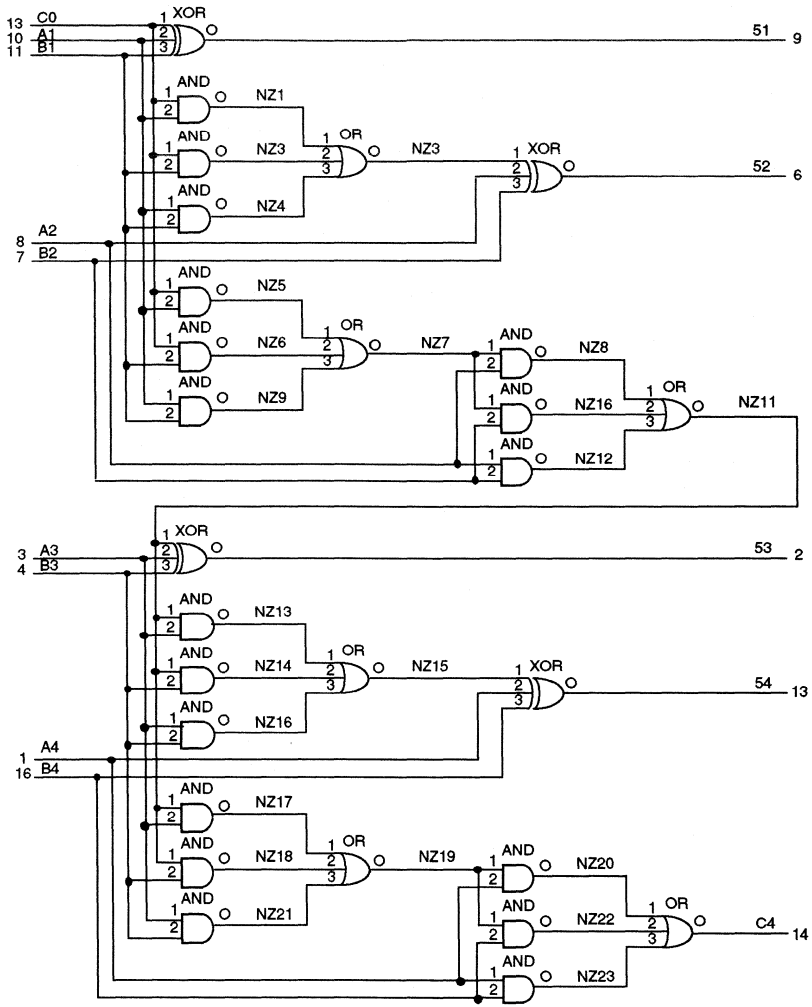
HX48 7448 BCD-to-Seven-Segment Decoder



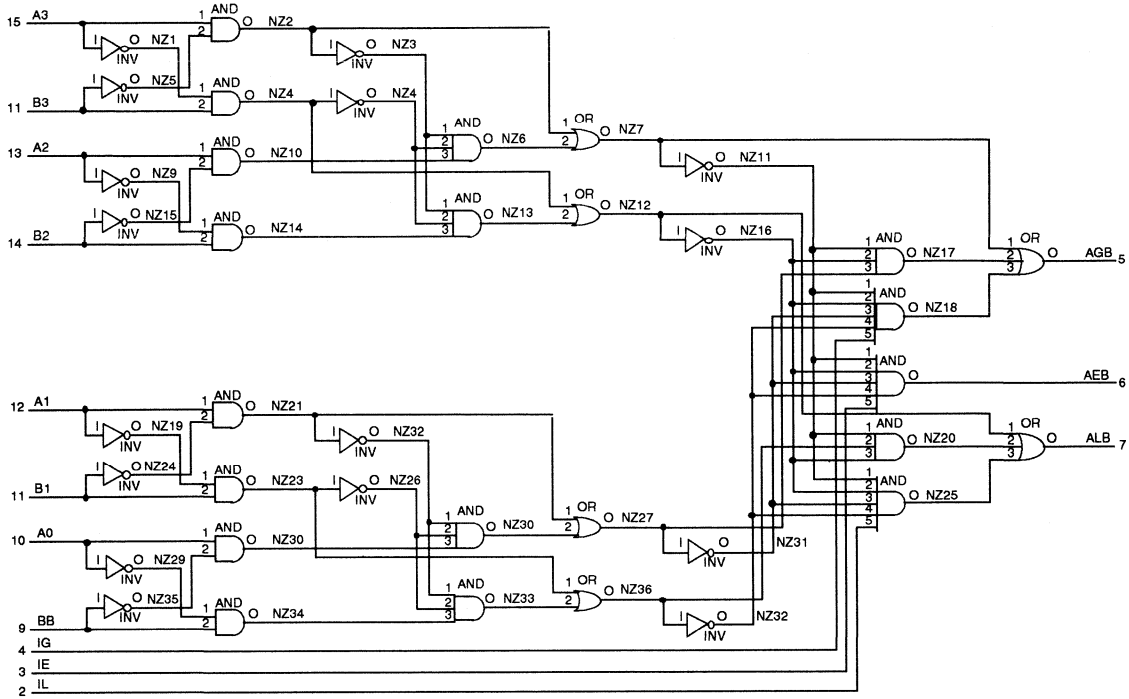
HX77 7477 2-bit Latch



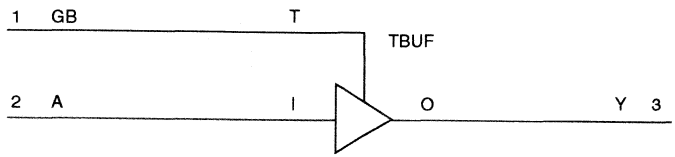
HX83 7483 4-bit Binary Adder with Fast Carry



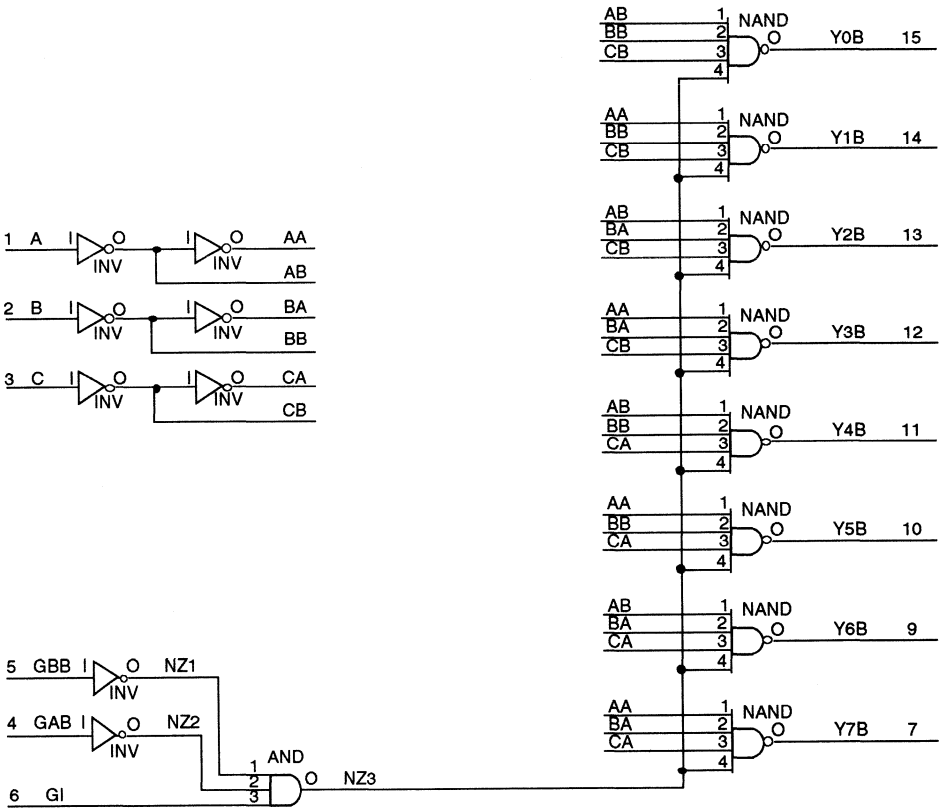
HX85 7485 4-bit Magnitude Comparator



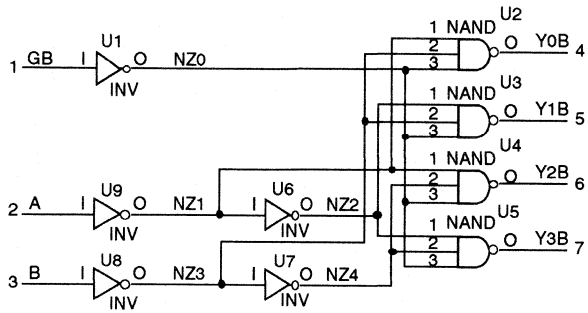
HX125 74125 Quad 3-State Bus Buffer



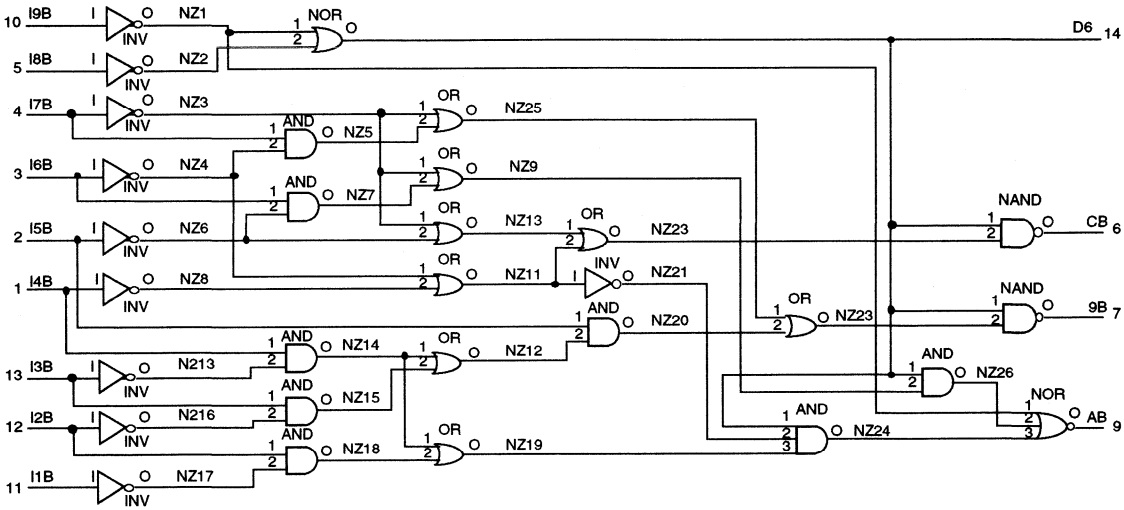
HX138 74138 1-of-8 Decoder/Multiplexer



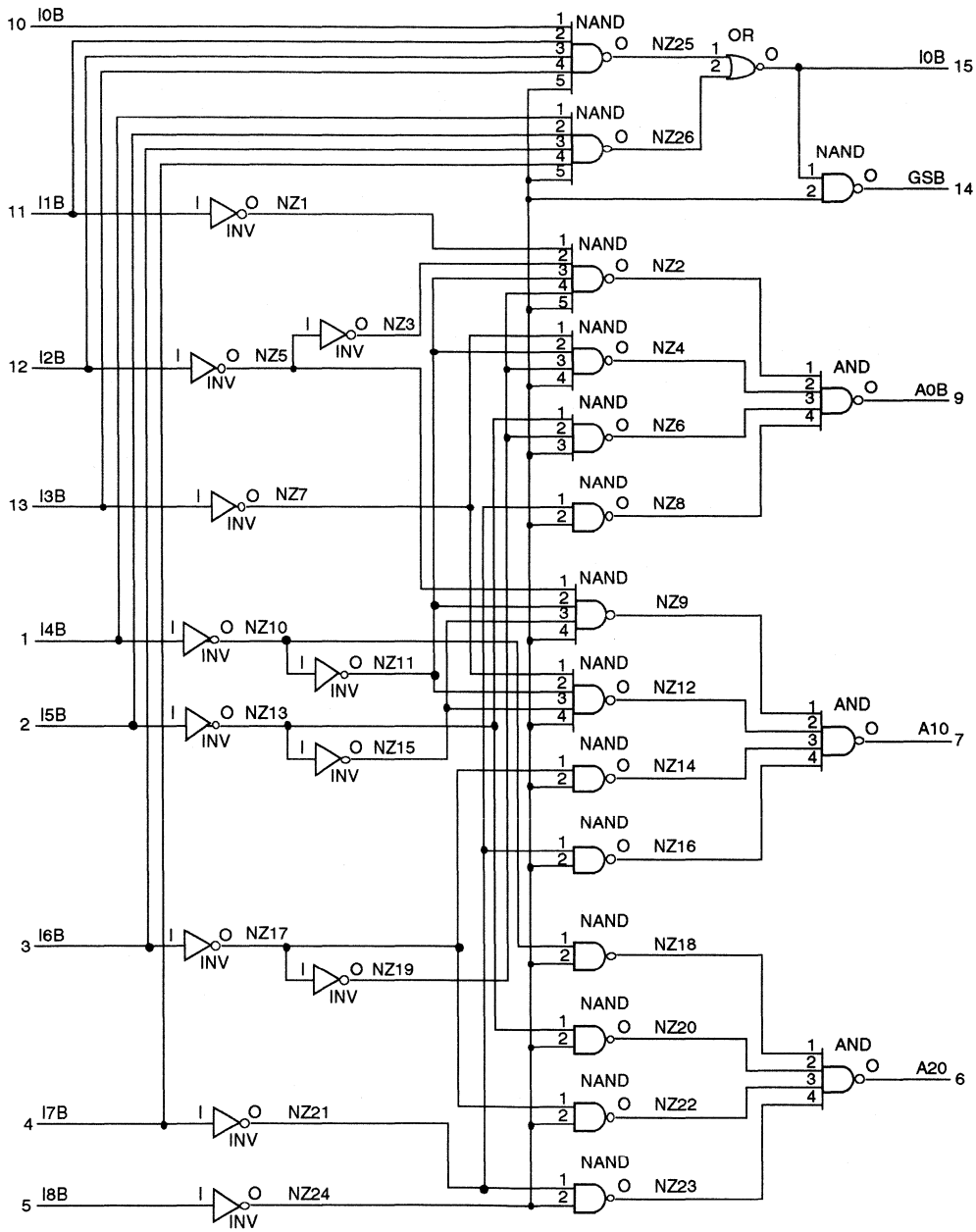
HX139 74139 Dual 1-of-4 Decoder



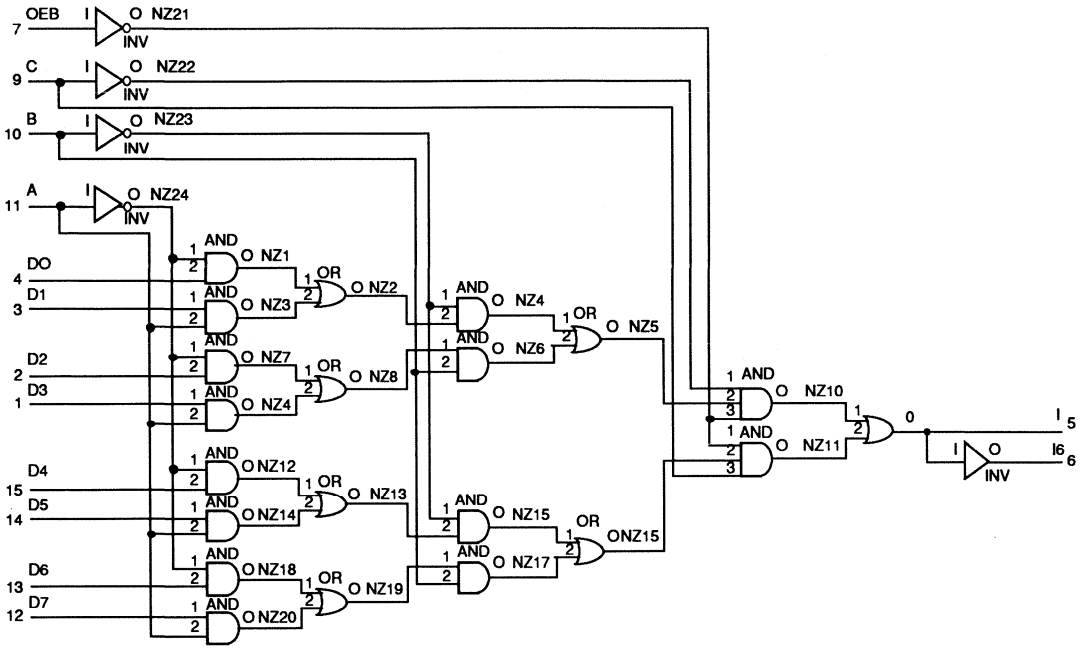
HX147 74147 10-to-4 Line Priority Decoder



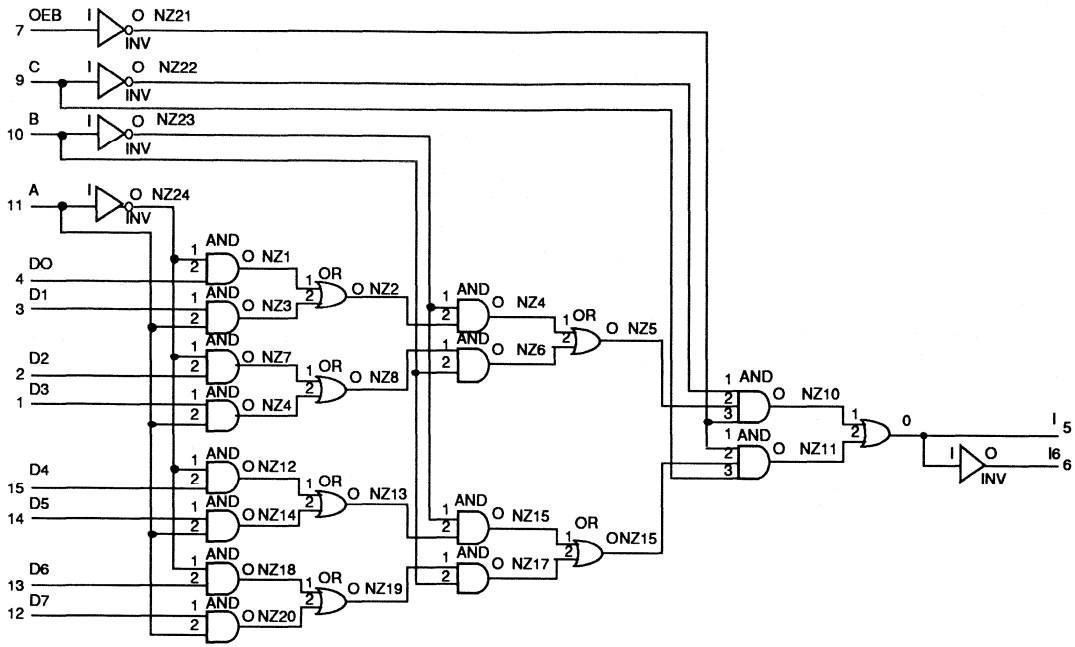
HX148 74148 8-to-3 Line Priority Encoder



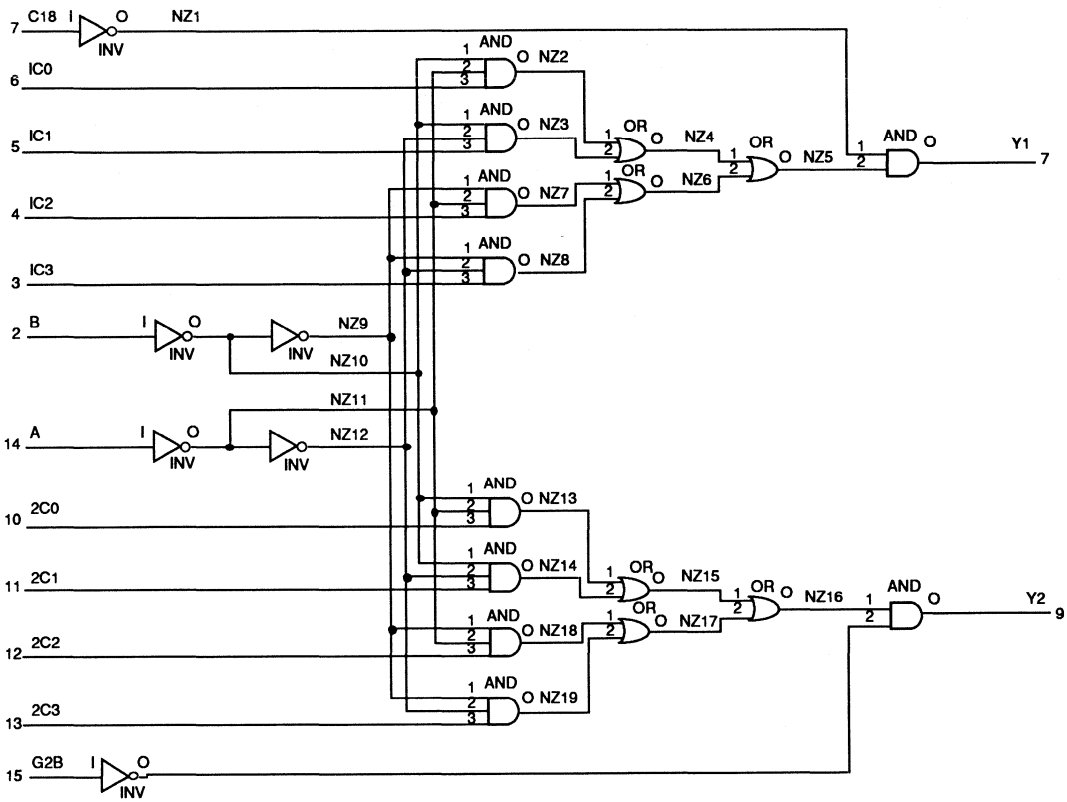
HX151 74151 8-Input Multiplexer



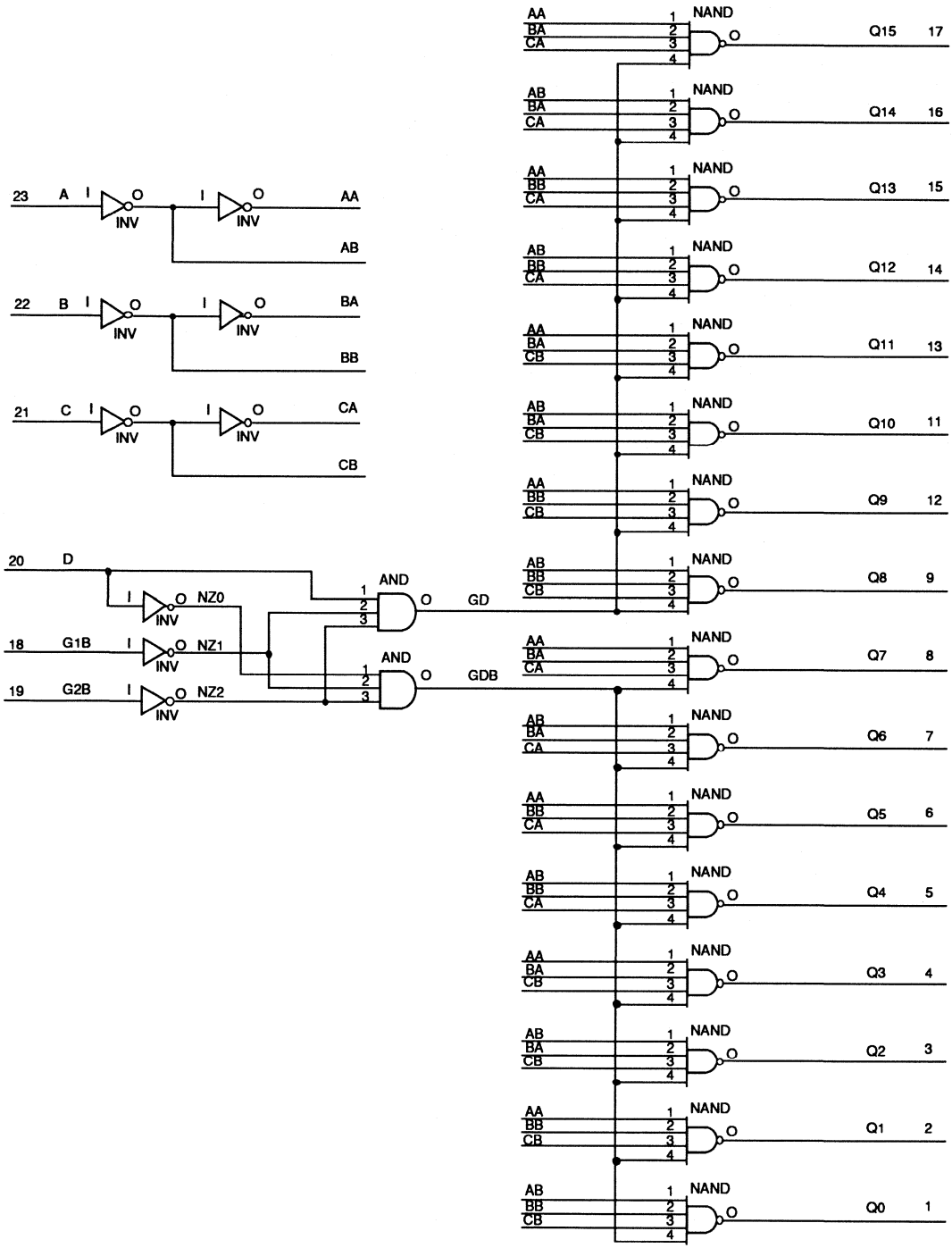
HX152 74152 8-Input Multiplexer



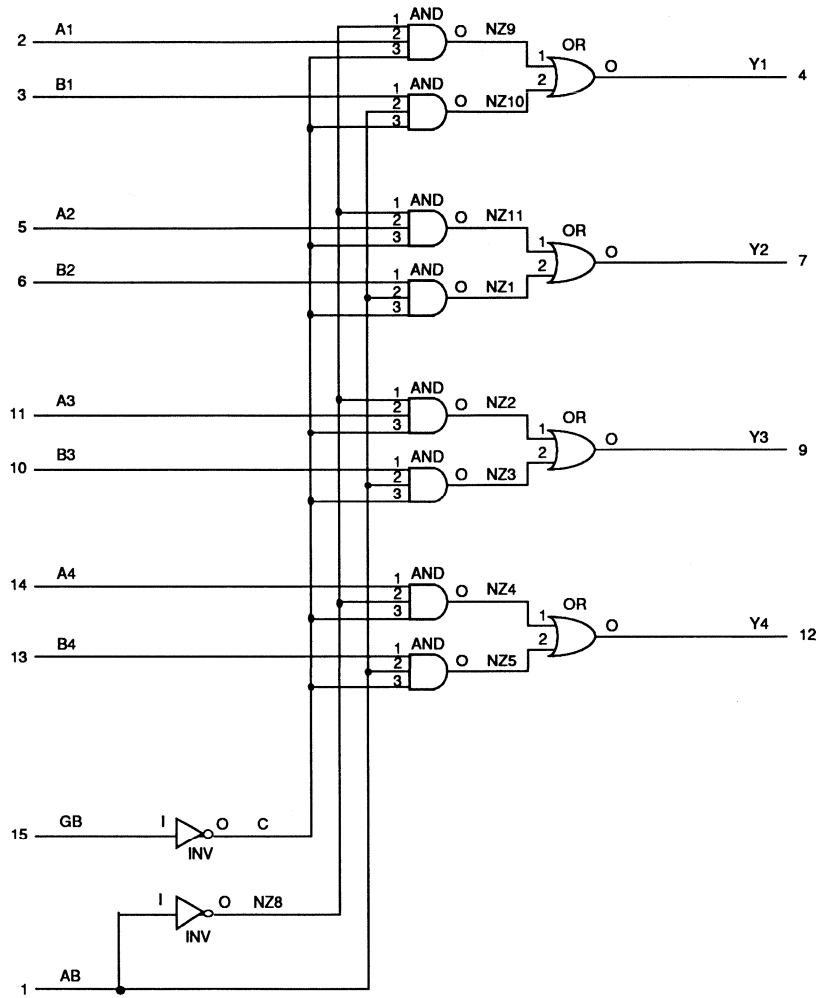
HX153 74153 Dual 4-Input Multiplexer



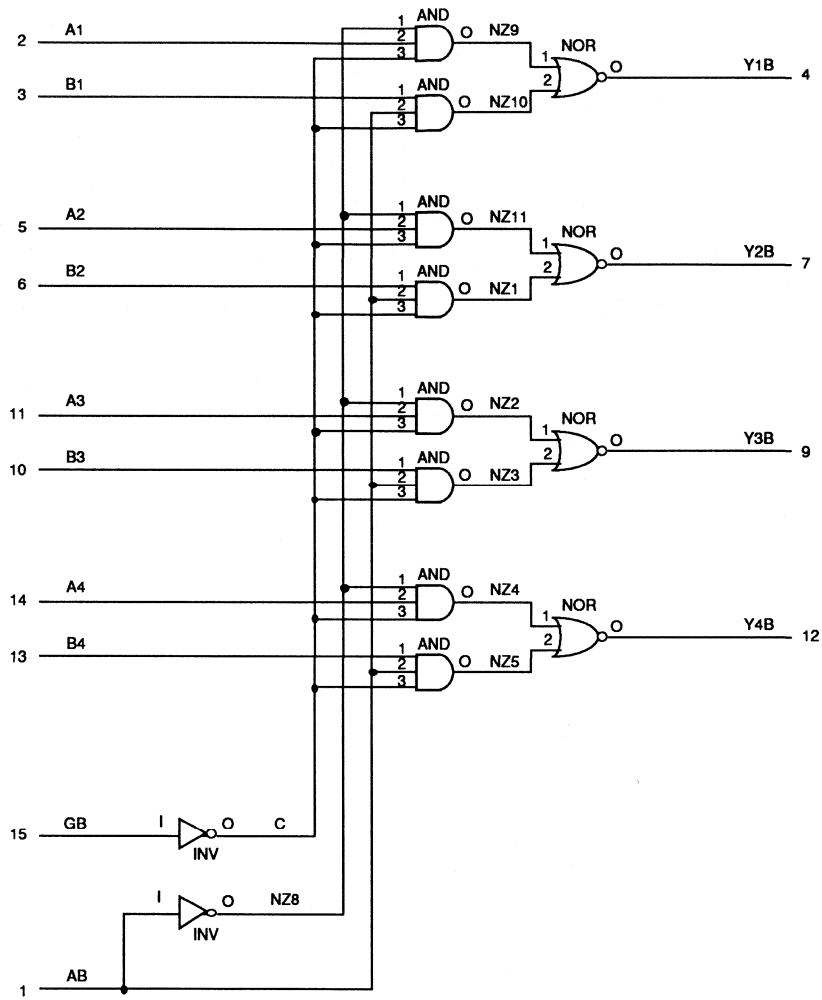
HX154 74154 1-of-16 Decoder/Multiplexer



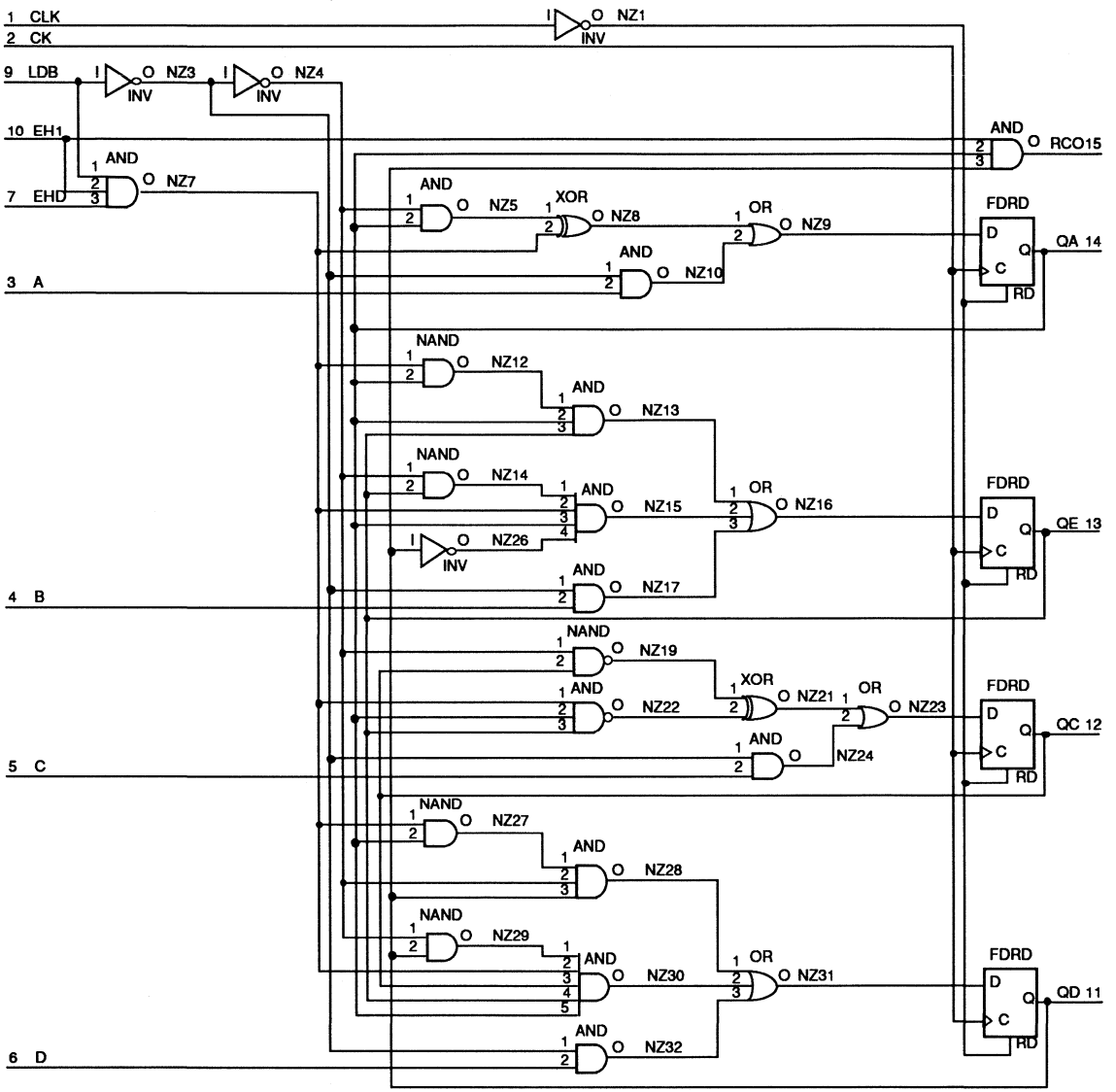
HX157 74157 Quad 2-Input Multiplexer



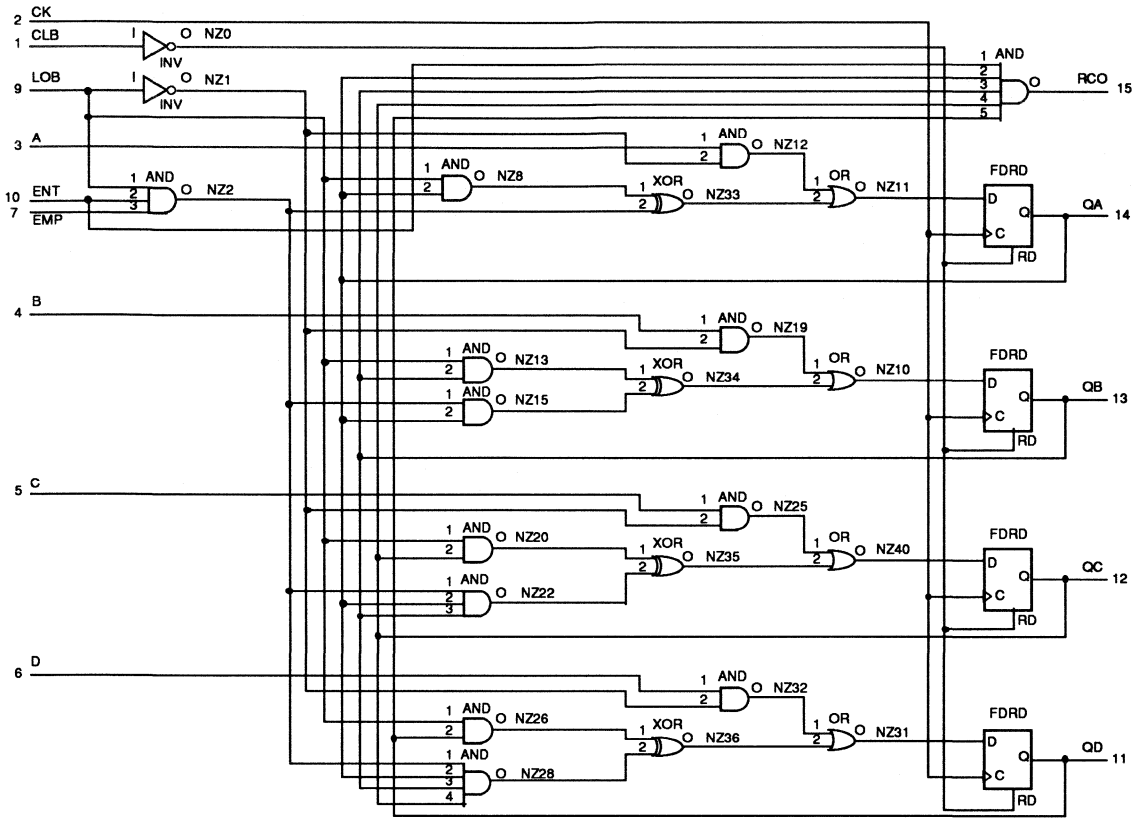
HX158 74158 Quad 2-Input Multiplexer



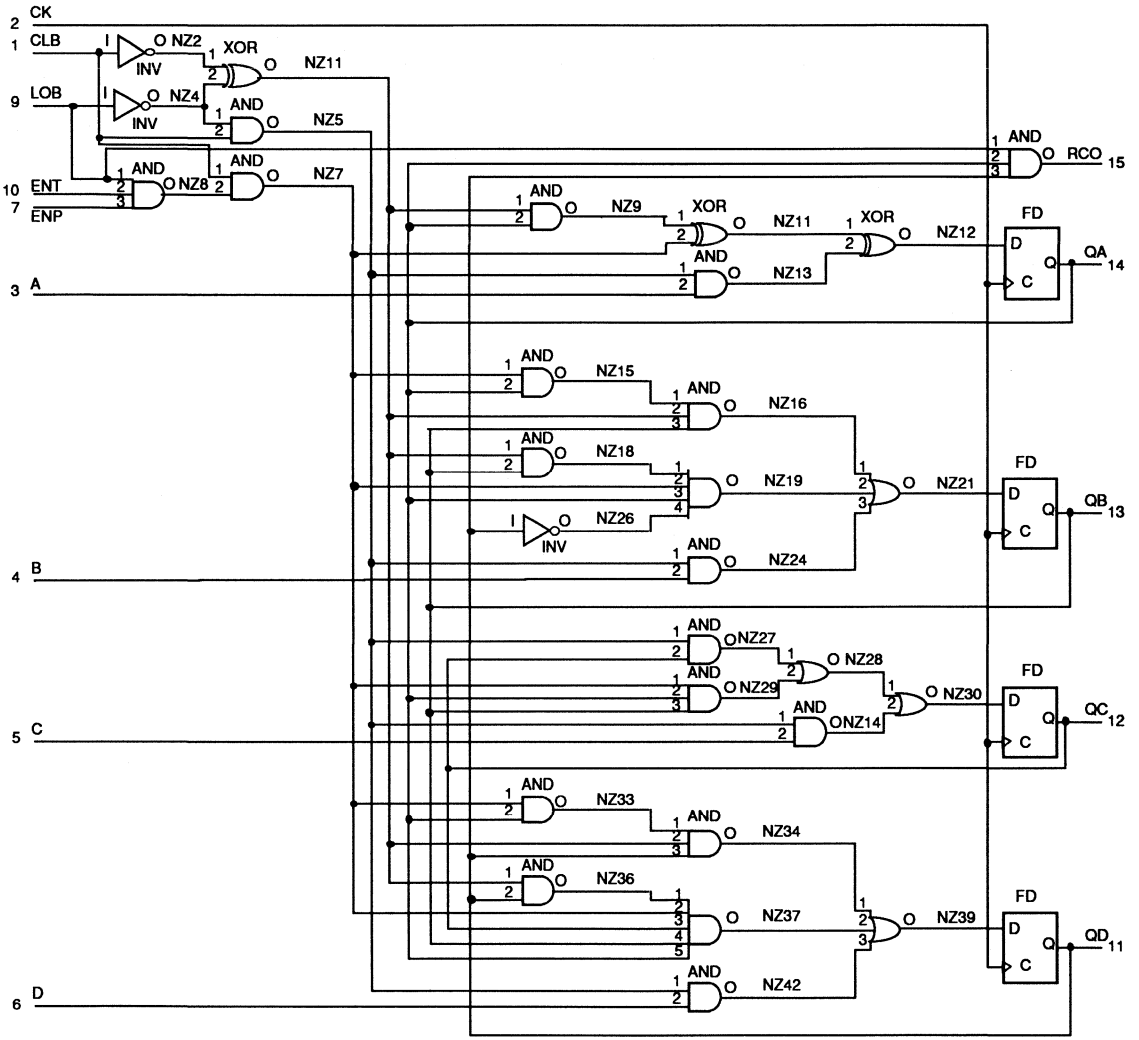
HX160 74160 Presettable Decade Counter



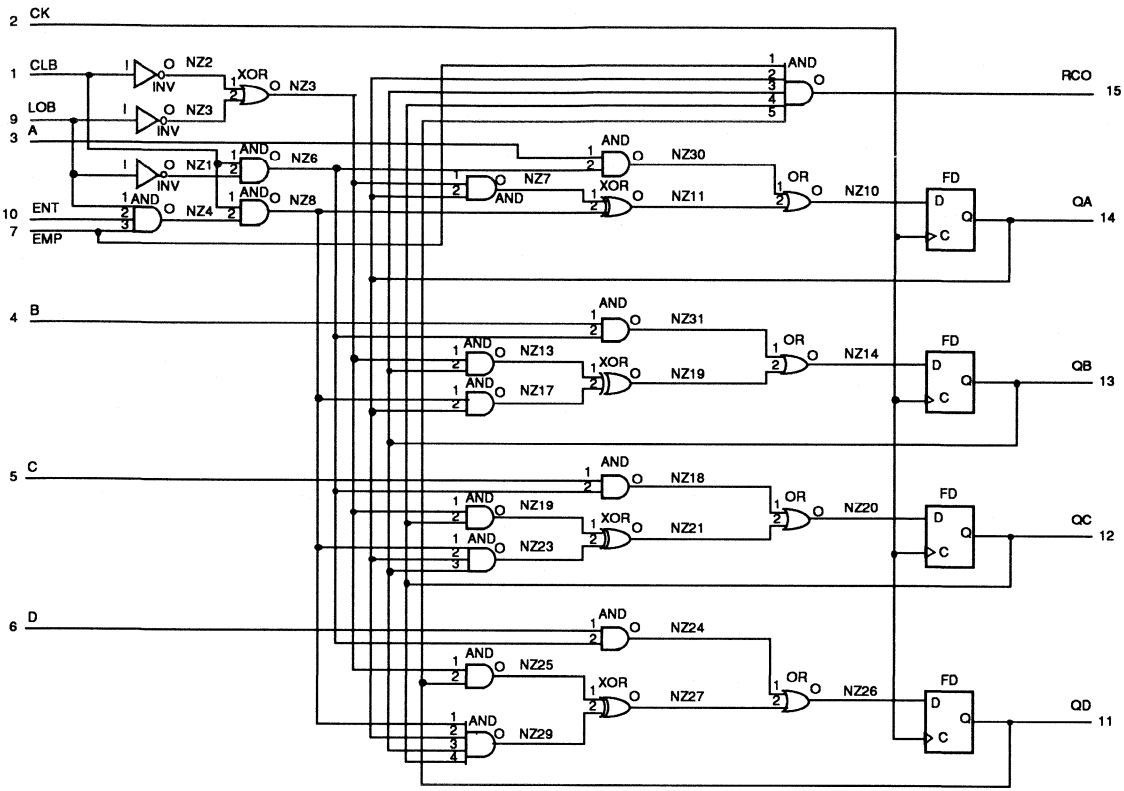
HX161 74161 Presettable Binary Counter



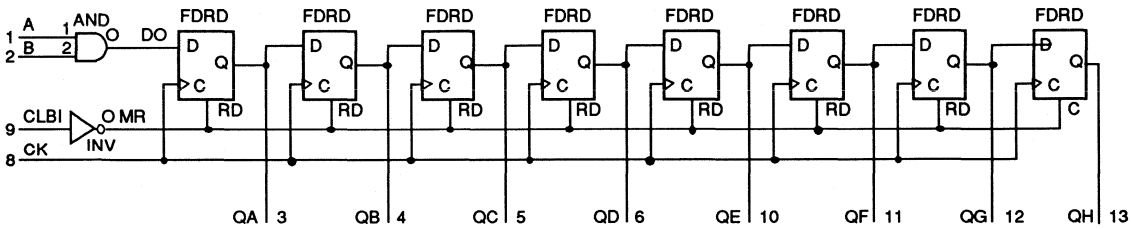
HX162 74162 Presettable Decade Counter with Synchronous Clear



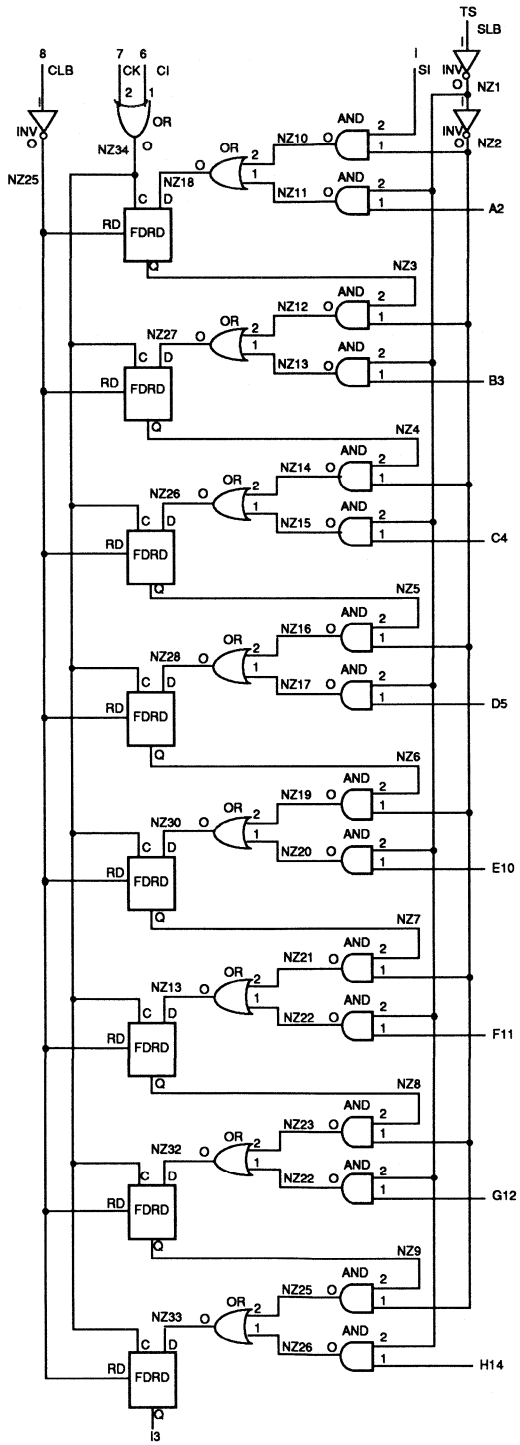
HX163 74163 Synchronous Binary Counter with Synchronous Clear



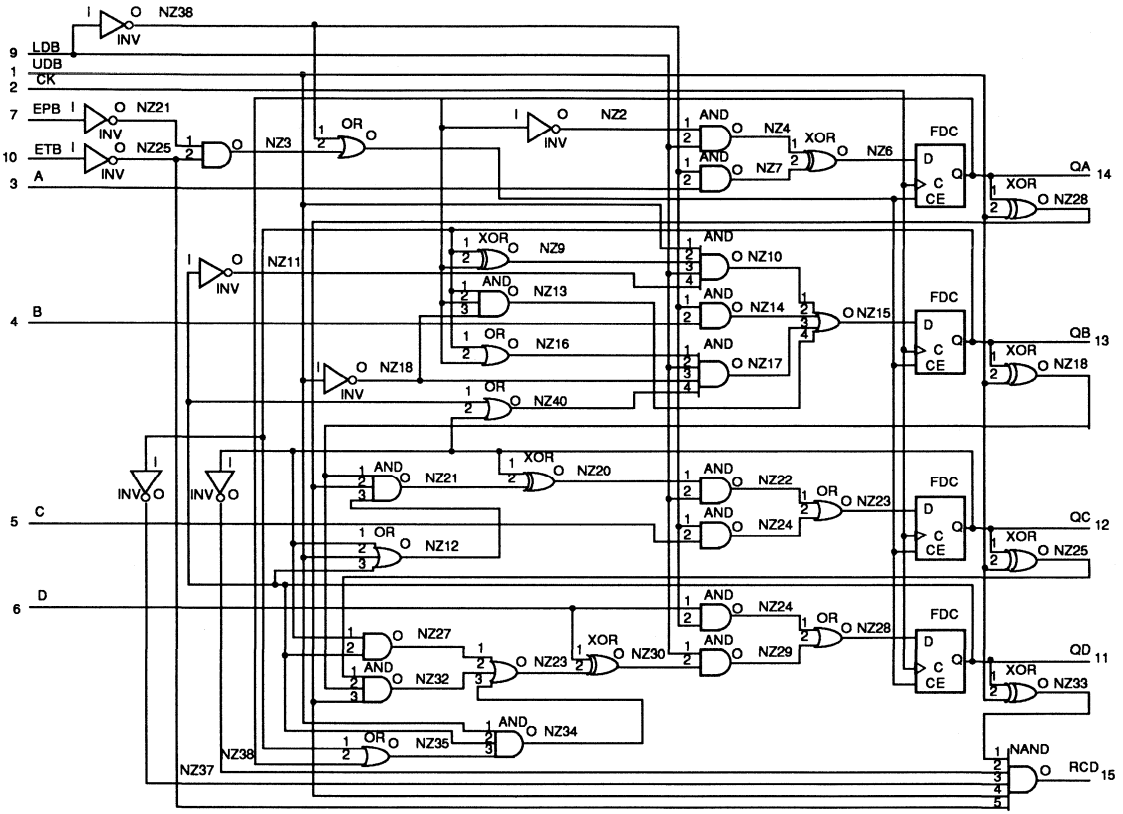
HX164 74164 8-bit Serial-In Parallel-Out Shift Register



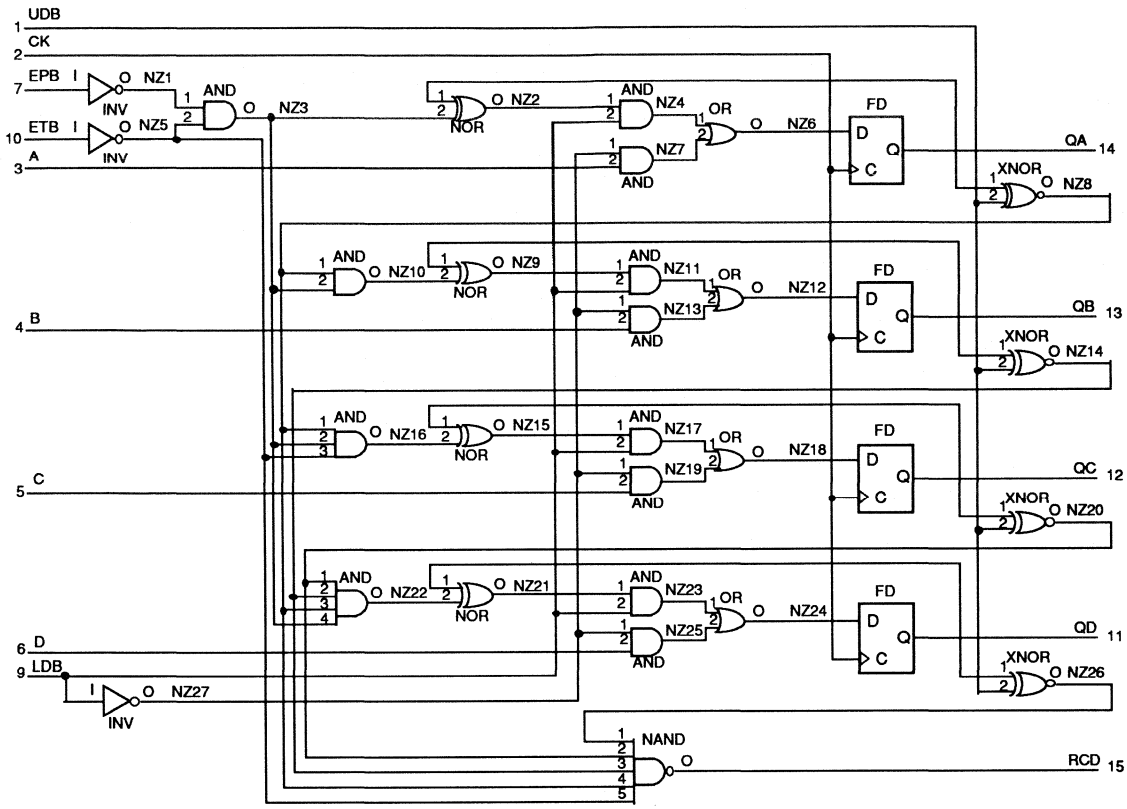
HX166 74166 8-bit Parallel Load Shift Register



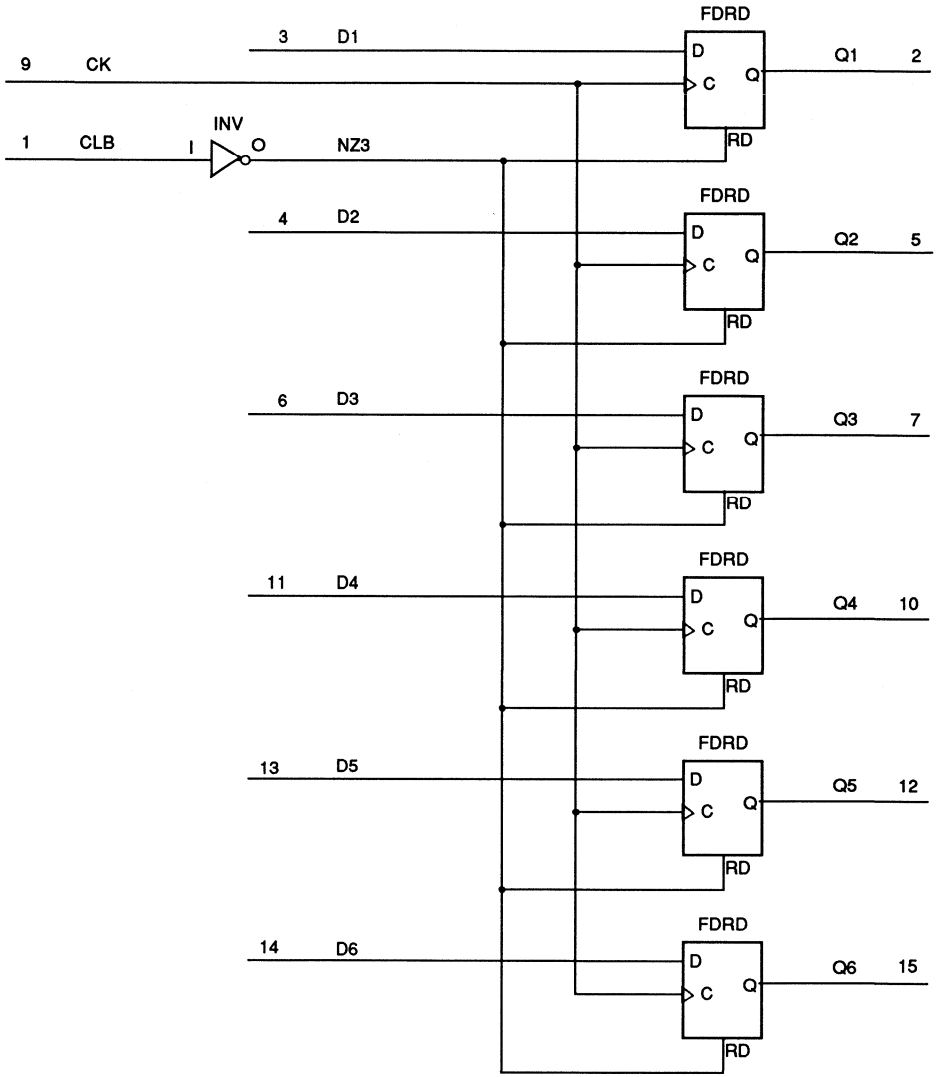
HX168 74168 4-bit BCD Synchronous Up/Down Counter



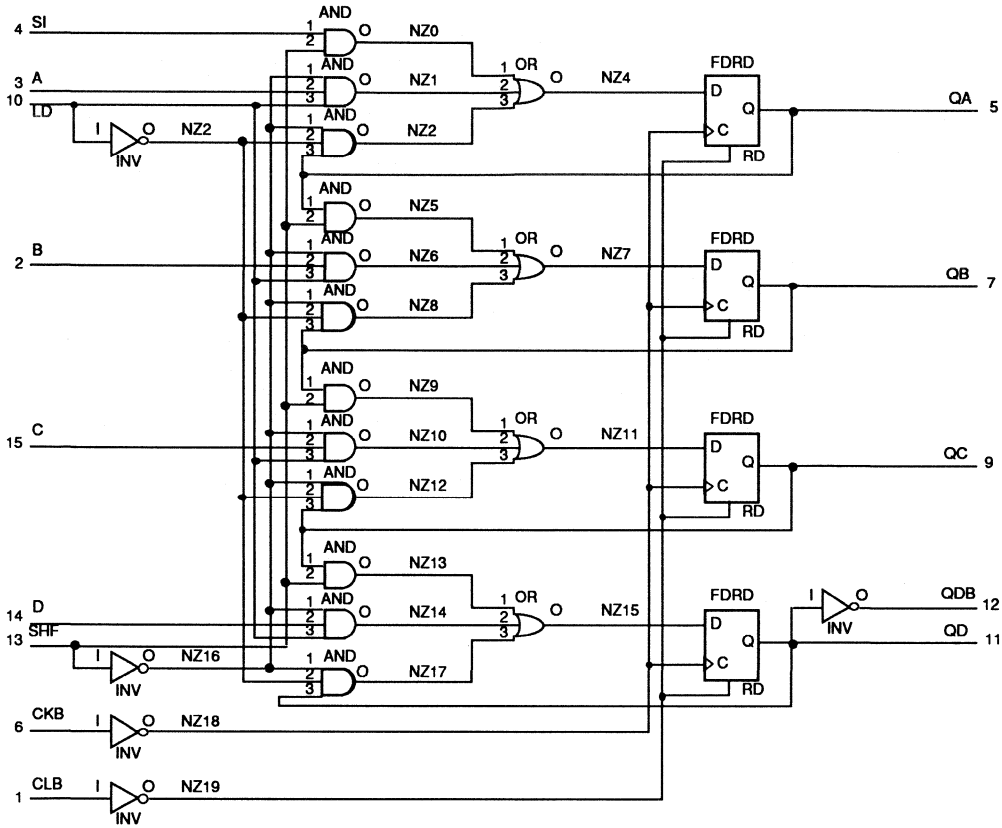
HX169 74169 4-bit Binary Synchronous Up/Down Counter



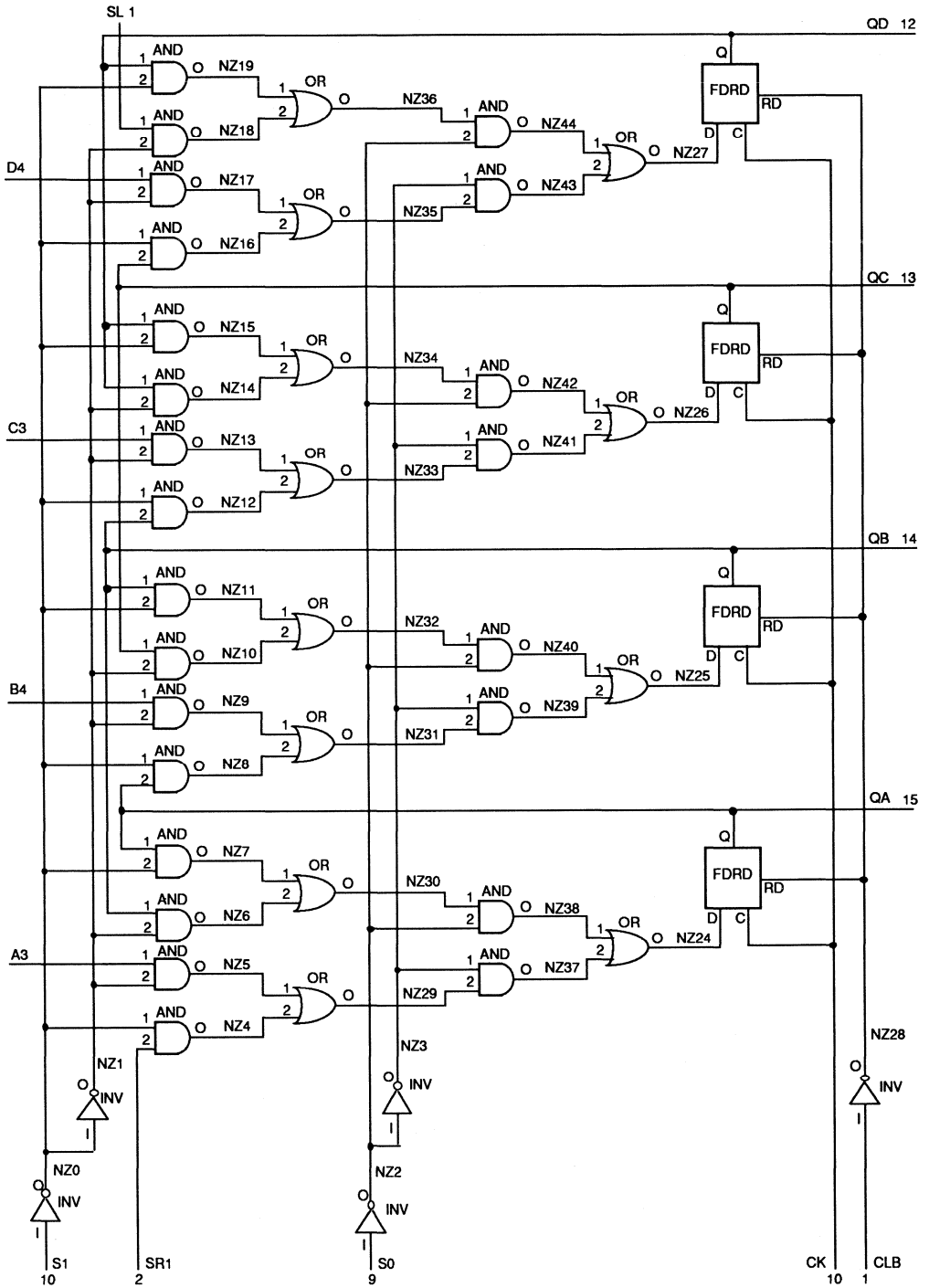
HX174 74174 Hex D Flip-Flop with Master Reset



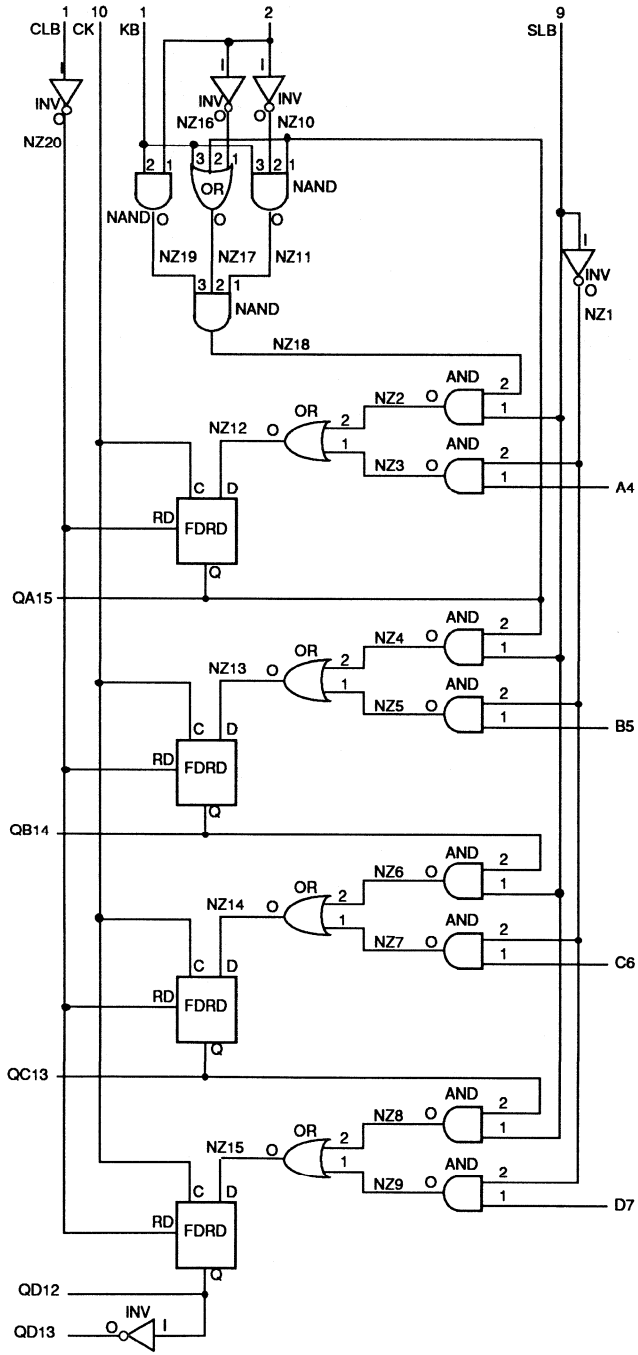
HX179 74179 4-bit Parallel-Access Shift Register



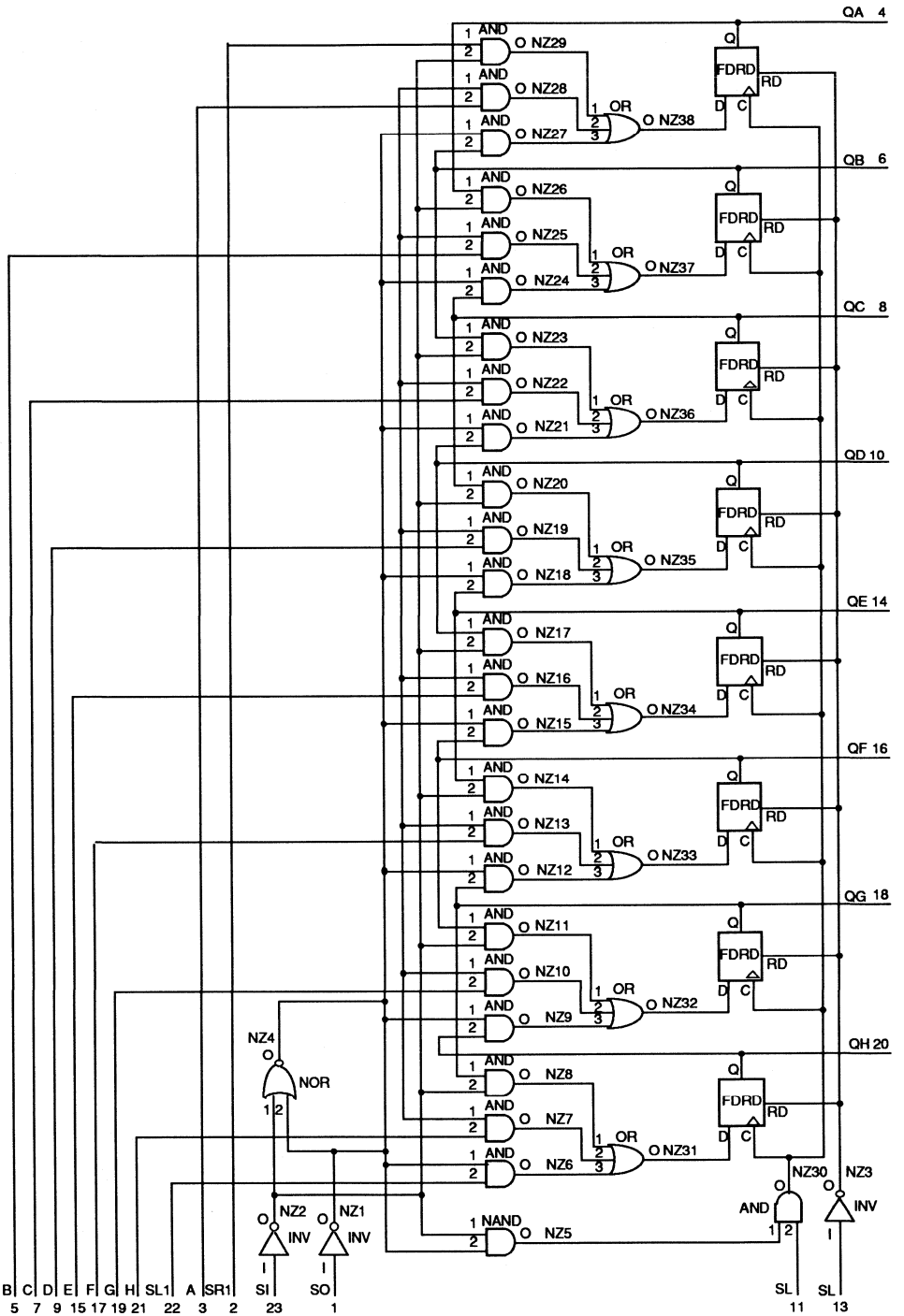
HX194 74194 4-bit Bidirectional Universal Shift Register



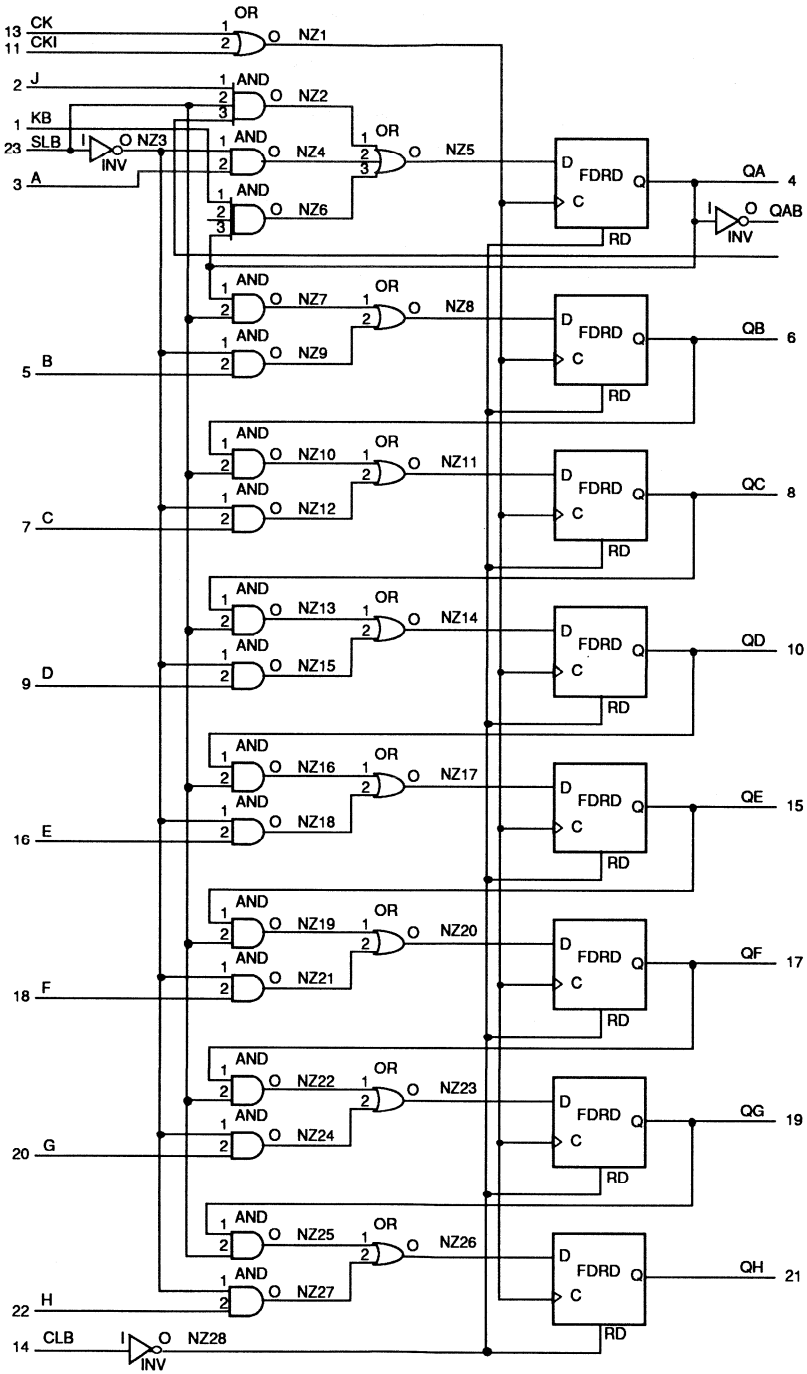
HX195 74195 4-bit Parallel-Access Shift Register



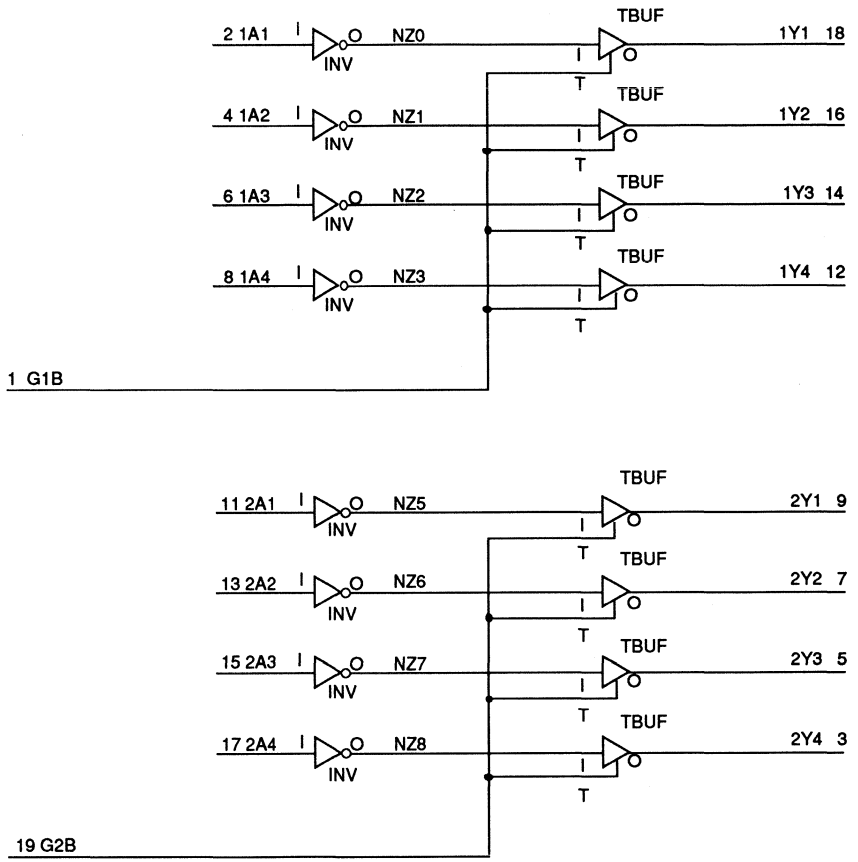
HX198 74198 8-bit Bidirectional Shift Register



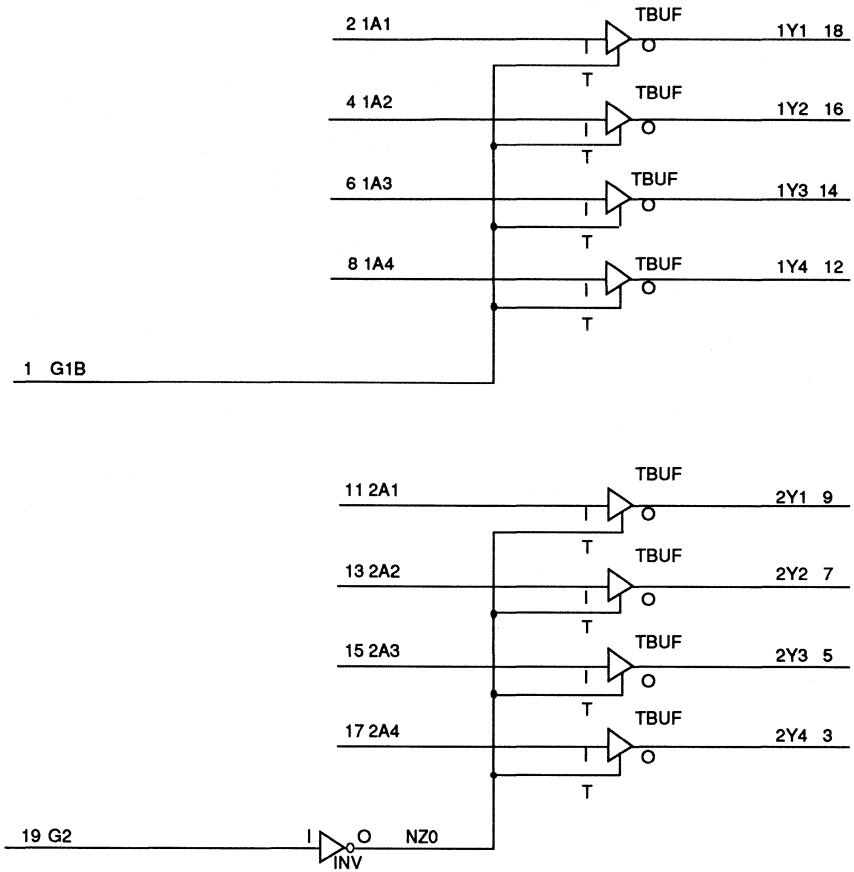
HX199 74199 8-bit Register with Clock Inhibit



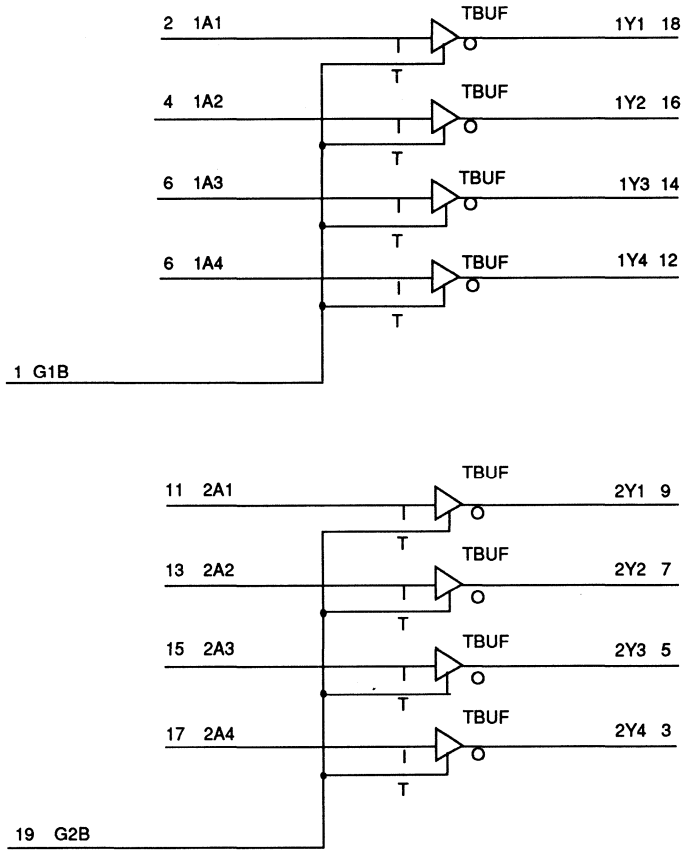
HX240 74240 Octal Inverting Buffer 3-State Output



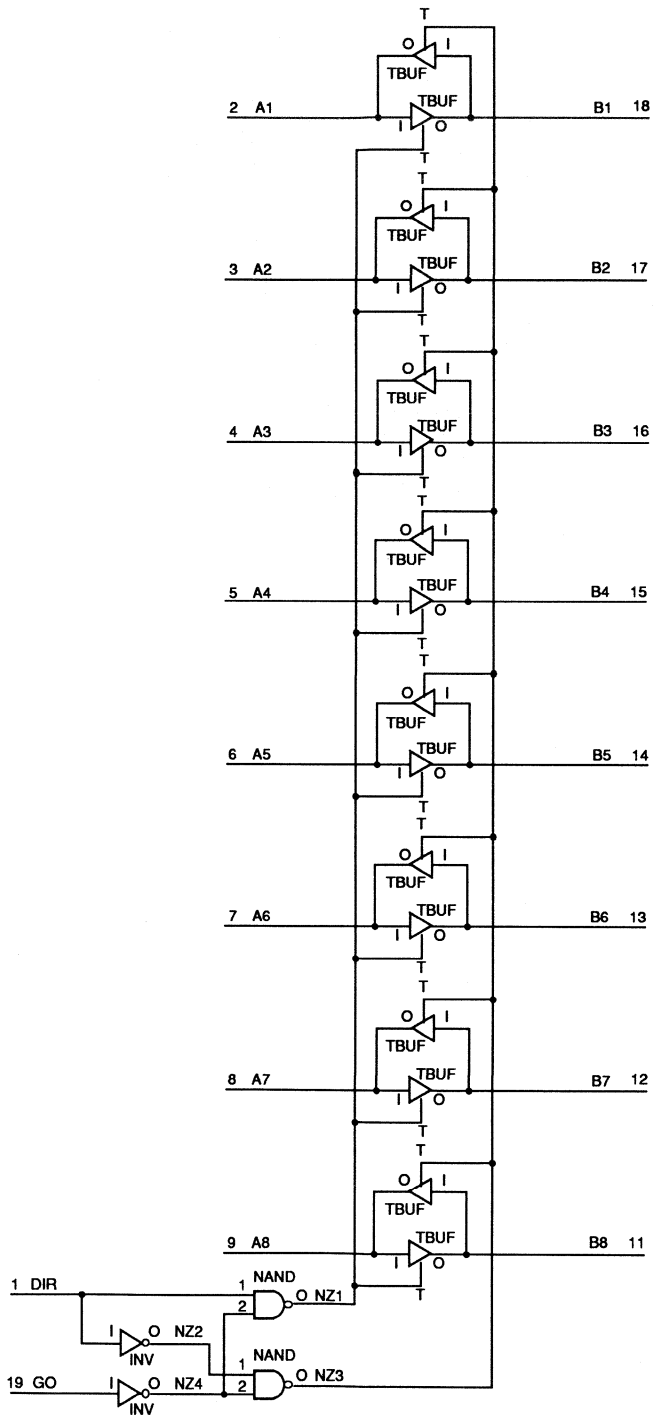
HX241 74241 Octal Noninverting 3-State Buffer



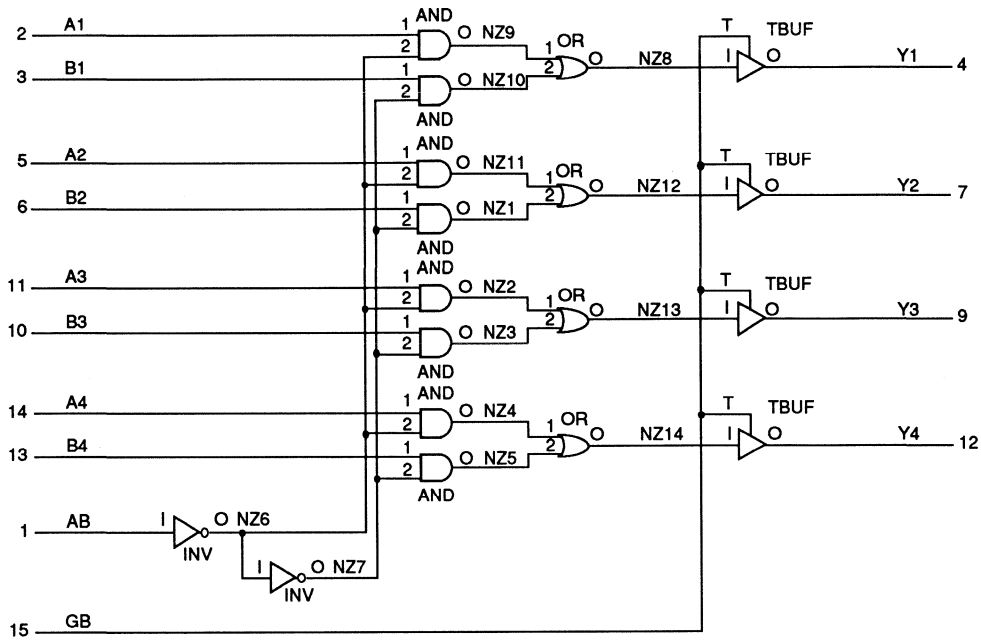
HX244 74244 Octal Noninverting Buffer with 3-State Outputs



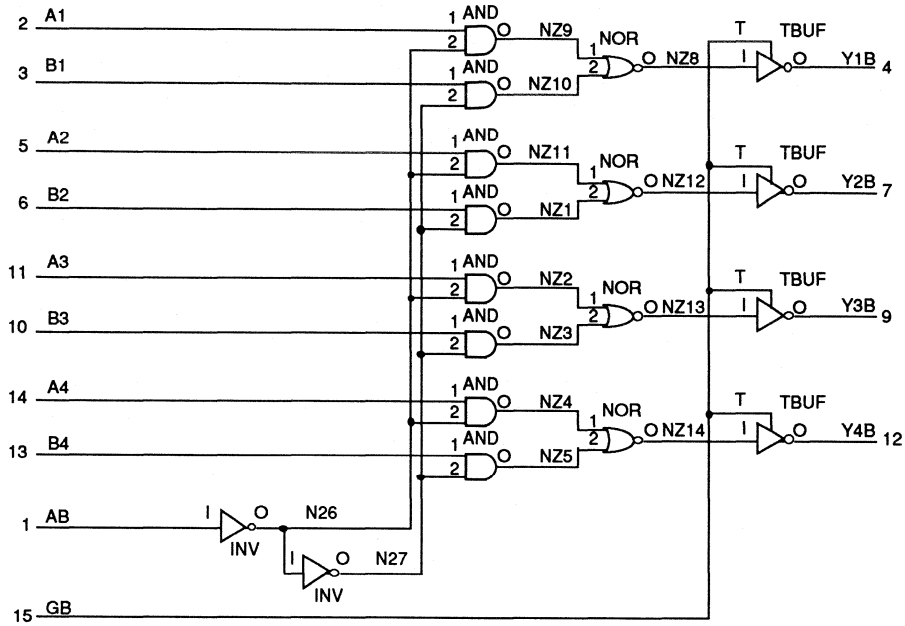
HX245 74245 Octal Bidirectional Transceiver



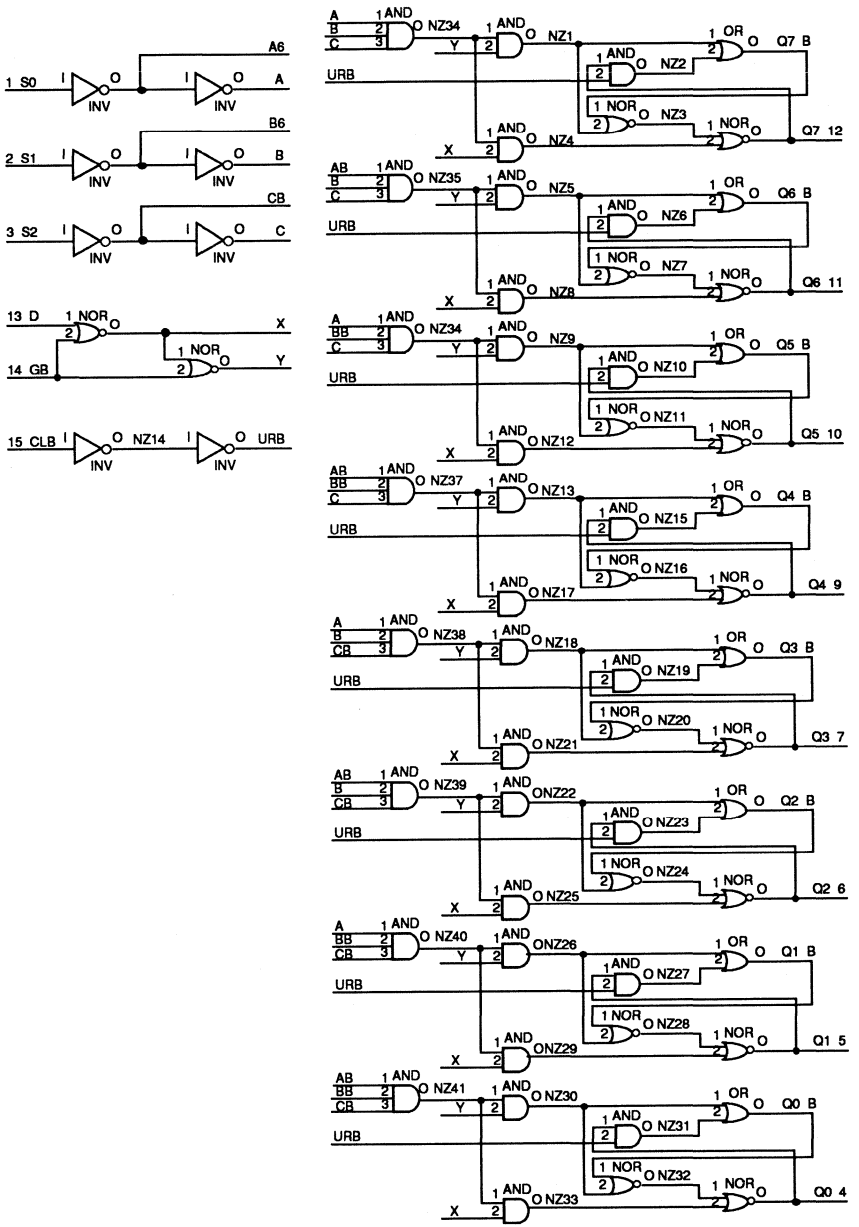
HX257 74257 Quad 2-Input Multiplexer



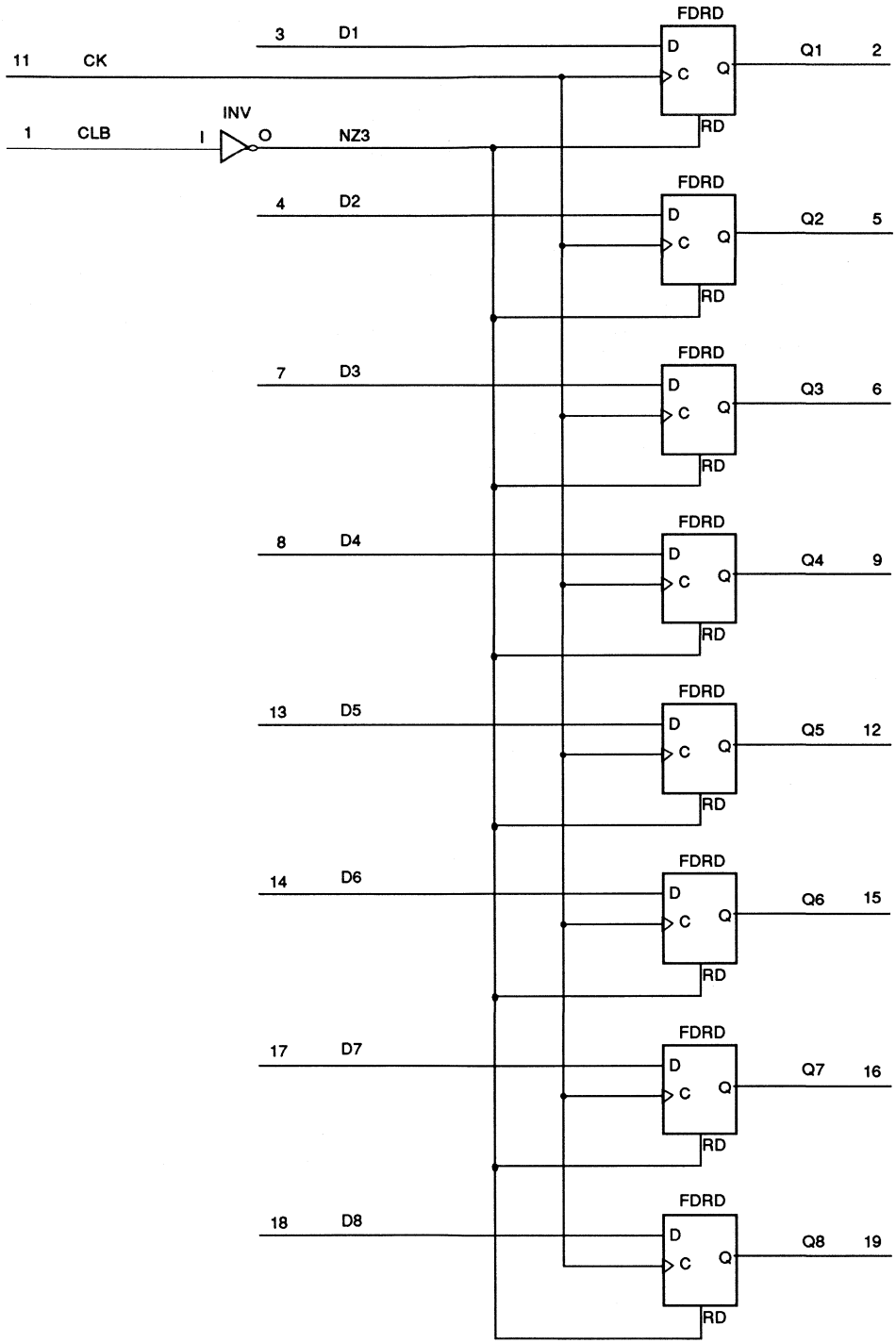
HX258 74258 Quad 2-Input Multiplexer



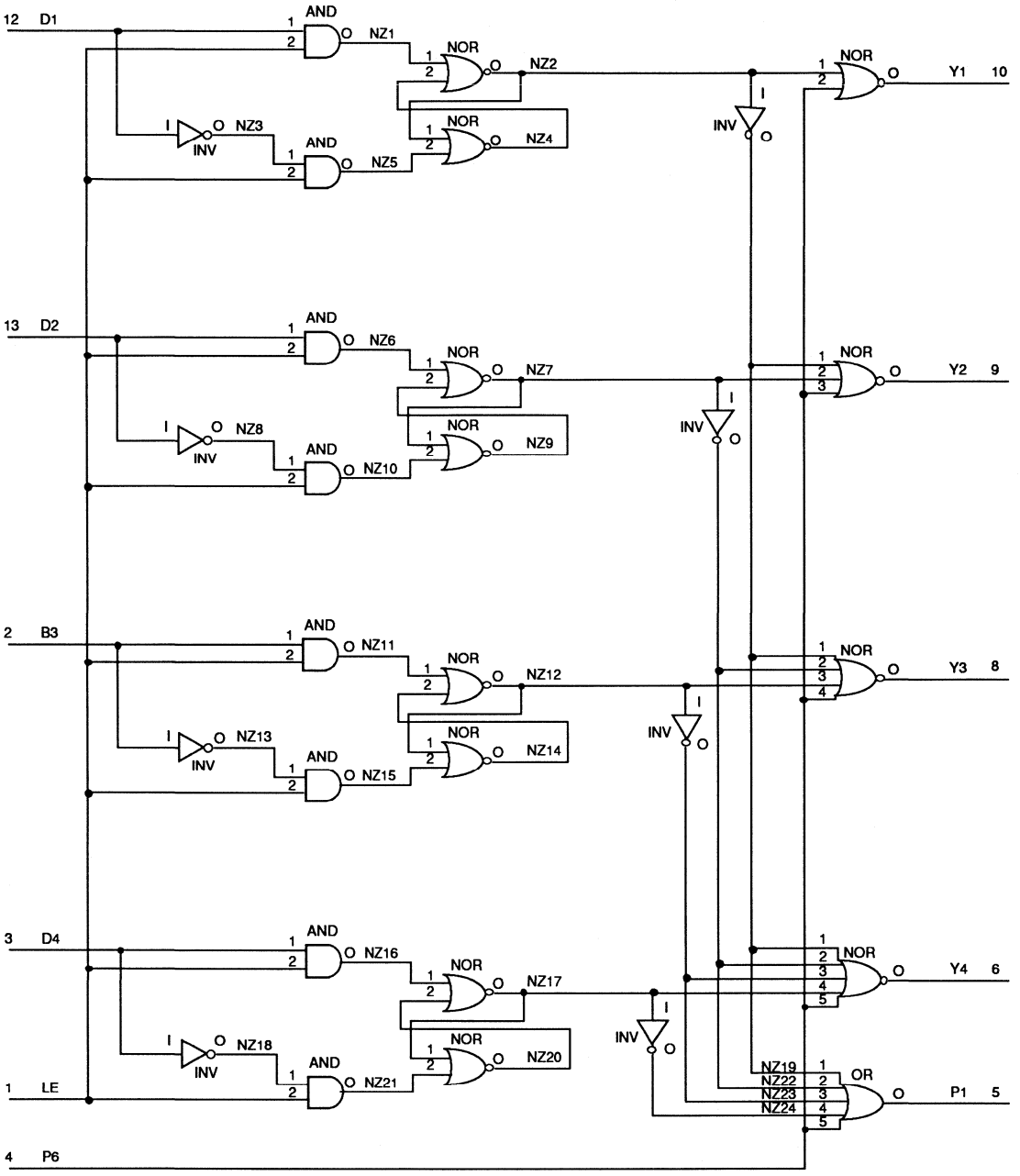
HX259 74259 8-bit Addressable Latch



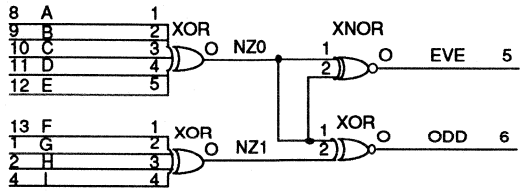
HX273 74273 Octal D Flip-Flop



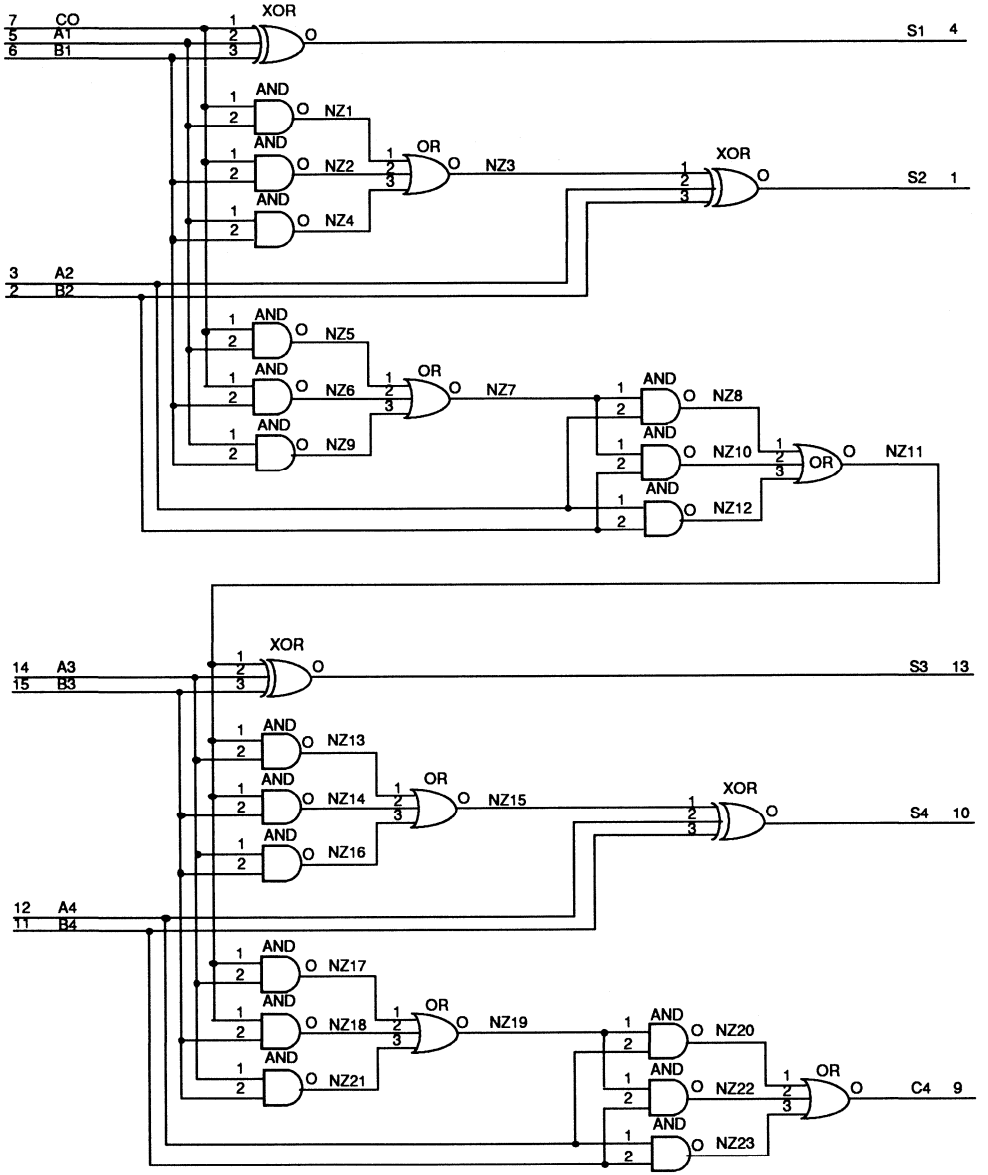
HX278 74278 4-bit Cascadable Priority Register



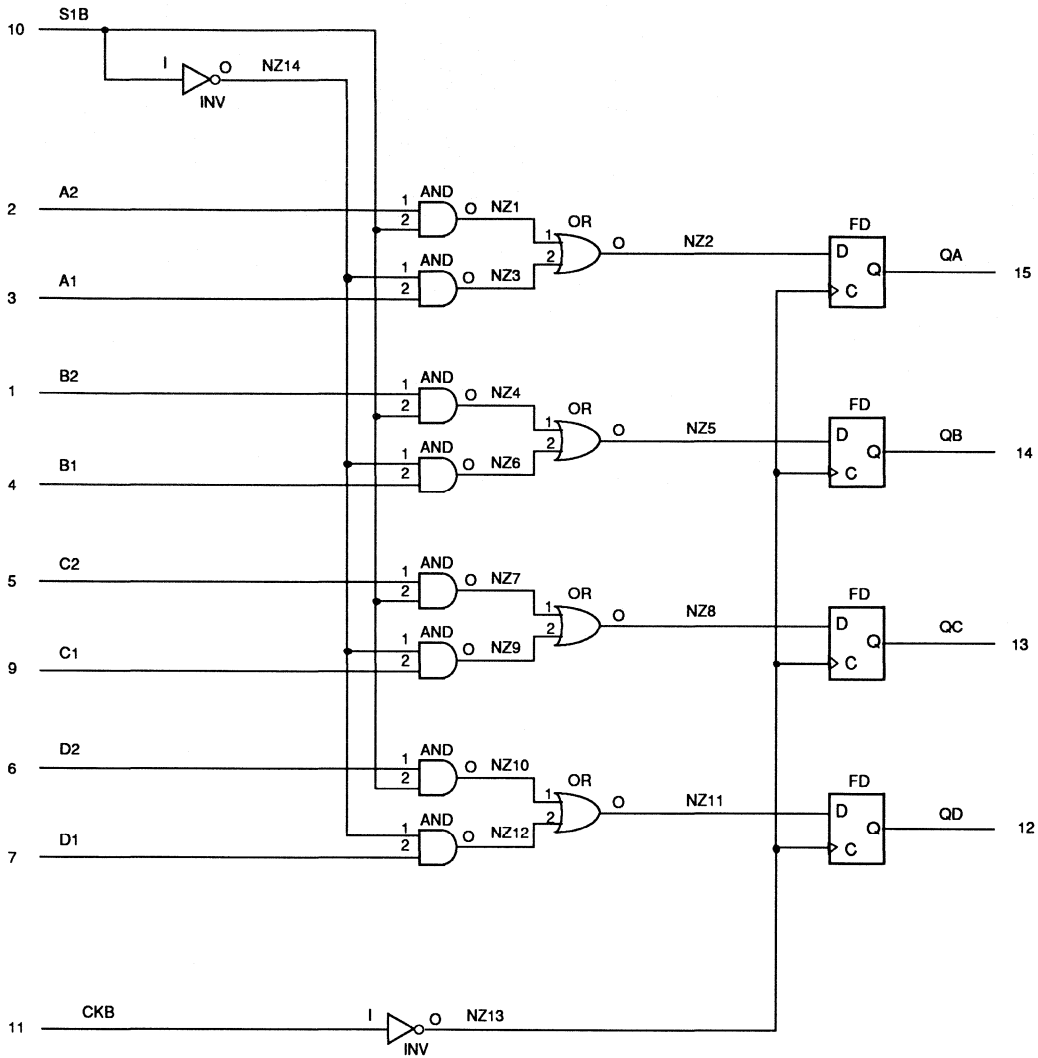
HX280 74280 9-bit Parity Checker/Generator



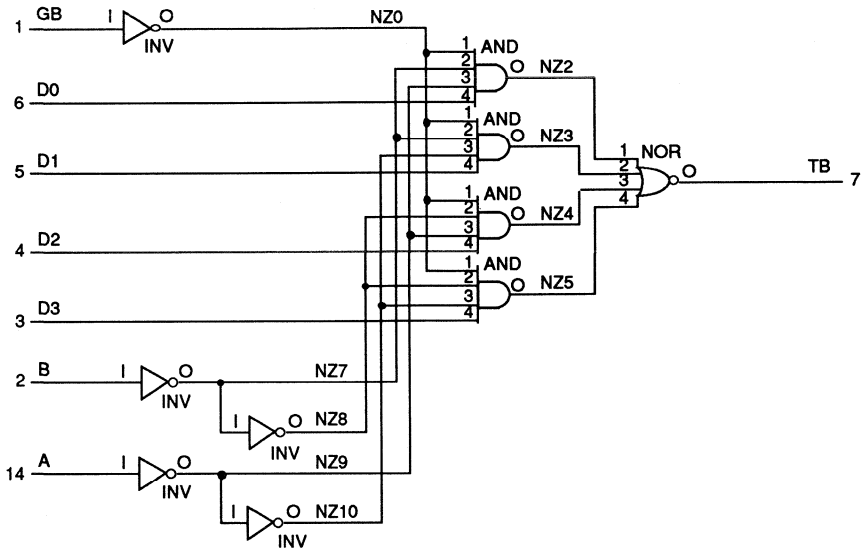
HX283 74283 4-bit Binary Full Adder



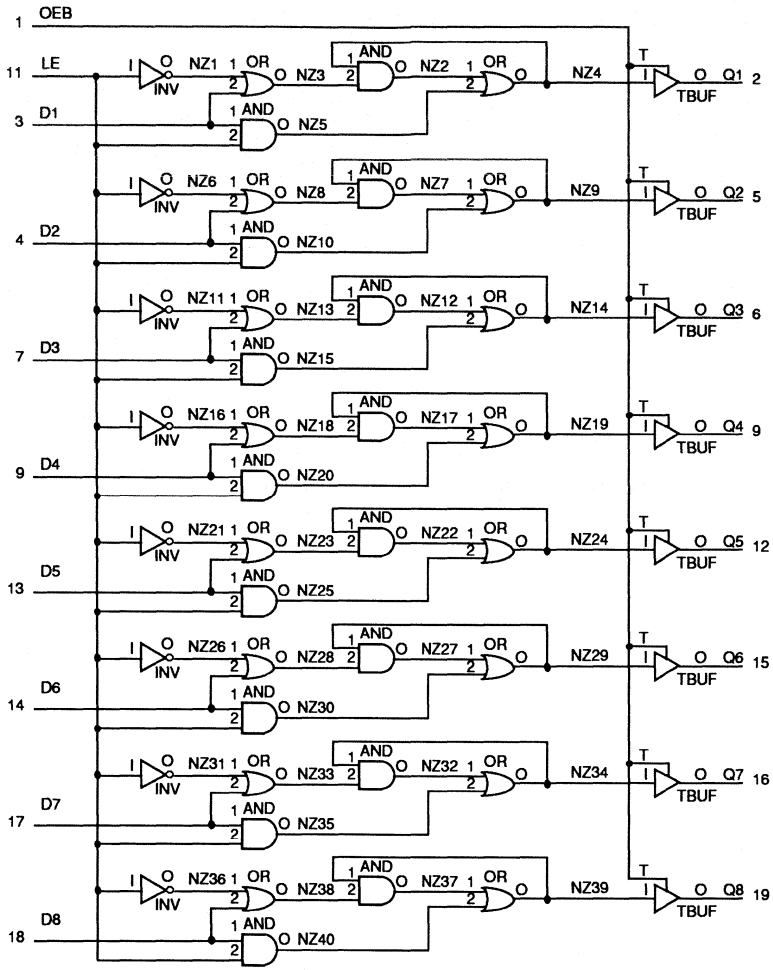
HX298 74298 Quad 2-Input Flip-Flop, Multiplexer with Storage



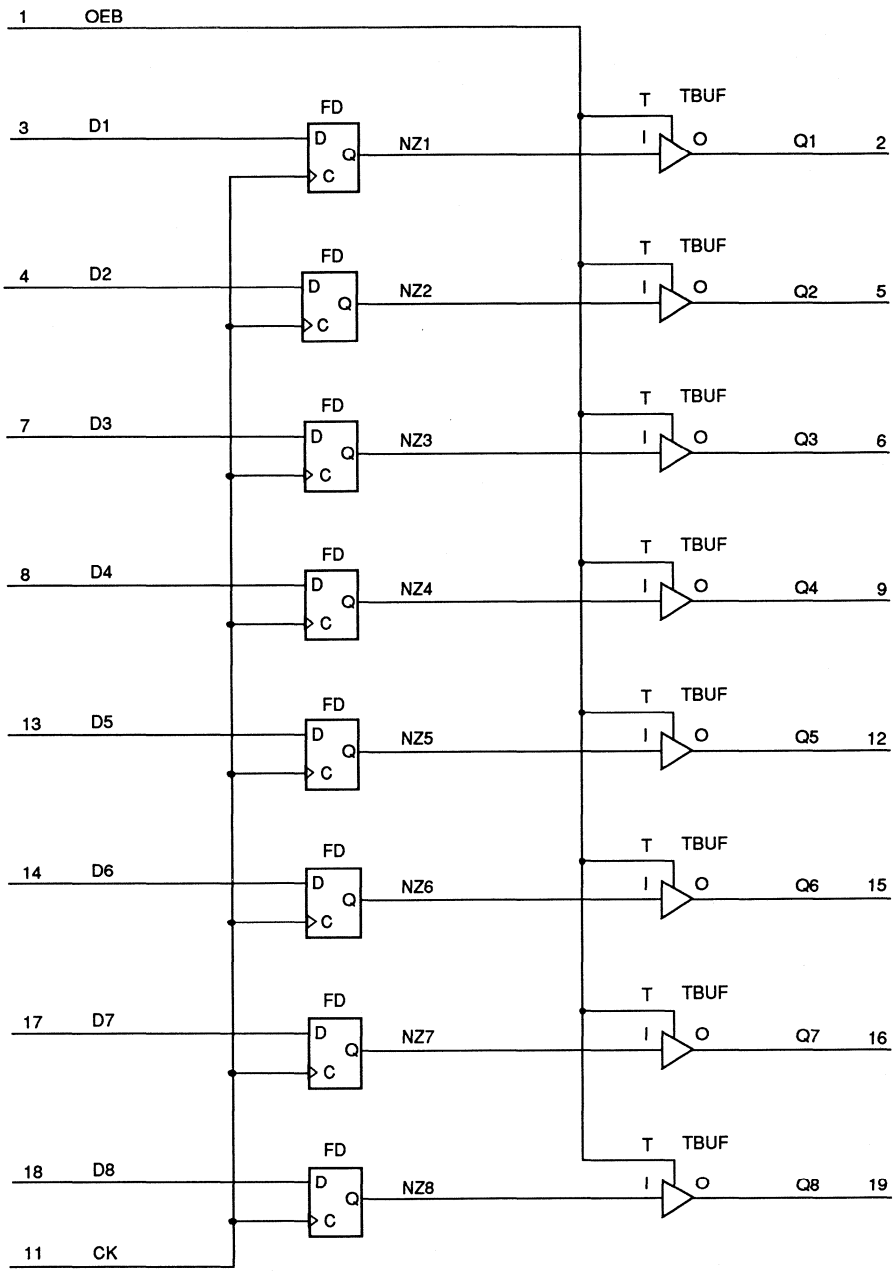
HX352 74352 Dual 4-to-1 Data Selector/Multiplexer



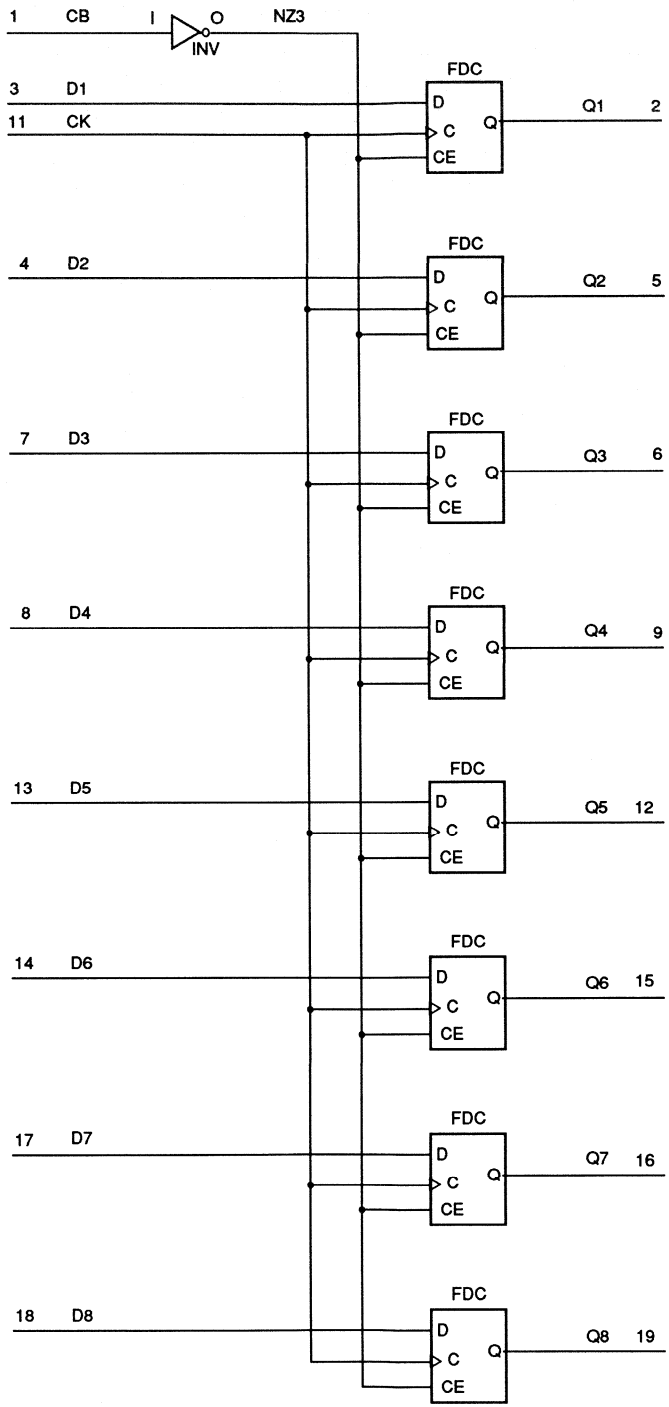
HX373 74373 Octal Latch with 3-State Outputs



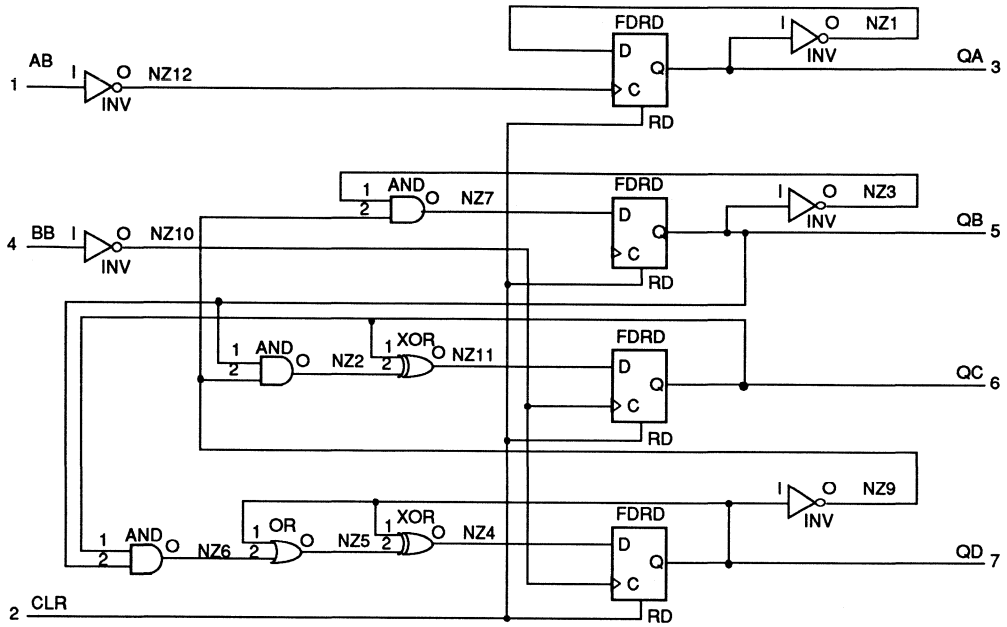
HX374 74374 Octal D Flip-Flops with 3-State Output



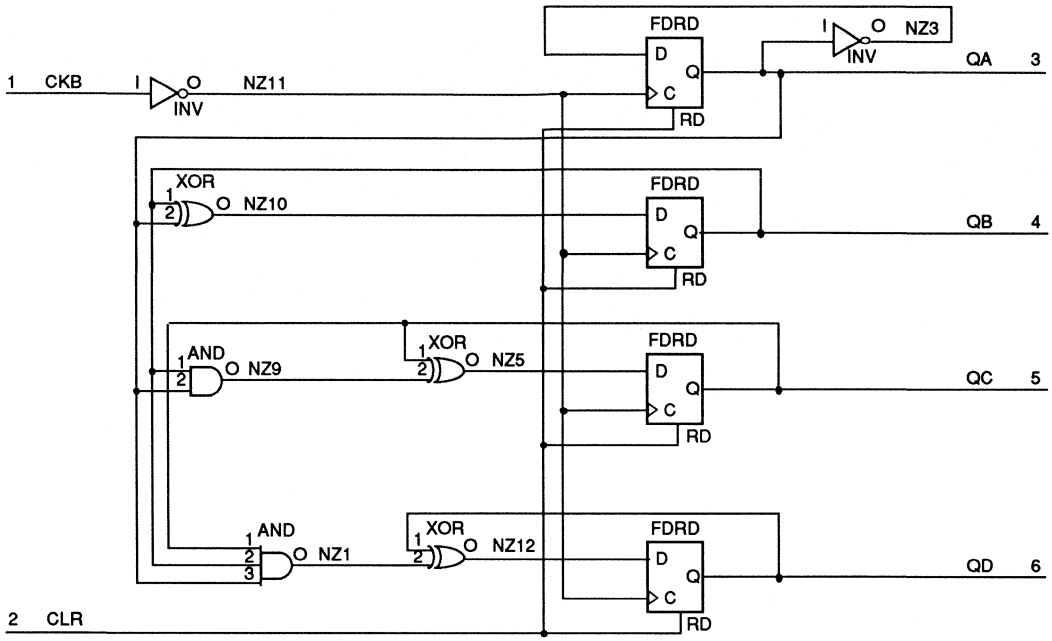
HX377 74377 Octal D Flip-Flops with Clock Enable



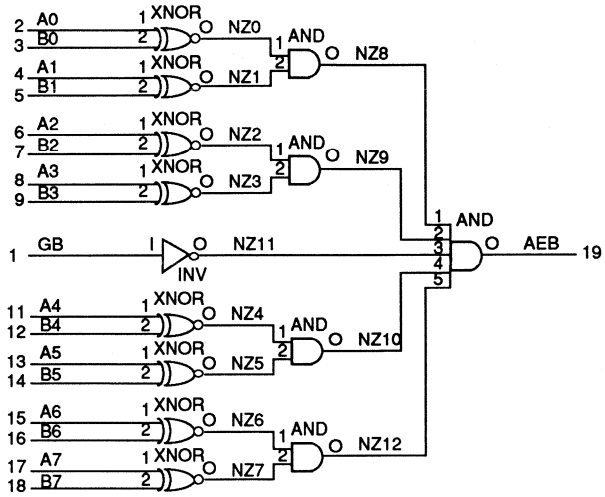
HX390 74390 Dual 4-bit Decade Counters with Clear



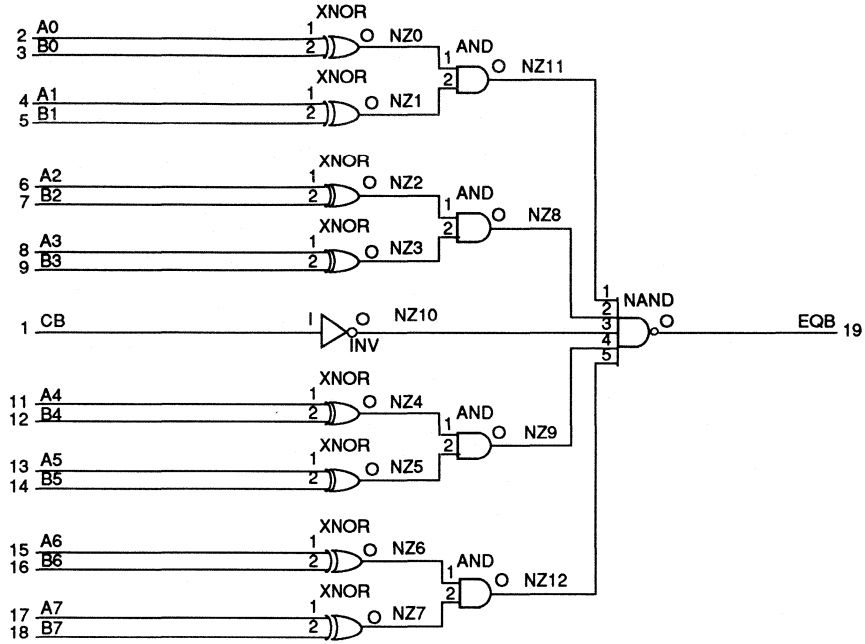
HX393 74393 Dual 4-bit Binary Counters with Clear



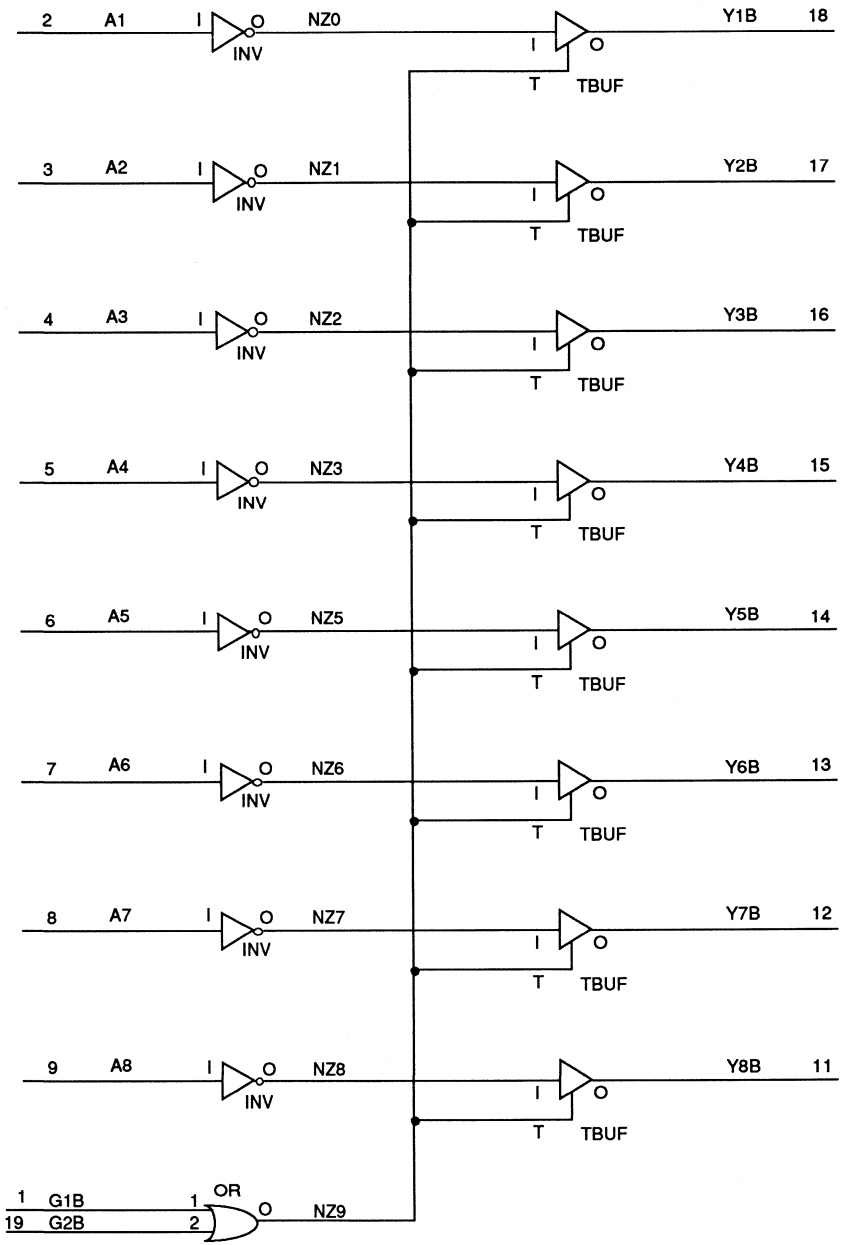
HX518 74518 8-bit Identity Comparator



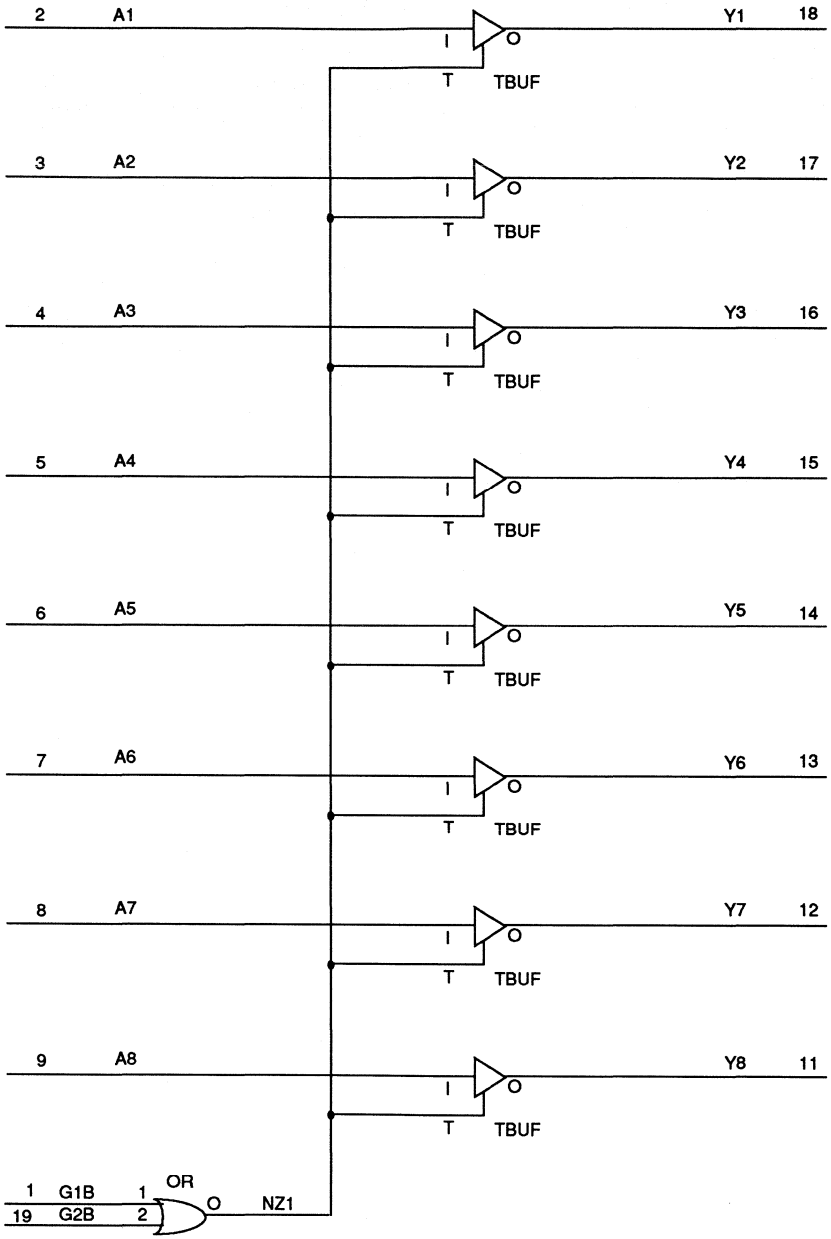
HX521 74521 8-bit Identity Counter



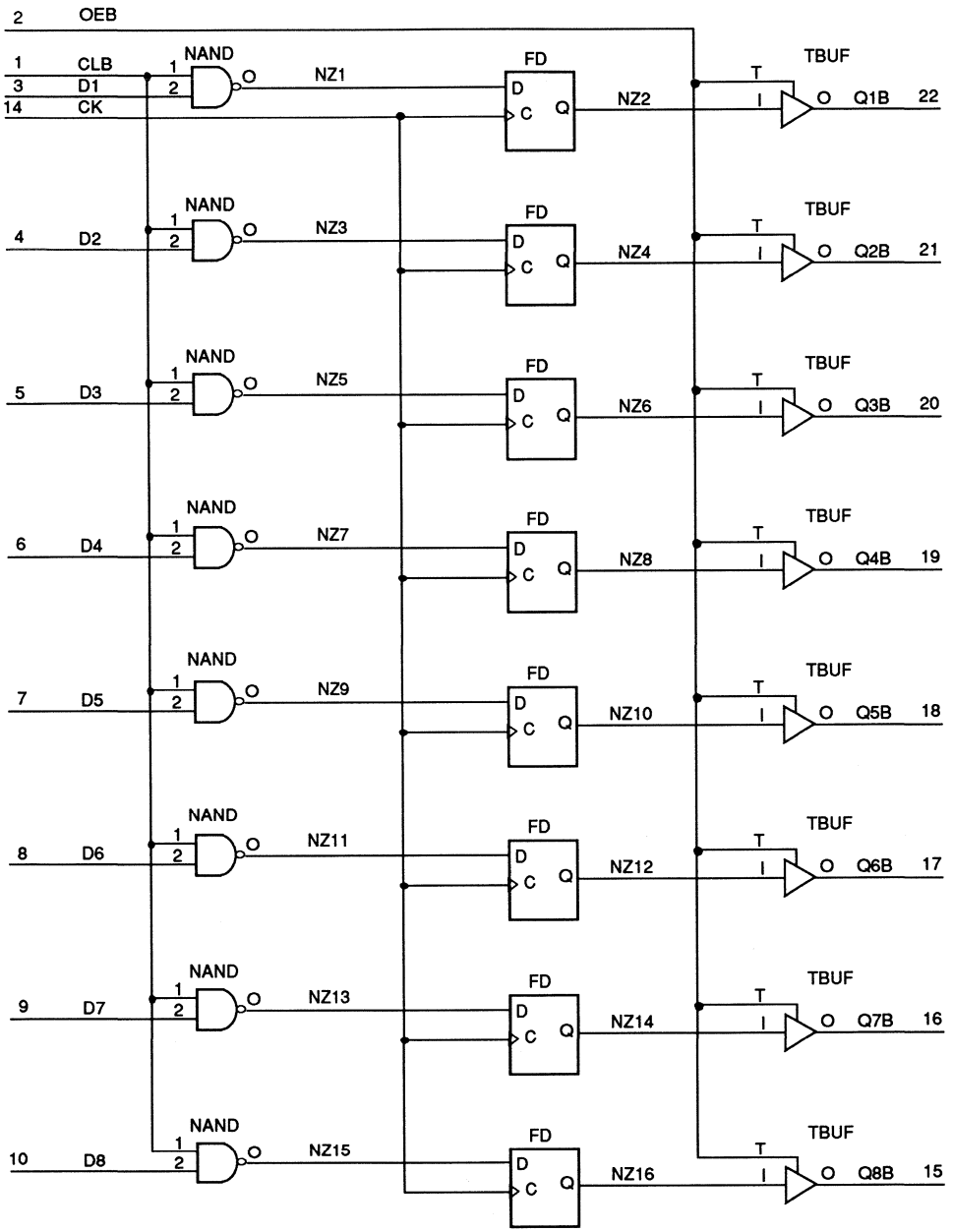
HX540 74540 Octal Inverting Buffer, 3-State Outputs



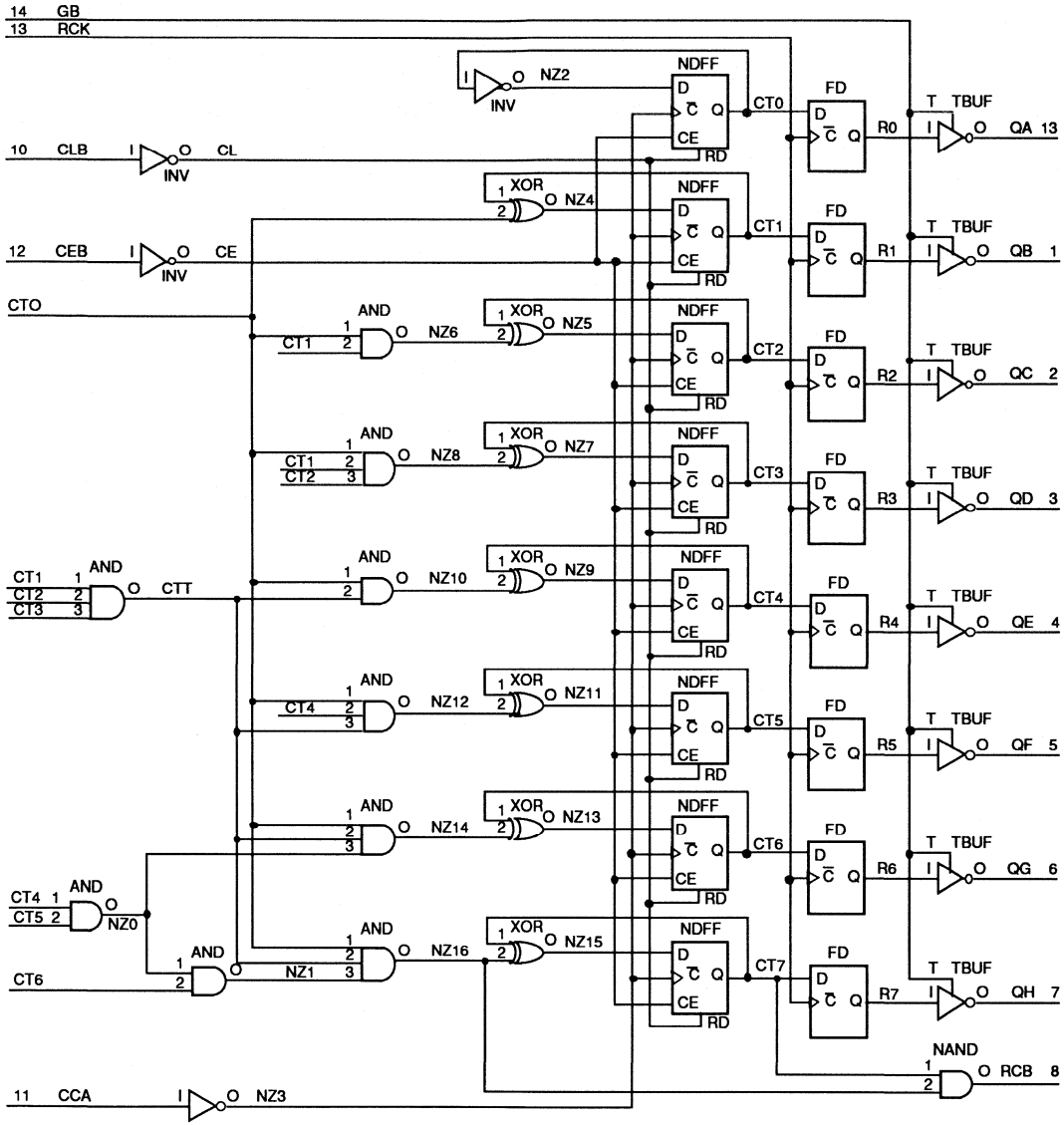
HX541 74541 Octal Noninverting Buffer, 3-State Outputs



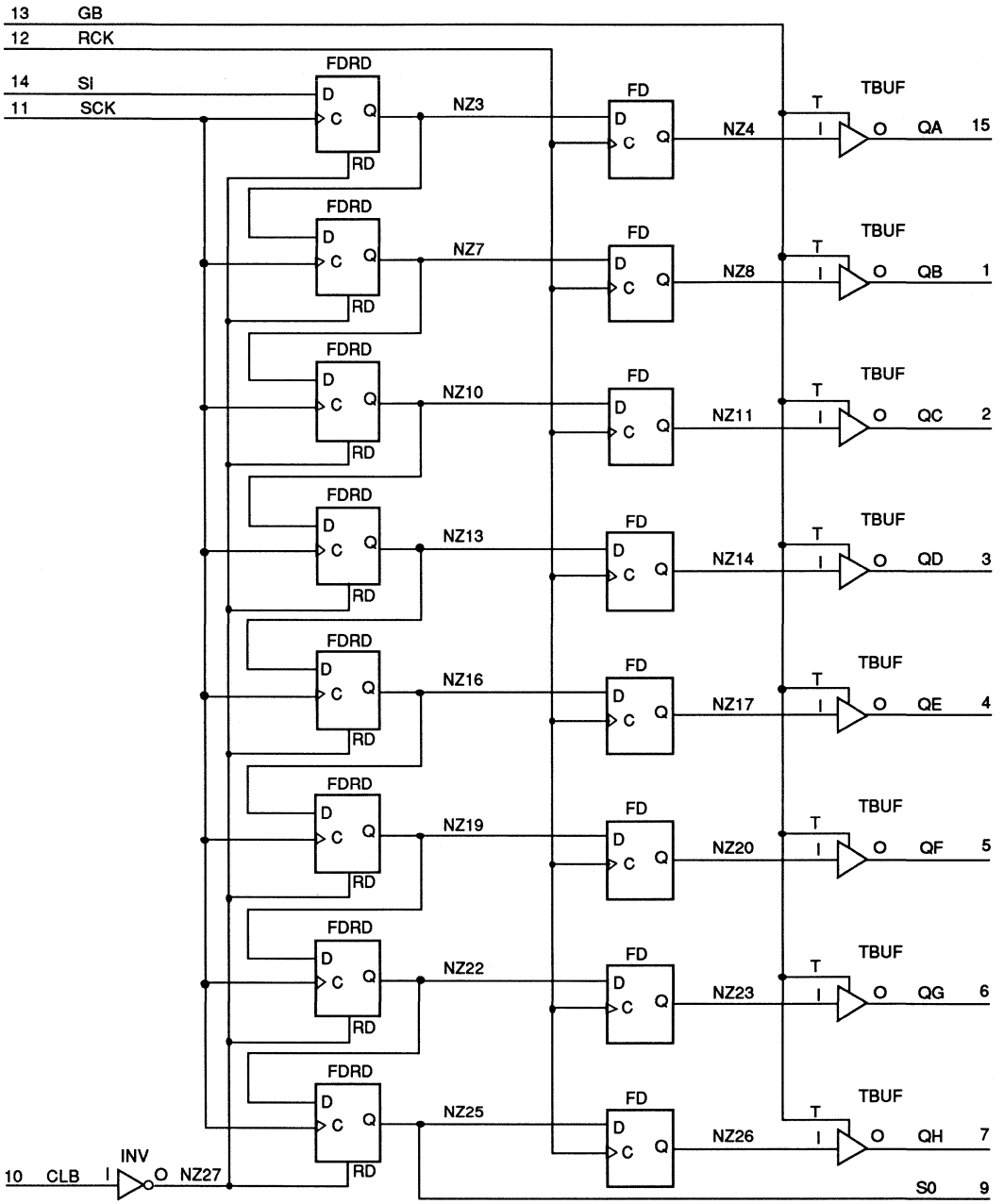
HX577 74577 Octal D Flip-Flop with Reset, 3-State Outputs



HX590 74590 8-bit Binary Counter with Output Register



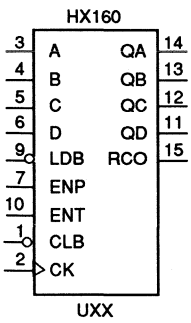
HX595 74595 8-bit Shift Register with Output Register



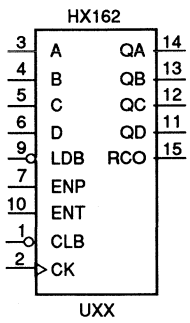
TTL Macros by Functions

Counters

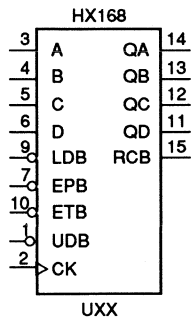
Decade



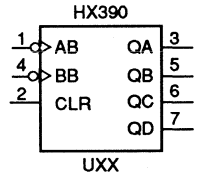
PRESETTABLE
W/ ASYN. RESET



PRESETTABLE
W/ SYNC. RESET

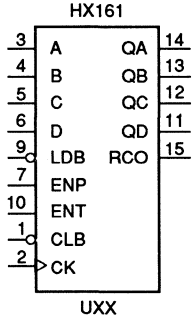


UP/DOWN
UDB = HIGH > UP

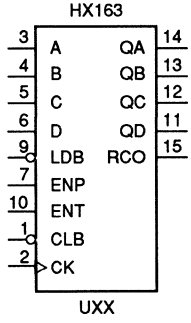


(SINGLE 4-BIT)
DIV BY 2 +
DIV BY 5
(SYNCHRONOUS)

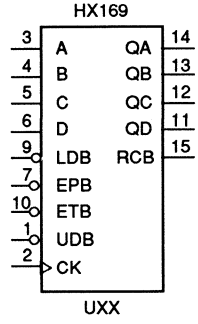
4-bit Binary



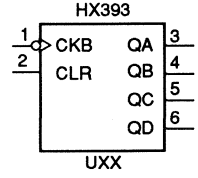
PRESETTABLE
W/ ASYN. RESET



PRESETTABLE
W/ SYNC. RESET

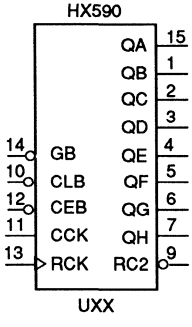


UP/DOWN
UDB = HIGH > UP



(SINGLE 4-BIT
SYNCHRONOUS)

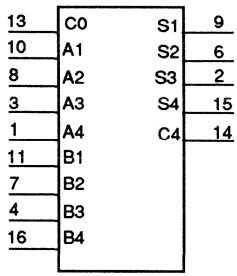
8-bit Binary



OUTPUT REGISTER
WITH 3-STATE BUFFERS

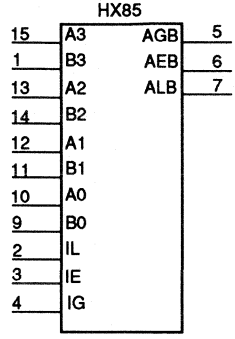
Adders, Comparators, Parity

HX83



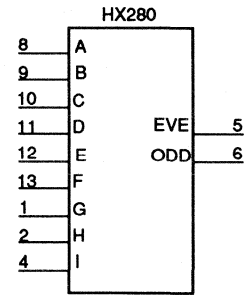
UXX
4-BIT ADD

HX85



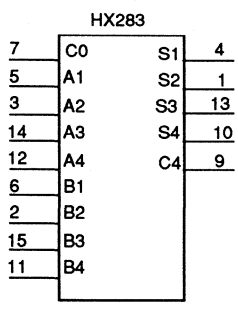
UXX
4-BIT COMPARE
(MAGNITUDE)

HX280



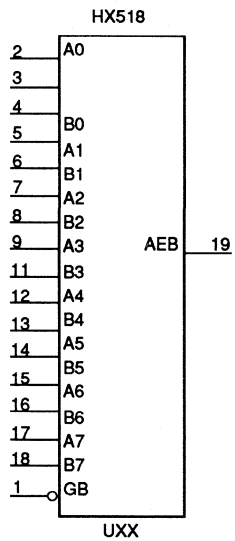
UXX
9-BIT PARITY

HX283



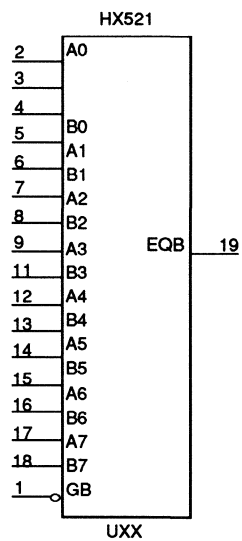
UXX
4-BIT ADD

HX518



UXX
8-BIT COMPARE
(IDENTITY)

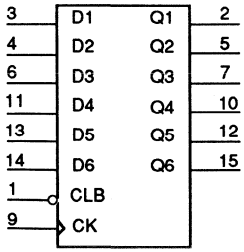
HX521



UXX
8-BIT COMPARE
(INVERTING)

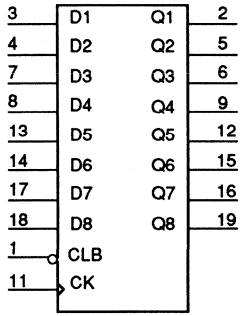
Data Registers

HX174



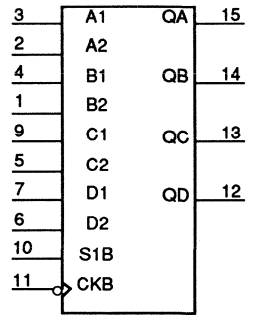
UXX
HEX D

HX273



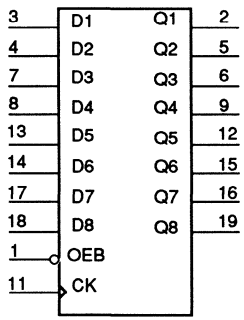
UXX
OCTAL D

HX298



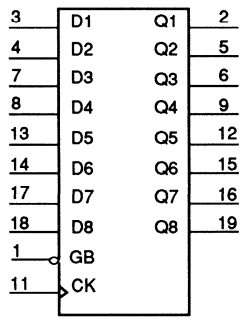
UXX
QUAD 2-INPUT

HX374



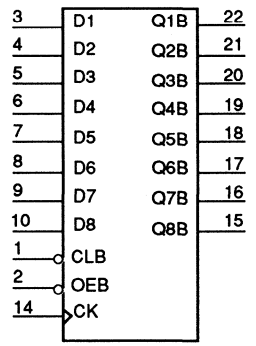
UXX
OCTAL D
WITH 3-STATE

HX377



UXX
OCTAL D
WITH CLOCK ENABLE

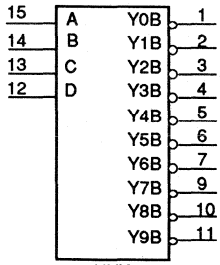
HX577



UXX
OCTAL D
WITH RESET
AND 3-STATE

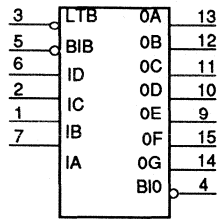
Decoders, Priority Encoders

HX42



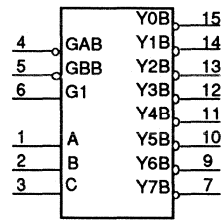
UXX
1-OF-10 WITH ENABLE
L = LSB

HX48



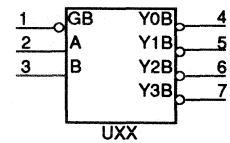
UXX
BCD-TO-7-SEGMENT

HX138



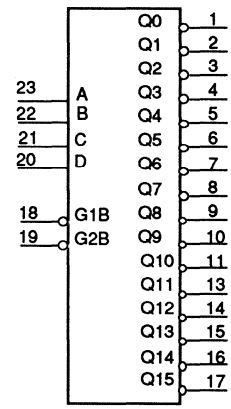
UXX
1-OF-8 WITH ENABLE
L = LSB

HX139



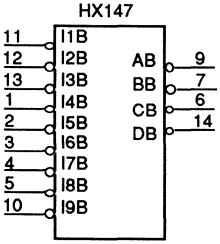
UXX
SINGLE
1-OF-4 WITH ENABLE
L = LSB

HX154



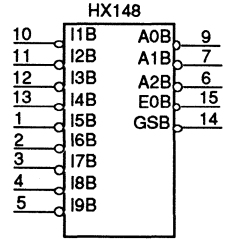
UXX
1-OF-16 WITH ENABLE
L = LSB

HX147



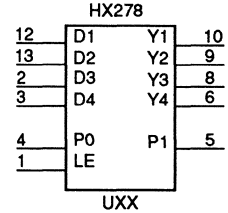
UXX
10-TO-4 LINE PRIORITY

HX148



UXX
8-TO-3 LINE PRIORITY

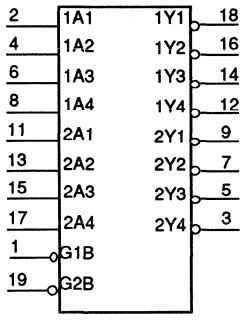
HX278



UXX
4-BIT CASCADABLE
PRIORITY REGISTER

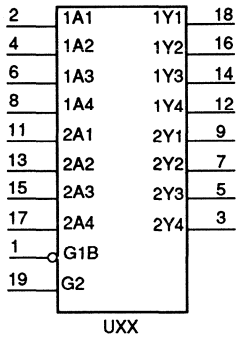
Interface Elements

HX240



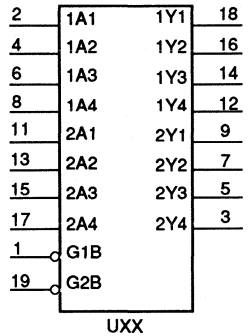
UXX
INVERTING OCTAL
BUFFER WITH 3-STATE

HX241



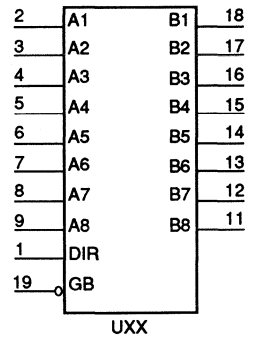
UXX
OCTAL BUFFER
WITH 3-STATE

HX244



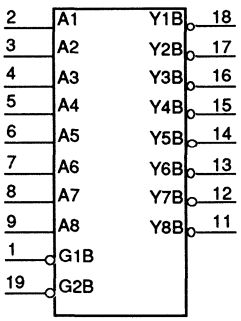
UXX
OCTAL BUFFER
WITH 3-STATE

HX245



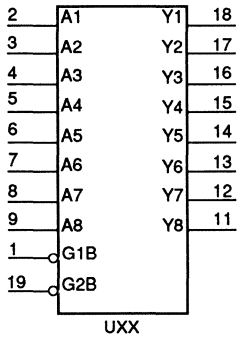
UXX
OCTAL TRANSCEIVER

HX540



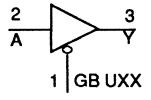
UXX
INVERTING OCTAL
BUFFER WITH 3-STATE

HX541



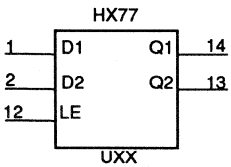
UXX
OCTAL BUFFER
WITH 3-STATE

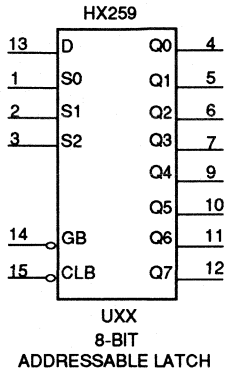
HX125

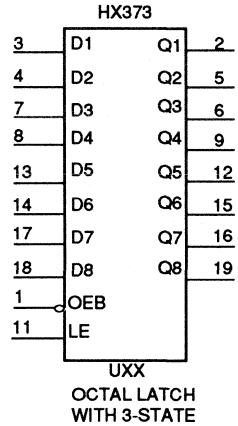


SINGLE-BIT
BUFFER SECTION

Latches

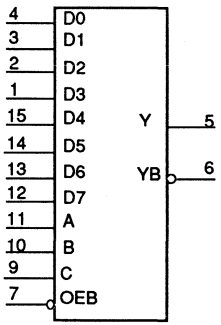

HX77
1 D1
2 D2
12 LE
Q1 14
Q2 13
UXX
DUAL D SECTION WITH
COMMON LOAD


HX259
13 D
1 S0
2 S1
3 S2
14 GB
15 CLB
Q0 4
Q1 5
Q2 6
Q3 7
Q4 9
Q5 10
Q6 11
Q7 12
UXX
8-BIT
ADDRESSABLE LATCH


HX373
3 D1
4 D2
7 D3
8 D4
13 D5
14 D6
17 D7
18 D8
1 OEB
11 LE
Q1 2
Q2 5
Q3 6
Q4 9
Q5 12
Q6 15
Q7 16
Q8 19
UXX
OCTAL LATCH
WITH 3-STATE

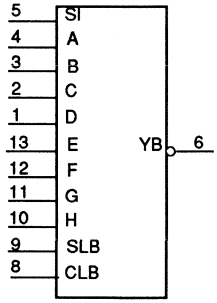
Multiplexers

HX151



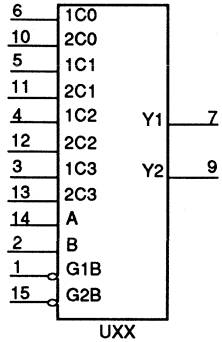
UXX
A = LSB
8-TO-1 WITH ENABLE

HX152



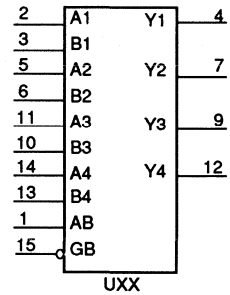
UXX
A = LSB
8-TO-1

HX153



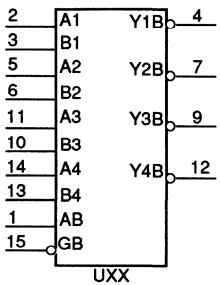
UXX
A = LSB
DUAL 4-TO-1 WITH ENABLES

HX157



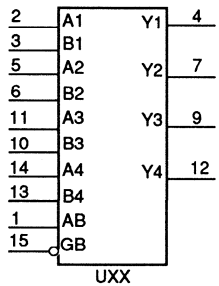
UXX
AB = HIGH > Yi = Bi
QUAD 2-TO-1 WITH ENABLE

HX158



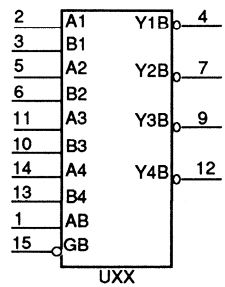
UXX
AB = HIGH > Yi = Bi
QUAD 2-TO-1 WITH ENABLE

HX257



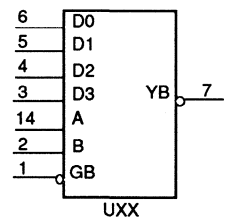
UXX
AB = HIGH > Yi = Bi
QUAD 2-TO-1 WITH ENABLE

HX258



UXX
AB = HIGH > Yi = $\overline{Bi}$
QUAD 2-TO-1 WITH ENABLE

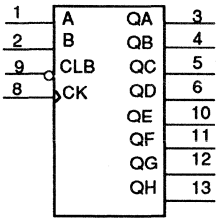
HX352



UXX
(SINGLE SECTION)
4-TO-1 WITH ENABLE
A = LSB

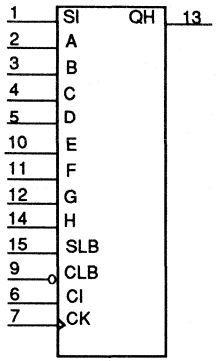
Shift Registers

HX164



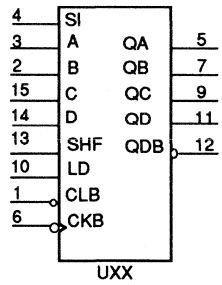
UXX
SERIAL-TO-PARALLEL

HX166



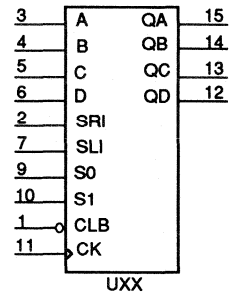
UXX
PARALLEL-TO-SERIAL

HX179



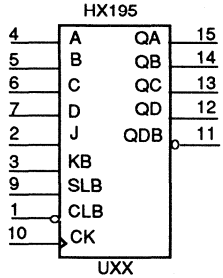
UXX
4-BIT REGISTER

HX194



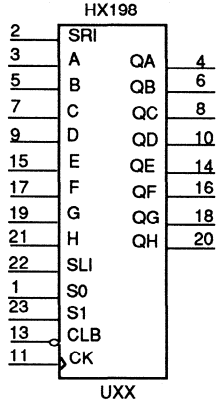
UXX
4-BIT BIDIRECTIONAL

HX195



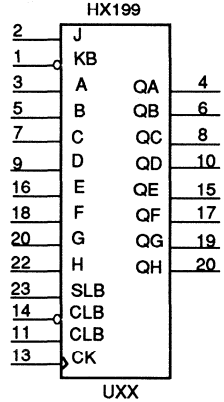
UXX
4-BIT LOAD/SHIFT

HX198



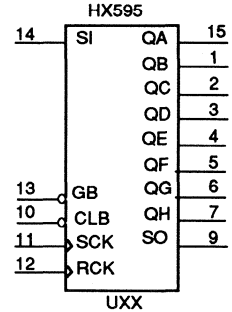
UXX
8-BIT LOAD/LEFT/RIGHT

HX199



UXX
8-BIT LOAD/SHIFT

HX595



UXX
8-BIT SERIAL WITH
3-STATE OUTPUT REGISTER

Custom TTL Macros

X74161U 4-bit Synchronous Loadable Binary Up Counter

Number of CLBs: 5

The X74161U macro is equivalent to the popular 74161 synchronous loadable binary up counter. When $\overline{PE}$ (parallel enable) is low, the rising clock edge loads parallel CET data into the Q outputs. When $\overline{PE}$, CET (count enable trickle), and CEP (count enable parallel) are high, the rising clock edge increments the counter. TC (terminal count) is high when the counter is in state F (all Qs high) and the CET input is high. When RD (reset direct) is high, the outputs are asynchronously reset to 0.

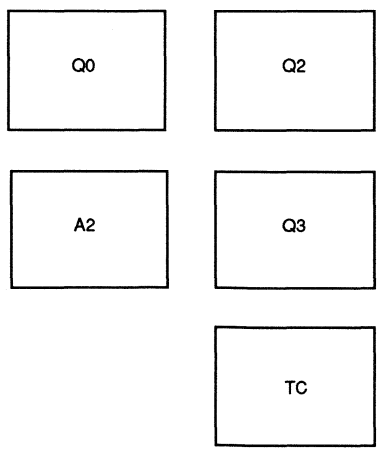
For the highest clock rate, this counter should be expanded by connecting each TC output (except the least significant) to the next significant CET input. The least significant TC should drive all more significant CEP inputs in parallel. The CET input of the second counter should be terminated high.

Best Speed

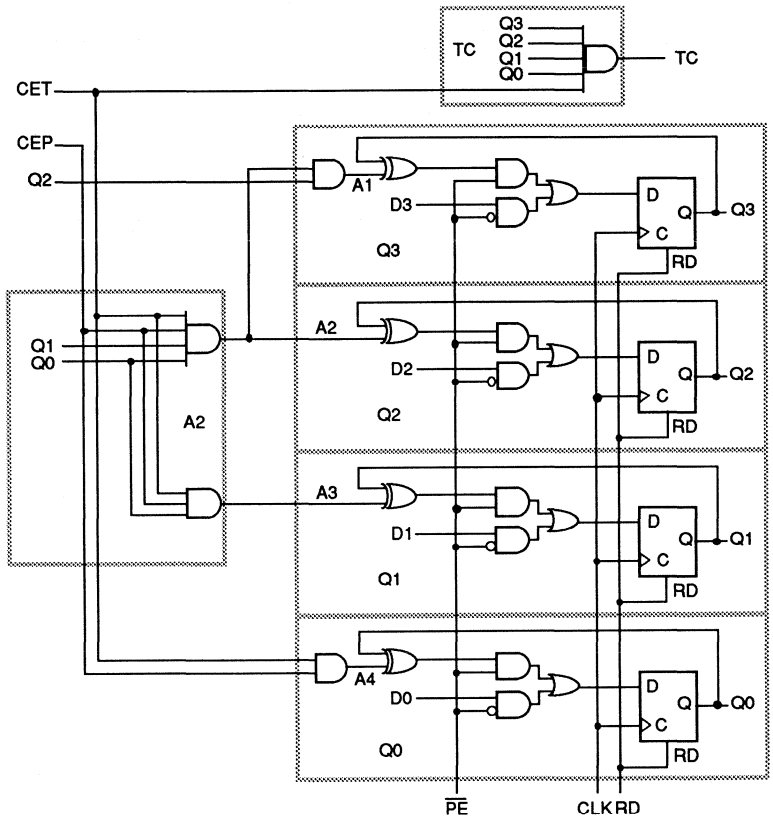
70 MHz FPGA: 33 MHz
100 MHz FPGA: 41 MHz

Place and Route Suggestions

The clock to TC delay is 25 ns. For a faster counter, decrease the net delays by editing the *LCA* file with the *XACT* design editor. The CLB placement is shown below.



X74161U 4-bit Synchronous Loadable Binary Up Counter (continued)



X74165A 8-bit Asynchronous Parallel Load Shift Register

Number of CLBs: 8

The X74165A macro is an asynchronous load variation of the 74165 8-bit parallel load shift register. When LE (latch enable) is high, the parallel data is loaded into the function generator. Then, when CE (clock enable) is high and $\overline{PE}$ (parallel enable) is low, the rising clock edge loads the latched data into the Q outputs. When CE and $\overline{PE}$ are both high, the rising clock edge causes data up (Q0 → Q1, etc.) and shifts SER (serial) into the Q0 output.

Inputs							Outputs	
CE	PE	SER	LE	D0...D7	A0...A7	CLK	Q0	Q1...Q7
L	X	X	X	X	X	X	$Q0_0$	$Q1_0...Q0_0$
H	L	X	L	X	a...h	↑	a	b...h
H	L	X	H	a...h	a...h	↑	a	b...h
H	H	L	X	X	X	↑	L	$Q0_n...Q6_n$
H	H	H	X	X	X	↑	H	$Q0_n...Q6_n$

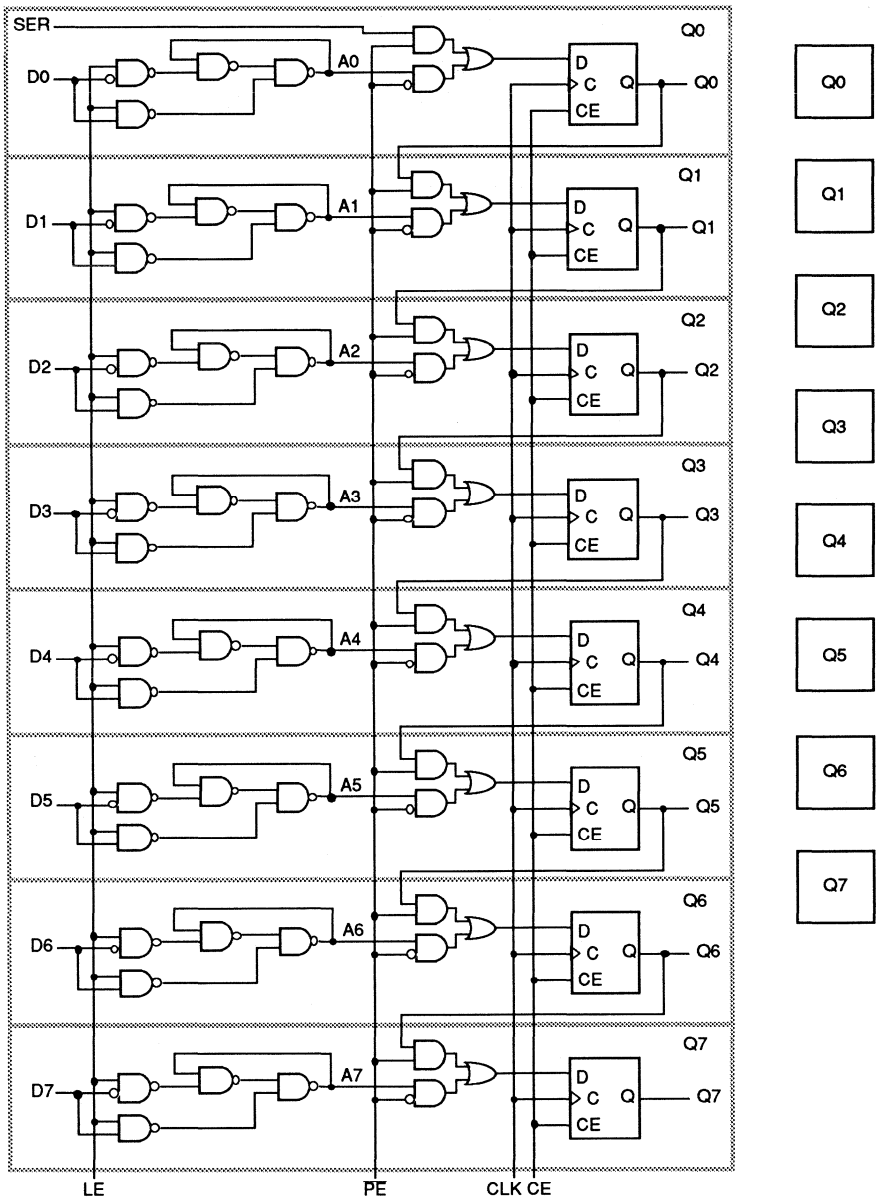
Best Speed

70 MHz FPGA: 55 MHz
100 MHz FPGA: 71 MHz

Place and Route Suggestions

In the 70 and 100 MHz FPGAs, LE requires setup times of 7 ns and 5 ns, respectively. To use the fast direct interconnects, place the CLBs in a row or column. The CLB placement is shown on the next page.

X74165A 8-bit Asynchronous Parallel Load Shift Register (continued)



X74165S 8-bit Synchronous Parallel Load Shift Register

Number of CLBs: 4

The X74165S macro is asynchronous variation of the 74165 8-bit parallel load shift register. When CE (clock enable) is high and $\overline{PE}$ (parallel enable) is low, the rising edge of the clock loads the parallel data into the Q outputs. When CE and $\overline{PE}$ are high, the rising clock edge shifts the outputs down and shifts SER (serial) into the Q0 output.

Inputs					Outputs		
CE	PE	SER	D0...D7	CLK	Q0	Q1	Q7
L	X	X	X	X	$Q0_0$	$Q1_0$	$Q0_0$
H	L	X	a...h		a	b	h
H	H	L	X		L	$Q0_nQ$	$Q6_n$
H	H	H	X		H	0_n	$Q6_n$

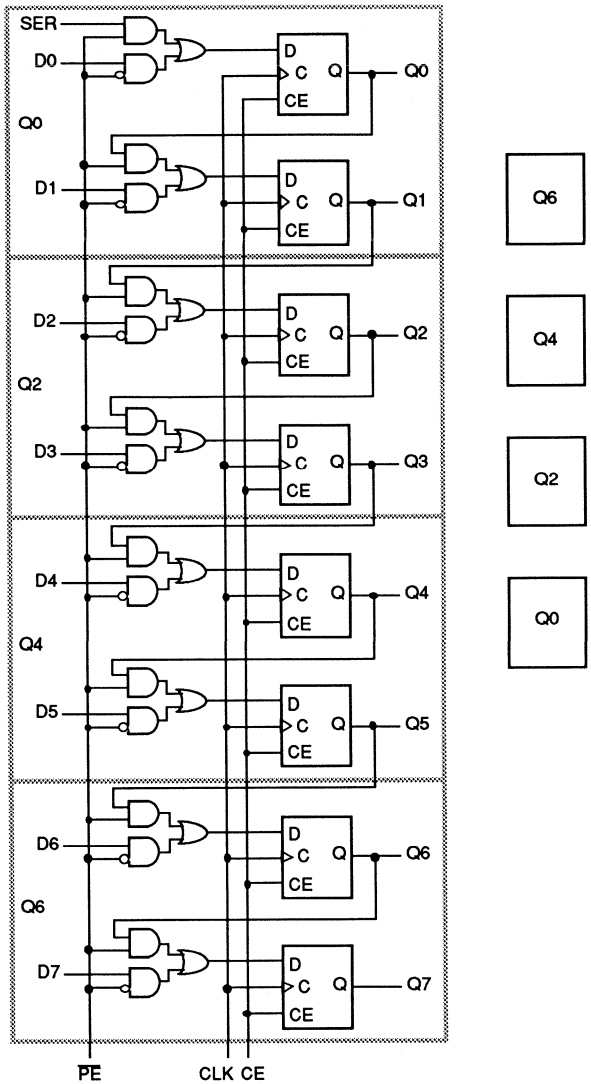
Best Speed

70 MHz FPGA: 55 MHz
100 MHz FPGA: 71 MHz

Place and Route Suggestions

To use the fast direct interconnects, place the CLBs in a row or column. The CLB placement is shown on the next page.

X74165S 8-bit Synchronous Parallel Load Shift Register (continued)



BRLSHFT4 4-Input Barrel Shifter

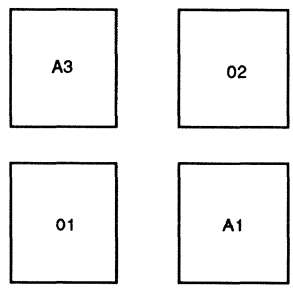
Number of CLBs: 4

The BRLSHFT4 macro is a 4-input barrel shifter with four data inputs, four data outputs, and two control inputs that specify rotation by 0, 1, 2, or 3 positions.

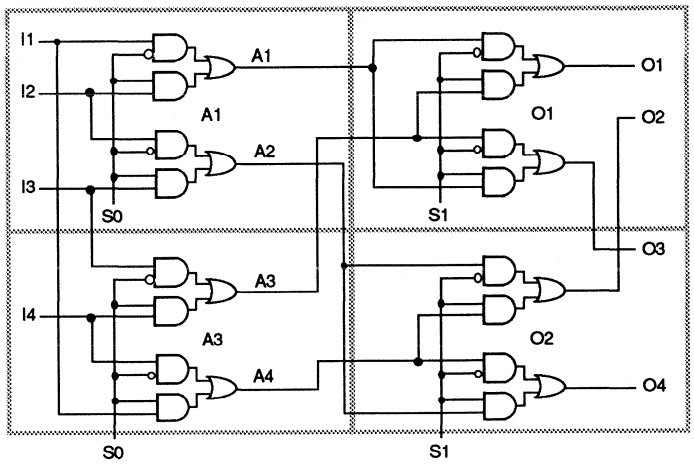
Inputs						Outputs			
S1	S0	I1	I2	I3	I4	Q1	Q2	Q3	Q4
L	L	a	b	c	d	a	b	c	d
L	H	a	b	c	d	b	c	d	a
H	L	a	b	c	d	c	d	a	b
H	H	a	b	c	d	d	a	b	c

Place and Route Suggestions

In the 70 MHz and 100 MHz FPGAs, the BRLSHFT4 has propagation delays of 17 ns and 13 ns, respectively, based on breadboard results. To use the fast direct interconnects, place the CLBs in the postion shown below.



BRLSHFT4 4-Input Barrel Shifter (continued)



BRM Binary Rate Multiplier

Number of CLBs: 1.5/bit

The BRM macro is a binary rate multiplier consisting of two macros: BRM and BRM2. BRM has two control inputs, A and B, which multiply the clock by 1/2 and 1/4, respectively. By cascading BRM2 onto BRM as shown, C and D multiply the clock by 1/8 and 1/16, respectively. If several control inputs are asserted, their sum multiplies the clock. For example, if A and B are both high, the clock is multiplied by 1/2 + 1/4 = 3/4. To multiply the clock by smaller fractions, additional BRM2 macros can be cascaded without speed loss.

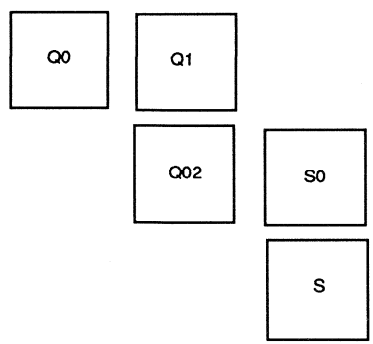
Inputs				
A	**B**	**C**	**D**	**Multiplication Factor**
H	L	L	L	1/2
L	H	L	L	1/4
L	L	H	L	1/8
L	L	L	H	1/16
H	H	H	L	1/2 + 1/4 + 1/8 = 7/8 (for example)

Best Speed

70 MHz FPGA: 34 MHz
100 MHz FPGA: 50 MHz

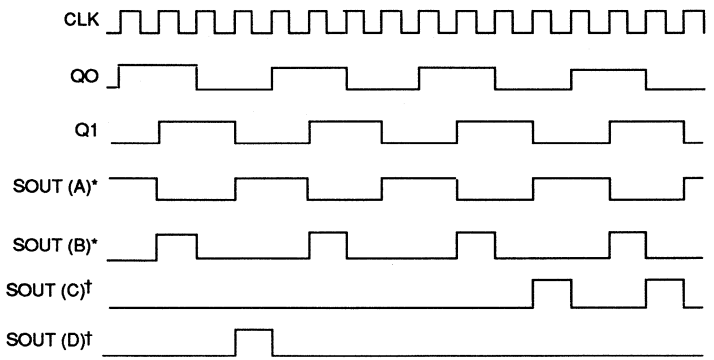
Place and Route Suggestions

The CLB placement is shown below.



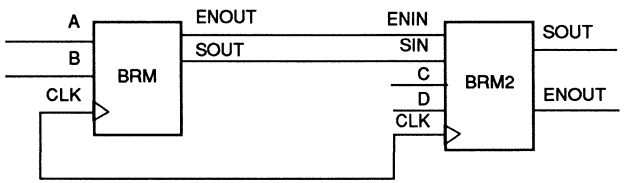
BRM Binary Rate Multiplier (continued)

Binary Rate Multiplier Waveforms

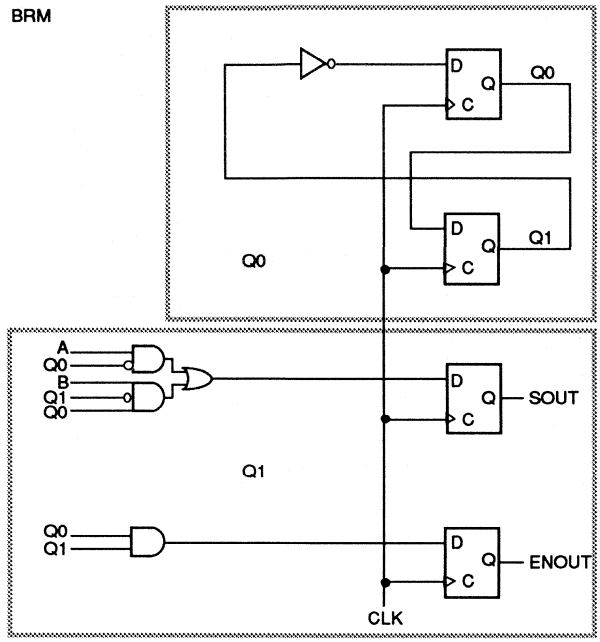


* SOUT of BRM with input in parentheses high and all other inputs low.
† SOUT of BRM2 with the input in parentheses high and all other inputs low.

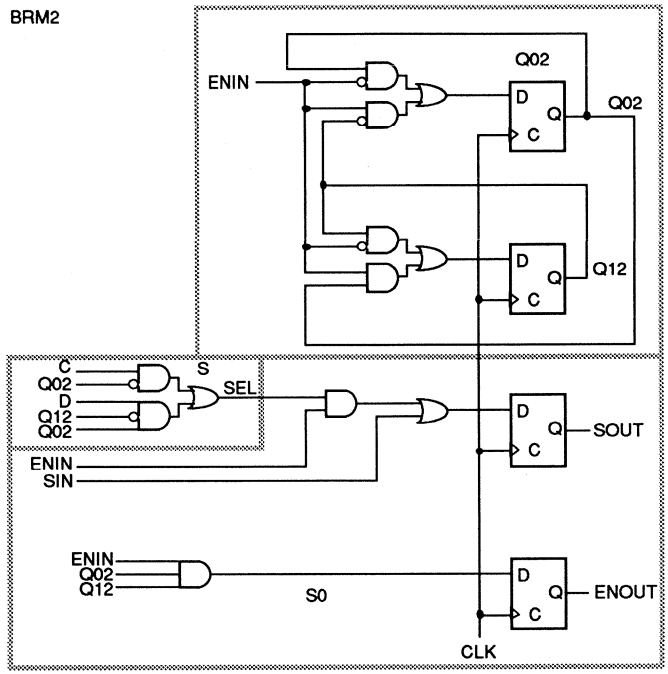
BRM1



BRM Binary Rate Multiplier (continued)



BRM Binary Rate Multiplier (continued)



CBINRIP 4-bit Binary Counter with Ripple Cascade

Number of CLBs: 2

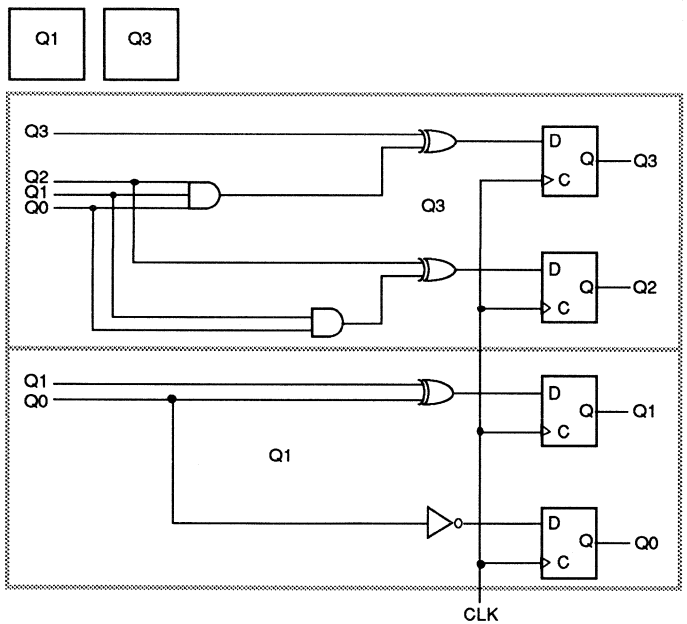
The CBINRIP macro is a 4-bit binary counter that counts from 0—15. Since terminal count (TC) is not decoded, the counter must be ripple-cascaded using Q3 as the falling-edge clock for the next higher 4-bit counter.

Best Speed

70 MHz FPGA: 50 MHz
100 MHz FPGA: 62 MHz

Place and Route Suggestions

The CLB placement is shown below.



CDECRIP 4-bit BCD Counter with Ripple Cascade

Number of CLBs: 2

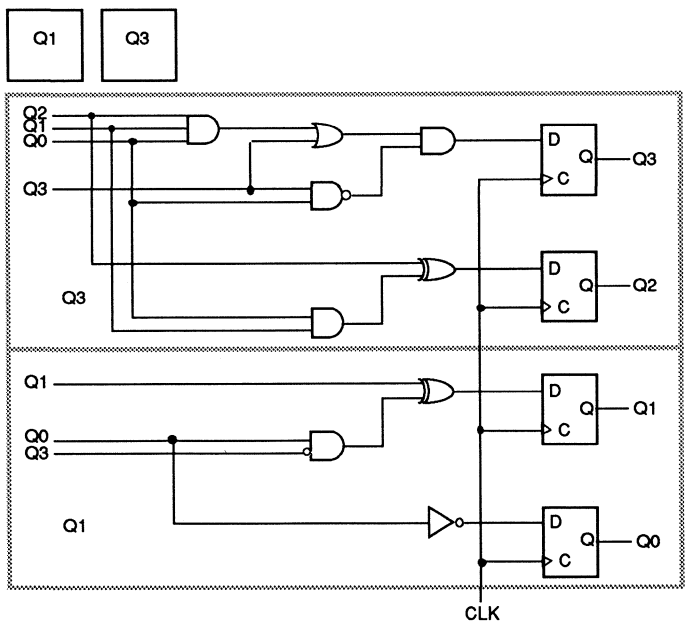
The CDECRIP macro is a 4-bit BCD ripple counter that counts from 0—9. Since terminal count (TC) is not decoded, the counter must be ripple-cascaded using Q3 as the falling-edge clock for the next higher 4-bit counter.

Best Speed

70 MHz FPGA: 50 MHz
100 MHz FPGA: 62 MHz

Place and Route Suggestions

The CLB placement is shown below.



C3BIT8 (7, 6, 5, 4) 3-bit Divide-by-8 (7, 6, 5, 4) Shift Register Counter

Number of CLBs: 2

The C3BIT8 is an efficient and fast synchronous counter with a built-in decoder/encoder. The counter consists of a 3-bit shift register with a 3-input function generator controlling the data input to the first bit. The function generator has been programmed for division by any number from 4—8. The order in which each of these macros count is shown below.

Counting Sequences

C3BIT8	C3BIT7	C3BIT6	C3BIT5	C3BIT4
000	000	000	000	000
100	100	100	100	100
110	110	110	110	010
111	011	111	001	
011	101	011	001	
101	010	001		
010	001			
001				

Notes:
Changing the dividing factor requires the inversion of some inputs. See specific macro drawing file.

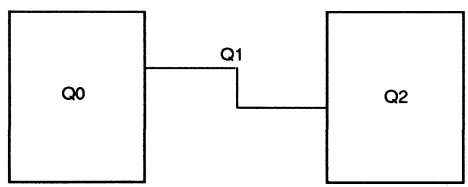
The pipelined decoder currently asserts the output TC when the counter is in state 001, but it can be reprogrammed to fit any desired asynchronous waveform.

Best Speed

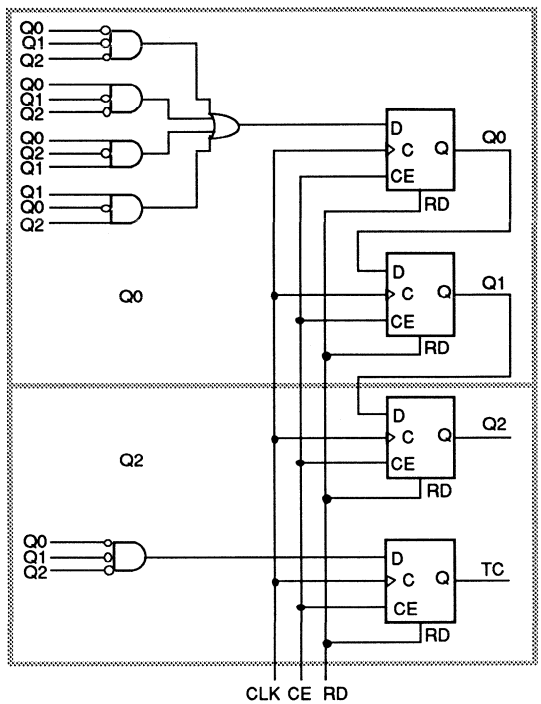
70 MHz FPGA: 45 MHz
100 MHz FPGA: 55 MHz

Place and Route Suggestions

The CLB placement is shown below.



C3BIT8 (7, 6, 5, 4) 3-bit Divide-By-8 (7, 6, 5, 4) Shift Register Counter
(continued)



C3BIT807 3-bit Divide-by-8 or -7 Shift Register Counter

Number of CLBs: 2

The C3BIT807 is an efficient and fast synchronous counter with a built-in decoder/encoder. The counter consists of a 3-bit shift register with a 3-input function generator controlling the data input to the first bit. It is programmed to count in the order shown below.

Counting Sequence

000
100
110
111
011
101
010
001

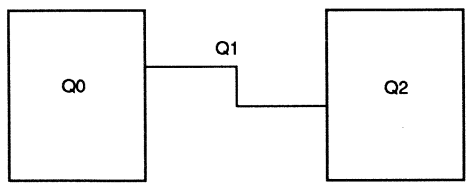
The pipelined decoder currently asserts the output TC when the counter is in state 001, but it can be reprogrammed to fit any desired asynchronous waveform. When the control input (CNTL) is high, the counter divides by eight. When CNTL is low, state 111 is skipped and the counter divides by seven.

Best Speed

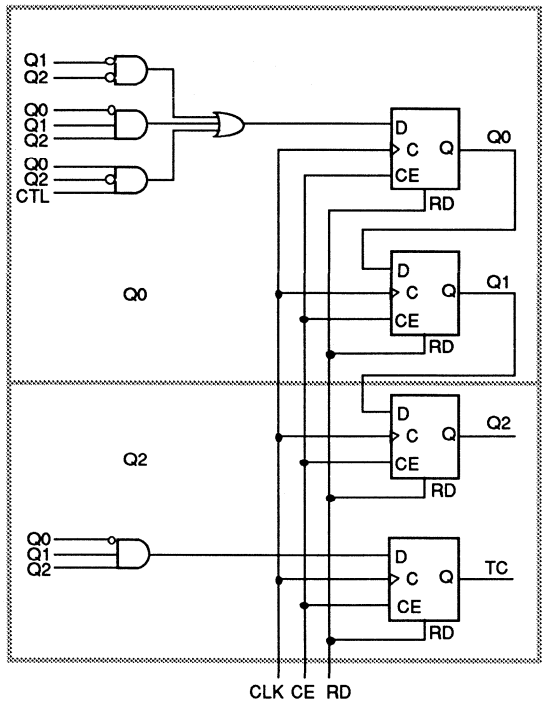
70 MHz FPGA: 45 MHz
100 MHz FPGA: 55 MHz

Place and Route Suggestions

The CLB placement is shown below.



C3BIT807 3-bit Divide-by-8 or -7 Shift Register Counter (continued)



C5BIT32 (31, 30, . . .9) 5-bit Divide-by-32 (31, 30, . . .9) Shift Register Counter

Number of CLBs: 3

The C5BIT32 is an efficient and fast synchronous counter with a built-in decoder/encoder. The counter consists of a 5-bit shift register with a 5-input function generator controlling the data input to the first bit. The function generator is programmed for division by any number from 9—32. The pipelined decoder currently asserts the output TC when the counter is in state 00001, but it can be reprogrammed for any synchronous waveform.

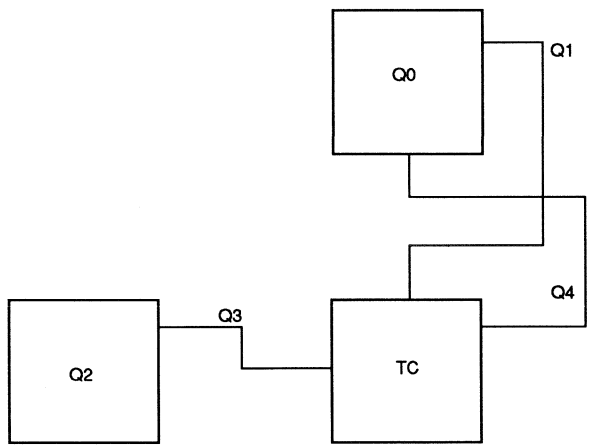
Best Speed

70 MHz FPGA: 38 MHz
100 MHz FPGA: 55 MHz

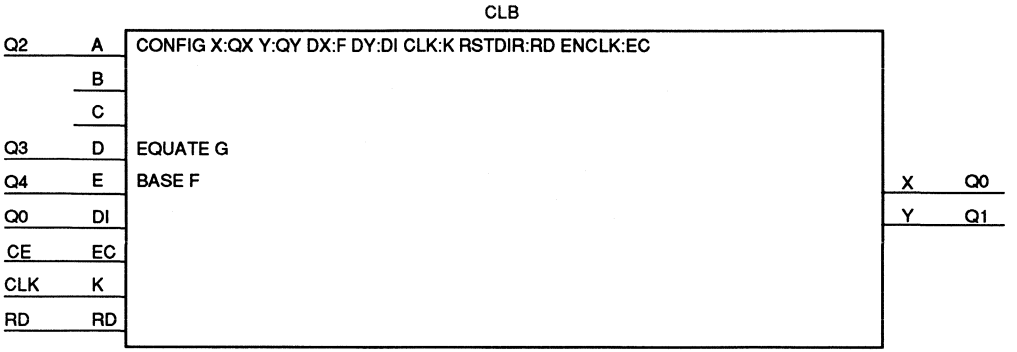
Place and Route Suggestions

To correctly partition this design, use CLB primitives instead of CLBMAPs because of the unorthodox use of DIN input. Both the CLB primitives and the schematic for the C5BIT32 are illustrated.

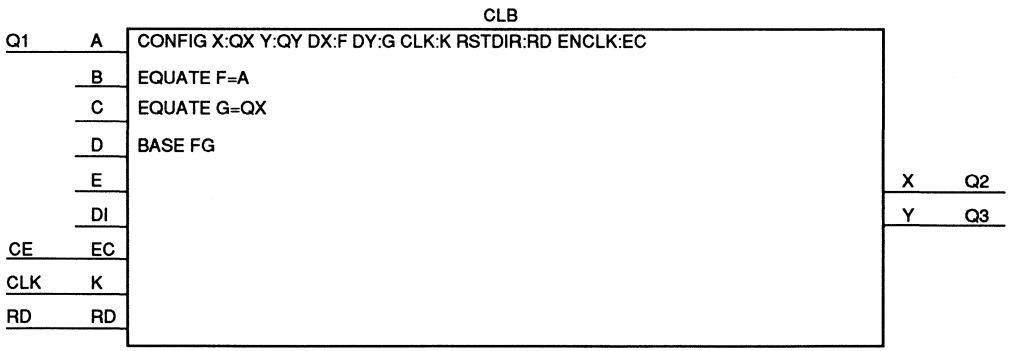
The CLB placement is shown below.

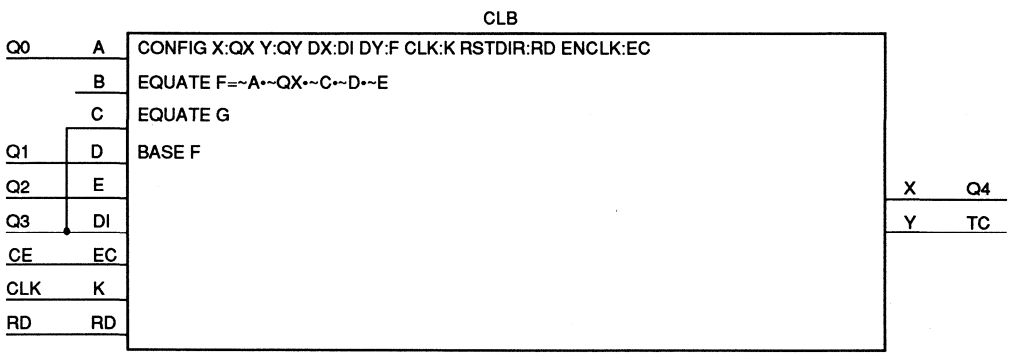


C5BIT32 (31, 30, . . .9) 5-bit Divide-by-32 (31, 30, . . .9) Shift Register Counter (continued)



EQUATE F= ~QX•~QY•~A•~D•~E+QX•~QY•~A•~D•~E+QX•QY•~A•~D•~E+
QX•QY•A•~D•~E+QX•QY•A•D•~E+~QX•QY•A•D•E+QX•~QY•A•D•E+
QX•QY•~A•D•E+~QX•~QY•A•D•E+QX•~QY•~A•D•E+~QX•QY•A•~D
•~E+~QX•~QY•~A•D•~E+~QX•~QY•A•~D•~E+~QX•QY•~A•~D•E+
~QX•QY•~A•D•~E+QX•~QY•A•~D•E





C5BIT32 (31, 30, . . .9) 5-bit Divide-by-32 (31, 30, . . .9) Shift Register Counter (continued)

Counting Sequence

The counting order for the C5BIT32 through the C5BIT9 is shown below.

Note: Changing the dividing factor requires the simple manipulation of the Boolean equations. See specific macro drawing file.

C5BIT32

00000	01111	10011	00010	10100	00011
10000	10111	11001	10001	01010	00001
11000	11011	01100	01000	10101	
11100	11101	10110	00100	11010	
11110	01110	01011	10010	01101	
11111	00111	00101	01001	00110	

C5BIT31

00000	10111	11001	10001	01010	00001
10000	11011	01100	01000	10101	
11000	11101	10110	00100	11010	
11100	01110	01011	10010	01101	
11110	00111	00101	01001	00110	
01111	10011	00010	10100	00011	

C5BIT30

00000	11111	01110	10110	01000	11010
10000	01111	00111	01011	00100	01101
11000	10111	10011	00101	10010	00110
11100	11011	11001	00010	01001	00011
11110	11101	01100	10001	10100	00001

C5BIT29

00000	01111	00111	01011	00100	01101
10000	10111	10011	00101	10010	00110
11000	11011	11001	00010	01001	00011
11100	11101	01100	10001	10100	00001
11110	01110	10110	01000	11010	

C5BIT28

00000	11111	01100	10001	10100	00110
10000	01111	10110	01000	01010	00011
11000	00111	01011	00100	10101	00001
11100	10011	00101	10010	11010	
11110	11001	00010	01001	01101	

C5BIT32 (31, 30, . . .9) 5-bit Divide-by-32 (31, 30, . . .9) Shift Register Counter (continued)

Counting Sequence (continued)

C5BIT27

00000	01111	10110	01000	01010	00011
10000	00111	01011	00100	10101	00001
11000	10011	00101	10010	11010	
11100	11001	00010	01001	01101	
11110	01100	10001	10100	00110	

CC5BIT26

00000	00111	01011	00100	10101	00001
10000	10011	00101	10010	11010	
11000	11001	00010	01001	01101	
11100	01100	10001	10100	00110	
01110	10110	01000	01010	00011	

C5BIT25

00000	01111	10110	01000	11010	
10000	00111	01011	00100	01101	
11000	10011	00101	10010	00110	
11100	11001	00010	01001	00011	
11110	01100	10001	10100	00001	

C5BIT24

00000	11110	01011	01000	10100	01101
10000	11111	00101	00100	01010	00110
11000	01111	00010	10010	10101	00011
11100	10111	10001	01001	11010	00001

C5BIT23

00000	11110	11011	10011	01011	00110
10000	11111	11101	11001	10101	00011
11000	01111	01110	01100	11010	00001
11100	10111	00111	10110	01101	

C5BIT22

00000	11110	11011	10011	10101	00011
10000	11111	11101	01001	11010	00001
11000	01111	01110	10100	01101	
11100	10111	00111	01010	00110	

C5BIT32 (31, 30, . . .9) 5-bit Divide-by-32 (31, 30, . . .9) Shift Register Counter (continued)

Counting Sequence (continued)

C5BIT21

00000	10110	10001	01001	11010	00001
10000	01011	01000	10100	01101	
11000	00101	00100	01010	00110	
01100	00010	10010	10101	00011	

C5BIT20

00000	11110	11011	10011	01011
10000	11111	11101	11001	00101
11000	01111	01110	01100	00010
11100	10111	00111	10110	00001

C5BIT19

00000	11110	11101	11001	00101
10000	01111	01110	01100	00010
11000	10111	00111	10110	00001
11100	11011	10011	01011	

C5BIT18

00000	11100	01111	11101	10011	00110
10000	11110	10111	01110	11001	00011
11000	11111	11011	00111	01100	00001

C5BIT17

00000	11100	10111	01110	11001	00011
10000	11110	11011	00111	01100	00001
11000	01111	11101	10011	00110	

C5BIT16

00000	11100	10011	01010	01101	00001
10000	01110	01001	10101	00110	
11000	00111	10100	11010	00011	

C5BIT15

00000	11100	01111	10101	00110
10000	11110	10111	11010	00011
11000	11111	01011	01101	00001

C5BIT32 (31, 30, . . .9) 5-bit Divide-by-32 (31, 30, . . .9) Shift Register Counter (continued)

Counting Sequence (continued)

C5BIT14

00000	00100	10100	11010	00011
10000	10010	01010	01101	00001
01000	01001	10101	00110	

C5BIT13

00000	11100	00111	01100	00001
10000	11110	10011	00110	
11000	01111	11001	00011	

C5BIT12

00000	01000	10010	10100	01101	00011
10000	00100	01001	11010	00110	00001

C5BIT11

00000	01000	01010	11010	00110	00001
10000	10100	10101	01101	00011	

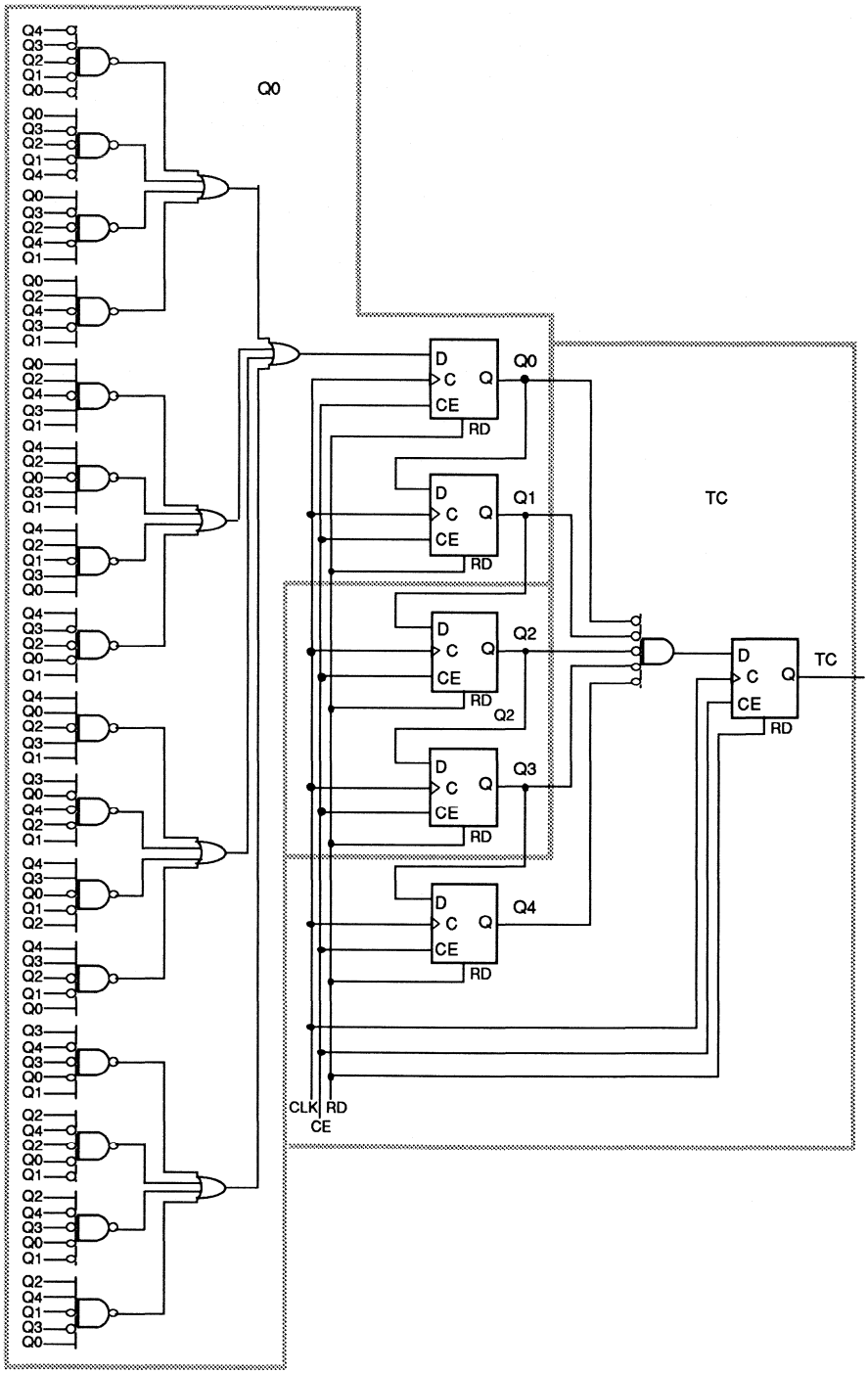
C5BIT10

00000	11000	11110	01111	00011
10000	11100	11111	00111	00001

C5BIT9

00000	11000	11110	00111	00001
10000	11100	01111	00011	

C5BIT32 (31, 30, . . .9) 5-bit Divide-by-32 (31, 30, . . .9) Shift Register Counter (continued)



C3SQUARE Divide-by-3 with 50% Duty Cycle

Number of CLBs: 2

The C3SQUARE macro is a synchronous divide-by-3 counter with a 50% duty cycle on the DIV output when driven by a 50% duty cycle clock.

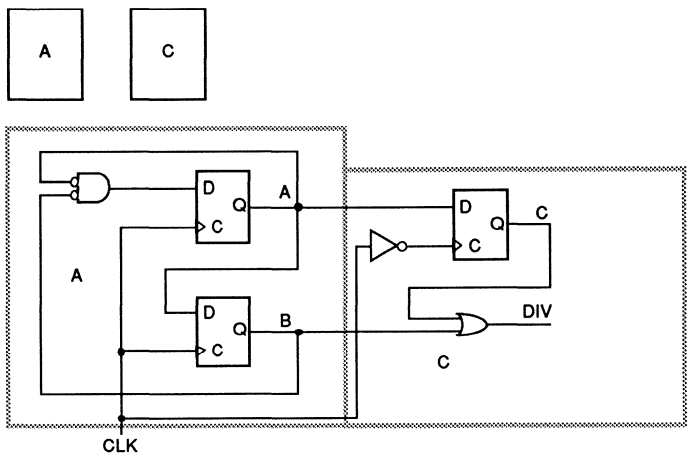
For the duty cycle to be exactly 50%, the net delays on A and B must be equal.

Best Speed

70 MHz FPGA: 27 MHz
100 MHz FPGA: 33 MHz

Place and Route Suggestions

The CLB placement is shown below.



C5SQUARE Divide-by-5 with 50% Duty Cycle

Number of CLBs: 2

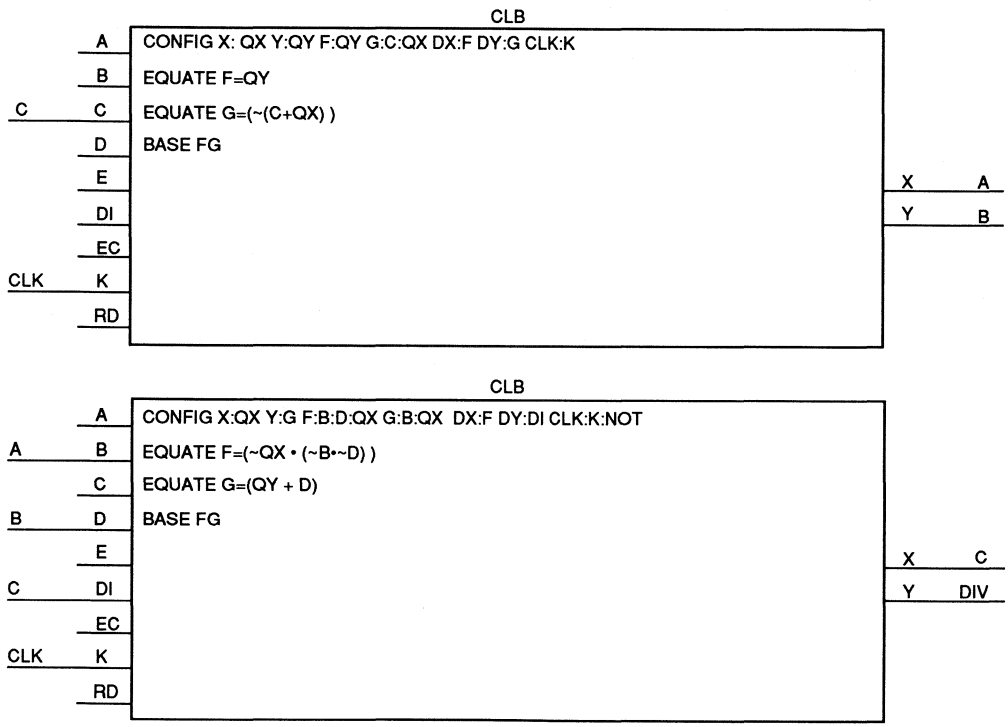
The C5SQUARE macro is a synchronous divide-by-3 counter with a 50% duty cycle on the DIV output when driven by a 50% duty cycle clock. (A ±2 ns error exists.)

Best Speed

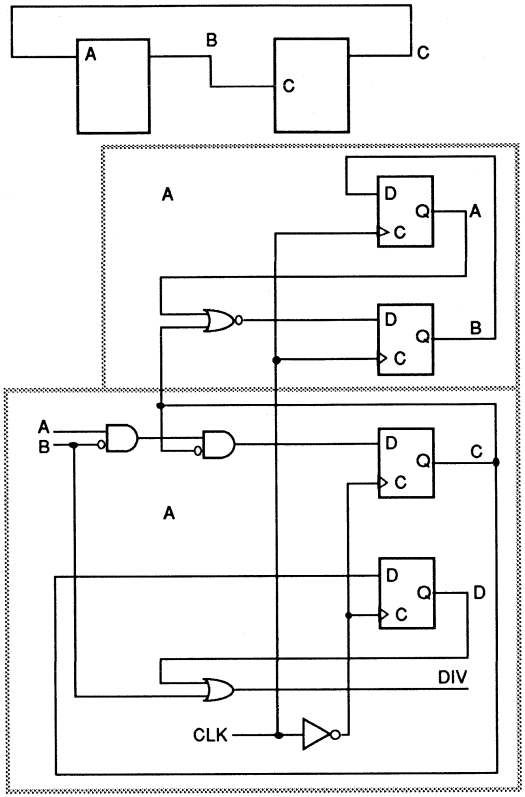
70 MHz FPGA: 27 MHz
100 MHz FPGA: 33 MHz

Place and Route Suggestions

The CLB placement is shown on the next page.



C5SQUARE Divide-by-5 with 50% Duty Cycle (continued)



C8UDLD 8-bit Loadable Up/Down Counter

Number of CLBs: 9

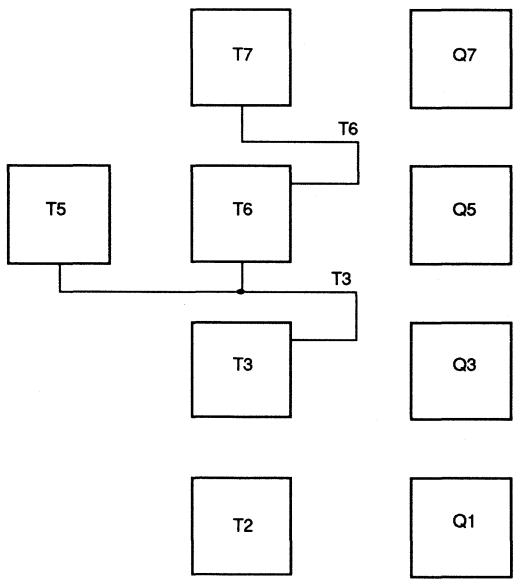
The C8UDLD macro is an 8-bit synchronously loadable up/down counter with a count enable (CE) input but without a terminal count output. When the parallel enable ($\overline{PE}$) output is low, data on the D0. . .D7 input is loaded into the Q0. . .Q7 outputs. When PE and CE are high, the counter content is incremented if the direction control (UP) is high and decremented when UP is low. For longer up/down counters, use the C16UDLD macro.

Best Speed

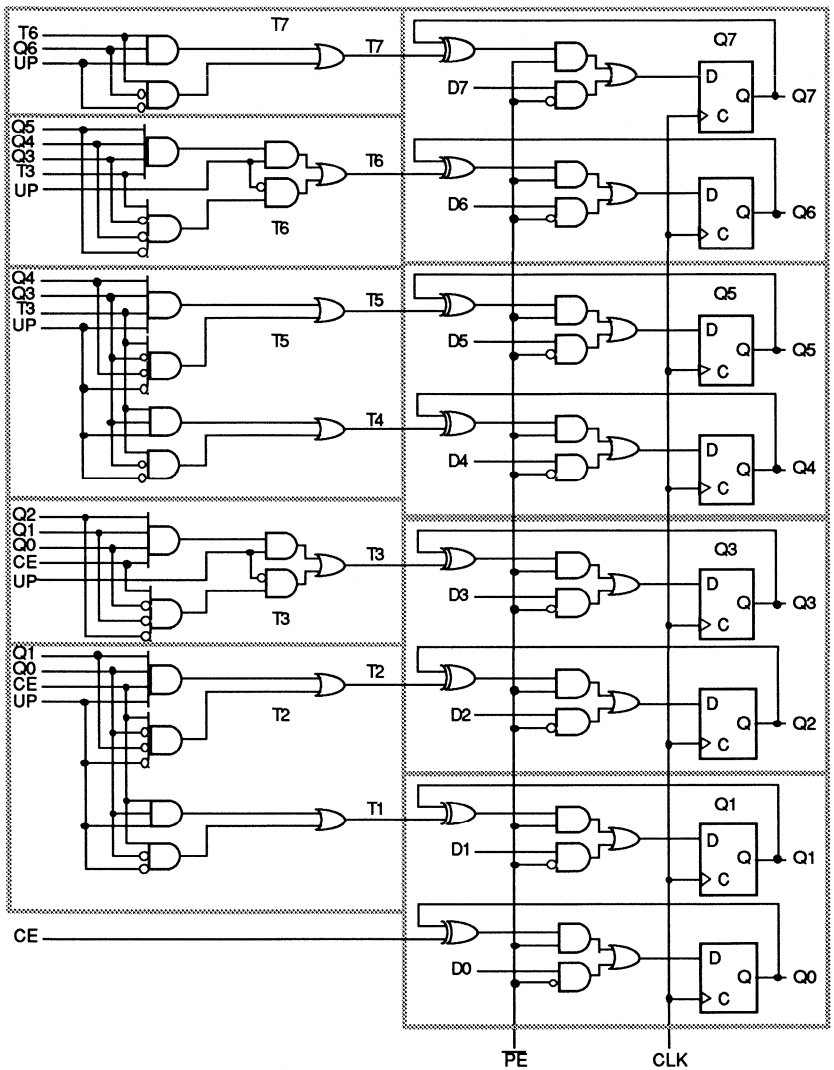
70 MHz FPGA: 11 MHz
100 MHz FPGA: 15 MHz

Place and Route Suggestions

The CLB placement is shown below.



C8UDLD 8-bit Loadable Up/Down Counter (continued)



C16DNLD 16-bit Loadable Binary Down Counter

Number of CLBs: 19

The C16DNLD macro is a 16-bit count-down variation of the popular 74161 synchronous loadable binary up counter. When the parallel enable ($\overline{PE}$) is low, the rising clock edge loads parallel data into the Q outputs. When $\overline{PE}$ and CE (clock enable) are high, the rising clock edge decrements the counter. TC (terminal count) is high when the counter is in state 0 (all Qs low) and the CET input is high.

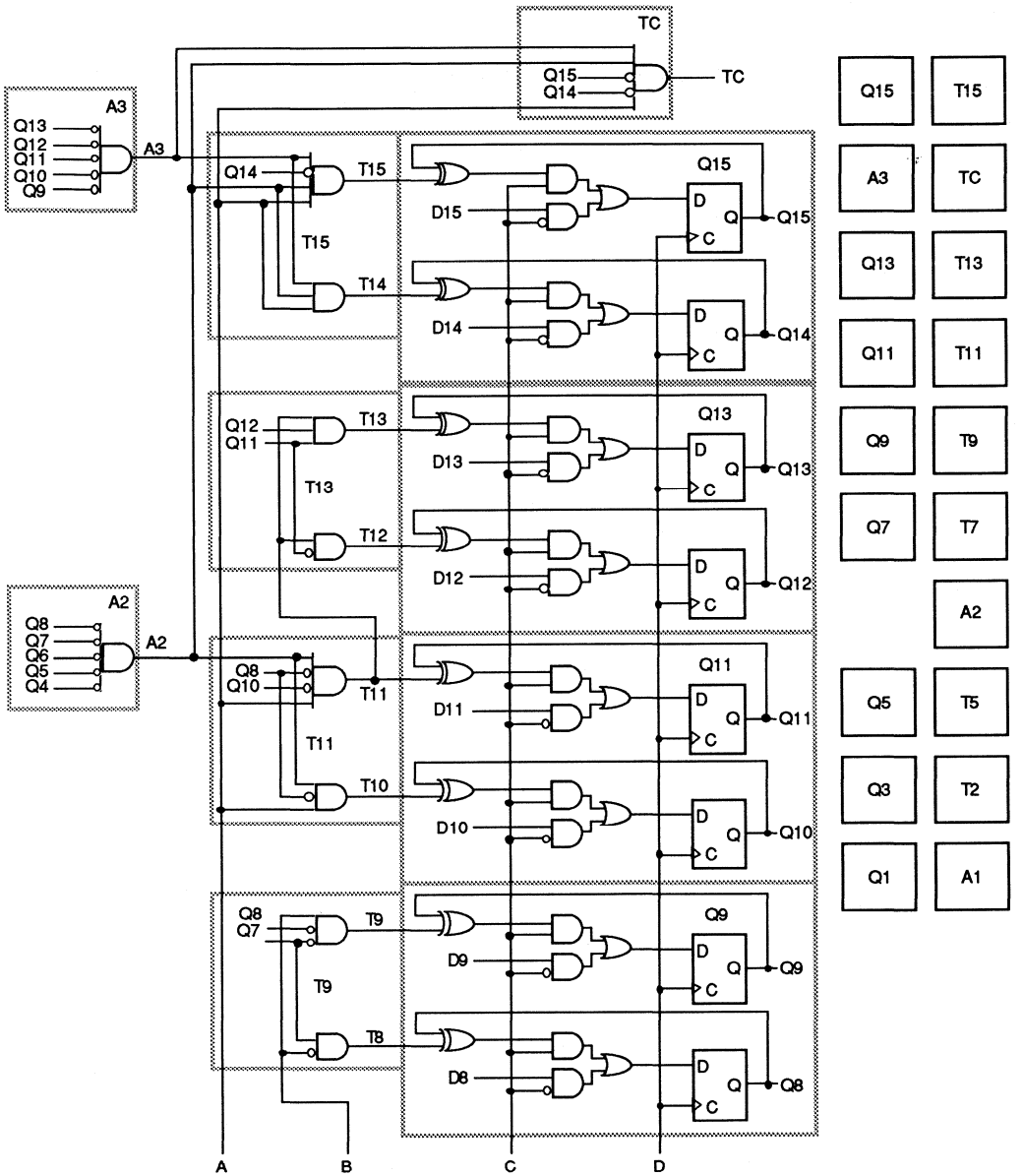
Best Speed

70 MHz FPGA: 17 MHz
100 MHz FPGA: 20 MHz

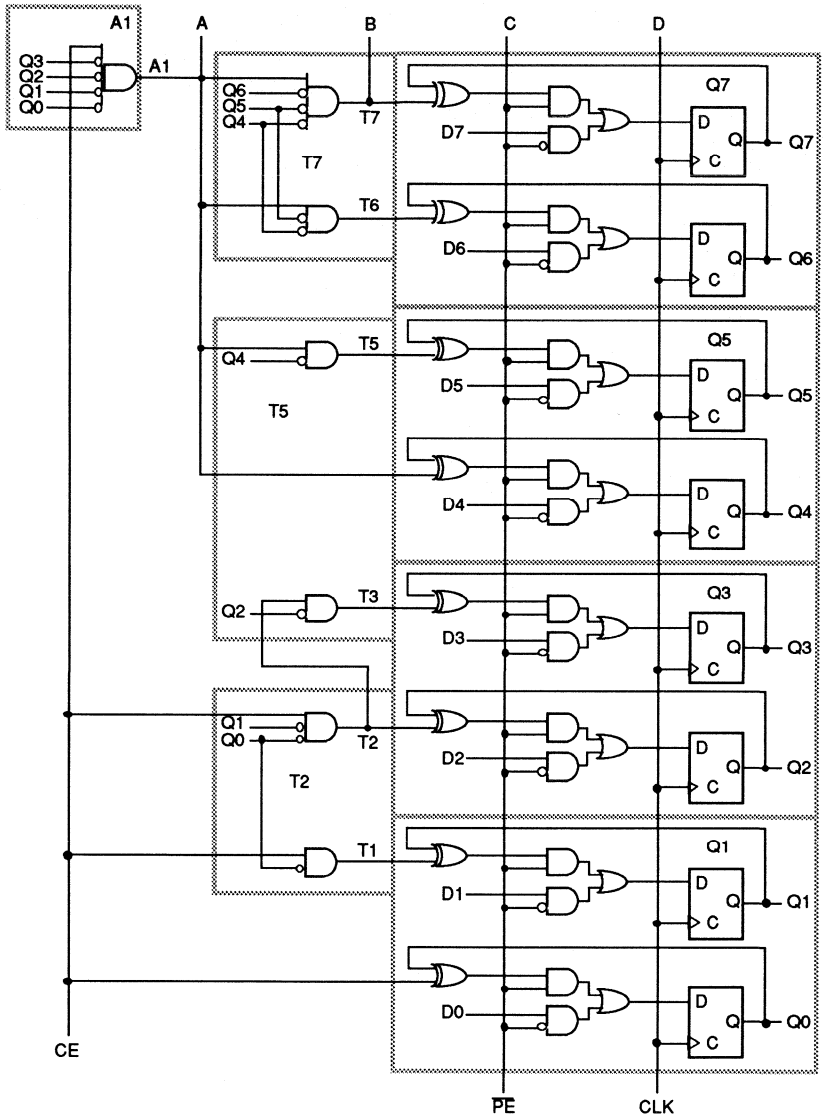
Place and Route Suggestions

The 70 and 100 MHz FPGAs have clock-to-TC delays of 36 ns and 31 ns, respectively. The C16DNLD counter can be cascaded to 32 bits by connecting the TC output of the lower 16-bit counter to the CE input of the higher 16-bit counter. The CLB placement is shown on the next page.

C16DNLD 16-bit Loadable Binary Down Counter (continued)



C16DNLD 16-bit Loadable Binary Down Counter (continued)



C16UDLD 16-bit Loadable Up/Down Counter

Number of CLBs: 18

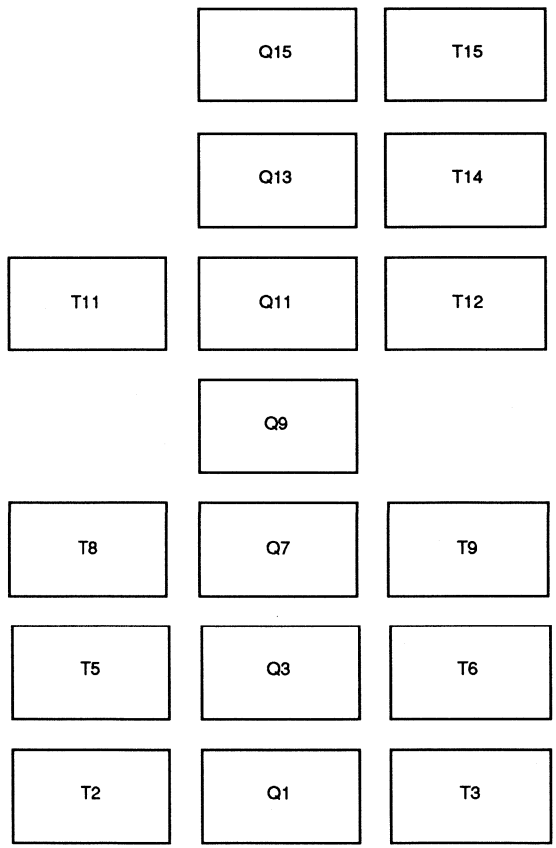
The C16UDLD macro is a 16-bit synchronously loadable up/down counter with a count enable (CE) input but without a terminal count output. When the parallel enable (PE) output is low, data on the D0. . .D17 input is loaded into the Q0. . .Q15 outputs. When PE and CE are high, the counter content is incremented when the direction control (UP) is high and decremented when UP is low.

Best Speed

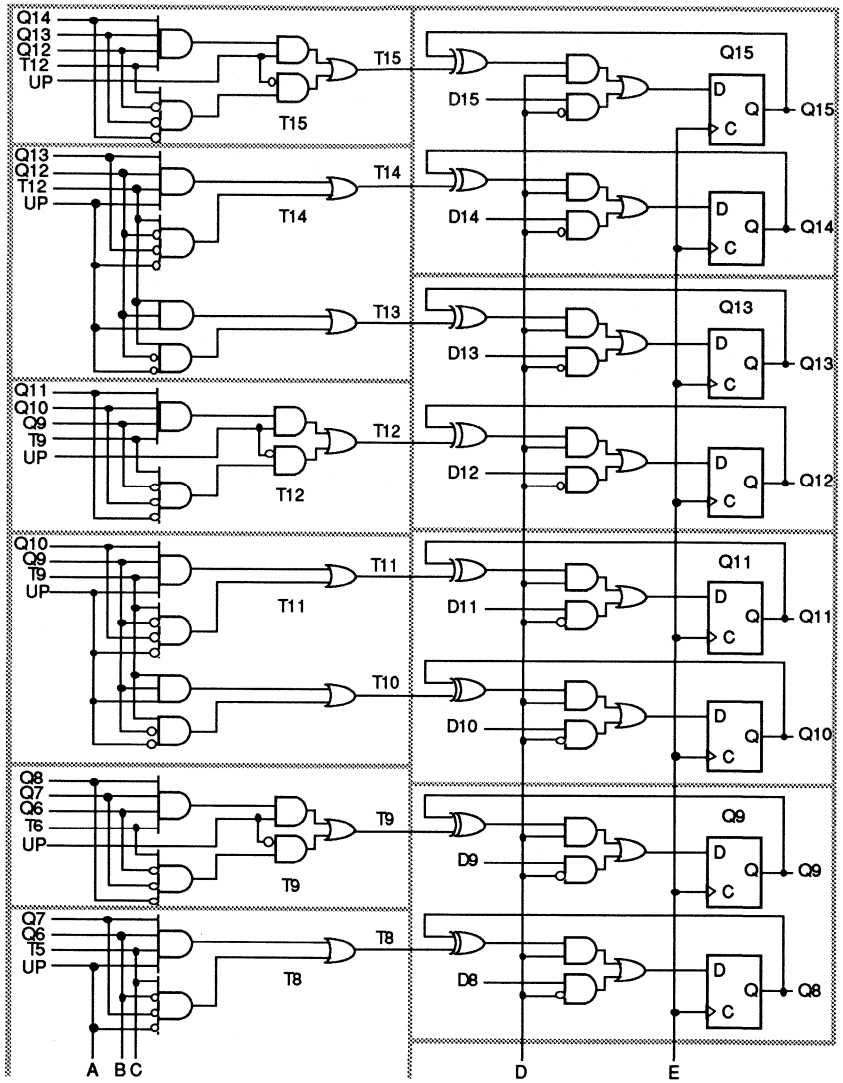
70 MHz FPGA: 18 MHz
100 MHz FPGA: 22 MHz

Place and Route Suggestions

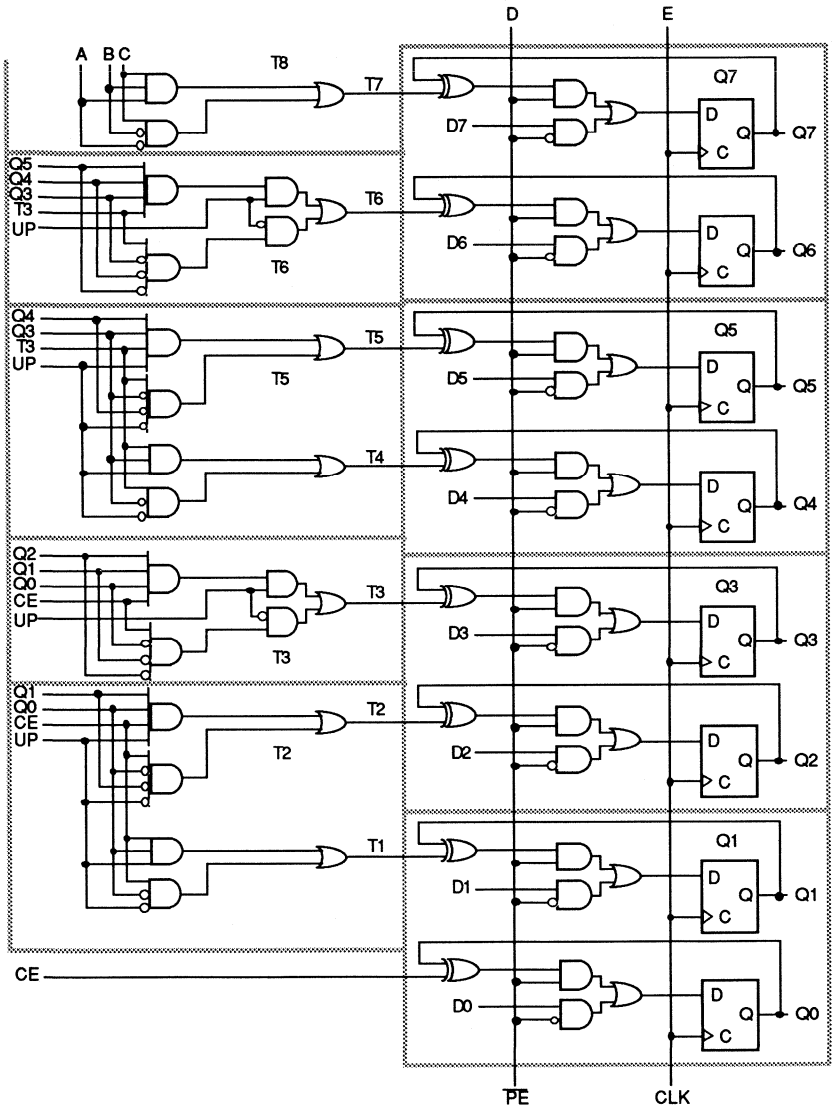
The CLB placement is shown below.



C16UDLD 16-bit Loadable Up/Down Counter (continued)



C16UDLD 16-bit Loadable Up/Down Counter (continued)



C16UPLD 16-bit Loadable Binary Up Counter

Number of CLBs: 19

The C16UPLD macro is a 16-bit variation of the popular 74161 synchronous loadable binary up counter. When parallel enable ($\overline{PE}$) is low, the rising clock edge loads parallel data into the Q outputs. When $\overline{PE}$ and CE (clock enable) are high, the rising clock edge increments the counter. TC (terminal count) is high when the counter is in state F (all Qs high) and the CET input is high.

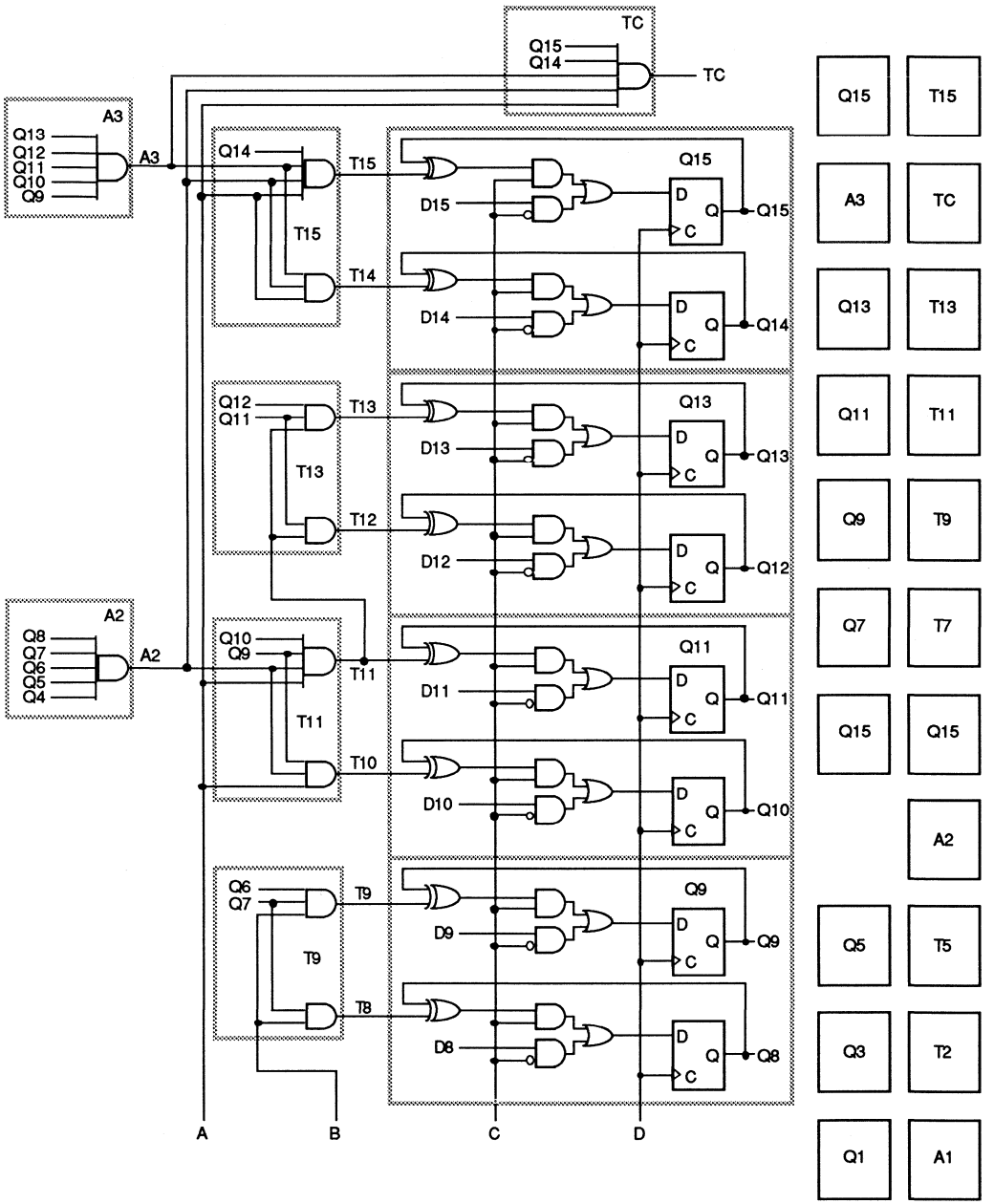
Best Speed

70 MHz FPGA: 17 MHz
100 MHz FPGA: 20 MHz

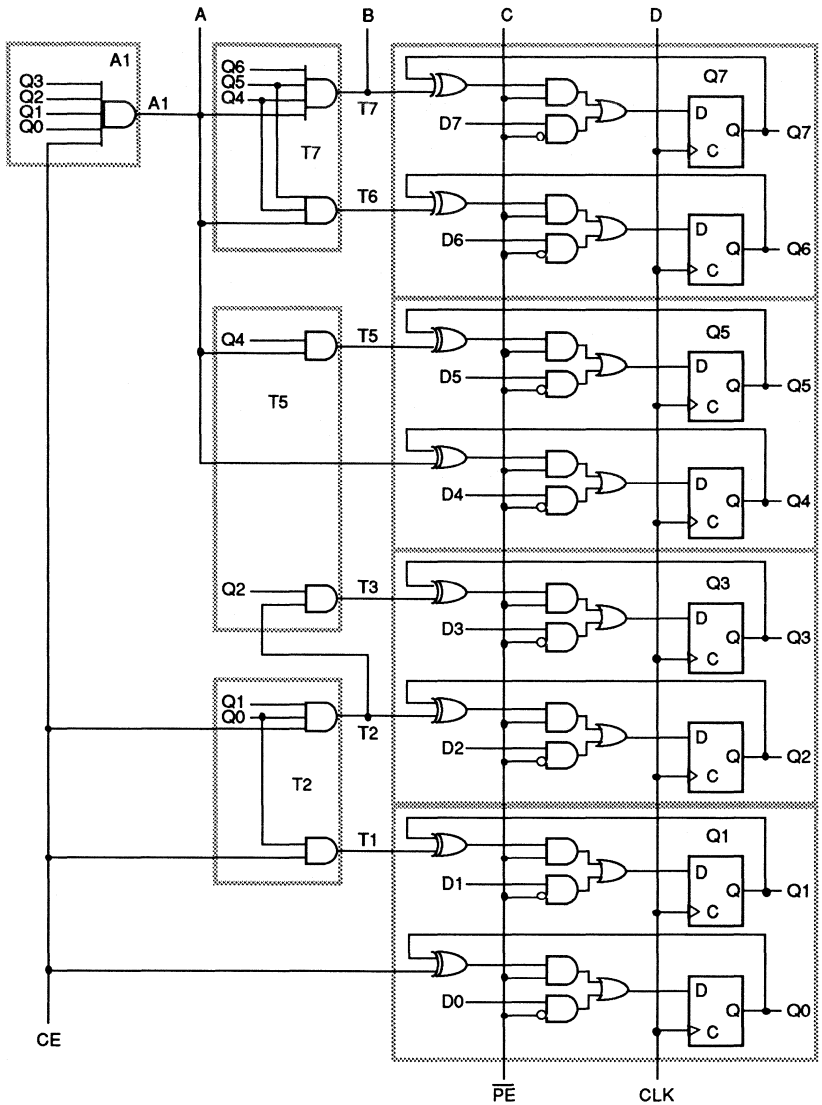
Place and Route Suggestions

The 70 MHz and 100 MHz FPGAs have clock-to-TC delays of 36 ns and 31 ns, respectively. The C16UPLD counter can be cascaded to 32 bits by connecting the TC output of the lower 16-bit counter to the CE input of the higher 16-bit counter. The CLB placement is shown on the next page.

C16UPLD 16-bit Loadable Binary Up Counter (continued)



C16UPLD 16-bit Loadable Binary Up Counter (continued)



M4-1C 4-input Multiplexer in One CLB

Number of CLBs: 1

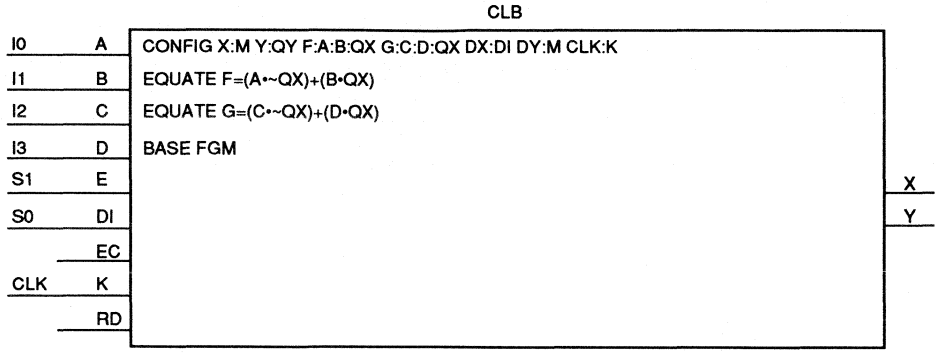
The M4-1C macro is a 4-input multiplexer in one CLB using the DIN pin as a pipelined select input (S0). A rising clock edge is needed to register the S0 input. The select inputs, S1 and S0, determine which of the inputs (I0, I1, I2, or I3) drive the outputs O and Q (direct and registered outputs, respectively). In the 70 MHz and the 100 MHz FPGAs, the M4-1C has I-to-O propagation delays of 8 ns and 7 ns, respectively.

Because the S0 input uses the DIN pin, it is necessary to represent this macro with a CLB primitive show instead of the schematic.

Inputs						Outputs	
S1	S0	I0	I1	I2	I3	O	Q*
L	L	L	X	X	X	L	L
L	L	H	X	X	X	H	H
L	H	X	L	X	X	L	L
L	H	X	H	X	X	H	H
H	L	X	X	L	X	L	L
H	L	X	X	H	X	H	H
H	H	X	X	X	L	L	L
H	H	X	X	X	H	H	H

* Q requires two clock pulses: one to register in S0 and another to register O.

M4-1C 4-Input Multiplexer in One CLB (continued)



MCOMP Magnitude Comparator, Small and Fast

A magnitude comparator is more complex than an identity comparator but simpler than an adder or subtractor. A magnitude comparator tells not only when two operands are equal, but also which one is greater when they are unequal. The 70 MHz and 100 MHz FPGAs have propagation delays of 8 ns and 7 ns, respectively.

Truth Table

The truth table is represented by the following equations:

A > B:

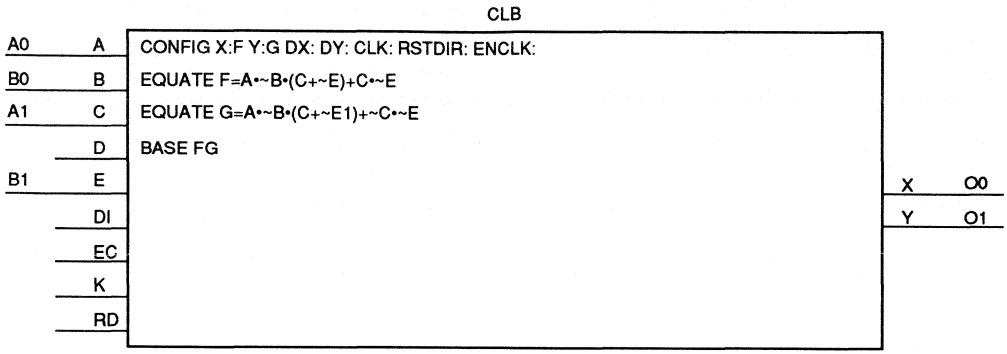
= A0* ~ B0 (A1 = ~ B1) + A1* ~ B1

A < B:

+ ~A0*B0* (~A1 + B1) + A1*B1

B1	A1	B0	A0	A > B	A < B
0	0	0	0	0	0
0	0	0	1	1	0
0	0	1	0	0	1
0	0	1	1	0	0
0	1	0	0	1	0
0	1	0	1	1	0
0	1	1	0	1	0
0	1	1	1	1	0
1	0	0	0	0	1
1	0	0	1	0	1
1	0	1	0	0	1
1	0	1	1	0	1
1	1	0	0	0	0
1	1	0	1	1	0
1	1	1	0	0	1
1	1	1	1	0	0

MCOMP Magnitude Comparator, Small and Fast (continued)



PHFRCOMP Phase/Frequency Comparator

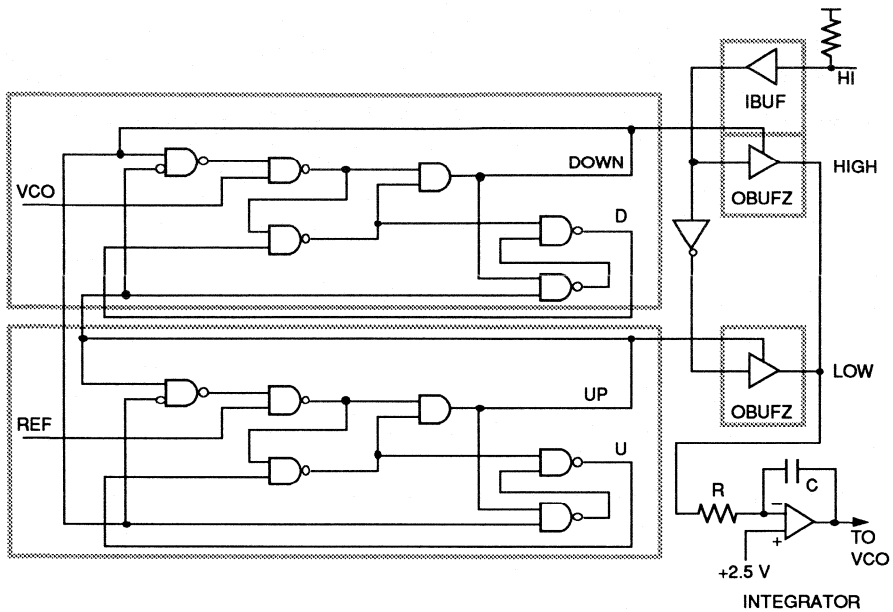
Number of CLBs: 2, plus 3 IOBs

The PHFRCOMP is used as part of a phase-locked loop (PLL) system. A voltage-controlled oscillator (VCO) frequency is compared against a reference (REF) frequency. Whenever the two frequencies or their phase differ, the comparator generates either a pump-down or a pump-up signal that can drive an integrator. The integrator, in turn, drives the VCO so that VCO and REF are in phase.

To use this macro, high and low should be connected to separate output pads and then wired together externally to an integrator, as shown in the schematic on the following page. In addition, since HI input is tied to a pull-up resistor, it requires assertion by an external source. HI could be connected to an unbonded pad (UPAD).

The PHFRCOMP has been breadboarded at 10 MHz with a phase error of less than 2 ns. It may be faster, however.

PHFRCOMP Phase/Frequency Comparator (continued)



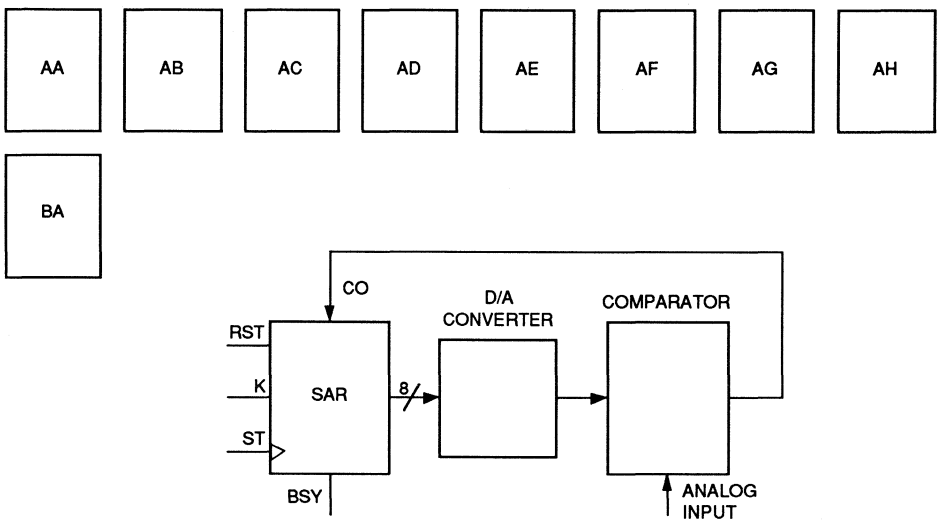
SAR Successive Approximation Register

Number of CLBs: 1/bit plus One CLB for the Controller

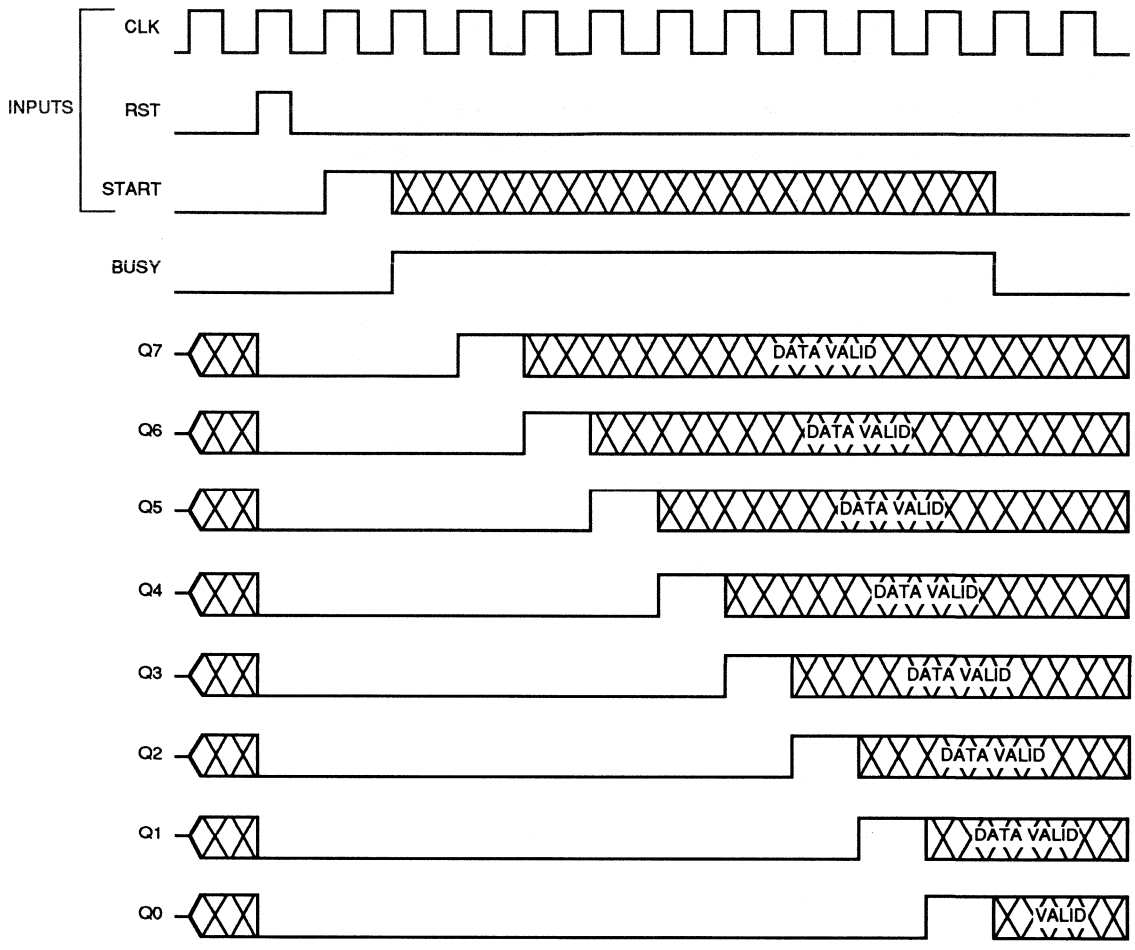
The successive approximation register, along with an external digital-to-analog converter and an analog comparator, can be used to implement a fast analog-to-digital converter. The SAR runs at 25 MHz in a 70 MHz part and at 38 MHz in a 100 MHz part. An n-bit conversion process takes n + 2 clock cycles.

A rising edge on START begins the process, and BUSY stays high until the conversion is complete. The conversion begins when START is high and BUSY is low. To retain data, START must be low before the conversion ends. Otherwise, when the next conversion begins, it overwrites the current data.

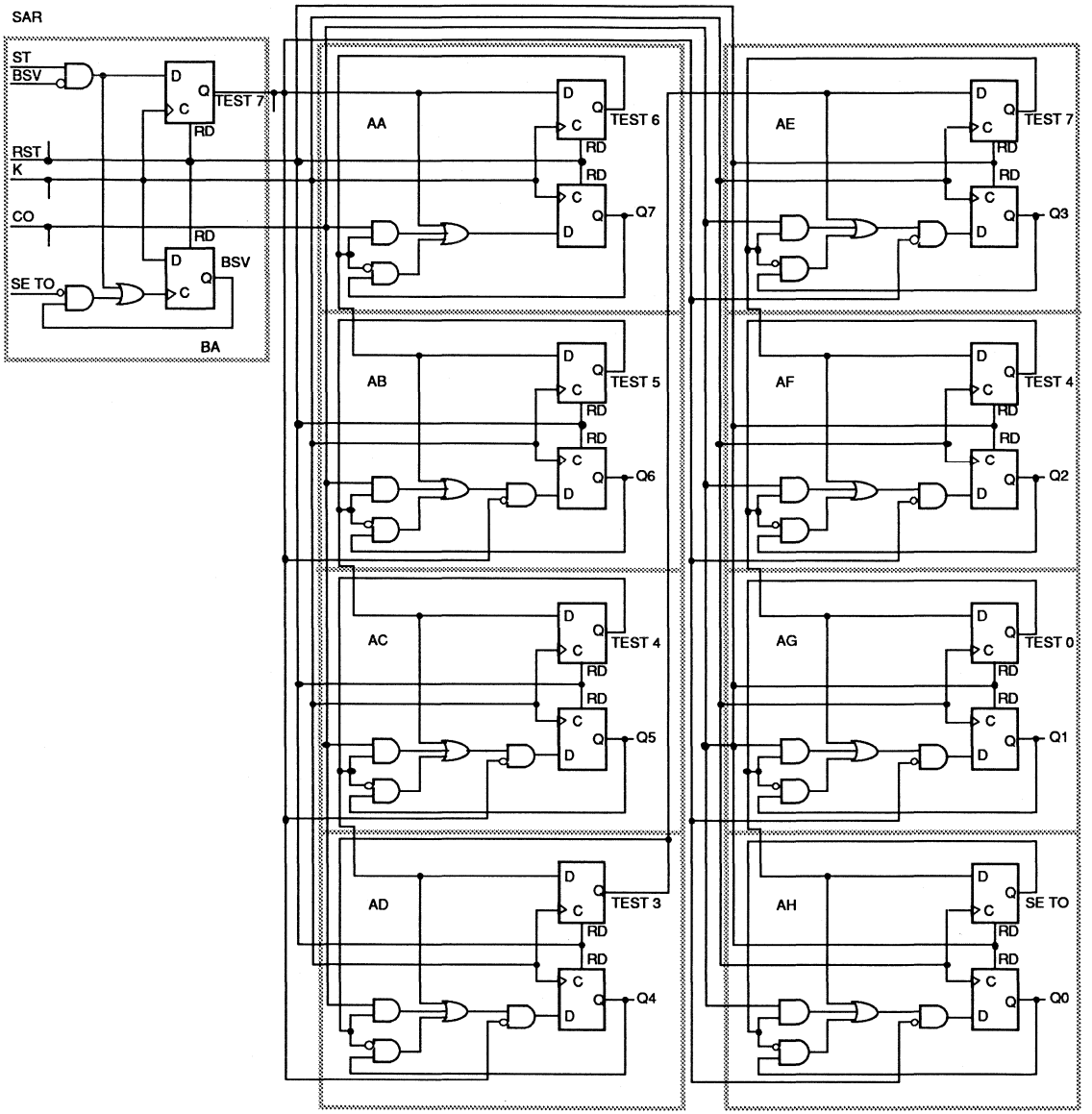
The CLB placement is shown below.



SAR Successive Approximation Register (continued)

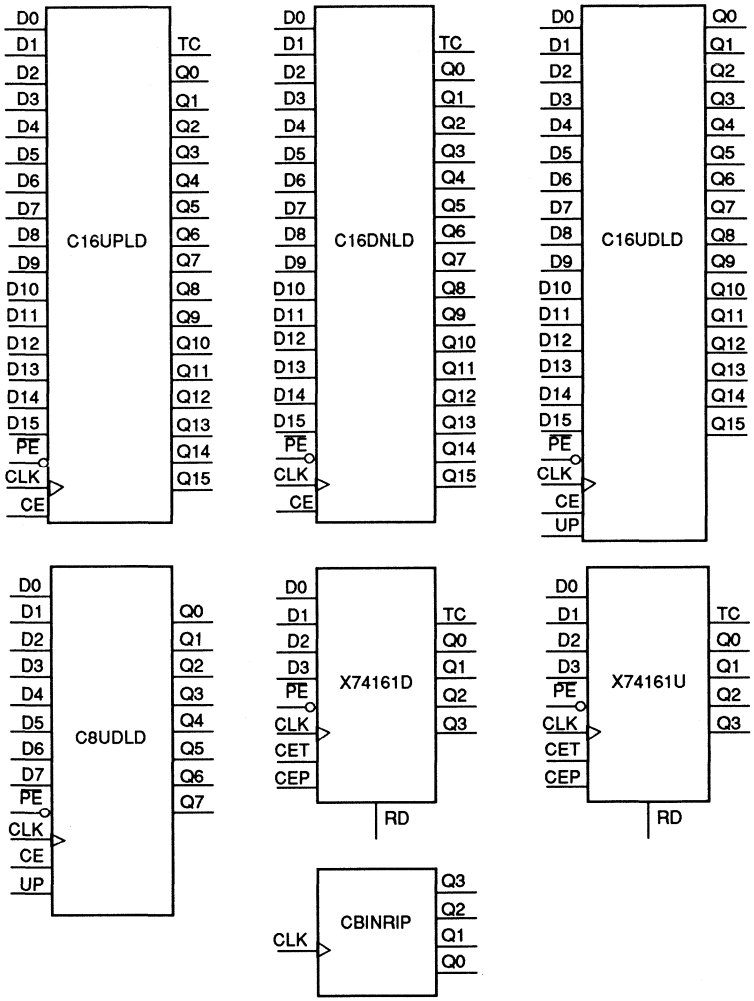


SAR Successive Approximation Register (continued)

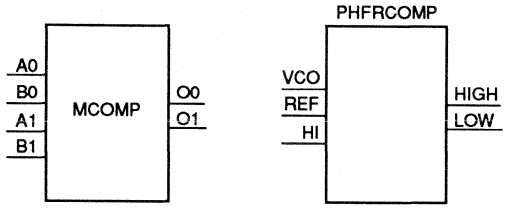


Custom Macro Symbols

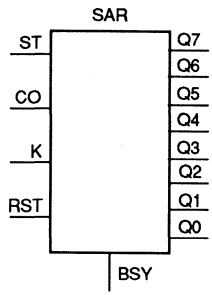
Binary Counters



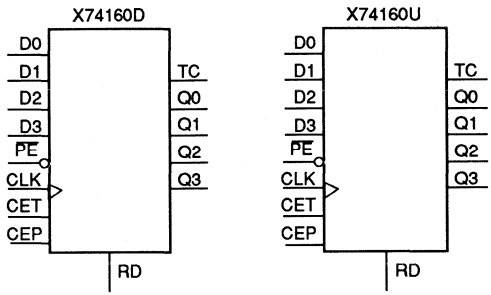
Comparator

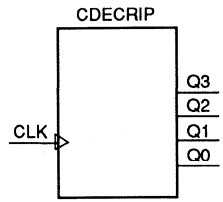


Data Register

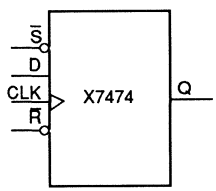


Decade Counters

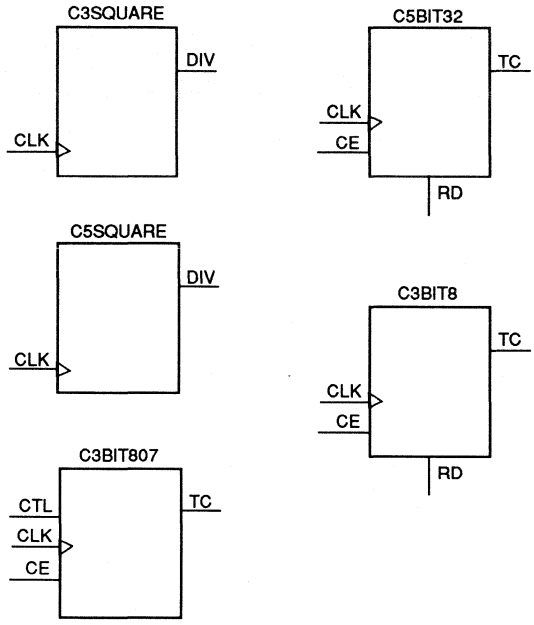




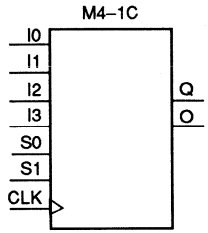
Flip-Flops



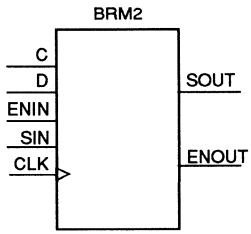
Divide by N

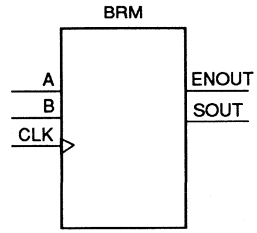

C3SQUARE
DIV
CLK
C5BIT32
TC
CLK
CE
RD
C5SQUARE
DIV
CLK
C3BIT8
TC
CLK
CE
RD
C3BIT807
CTL
CLK
CE
TC

Multiplexer

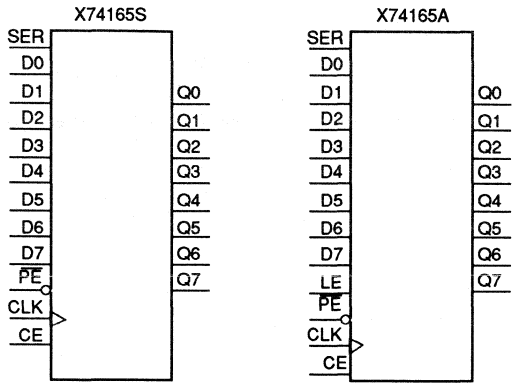


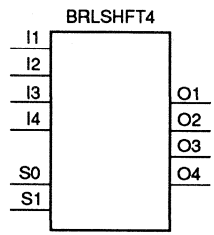
Rate Multiplexer





Shift Register





Section 5.
Applications

5. Applications

Table of Contents

Contents **Page**

Introduction

The following pages show examples of systems and subsystems solutions using AT&T field-programmable gate arrays. Some of these designs have been implemented and a few are in production, but most are conceptual designs. They are intended to demonstrate the device capabilities, to highlight special advantages, to emphasize the best design methods, and, in general, to stimulate the designer's imagination.

A field-programmable gate array (FPGA) can implement virtually any digital design. AT&T 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 FPGA design while paying very little attention to the architectural details of the logic cell array.

Such an approach, however, will not always achieve the highest possible performance and the lowest possible cost. For specific, well-structured designs, it may pay to work out a good match with the FPGA architecture. This chapter gives several examples of such solutions. The ATT3000 familly of FPGA devices have inherent features different from those of LSTTL MSI circuits, *PAL* devices, or conventional gate arrays. These four technologies all have different structures, which lead to different strengths and weaknesses when they are being used to implement any specific type of logic.

TTL-MSI was originally defined to fit into a 16-pin package and to provide maximum flexibility so that each standard part could be used in a myriad of applications. Some functions are, therefore, overdesigned (counters and shift registers have parallel inputs and outputs, when few applications need both), and some are crippled by the 16-pin limitation (notably the up-down counters).

PAL devices suffer from the rigidity of the AND-OR architecture and from the fixed assignment of flip-flops to output pins. While the number of inputs is generous, ideal for wide decoding, the limited number of product terms that can be ORed together makes many designs inefficient and slow. The number of flip-flops available in *PAL*s is very limited.

Gate Arrays

Gate arrays offer maximum flexibility and a high level of integration, but burden the user with high risk, high cost, and a long delay from finished design to working prototype. Generating test vectors and worrying about testability is another price the gate array user has to pay.

Field-Programmable Gate Arrays

Field-programmable gate arrays offer a very large number of flip-flops (128 in the CLBs and another 128 in the IOBs of the ATT3020, a total of 928 in the ATT3090). Unlike the situation with gate arrays, these FPGA flip-flops cannot be, or need not be, traded off against logic. Combinatorial logic coexists with the flip-flops in the form of function generators. The function generators are surprisingly versatile pieces of logic, unlimited in their flexibility, limited only by their fan-in of four or five signals.

When the logic has five inputs or less and is interspersed with flip-flops driven by a common clock, the FPGA devices are extremely efficient. Certain high fan-in functions like ALUs tend to be less efficient, and bus-oriented designs must be routed carefully to utilize the long lines of the 3-state drivers of the ATT3000 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 FPGA design, and some want to know the definition of equivalent gates. While the data sheets provide worst-case guaranteed parameters, many designers need additional information about input and output characteristics, power consumption, crystal oscillator design, and the exact interpretation or certain ac parameters. CLB flip-flops show excellent recovery from metastable problems, an important concern with asynchronous interfaces.

Combinatorial Functions

The five-input function generator of the ATT3000 family CLBs offers unlimited flexibility to implement any one of the more than four billion (2^{32}) possible functions of up to five variables in one CLB, all with the same combinatorial delay. The logic designer should take advantage of this flexibility while avoiding the possible speed penalty imposed by the limitation to only five 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 five-input majority function would use 29 gates when implemented with two-input NANDs and inverters, but it fits into the combinatorial portion of one ATT3000-series CLB.

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

ALUs consume many *FPGA* device resources, but adders or subtractors can be implemented quite efficiently, even using carry-look-ahead for functions that exceed a width of 8 bits.

Sequential Functions

FPGAs offer an abundance of flip-flops, up to 928 in the ATT3090. Each CLB flip-flop 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 half of an ATT3020, with 100% utilization of the CLB flip-flops.

Using the fast flip-flops and distributed logic in the FPGA to the best advantages, a synchronous presettable counter of arbitrary length has been demonstrated to run at 40 MHz. This is much faster than any available popular microprocessor peripheral counter/timer.

State-machine design is another example in which the creative use of CLB resources can result in a straightforward and easily understood solution.

As explained in the beginning of this chapter, the CLB flip-flops are metastable-resistant; they resolve metastable situations typically within a few nanoseconds. Designers are nevertheless encouraged to avoid asynchronous designs whenever possible. The combination of very fast CLB flip-flops with relatively slow and layout-dependent interconnects can lead to internal decoding spikes and glitches that cannot be observed with an oscilloscope. However, they can play havoc with internal asynchronous logic. The high-speed, low-skew global clock lines and the individual clock enable inputs on each CLB favor synchronous design approaches that are inherently safer and more predictable.

System Descriptions

FPGAs are universal programming building blocks that are used in a wide variety of systems. A *PS/2* micro channel controller and a DRAM controller/error corrector demonstrate the versatility of the FPGA in speed-critical applications. Some of the designs are available, as indicated, and can be obtained by calling the AT&T Microelectronics hotline.

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

Application Note
June 1992

AT&T Microelectronics

FPGA Migration

Introduction

Migrating FPGA designs to either gate arrays or hard-wired FPGAs often creates timing problems that can easily elude ASIC logic designers. Timing errors relating to routing delays are especially prevalent in asynchronous designs based on complex, high gate count devices that make routing particularly difficult. They often trigger long delays on some paths and short ones on others, leading to race conditions and glitches in the design when migrated, even when a prototype in the board operates.

Functional Description

FPGA to Gate Array Timing

Some gate array manufacturers now claim that they will convert an FPGA to their gate array automatically, without user-supplied test vectors. Although this may work in a few circuits that are completely synchronous, the chances for first-time success are significantly increased by timing simulations done on both the FPGA and the gate array. Some of the timing problems created during the migration are documented in this application note.

Figure 1 shows an example of one of the many different race conditions that can occur. In the FPGA, gated clock's AND gate has an input with 50 ns of routing delay; the other, 10 ns. The intervals before and after the routing delay are represented by A and A' respectively, with the 50 ns delay between the two. Likewise, B and B' bracket the 10 ns delay for B routing to the AND gate.

Although A's routing delay is that long, a glitch doesn't occur at the output and the chip is not clocked. The reason is that one or the other of C's inputs is always low.

However, in a gate array or hard-wired FPGA, those same delays could both become 5 ns. In this instance, assume both A and B inputs come from a flip-flop. Instead of a 50 ns delay at A' and 10 ns at B', the ASIC logic designer winds up with both A and B at high states since both are 5 ns delays. As a result, a glitch occurs on the output of the AND gate, possibly causing an unwanted clock pulse.

Other routing delay problems relating to FPGA migration may appear alike to some ASIC logic designers. But each problem has its own unique variations and characteristics that still adversely affect a gate array conversion.

Figures 2, 3, and 4, in which D represents major FPGA routing delays, show that the FPGA routing delays are considerably larger than those of gate arrays and can easily be larger than gate delays in data paths.

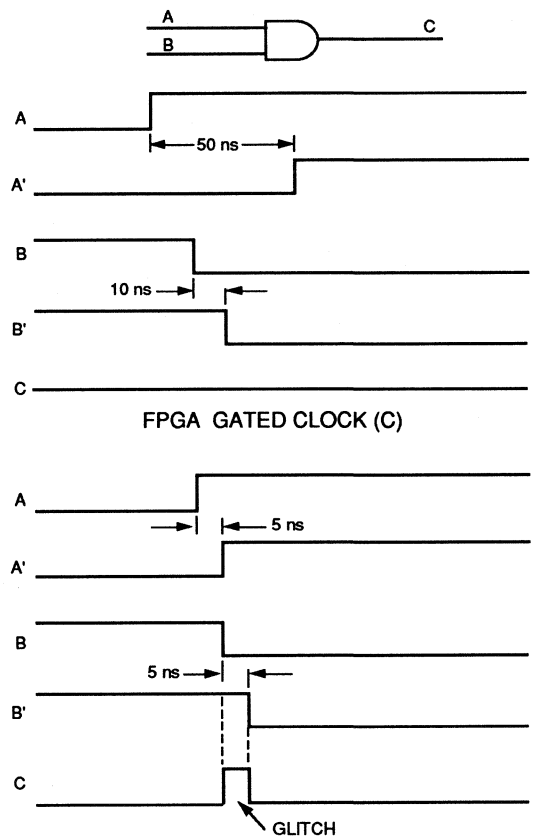
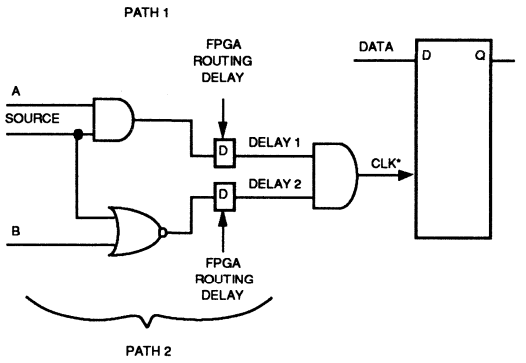


Figure 1. Race Conditions

Functional Description (continued)

Conversion Confusion

Some confusion exists today over synchronous and asynchronous circuits. For instance, some designers might regard the circuit in Figure 2 as a single clock, which it is. However, it is a single clock that can take two different paths to get to the clock on the flip-flop.



* CLK = (A x source) x ($\overline{B + source}$). If A = 1 and B = 0, CLK can be affected by both Path 1 and Path 2.

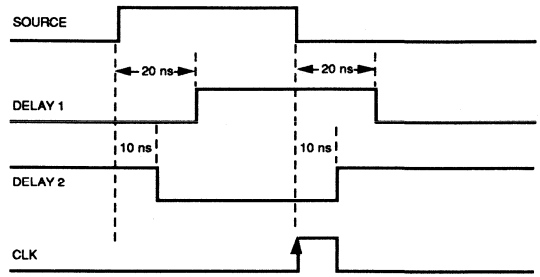
Figure 2. Asynchronous Circuit

While many designers would not consider this clock to be asynchronous, it is: not because the circuit has two different clocks, but because the clock can have two different paths to get to the flip-flop. Again, due to the large routing delays in FPGAs, this circuit may not act the same when converted to a gate array, with smaller routing delays.

The same holds true for the next example shown in Figure 5. Some designers may regard it as synchronous since only one clock source is involved.

As shown in Figure 2, Path 1 and Path 2 go through two different types of FPGA delay cells, each with nets labeled DELAY 1 and DELAY 2, respectively. If Input A is 1 and Input B is 0, the clock can be affected by both Path 1 and Path 2.

In the Figure 3 waveform, the FPGA design is assigned 20 ns for the Path 1 delay and 10 ns for the Path 2 delay. That results in a 10 ns wide pulse on the falling edge of the source for the clock line to the flip-flop.

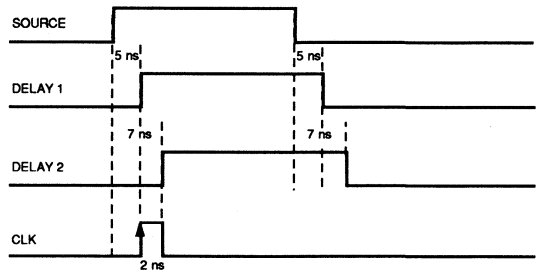


Note: If A = 1, B = 0, DELAY 1 = 20 ns, and DELAY 2 = 10 ns, there will be a 10 ns clock pulse on the falling edge of SOURCE.

Figure 3. FPGA Waveform

However, in the gate array implementation shown in Figure 4, the Path 1 delay becomes 5 ns and Path 2 is 7 ns. In effect, the two can skew in opposite directions.

The gate array thus experiences a 2 ns wide pulse on the rising edge of the source, possibly clocking the flip-flop. Since the flip-flop will be clocked on the opposite edge of the source, the appropriate data may not be at the data pin, and therefore will not be latched. For example, a 1 would be latched in the FPGA in Figure 3, and a 0 would be latched by the gate array in Figure 4.



Note: DELAY 1 = 5 ns, DELAY 2 = 7 ns. The gate array now has an unwanted glitch on the rising edge of SOURCE.

Figure 4. Gate Array Implementations

Functional Description (continued)

Figure 5 shows a similar problem. However, here the example is a clock with sequential reconvergent sources. The difference is that Input A is clocked into flip-flop 1 by SOURCE, with Output B from the flip-flop going through FPGA routing DELAY 1. Routing DELAY 2 comes from the source clock and goes to the input of the OR gates. These two could conceivably bear major delay differences.

Since flip-flop 1 is clocked with the same clock (SOURCE) going to the OR gate, one assumes that the clock gets to the OR gate before Output B of the flip-flop. That's not necessarily the case with an FPGA.

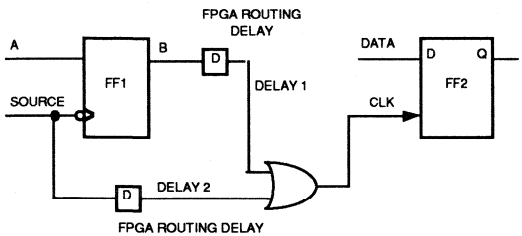


Figure 5. Clock with Reconvergent Sources

Thus, in Figure 6, assume that A = 1, B = 0, the delay of B to the OR gate is 5 ns, and the delay of SOURCE to the OR gate, after it has reached flip-flop 1, is 10 ns.

Since DELAY 1 and DELAY 2 are OR'd together, when the source clock falls one is always high at any time. Thus, there is no glitch on the falling edge of the source clock. It's another matter with gate arrays since timing is different.

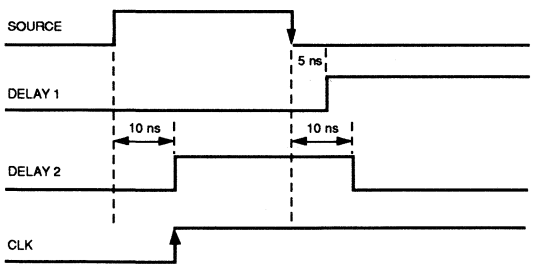


Figure 6. Source Clock without Glitch

Figure 7 shows DELAY 1 as 5 ns and DELAY 2 as 2 ns. DELAY 1 is larger than DELAY 2; therefore, the clock going to flip-flop 2 will glitch on the falling edge of the source clock.

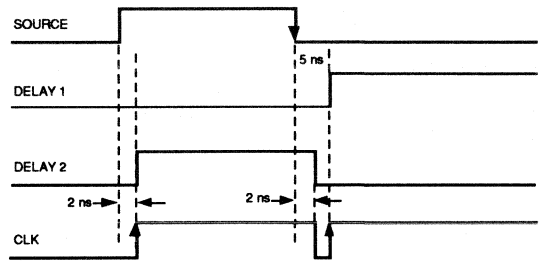


Figure 7. Source Clock with Glitch

Clock/Data Races

Figure 8 shows an asynchronous circuit involving flip-flops having different clocks and three different delays. DELAY 1 extends from clock 1 to flip-flop 1, and DELAY 2 extends from clock 2 to flip-flop 2. DELAY 3 comprises the total delay of the logic and the delay cell, and goes between the flip-flops.

If both clocks are active about the same time, there may be a clock/data race condition at the second flip-flop. This is aggravated when converting an FPGA design to a gate array because there are four delays that could change significantly. They are logic delays between the two flip-flops, routing delays of this logic between the two flip-flops, and the routing delays of both CLK1 and CLK2.

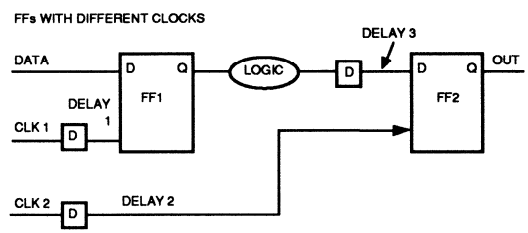


Figure 8. Asynchronous Circuit with Flip-Flops

Functional Description (continued)

For example, in Figures 9 and 10, since CLK2's delay in the gate array is speeded up, data to flip-flop 2 does not get there in time to be clocked by CLK2, and OUT remains a 0 instead of going high, as happens in FPGA.

Also, asynchronous presets and clears pose the same problems as the clocks on flip-flops, as explained in Figures 2—8. If there is an asynchronous preset or clear on a flip-flop, a circuit can experience the same glitches and other problems as those involving flip-flop clocks when converted to a gate array.

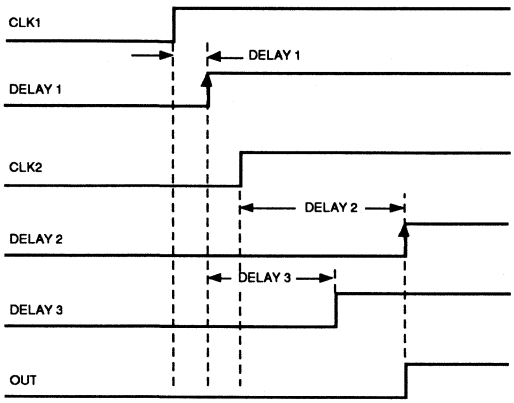


Figure 9. FPGA Design

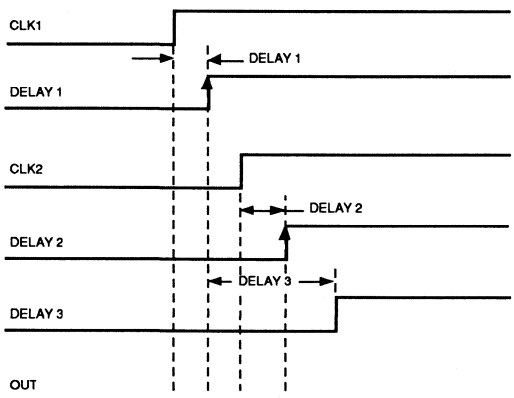


Figure 10. Gate Array Design

Figure 11 shows yet another routing delay problem, this time involving mixed data and clock on the same sequential element. The paths from SOURCE to the clock and data pins of the flip-flop cause a clock/data race condition.

In this example, since it's a positive edge flip-flop, a race condition exists between incoming data and the clock when the clock rises.

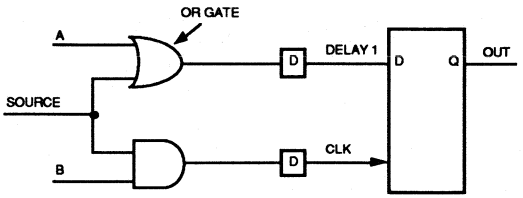


Figure 11. Mixed Data and Clock

For the FPGA implementation, Figure 12 shows that if DELAY 1 is 10 ns, and the delay to CLK is 20 ns, the data arrives in time for the flip-flop to clock it. In Figure 13, the delays change when converting the FPGA to a gate array. DELAY 1 is larger than the clock delay, so when the source signal rises, the flip-flop is clocked before the data gets there.

Lastly, routing delay problems can be related to asynchronous loops used in an FPGA. Generally, they are included in a circuit for their timing characterisitcs. Sometimes asynchronous loops are added as ring oscillators, as timing chains, or as long delay paths to get FPGAs to operate. When converting from an FPGA to a gate array, this circuit can have the same function, but the timing associated with it may be lost.

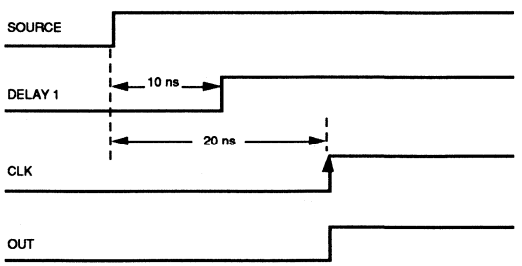


Figure 12. Mixed Data and Clock FPGA Design

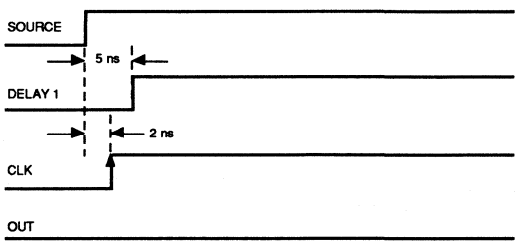


Figure 13. Mixed Data and Clock Gate Array Design

Functional Description (continued)

No Plug and Play

For many FPGA applications, a prototype FPGA is created, placed in the system, then verified by either performing system diagnostics or by allowing normal system operation. If a problem is found, a new FPGA prototype is created, generally in a matter of minutes to hours, and the system is verified. This loop is continued until the FPGA is proven to work correctly. Although this works well for prototyping, if the FPGA is to be used in a production environment, a worst-case timing simulation should be performed.

The FPGA used in the prototype system is generally guaranteed to be faster than the worst-case rated values. However, a set of production parts may be much slower than the particular part used for prototyping, but will still be faster than the worst-case rated values. Some of the production parts may, therefore, fail in the system. Since the user has no way of determining the absolute speed of the given part, a timing simulation needs to be done. When timing simulations have been performed on the FPGA, these same timing simulations can be performed on the resulting gate array. This significantly improves the chances of the gate arrays working the first time. The test vectors from these simulations can then be used for testing the gate arrays.

Application Note
June 1992

AT&T Microelectronics

Estimating Size and Performance

Introduction

Field-programmable gate arrays are available in a range of densities and speed grades. Before committing resources to design implementation, the user should make an estimate to determine which FPGA best fits the specific application. Size and performance estimates cannot be expected to provide exact details, but they provide useful guidelines for device selection and cost estimates. A complete design is always the final test for both density and performance.

Design-fit estimates can be done in two steps. The first is a quick I/O and storage element count, with no regard for performance. The second step counts logic blocks based on details of the intended circuit, and includes gross performance estimates, still without regard for routing delays. Performance estimates should always be considered best-case, recognizing that actual system performance can only be verified on a completed design.

Step 1: I/O and Storage Element Fit

A quick initial estimate of how a system fits a specific logic cell array device can be made by counting the required input and output pins and internal storage elements. Table 1 lists the ATT3000-series FPGA devices and their respective I/O and storage element counts. To estimate a fit, first count the required inputs and outputs and compare the total with the I/O pin count of the desired device. If the desired functions require more I/O than listed for a device, the designer must either select a larger device or package, or reduce the I/O requirement.

Table 1. I/O and Storage Element Summary

Device	Maximum I/O	Logic Block Storage	I/O Block Storage
ATT3020	64	128	128
ATT3030	80	200	160
ATT3042	96	288	192
ATT3064	120	448	240
ATT3090	144	640	288

If the desired FPGA device has enough I/O pins, the next step is to count the required storage elements. Table 1 shows both logic-block storage elements and I/O-block storage elements. Logic-block storage elements should be considered first, since they are the most flexible. If the required number of storage elements is less than the number of logic storage elements, the desired functions can probably be performed in the chosen FPGA device.

In some cases, the I/O-block storage elements can also be used to meet storage-element requirements. In particular, if the number of additional storage elements required beyond the available logic storage elements is less than the number of unused I/O pins, then the desired functions may still fit into the chosen device.

The following two examples illustrate the step one quick-estimation procedure:

Table 2. Eight-bit Microprocessor Peripheral (Example 1)

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

Function	Storage Elements
Control registers (assume 8 bits)	16
Buffered input shift register	16
Miscellaneous control logic	10
Total	42

Step 1: I/O and Storage Element Fit
(continued)

Table 3. Memory Controller for a 32-bit High-Performance Processor (Example 2)

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

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

* Based on this I/O count, the ATT3042 with 96 pins would be marginal. An ATT3042 with up to 120 I/O pins may be required.

With two storage elements per logic block, the ATT3042 can provide up to 288 storage elements. Based on this estimate, the desired functions should fit into the device. Some caution is indicated for two reasons. First, the I/O count is very near the limit of the device. This could cause some routing congestion in the I/O area, making a higher pin-count device a better choice. Second, high performance requires making the best use of device features. The 32-bit bus may impose critical performance requirements. Only the ATT3064 and ATT3090 permit a 32-bit internal bus, based on the number of available long lines. Choosing the ATT3064 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 index listing in the Software Development System (Section 4) of this data book may be used to determine specific CLB counts for each function to be implemented.

The macro list shows the various gates and functions available with each design library. Each entry in the list includes the required number of logic blocks to implement that function. To develop a rough block count, the designer simply tabulates all of the blocks required by each of the functional elements in the design. Figure 1 shows a portion of a schematic and the block count from the macro list.

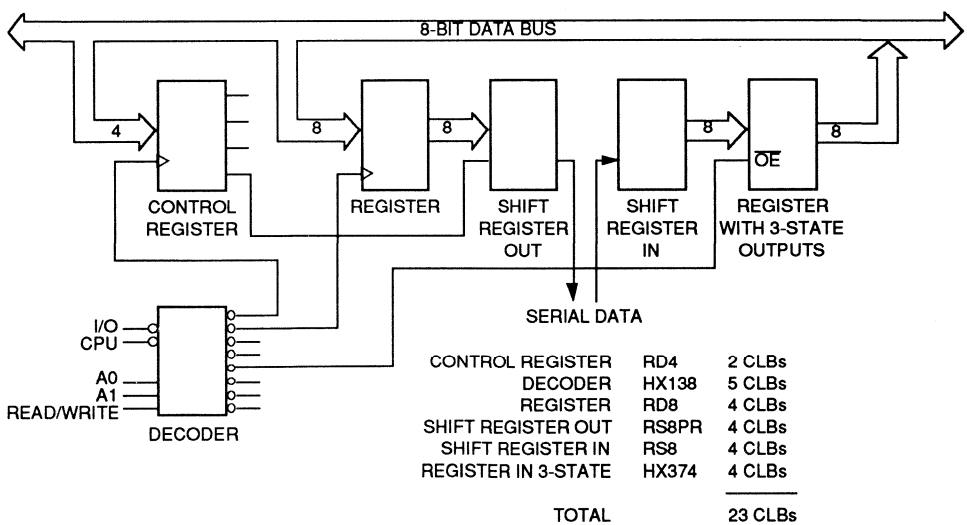


Figure 1. Portion of Schematic with Block Count

Step 2. Logic Block Requirements

(continued)

In many schematics, there are collections of random gates that need to be considered, along with the higher-level functions such as counters, decoders, and multiplexers. The following technique can be used to estimate the logic blocks required for random logic. Begin at an output point and move back along the path collecting gates until the number of inputs is five for ATT3000-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 ATT3000 family. For complex equations using more than five inputs, a conservative estimate is to use three blocks per output pin.

Decisions about the appropriate device can be reviewed as more information is collected. Block count estimates which are near the limit of a device, either in block count or in I/O and storage element count, may suggest use of the next higher-density device.

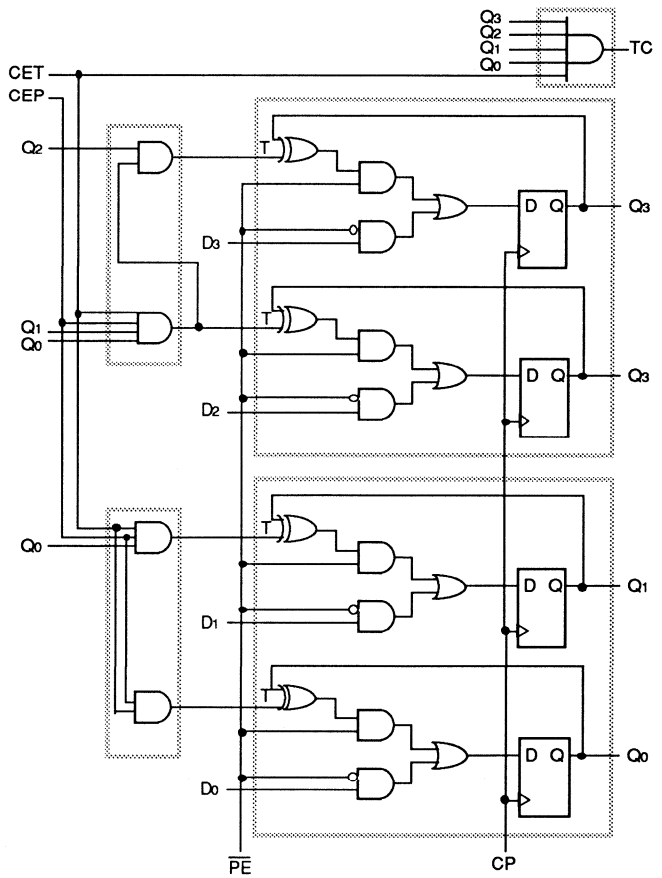


Figure 2. Five CLBs Are Required to Implement a 74161 Binary Counter

Estimating Performance

After selecting the right FPGA device based on logic resources, an estimation of performance is often the next step. If the system clock rate is less than 20% of the flip-flop toggle rate of the selected device, then the performance goals can usually be met easily. In cases of higher system clock rates or very complex functions, a more detailed analysis may be required.

The macro library for each device family includes the number of logic-block levels used for each listed function; the FPGA data sheet specifies the block delay for each level. Some routing delay, which can add 25% to 50%, must be added to the block delay.

As an example, a circuit might have three levels of blocks in the path from one clock edge to another. For a device with 10 ns block delays, this gives 30 ns delay from the first clock to the setup required for the next clock. Allowing 30% for routing (10 ns) and 8 ns for setup gives a total delay of 48 ns. This should permit operation at a system clock rate of up to 20 MHz.

Summary

The final determination of whether a logic device meets the goals for integration and performance can come only after the design has been completed. For field-programmable gate arrays, estimating logic capacity and performance should precede device selection. If the design fits, the *XACT* Development System and the simplicity of in-system design verification ensures cost-effective and rapid design implementation.

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

AT&T Microelectronics

Designing with the ATT3000 Family

Clocking

Global and Alternate Clocks Buffers

There are two high fan-out, low-skew clock resources. The global clock originates from the GCLK buffer in the upper left corner of the chip, and the alternate clock originates from the ACLK buffer in the lower right corner of the chip.

These resources drive nothing but the K pins (clock pins) of every register in the device. They cannot drive logic inputs. In the rare case where this connection is required, tap a signal off of the input to the block 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 fan-out clocks.

I/O Clocks

There are a total of eight different I/O clocks, two per edge on each of the four edges.

I/O storage elements can be configured to be latches or flip-flops. Clocking polarity is programmable per clock line, not per IOB. A clock line that triggers a flip-flop on the rising edge can be an active-low latch enable (latch transparent) and vice versa.

Crystal Oscillator

The crystal oscillator connects to alternate clock buffer, ACLK, not to GCLK.

3-State Buffers

Active-high 3-state is the same as active-low enable.

In other words, a one on the T pin of a TBUF or an OBUFZ 3-states the output, and a zero enables it.

Input/Output Blocks (IOBs)

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

IOB pull-up resistors cannot be used with IOB outputs, only on pins that are inputs exclusively.

Configurable Logic Blocks (CLBs)

CLBs have two flip-flops (not latches). They share a common clock, a common reset, and a common clock-enable signal.

Asynchronous preset can be achieved by the asynchronous reset, by just inverting D and Q of the flip-flops.

Routing Resources

Horizontal Long Lines

The number of horizontal long lines (HLL) per device is double the number of rows of CLBs.

The number of TBUFs that drive each horizontal long line is one higher than the number of columns on the device.

Part Name	Row x Column	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

Routing Resources (continued)

Horizontal Long Lines (continued)

T and I pins of TBUFs have limited interconnect resources.

Never use fewer than four TBUFs per horizontal long line. Using fewer than four TBUFs for multiplexing applications 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

Application Note
June 1992

AT&T Microelectronics

Additional Electrical Parameters

Introduction

The logic cell array 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.

Table 1. Pull-Up Resistor Values

Pull-Up Resistor	Value	Unit
IOB Pull-ups	40 to 150	kΩ
DONE Pull-up	2 to 8	kΩ
Long Line Pull-up (each)	3 to 10	kΩ

Inputs

Hysteresis

All inputs, except PWRDN, and XTL2 when configured as the crystal oscillator input, have limited hysteresis, typically in excess of 200 mV for TTL input thresholds, and in excess of 100 mV for CMOS thresholds.

Required Input Rise and Fall Times

For unambiguous operation, the input rise time should not exceed 200 ns; the input fall time should not exceed 80 ns.

These values were established through a worst-case test with internal ring oscillators driving all I/O pins except two, thus generating a maximum of on-chip noise. One of the remaining I/O pins was then tested as an input for single-edge response, the other one was the output monitoring the response. This specification may, therefore, be overly pessimistic, but, on the other hand, it assumes negligible PC board ground noise and good Vcc decoupling.

Outputs

All ATT3000 FPGA outputs are true CMOS with n-channel transistors pulling down and p-channel transistors pulling up. Unloaded, these outputs pull rail to rail.

Table 2. dc Parameters

Parameter	Value	Unit
Output Impedance		
Sinking, Near Ground	25	Ω
Sourcing, Near Vcc	50	Ω
Output Short-circuit Current		
Sinking Current by the FPGA	96	mA
Sourcing Current by the FPGA	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.

Table 3. ac Parameters

Parameter	Value		Unit
	Fast*	Slow*	
Unloaded Output Slew Rate	2.8	0.5	V/ns
Unloaded Transition Time	1.45	7 .9	ns
Additional Rise Time for 812 pF	100	100	ns
Normalized	0.12	0.12	ns/pF
Additional Fall Time for 812 pF	50	64	ns
Normalized	0.06	0.08	ns/pF

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

FPGA 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 designers will usually have great difficulty estimating the average frequency on other nodes.

Two extreme cases are the following:

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 exercised at the clocking speed.

Table 4. FPGA Power Output

Parameter	Dynamic Power (mW/MHz)
Output with 50 Ω Load	1.9
Global Clock (ATT3020)	1.7
Global Clock (ATT3090)	3.6
CLB with Local Interconnect	0.36
Horizontal Long Line (ATT3020)	0.09
Horizontal Long Line (ATT3090)	0.15
Vertical Long Line (ATT3020)	0.19
Vertical Long Line (ATT3090)	0.075

* Add 2.5 mW/MHz for every 100 pF of additional load.

Table 5. Example of Power Output Using the ATT3020

Parameter	Dynamic Power (mW)
3 Outputs at 5 MHz	28
20 Outputs at 0.1 MHz	4
Global Clock at 20 MHz	34
10 CLBs at 5 MHz	18
40 CLBs at 0.2 MHz	3
16 Vertical Long Lines at 1 MHz	1
20 Inputs at 4 MHz	6
Total	94 mW

CCLK Frequency Variation

Configuration clock (CCLK) is the internally generated free-running clock that is responsible for shifting configuration data into and out of the device.

Table 6. Configuration Clock Frequencies

Vcc	T (°C)	Freq (kHz)
4.5 V	25	687
5.0V	25	691
5.5 V	25	695
4.5 V	–30	966
4.5 V	130	457

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.

Crystal Oscillator

The on-chip oscillator circuit consists of a high-speed, high-gain inverting amplifier between two device pins, requiring an external biasing resistor R1 of 0.5 MΩ to 1 MΩ.

A series-resonant crystal Y1 and additional phase-shifting components R2, C1, and 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 MHz to 24 MHz

Third overtone operation from 20 MHz to 75 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.

Table 7. Crystal Oscillator Parameters

Freq (MHz)	L (µH)	C (pF)	LC Tank Freq (MHz)	R2 (Ω)	C1 (pF)
32	1	60	20.6	430	23
35	1	44	24.0	310	23
59	1	31	28.6	190	23
72	1	18	37.5	150	12

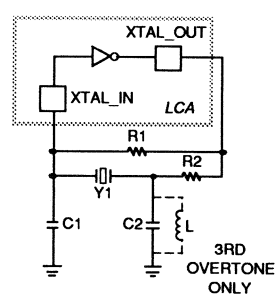


Figure 1. Crystal Oscillator

AT&T Microelectronics

FPGA Performance

Estimating CLB Performance

Since the delays in FPGA-based designs are layout dependent, the data sheet cannot give all of 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 designers want to evaluate the possible performance, well before they have finished the design.

Here are some guidelines for ATT3000 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:

Parameter	-50	-70	-100	-125
Clock-to-Output (ns)	12	8	7	6
Routing (ns)	1	1	1	1
Logic Setup (ns	12	8	7	6
Clock Period	25	17	15	13
Clock Freq. (Mhz)	40	59	67	77

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

Parameter	-50	-70	-100	-125
Clock-to-Output (ns)	12	8	7	6
Routing (ns)	1	1	1	1
Logic Setup (ns	12	8	7	6
Clock Period	36	24	20	18
Clock Freq. (Mhz)	28	42	50	56

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

Parameter	-50	-70	-100	-125
Clock-to-Output (ns)	12	8	7	6
Routing (ns)	12	8	6	6
Logic Delay (ns)	14	9	7	6
Routing (ns)	1	1	1	1
Logic Setup (ns)	12	8	7	6
Clock Period	25	17	15	13
Clock Freq. (Mhz)	40	59	67	77

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. Designing with typical numbers should be avoided.

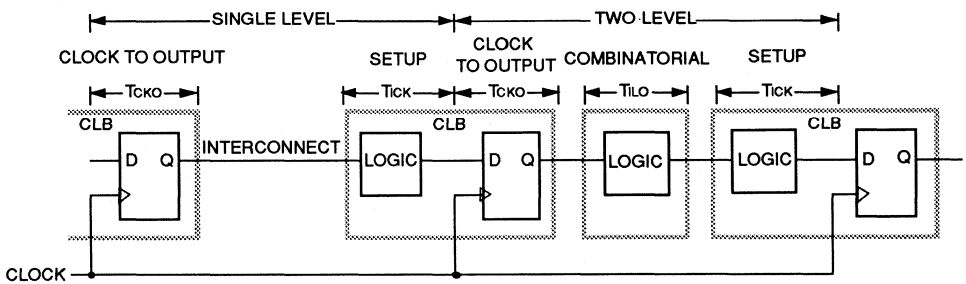


Figure 1. Critical Timing Parameters for Clocked CLB Driving Clocked CLB Directly (Single Level) and Driving It Through Additional Combinational Logic (Two Level)

Designing for Highest Data Transfer Rate Between ATT3000 Family FPGAs

Worst-case analysis of a synchronous data transfer between ATT3000 family devices postulates that the sum of clock-to-output propagation delay of the sending device, plus the input-to-clock setup 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 FPGA without specifying certain restrictions:

On the transmitting FPGA , the clock-pin to output-pin propagation delay is minimized if TCLKIN or BCLKIN are chosen as clock inputs. They are CMOS-level only, and offer the shortest on-chip clock delay.

The clock-pin to output delay is then:

2 + 3.5 + 9 = 14.5 ns for the 3020-125

2 + 3.9 + 10 = 15.9 ns for the 3020-100

3 + 4.5 + 13 = 20.5 ns for the 3020-70

On the receiving FPGA, the input-pin to clock-pin setup time is the specified I/O pad input set-up time (parameter TPICK in the IOB switching characteristic table of the ATT3000 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 setup time is the following:

16 – 2 – 3.5 = 10.5 ns for the 3020-125

17 – 2 – 3.9 = 11.1 ns for the 3020-100

20 – 3 – 4.5 = 12.5 ns for the 3020-70

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

14.5 + 10.5 = 25 ns, i.e., max 40 MHz for the 3020-125

15.9 + 11.1 = 27 ns, i.e., max 37 MHz for the 3020-100

20.5 + 12.5 = 33 ns, i.e., max 30 MHz for the 3020-70

Bypassing the input flip-flop in the IOB and going directly to the DI input of the closest CLB is another, nonobvious way of improving performance by 8 ns for the 3020-125 device, by 9 ns for the 3020-100 device, and by 10 ns for the 3020-70 device.

If this is not fast enough, there are design methods that can improve the performance. Assume a 100 MHz device. The easiest and safest method is to increase the clock delay on the receiving FPGA, thus reducing the apparent input setup time. Changing to a direct input (instead of TCLKIN) adds 2 ns to the clock delay and subtracts it from the input setup time.

More aggressive methods of increasing clock delay inside or outside the receiving FPGA must be used with care, since they might reduce the best case setup 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.

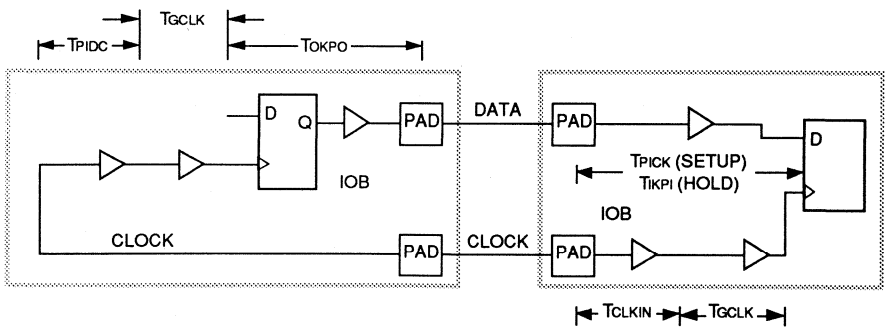


Figure 2. Critical Timing Parameters for Data Transfer Between FPGAs

Input Setup Time on a ATT3000 Family FPGA Is Better Than the Specification

The ATT3000 family data sheet specifies a worst-case setup time of 17 ns for the 100 MHz speed grade, but this is the data input pad setup 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 setup value in order to arrive at the true system's setup time as seen on the device package pins (pads) for data and clock. Since the internal clock delay can be manipulated by the user, AT&T cannot specify the system's setup time.

The shortest possible clock delay from the package pin to the IOB clock is achieved by selecting the CMOS compatible clock inputs TCLK or BCLK. The guaranteed max value for their delay is 4.9 ns (ATT3020-100), the sum of 2 ns for pad-to-CLKIN plus 2.9 ns for the clock buffer and clock distribution.

AT&T does not guarantee any shortest values for all of these parameters. An unrealistic worst-case analysis might, therefore, assume two extreme values:

- 17 ns setup time for a slow data input with an infinitely fast clock path
- 4.9 ns hold time for an infinitely fast data input combined with a slow clock path

That is a meaningless mathematical exercise. In reality, all of 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 setup time for the data input with respect to the CMOS compatible clock input is, therefore, 14.4 ns (17 ns minus 54% of 4.9 ns).

What is the shortest setup time, and 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 setup time with respect to the CMOS compatible clock input is, therefore, 1.2 ns (10% of 17 ns minus 4.9 ns). This is still a positive value, sometimes called a negative hold time. There will never be a hold time requirement if the user selects the CMOS-compatible clock-input option.

No Guaranteed Minimum 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 minimum value would be surprisingly large. There are five reasons:

1. Temperature. CMOS propagation delays decrease approximately 0.3% per degree C.
2. Supply Voltage. CMOS propagation delays are just about inversely proportional to 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. FPGAs are sorted into a few speed classes. A part marked 50 MHz might have barely missed the 70 MHz specification in only a few or perhaps only one parameter. IC manufacturers may sometimes mark down (call a 70 MHz part a 50 MHz part) in order to adjust production yield to market demand. This increases the spread even more.
5. Process Evolution. As IC technology improves, smaller geometries reduce not only device size and cost, but also propagation delay. Tight minimum specifications would be a hindrance to progress.

Finally, it can be argued that a proper synchronous design is insensitive to minimum propagation delay values. When the clock skew is small (AT&T clock networks guarantee extremely small clock skew values, less than 2 ns over a big chip like the ATT3090), the designer can safely ignore the minimum delay issue. No AT&T 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 FPGA where delays track, and the temperature is the same for all elements.

AT&T Microelectronics

Delay Tracking

Tracking Chip Delays

Delays on a chip vary, relative to each other.

When one delay value is known exactly, a certain tolerance must be assumed for any other delay value on the same chip.

Figure 1 explains delay tracking, i.e., the correlation between any two delays, called A and B, on the same chip. Delay A is represented on the horizontal axis; delay B, on the vertical axis. 100% represents the guaranteed worst-case maximum value for either of the two delays.

No delay can ever be shorter than 10% of the specified maximum value; the horizontal and vertical strips parallel to both axes, therefore, represent impossible combinations. If the correlation between delay parameters were perfect, all possible combinations would be on the diagonal line. If there were no correlation at all, the whole remaining square would represent valid combinations. AT&T assumes a correlation factor of 70%; all possible combinations thus fall onto the unshaded part of the square.

Examples:

If one delay is at its maximum value, the other is between 70% and 100% of its maximum value. If one delay is at 50% of its maximum value, the other one can be anywhere between 71% and 35% of its specified maximum value.

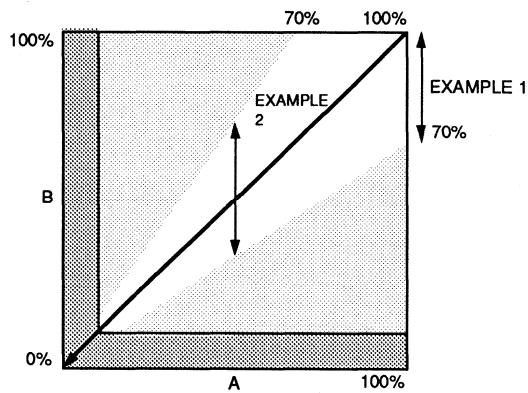


Figure 1. Delay Tracking Example

Warning: Typicals Are Hazardous to Your Designs

All AT&T parameter specifications in this data book, shown on the screen and in any simulator, are worst-case values, guaranteed over the full range of operating conditions. Delays in all CMOS devices are inevitably longest at the highest temperature and the lowest supply voltage, and delays are also affected by variations in the manufacturing process.

A clear description of these delay variations is shown in Figure 2 below where the typical delay, exhibited by an average device at 25 °C and 5.0 V is slightly more than half the worst-case delay for a commercial temperature range product and is less than half of the worst-case military value.

Designers should regard typical values as meaningless averages, taken under favorable operating conditions. Nobody should base a design on typical values, but some manufacturers still use this misleading way to specify device performance.

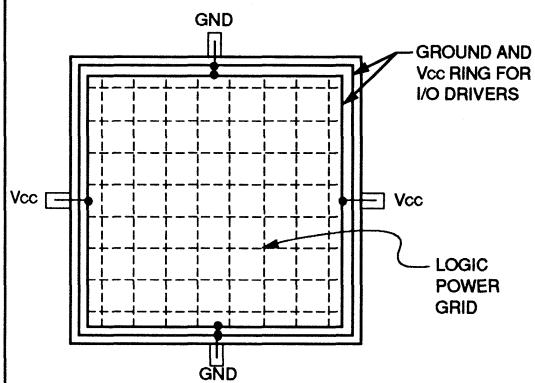


Figure 2. FPGA Power Distribution

AT&T Microelectronics

Start-Up and Reset

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 FPGAs 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 (active-low) signal.

Synchronous Reset After Configuration

After configuration is completed, the FPGA 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:

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

End of configuration before first system clock:

> LDC pin goes active-high, Q stays low, D stays high.
> $\overline{\text{RESET}}$ is still pulled high by external resistor.

Result of first system clock after end of configuration:

> Q is clocked high, which forces D low.
> Output driver goes active-low and forces $\overline{\text{RESET}}$ low. This resets the whole chip until the low on Q causes $\overline{\text{RESET}}$ to be pulled high again. The whole chip has thus been reset by a short pulse instigated by system clock.

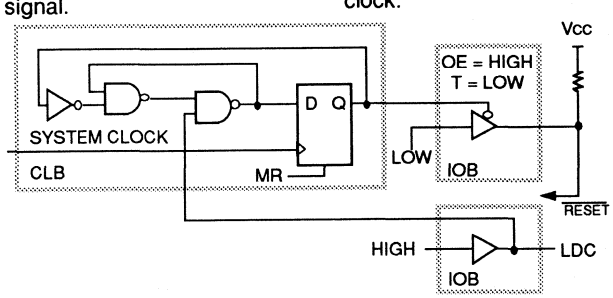


Figure 1. Synchronous Reset

Application Note
June 1992

AT&T Microelectronics

Metastable Recovery

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 1970's, it has developed into a scary subject for every designer of asynchronous interfaces. Now AT&T offers data and a demonstration kit to help users analyze and predict the metastable behavior of FPGAs.

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 chance to a valid logic state.

The basic phenomenon is unavoidable, 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.

AT&T has evaluated the ATT3020 CLB flip-flop with the help of a mostly self-contained circuit on the demonstration board that is available to any AT&T customer.

The result of this experimental evaluation shows the AT&T 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:

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).

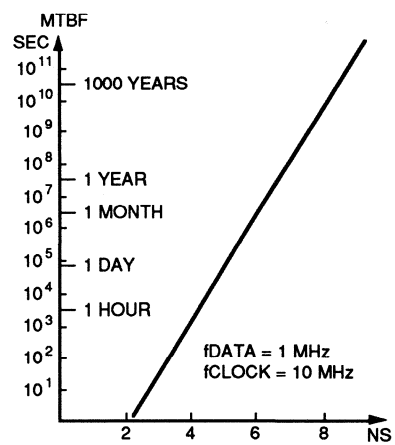


Figure 1. Metastable MTBF as a Function of Additional Acceptable Delay

Measuring Metastable Recovery

The excellent metastable recovery rate of FPGA flip-flops was measured in a working ATT3020-70 on an 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 setup 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 setup 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 of 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 of 40; see Figure 1 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 was wrong by a factor of 10, it would only move the graph horizontally by 0.624 ns.

Over the past fifteen years, many attempts have been made to pinpoint, 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 FPGA flip-flops to be superior to any other circuit documented so far. There is a reason: FPGA 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 FPGAs offer the closest thing to an acceptable solution.

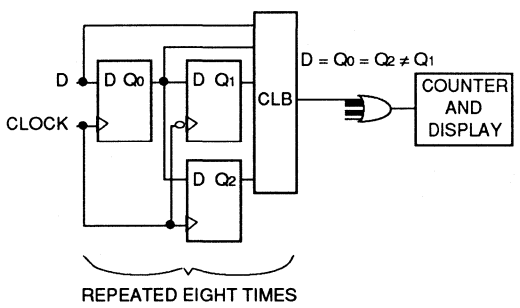


Figure 2. Metastable Delay Measuring Circuit

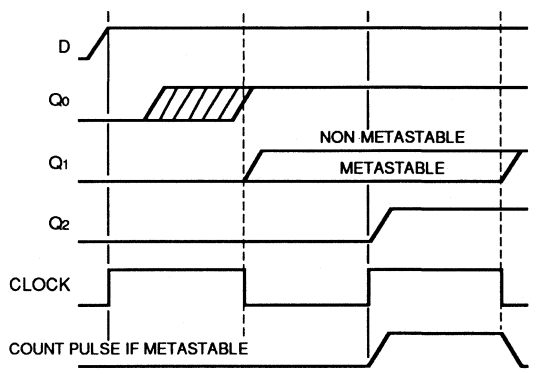


Figure 3. Metastable Detection

Application Note
June 1992

AT&T Microelectronics

Battery Backup for Field-Programmable Gate Arrays

Low-Power, Nonoperational Status

Field-programmable gate 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 nonoperational 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 powerdown ($\overline{\text{PWRDWN}}$) pin.
- Switching between primary Vcc supply and battery. Important considerations are the following:
  - — Ensure 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 re turned to a normal level. $\overline{\text{PWRDWN}}$ edges must not be slow rising or falling.
  - — Ensure glitch-free switching of the power connections to the FPGA from the primary Vcc to the battery voltage and back.
  - — Ensure that during normal operation the FPGA Vcc is maintained at an acceptable level, 5.0 V ± 5% (±10% for industrial and military).

Figure 1 shows a circuit developed by Shel Epstein of Epstein Associates in Wilmette, IL. Two 1N5817 Schottky diodes power the FPGA from either the 5.2 V supply or a 3 V Lithium battery. A SEIKO S8054 3-terminal power monitor circuit measures Vcc and pulls $\overline{\text{PWRDWN}}$ low.

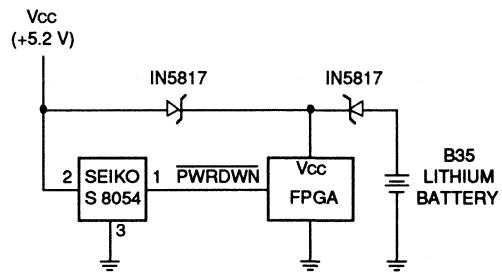


Figure 1. Battery Backup Circuit

Application Note
June 1992

AT&T Microelectronics

Compact Multiplexer and Barrel Shifter

Four-Input Multiplexer in One CLB

Since the function generator in the ATT3000-series CLB has only five inputs, it cannot directly implement a four-input multiplexer, which requires four data inputs and two select inputs.

Registering one of the select inputs in the same CLB frees up one input and puts a complete four-input multiplexer into one CLB. It is even possible to register the multiplexer output.

This nonobvious trick increases the apparent delay of the registered select input, but that will be acceptable in the majority of applications. Since it reduces not only the size but also the through-delay of the four-input multiplexer by 50%, this approach is definitely worth considering.

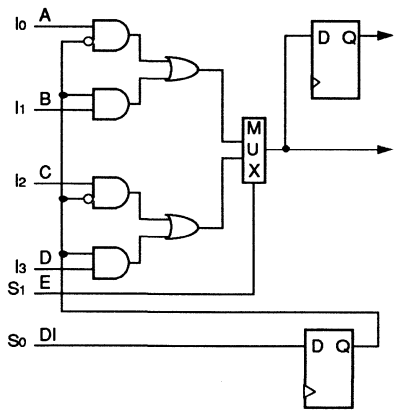


Figure 1. Four-Input Multiplexer

4-Bit Barrel Shifter in Only Four CLBs

A four-input barrel shifter has four data inputs, four data outputs, and two control inputs that specify rotation by 0, 1, 2 , or 3 positions. A brute force design would use four four-input multiplexers, since each output can receive data from any input. Each four-input multiplexer requires two ATT3000-family CLBs, for a total of eight CLBs.

There is, however, a smarter method that reduces the design to only four CLBs. The key to this approach lies in the signal crossovers at the input and output of the second-level CLBs.

8-Bit Barrel Shifter in 12 CLBs

The 4-bit barrel shifter design can be extended to 8 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.

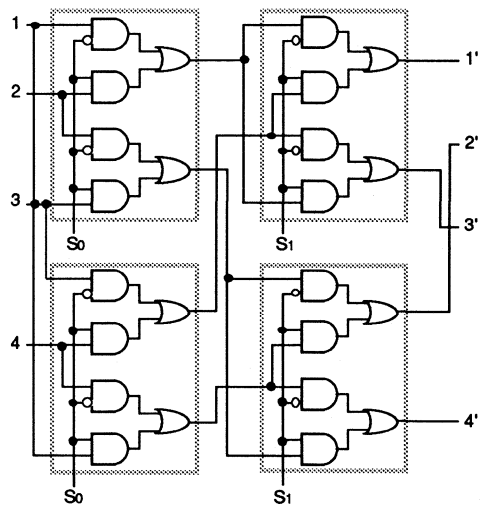


Figure 2. 4-Bit Barrel Shifter

AT&T Microelectronics

Majority Logic, Parity

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 FPGAs can generate any function of five variables at the same cost and the same delay, they can easily decode majority logic. The output F, G is low when none, one, or two inputs are high; it is high when three, four, or five inputs are high.

Majority logic is a special case of N-of-X decoding. An ATT3000-series CLB can directly encode any N of five inputs active. This concept can be cascaded so that three CLBs encode any N of seven inputs active.

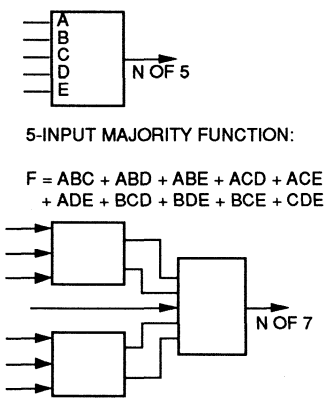


Figure 1. N of X, e.g., Majority Logic

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

Parity

Two CLBs can generate the parity for nine inputs or can check a 9-bit input for odd or even parity with a through-delay of two cascaded CLBs. Three CLBs can check 13 inputs; four CLBs can check 17 inputs; five CLBs can check 21 inputs; six CLBs can check 25 inputs; all with the same delay of two cascaded CLBs.

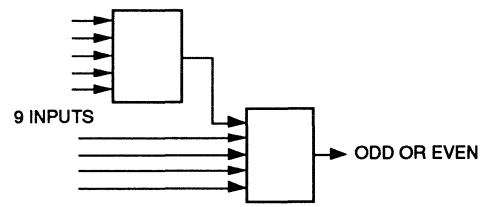


Figure 2. Nine-Input Parity

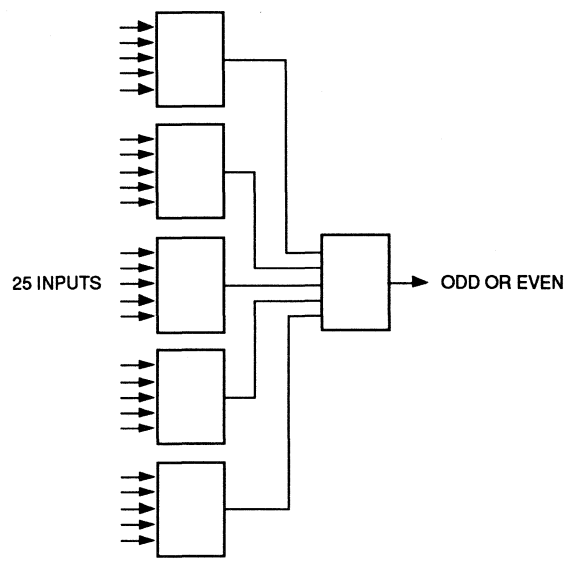


Figure 3. 25-Input Parity

AT&T Microelectronics

Multiple Address Decoding

Decoding Methods

An ATT3000-series CLB can decode a 5-bit address in any conceivable way, or it can decode a 4-bit address in two different ways, each without any restrictions.

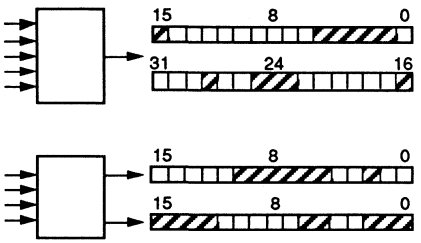


Figure 1. Multiple-Address Decoding

Three ATT3000-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.

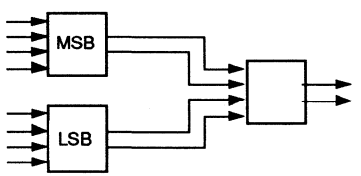


Figure 2. Three-Address Decoding

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—3	generate LS = 1
inputs 4—8	generate LS = 2
inputs 9—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—F	generate MS = 0
input 2	generates MS = 1
inputs 3—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.

AT&T Microelectronics

Binary Adders, Subtractors, and Accumulators

Implementation Trade-offs

There are many different ways to implement binary adders, subtractors, and accumulators with FPGAs using different trade-offs between size and speed.

Most compact, but slowest, is the bit-serial function that operates on one bit pair per clock cycle, generating sum and carry. The sum is fed back into the shift register, and the carry is stored for the subsequent bit time.

The most compact combinatorial (parallel) adder, subtractor, or accumulator consists of cascaded CLBs. Each CLB 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 five-input function generator of the ATT3000 family can add a carry to two operator pairs. Three CLBs can thus handle two input bit pairs, generating two sum outputs and the outgoing carry. A 16-bit function requires 24 CLBs. It performs an operation in eight combinatorial delays.

Carry propagate and carry generate are intermediate signals that can speed up the operation as shown on pages 5-33 and 5-34. Such a 16-bit function requires 30 CLBs. It performs an operation in six combinatorial delays.

The concept of carry propagate and carry generate has been made popular by the 74181 ALU and its descendents.

These signals can reduce the ripple carry delay. Both CP and CG are outputs from an arithmetic block (often of 4 bits). Both of 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 1990 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 ATT3000 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 AT&T adder structures can also be accumulators without any size or speed penalty. Conventional gate arrays and other gate-array-like structures usually configure flip-flops out of gates. The flip-flop setup time must then be added to the combinatorial propagation delay. FPGAs hide the flip-flop set-up time in the combinatorial propagation delay of the CLB. Adders and accumulators thus operate at the same speed.

AT&T Microelectronics

Adders and Comparators

Description

The FPGA structure accommodates 2-bit adders very efficiently. An ATT3000-series CLB can even include an additional control input, either ADD/SUBTRACT or ADD ENABLE.

A 2-bit adder requires three ATT3000-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 4-bit adder in six CLBs with a through-delay of two CLBs, i.e., 25 ns (allowing for some interconnect delay).

Four 2-bit adders can be cascaded to form a byte-wide adder, using 12 CLBs with a through-delay of four CLBs, but there is also a slightly faster design using a carry look-ahead technique. The third and fourth di-bit adders are changed; they no longer generate carry out, but now each generates two outputs as a function of the four A and B inputs (ignoring 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.

For 8 bits, this look-ahead carry scheme is 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).

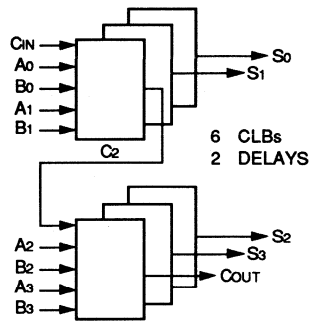


Figure 1. Four-Bit Adder

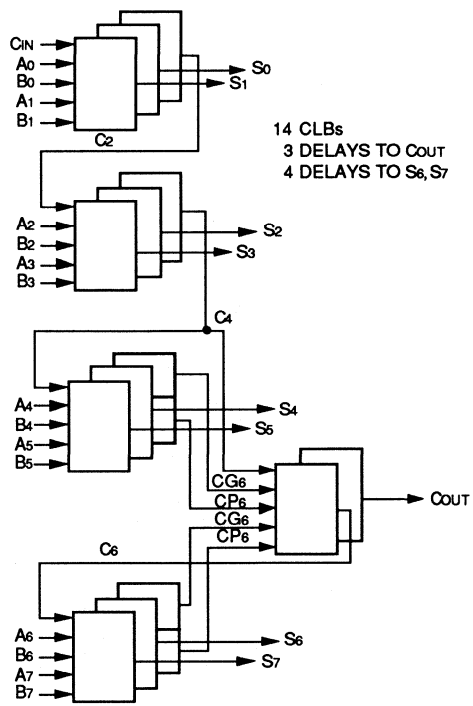


Figure 2. Eight-Bit Adder with Carry Look-Ahead

Description (continued)

A 16-bit adder benefits from carry look-ahead. Simply cascading di-bit adders uses 24 CLBs at a max propagation delay of eight CLBs from CIN to COUT or to S14, 15.

A look-ahead carry scheme uses 30 CLBs at a max prop delay of five CLBs from CIN to COUT (six delays to S14, 15).

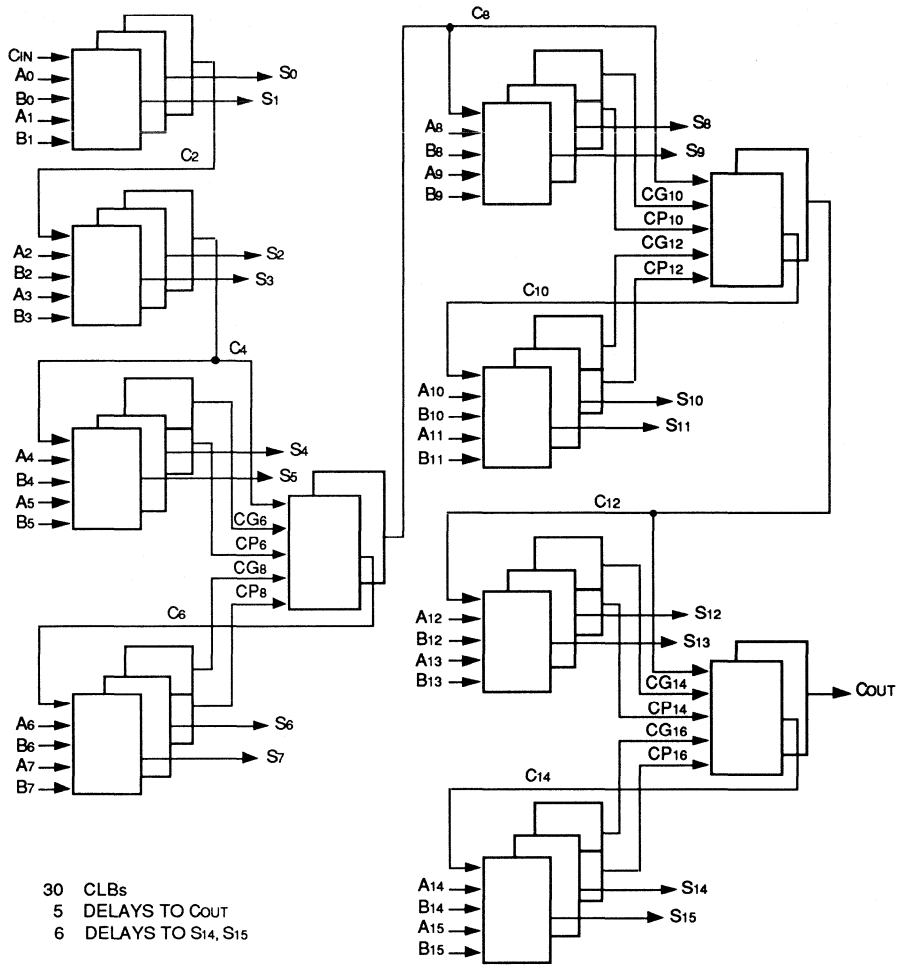


Figure 3. 16-Bit Adder with Carry Lookahead

Description (continued)

Adder Logic Truth Tables

After adjusting the subscripts appropriately, the truth table for the three CLBs generating S2 and S3 is identical with that for the CLBs generating S0 and S1; and the truth table for the bottom three CLBs is identical to that of the three CLBs generating S4 and S5.

The 16-bit adder is a natural extension of the 8-bit adder.

Table 1. Adder Logic Truth Table

Outputs	Inputs				
	C4	A0	B0	A1	B1
S0 =1	1	0	0	x	x
	0	1	0	x	x
	0	0	1	x	x
	1	1	1	x	x
S1 =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
C2=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

Table 2. Adder Logic Truth Table

Outputs	Inputs				
	C_4	A_4	B_4	A_5	B_5
S4 =1	1	0	0	x	x
	0	1	0	x	x
	0	0	1	x	x
	1	1	1	x	x
S5 =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
CG6 =1	x	x	x	1	1
	x	1	1	1	0
	x	1	1	0	1
CP6 =1	x	0	1	1	0
	x	1	0	1	0
	x	0	1	0	1
	x	1	0	0	1

Table 3. Adder Logic Truth Table

	C_{IN}	Low CP_n	CG_n	High CP_{n+2}	CG_{n+2}
$Carry_n$	x	x	1	x	x
	1	1	x	x	x
$Carry_{n+2}$	x	x	x	x	1
	x	0	1	1	x
	1	1	0	1	x

Description (continued)

The CLB architecture is ideally suiited for bit-serial arithmetic, where the function generator performs the serial arithmetic (LSB first) and the associated flip-flop stores the carry or borrow.

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, stays set until the end of the word, and 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.

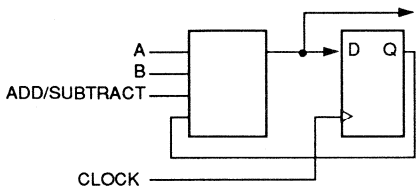


Figure 4. Serial Adder/Subtractor

A bit-serial magnitude comparator distinguishes between A > B and A < B. It can operate LSB first or MSB first, if the logic is adjusted:

LSB first: Start with both flip-flops reset
if A > B set Qx, reset Qy
if A < B set Qy, reset Qx

MSB first: Start with both flip-flops reset
if A > B and Qy = 0: set Qx
if A < B and Qx = 0: set Qy

Result in both cases:

Qx	Qy	
0	0	A = B
0	1	A < B
1	0	A > B
1	1	Impossible

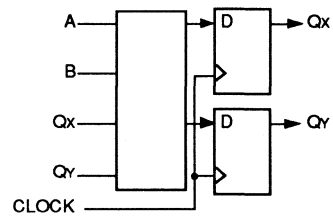


Figure 5. Serial Magnitude Comparator

AT&T Microelectronics

Conditional Sum Adder Adds 16 Bits in 33 ns

Conditional Sum Adder

This circuit is based on a 1960 paper by J. Sklansky (see page 5-28). 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.

Of the 41 CLBs, 27 each generate one function of up to five variables, and 14 each generate two functions of four variables.

In accordance with the original paper, all subscripts denote the binary position (weight) and superscripts describe the assumed input condition:

0: carry into this position is assumed inactive

*0: carry into the position one lower is assumed inactive

1: carry into this position is assumed inactive

*1: carry into the position one lower is assumed inactive

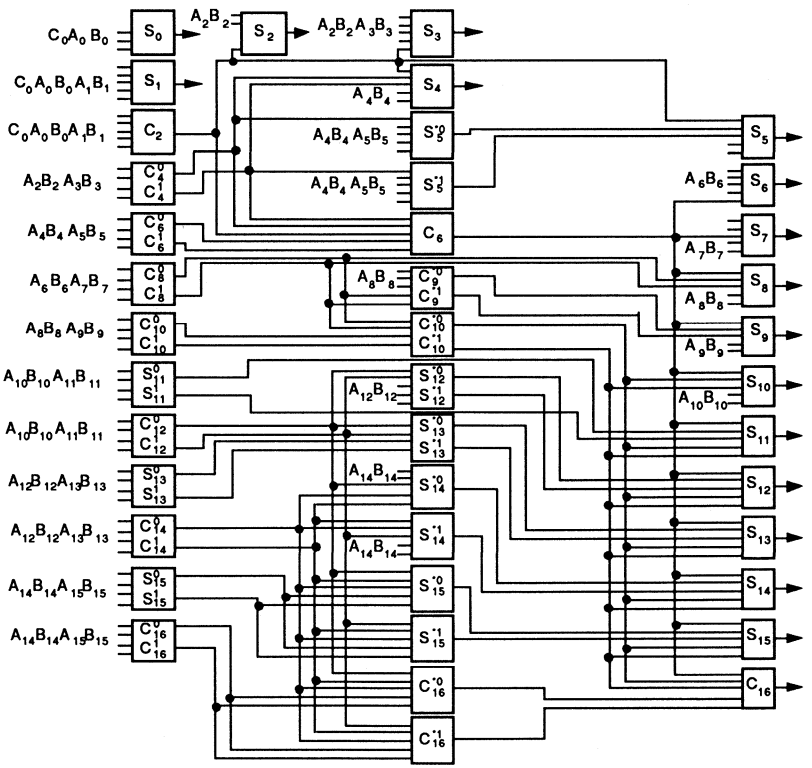


Figure 1. Conditional Sum Adder

AT&T Microelectronics

Building Latches Out of Logic

Latch Designs

Since the ATT3000 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 permits an amazing diversity of latch designs; here are several ideas:

With F fed back to close the feedback path, there are four control inputs left. They might be called set, reset, data, and enable, defined such that S and R are independent of enable, but D is activated by it. Any of these four inputs can be defined as active-high or active-low. This results in 16 different latch designs, all with the same basic characteristics and the same timing.

We can also eliminate D and have two enables, affecting S and R (again 16 different flavors); or we use multiple S and multiple R, either ORed, or ANDed, or XORed.

We can also have two D inputs, each with its own enable; or we can have two D inputs, a select input and an enable input; or we can have an enable and three D inputs defined in any arbitrary way.

Majority gating could be one way: if none or one is active, reset the latch; if two or three are active, set the latch. Or, if none is active, reset; if one or two are active, hold; if three are active, set. Or we can assign positive or negative weights to the D inputs.

Since there are 65,536 different functions of four variables, there are many different ways to define a latch, not counting pin rotations and active-high/active-low variations.

All of these latches have the same timing characteristics: propagation delay from input to output = 14 ns/9 ns for the 50 MHz/70 MHz part. Setup time to the end of nable, or min.

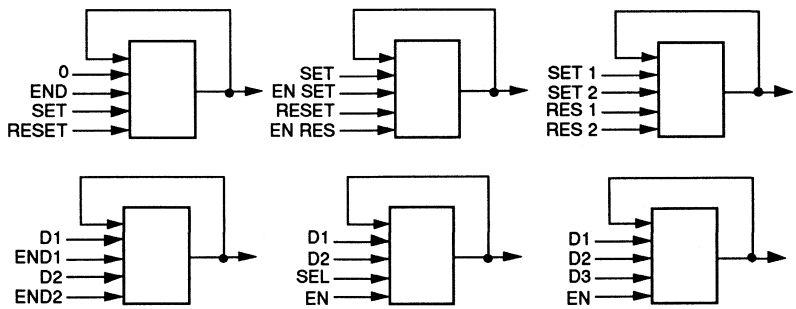


Figure 1. Latched Logic

Application Note
June 1992

AT&T Microelectronics

Synchronous Counters: Fast and Compact

Fully Synchronous 4-Bit Counter Uses Only Two CLBs to Count Any Code

This 4-bit counter operates synchronously and has a count enable (CE) input. Count length, count direction, and even the code sequence can be selected through configuration. There are 15, i.e., more than 1012 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.

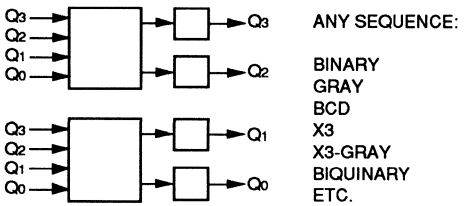


Figure 1. Synchronous 4-Bit Counter in Two CLBs

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.

Table 1. Counter Codes

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		
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 ATT3000-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 five-input combinatorial function F5 determines the modulus (there are no illegal or hang-up states). The five-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

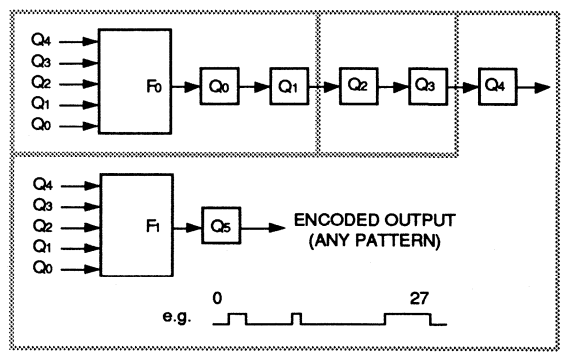


Figure 2. Synchronous 5-Bit Counter In Three CLBs

AT&T Microelectronics

30 MHz Binary Counter Uses Less Than One CLB per Bit

Counter Segmentation

Borrowing the concept of count-enable trickle/count-enable parallel that was pioneered in the popular 74160 TTL-MSI counter, a fast nonloadable synchronous binary counter of arbitrary length can be implemented efficiently in the ATT3000-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 of its count-enable (CE) inputs are active. CEO is active when all three bits are set **and** CET is high. CEP does not affect CEO.

The least-significant tri-bit thus stops the remaining counter chain for seven out of eight incoming clock pulses, allowing ample time for the CEO-CET ripple-carry chain to stabilize. Max clock rate is determined by the first tri-bit's clock-to-CEO delay (T_{CKO} + T_{ILO}), plus the CEP input setup time for all other tri-bits (TICK), plus the routing delay of the CEP net. In a 70 MHz device, this sum can be below 32 ns. The higher tri-bits are not speed critical if they propagate the CET signal in less than eight clock periods, easily achievable for counters as long as 20 tri-bits, i.e., 60 bits.

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

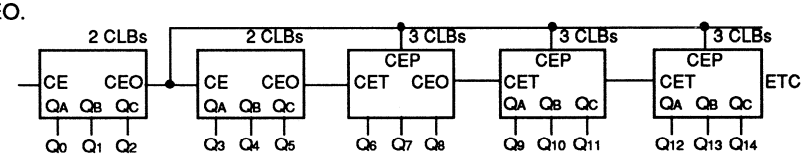


Figure 1. 30 MHz Nonloadable Binary Counter, Expandable up to 60 Bits

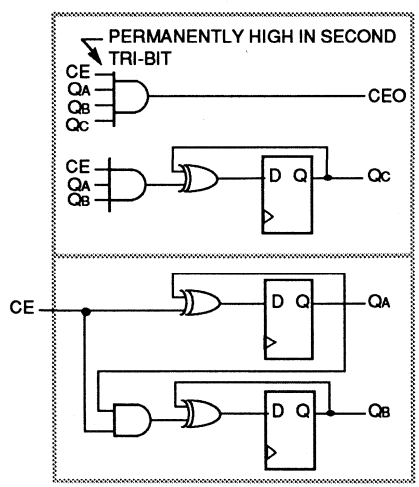


Figure 2. First and Second Tri-Bits Use Two CLBs Each

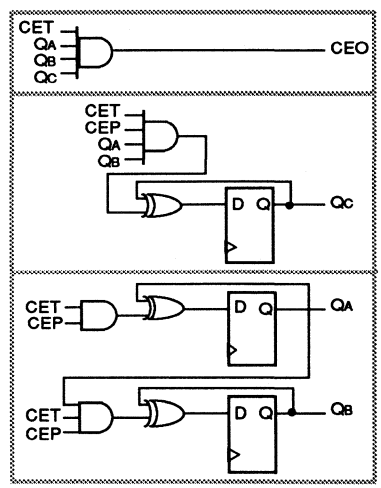


Figure 3. All More Significant Tri-Bits Use Three CLBs

AT&T Microelectronics

Up/Down Counter Uses One CLB per Bit

Cascaded Toggle Control

A fully synchronous resettable but nonloadable up/down counter of arbitrary length can be implemented with only one 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 MHz 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 ATT3000 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 MHz part can count 10 MHz, guaranteed worst case.

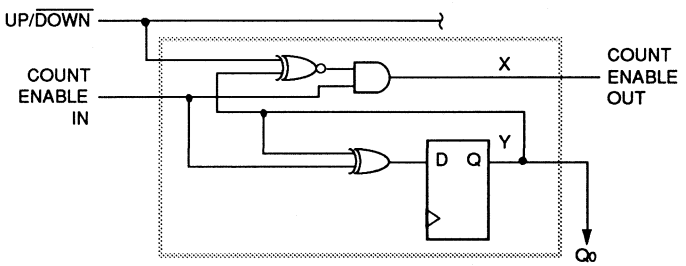


Figure 1. Up/Down Counter Implemented with One CLB per Bit

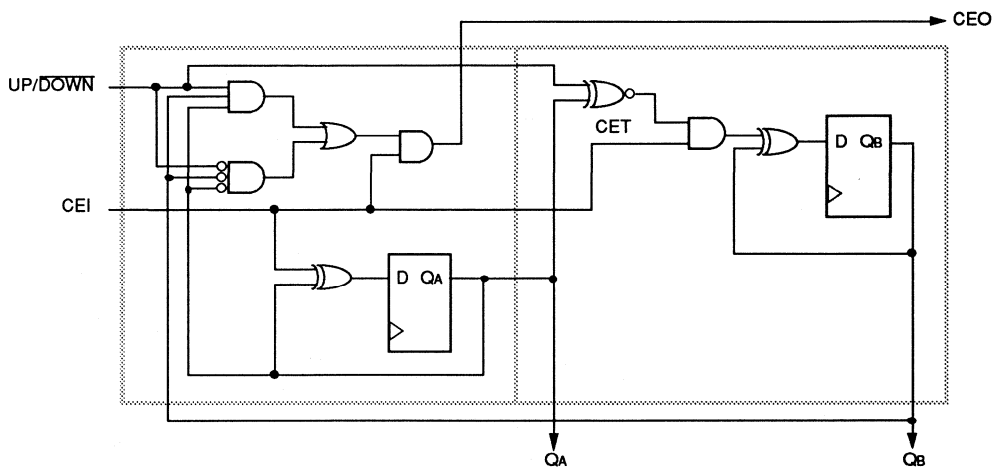


Figure 2. Up/Down Counter Implemented with Reduced Delays

AT&T Microelectronics

Loadable Up/Down Counter Uses One CLB per Bit

Description

The five-input function generator of the ATT3000 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 nonloadable 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 MHz 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.

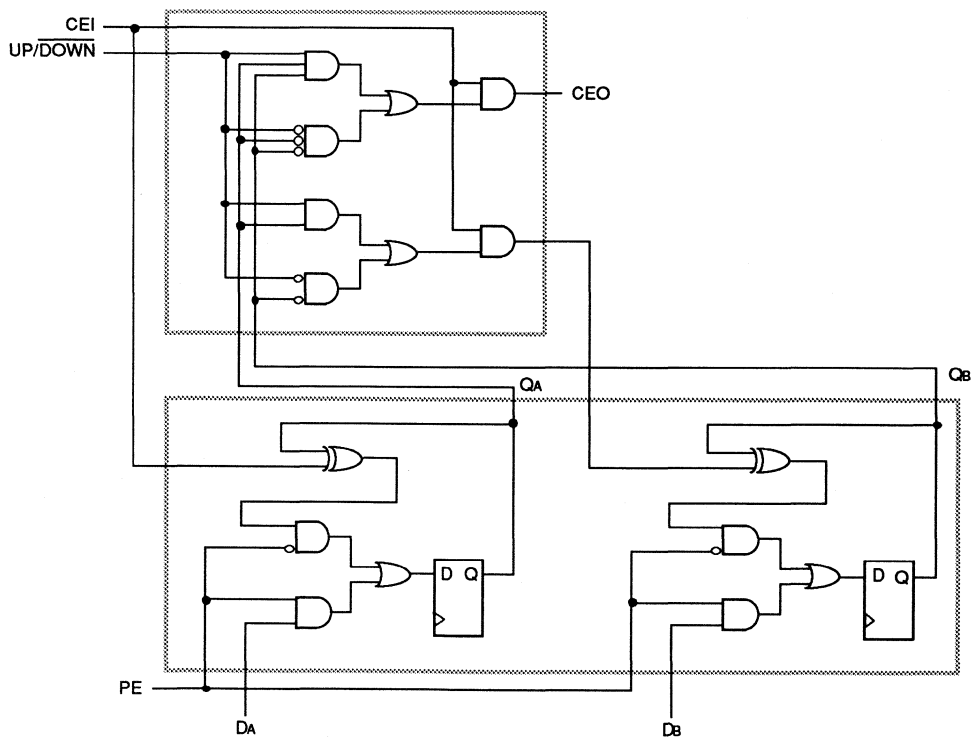


Figure 1. Loadable Up/Down Counter Schematic

Application Note
June 1992

AT&T Microelectronics

30 MHz Binary Counter with Synchronous Reset/Preset

Description

In many applications, design modularity is more important than highest clock speed and best space efficiency. A counter design is described here that uses identical CLB primitives, one CLB per bit. The count-enable trickle/count-enable parallel concept, introduced by the 74160 family, is changed here to a 1-bit clock size. Any block increments only if both count enables are high, but the outgoing carry (COUT) is not a function of CEP. The CEP input prevents erroneous counts while the ripple carry chain is settling.

A shorter counter (6 bits or less) drives the CEP net from the Q0 output, achieving a 40 MHz speed. A longer counter generates a 1-in-4 duty cycle on CEP and runs at 30 MHz up to 12 bits long, or at 25 MHz up to 18 bits long as shown below. To achieve this performance, CEP and R must be driven by long lines.

Figure 3 shows a variation of the circuit in Figure 2, where the synchronous reset input ($\overline{R}$) is changed to a synchronous preset ($\overline{P}$). Any counter chain can use a mixture of these two circuits to preset the counter to an arbitrary predetermined value.

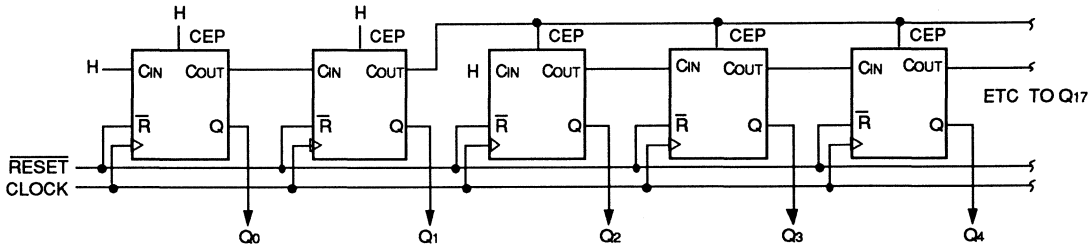


Figure 1. Long Counter (up to 18 Bits)

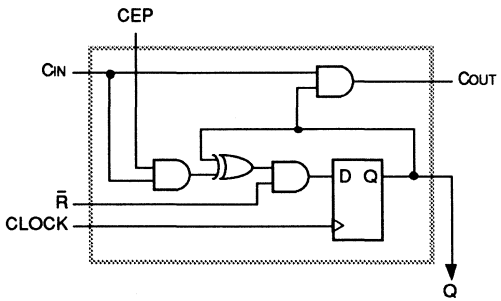


Figure 2. CLB Primitive with Reset, One per Bit

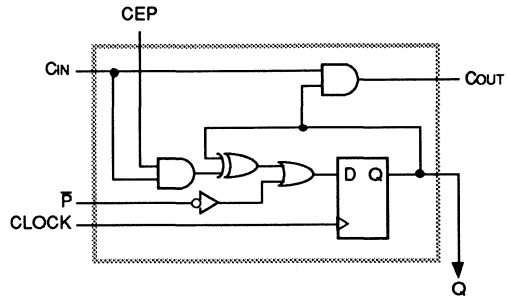


Figure 3. CLB Primitive with Preset, One per Bit

Application Note
June 1992

AT&T Microelectronics

Fast Bidirectional Counters for Robotics

Description

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 m/s and a resolution of 1 µm, 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 AT&T suggests a design that packs three 22-bit counters into one FPGA, the ATT3020. 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 FPGA clock, it is easily synchronized to the microprocessor clock. At a 20 MHz clock rate, it recirculates once and can be incremented, decremented, and also read or preset, once per microsecond.

Communication between these two parts of the counter is through a carefully controlled mailbox. Whenever the 4-bit up/down counter reaches plus or minus 8, it sets a carry or a borrow flip-flop. The shift register counter accepts these inputs synchronously, with a max delay of 1 µs.

When the microprocessor wants to read the counter, it first disables the interaction between the two parts of the counter. Then both parts are transferred into 24 output registers, and the counter interaction is enabled again. This mechanism ensures reliable read-out, even if the counter is oscillating around certain critical values.

The problem of a traditional up/down counter is that it can oscillate between two values where all (or most) counter bits change at the incoming count rate. This makes a reliable microprocessor interface virtually impossible.

In this design, the most significant 20 bits of the counter do not have this problem, and the least-significant four bits count in a grey code, where only 1 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.

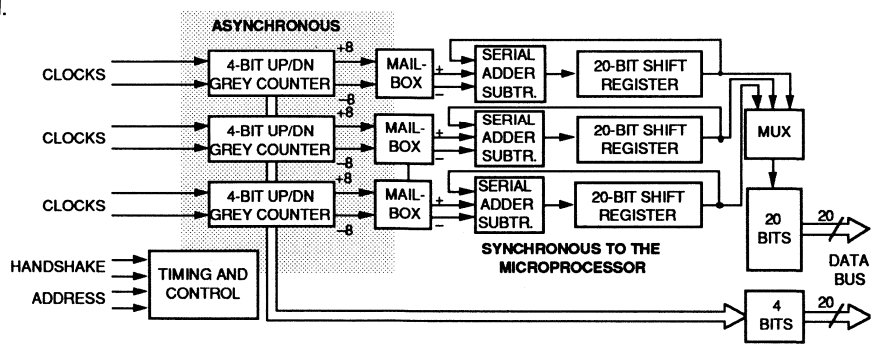


Figure 1. Triple 22-Bit Up/Down Counter with Microprocessor Interface

Application Note
June 1992

AT&T Microelectronics

40 MHz Presettable Counter

Description

A new counter architecture, described here, is used to implement a very high-speed, presettable, up-to-40-bit long, binary counter in an ATT3020 FPGA device. The design can easily be modified to implement two 20-bit counters or the equivalent BCD counters.

Traditional counter designs always represent a compromise between two conflicting goals: highest clock speed/event resolution on one hand, sophisticated features (like preset to any arbitrary value, or decode any state) on the other hand.

Asynchronous ripple counters offer highest speed, but cannot be decoded in one clock period, thus cannot be made programmable.

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

Decoding terminal count (TC) to preset the counter again poses a similar problem. The design described here separates the two functions of the carry chain as follows:

- One propagates the carry signal from the less-significant to the more-significant bit positions, and causes the appropriate flip-flop to toggle.
- One cascades the decoding of the terminal count of the whole counter and generates a parallel enable signal.

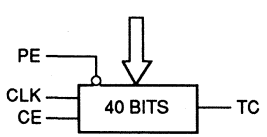


Figure 1. Cascaded TC Decoding

Cascaded TC Decoding

The TC decoder must receive inputs from all counter bits, but only the LSB timing is critical; the more-significant bits have been stable for many clock periods. TC can, therefore, be decoded in a slow gating chain that starts at the most significant end of the counter.

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 2 bits (a di-bit) long, implemented as a synchronous presettable down-counter, with carry in (= count enable), parallel enable, and two data inputs. Terminal count (0.0) is decoded with an additional input coming from the next higher section. The least significant section decodes the state prior to TC; its output activates the parallel enable for all counters. The carry function between sections is pipelined. The carry flip-flop is set when carry in is active and the di-bit is in state 00. The carry flip-flop stays set for only one clock period; its output drives the carry-in function of the next higher section. As a result of this pipelining, the counter can be made arbitrarily long without any speed penalty. Note that each di-bit, except the first, makes its transition n clock pulses later than required by the binary code sequence (n is the relative position of the di-bit, n = 0 for the input di-bit).

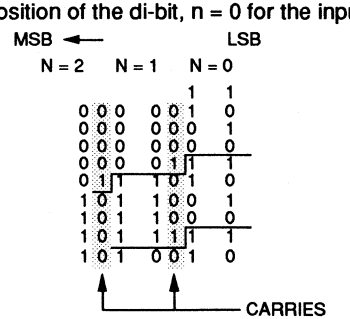


Figure 2. Carry Propagation

Carry Propagation (continued)

This code violation has no impact on TC decoding. This counter can be four times faster than presently available standard microprocessor peripherals like the 8254 and 9513. Typical applications are in instrumentation and communications; for example, as the frequency-determining counter in a phase-locked-loop frequency synthesizer.

Summary

Unlike the speed of conventional synchronous counters, the speed of this design is independent of its length. All speed-critical paths are single level; their interconnect delay can be kept below 9 ns, which means that even a 70 MHz device can count at a 40 MHz rate (worst case).

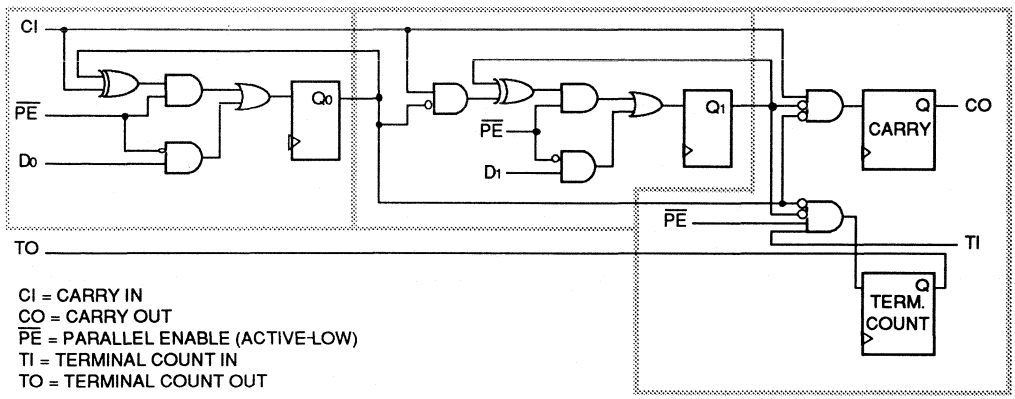


Figure 3. Any Di-Bit Except the Least Significant

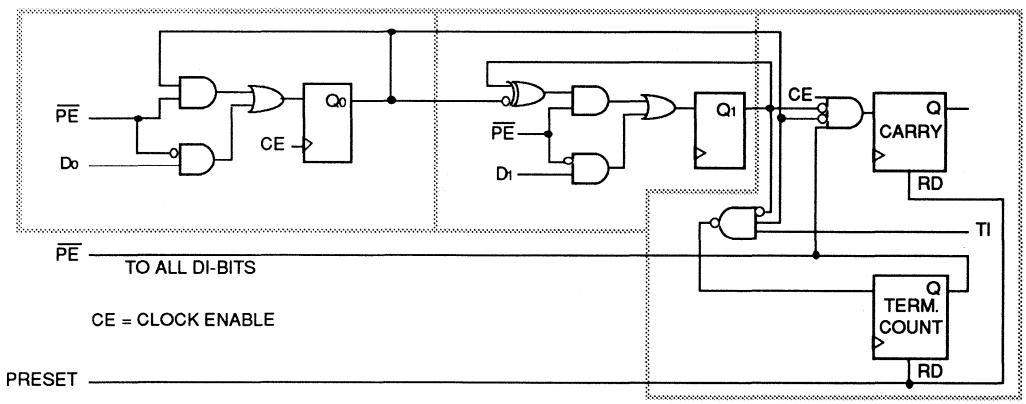


Figure 4. Least Significant Di-Bit

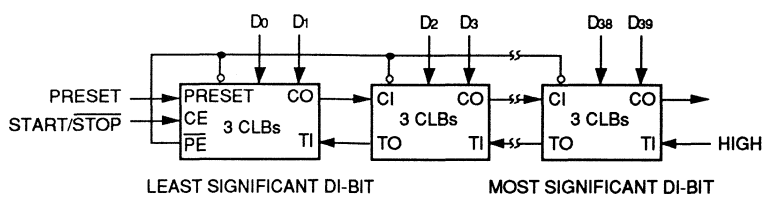


Figure 5. Synchronous Presettable Counter — 40 Bits In 60 CLBs

Summary (continued)

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

What is the function of the TC pipeline flip-flop, formerly called Q3?

The unconventional idea behind this counter design is that terminal count decoding can be rippled from the MSB to the LSB, i.e., against the direction of carries. This is possible because the high-order bits reach 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. Assume the worst case where the preset value is all zeros except a single one in the MSB position.

When this counter reaches the all-zero terminal count, $\overline{PE}$ is activated and the counter is preset. This action should obviously deactivate 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 deactivate TC within 25 ns.

The TC pipeline flip-flop and the inclusion of $\overline{PE}$ in the AND gate that detects TC reliably deactivate TC and, thus, $\overline{PE}$ one clock after they have been activated. This has one-side effect, however: it makes it illegal to preset the counter to very small numbers (less than 10 for a 20-bit counter), since the TC pipeline takes that many clock pulses to become active again.

In the unlikely case where this might cause a problem, most TC pipeline flip-flops can be eliminated. They were inserted to simplify modeling and because they are available for free.

Why is the least significant di-bit different?

To achieve a 40 MHz clock rate, the $\overline{PE}$ signal must be made as fast as possible. It has to come directly from a flip-flop output so that the sum of clock-to-output delay, routing delay, and input setup 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 PE so that the asynchronous clear input can be used to force the counter into loading.

For operation below 30 MHz, the least significant di-bit can be like all the other di-bits, but $\overline{PE}$ must be excluded from the AND gate generating $\overline{PE}$, and the user may want to adjust the polarity of the last TC pipeline flip-flop to facilitate the preset function mentioned above.

Where should this design be used?

This counter design achieves high performance by using several logic tricks. It generates incorrect outputs when undigested carries sit in the carry flip-flops. That makes this design useless for any parallel application like DMA counters.

For the intended application, timebase counters or frequency synthesizers, this design offers the highest possible count speed.

AT&T Microelectronics

Asynchronous Preset in ATT3000 CLBs

Overview

The ATT3000 CLB can be reconfigured for asynchronous preset capability . 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 an FPGA. Note that the flip-flop will now come out of configuration in the apparent preset state.

2. If the circuit needs both asynchronous preset and clear, chances are that the function can be performed by a latch. The ATT3000 CLB can implement complex latches in its function generators (see page 5-33).

3. If the circuit really needs asynchronous preset and clear (or asynchronous data transfer) in a flip-flop, the problem must be solved on a system level.

The design can usually be transformed into a synchronous solution where all flip-flop changes occur as a result of the same clock edge.

Truly asynchronous parallel data transfer into several clocked flip-flops simultaneously is inherently unreliable and must be avoided. If, however, the transfer pulse is synchronized with the clock, it should not be too difficult to change the design to utilize the clock for loading.

AT&T Microelectronics

Frequency/Phase Comparator for Phase-Locked Loops

Description

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, and pump-down when it's too high. The multiple feedback network ensures 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 FPGA solution has been breadboarded at 10 MHz. It achieved a phase error of less than 2 ns.

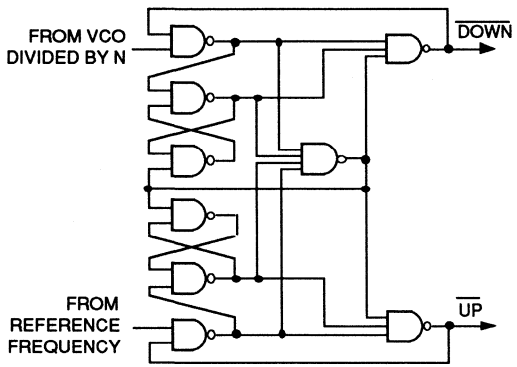


Figure 1. Digital Frequency/Phase Detector

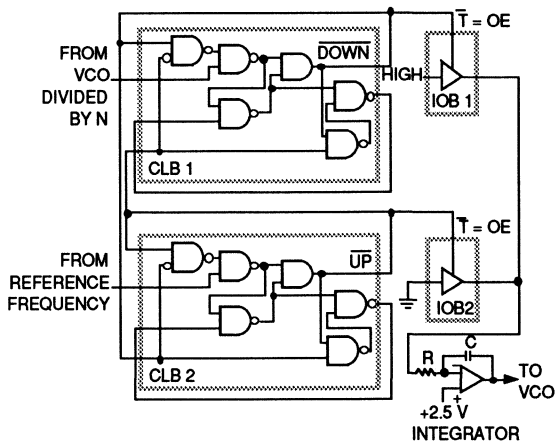


Figure 2. Frequency/Phase Detector Using Four Blocks

AT&T Microelectronics

Gigahertz Presettable Counter

Description

Some frequency synthesizers for communications, e.g., cellular telephone networks, require a clock frequency of hundreds of megahertz, up to a gigahertz. Obviously, the FPGA cannot operate quite that fast, but with the help of a two-modulus prescaler, the FPGA 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 two-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 FPGA) 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 FPGA 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-FPGA combination can divide by any integer number higher than the values above.

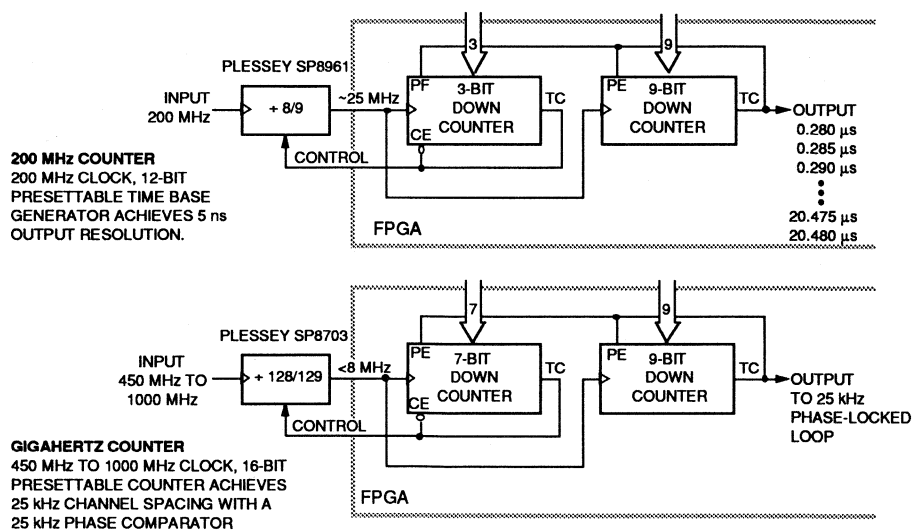


Figure 1. 200 MHz Counter and Gigahertz Counter

Description (continued)

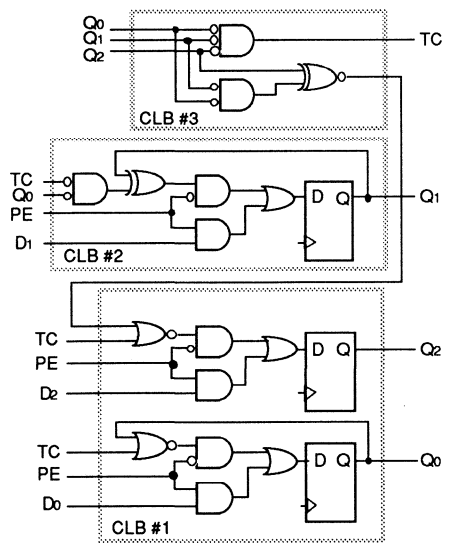
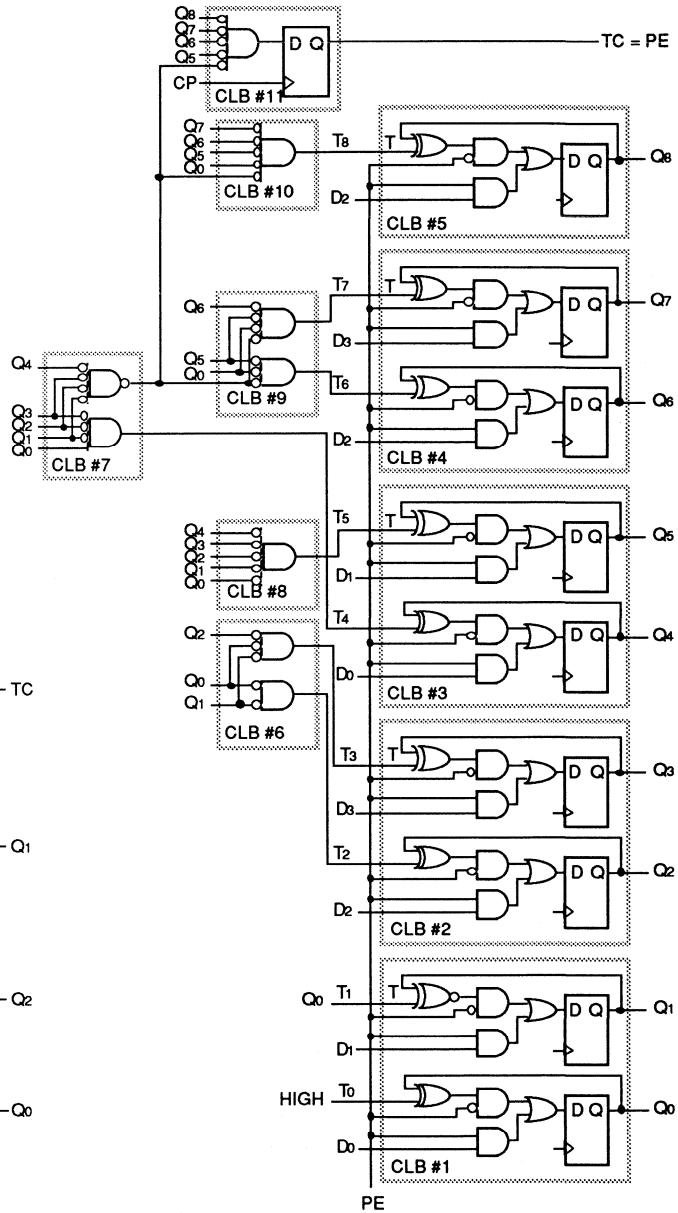


Figure 2. 3-Bit Presettable Down Counter with Pipelined Terminal Count, Locking Up on TC



Figre 3. 9-Bit Presettable Down Counter with Decoded Terminal Count

AT&T Microelectronics

83 MHz Presettable Counter or Programmable Delay

Description

The ÷8/9 prescaler described on page 5-45 can also be implemented inside an FPGA. The highest clock frequency for a 125 MHz part is 83 MHz, i.e., the output delay can be programmed with a granularity of 12 ns.

The implementation ÷8/9 prescaler comprises a simple three-stage binary counter followed by an extra bit. The ENABLE bit halts the 3-bit counter for one clock period in the all-ones state. The prescaler thus divides by 8 instead of 9. The MSB is stored inverted form so it can provide a rising clock edge at the end of the count sequence.

This design demonstrates the high performance possible with FPGAs when the user is willing to optimize the system design to fit the available logic.

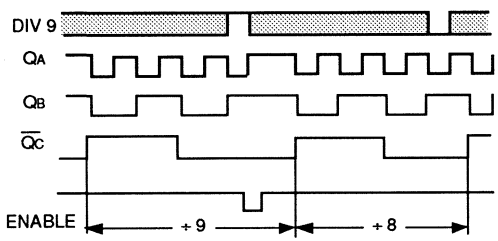


Figure 1. 83 MHz Presettable Counter Timing Diagram

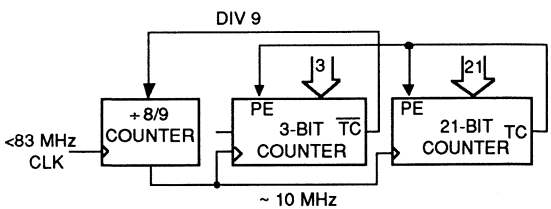


Figure 2. 24-Bit Frequency Division with Pulse-Swallowing Prescaler

The high clock resolution of 83 MHz is partly due to a system trick (pulse swallowing) and partly due to the inherent flexibility and high speed of the CLB function generators.

The prescaler can be implemented in two CLBs. One CLB contains Q_A and Q_C, and the other Q_B and the ENABLE bit. Q_A and ENABLE are the critical signals.

A conventional 24-bit presettable counter would be limited to a clock rate of 13 MHz. This pulse-swallowing design is more than six-times faster.

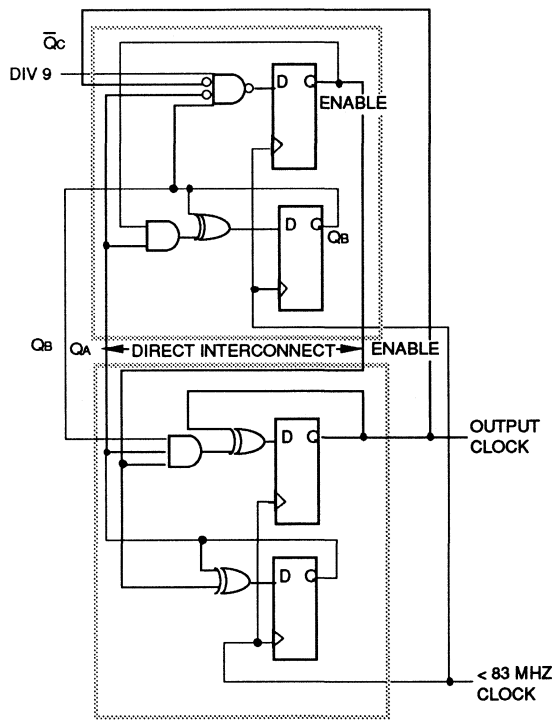


Figure 3. Divide by 8/9 Pulse-Swallowing Prescaler

AT&T Microelectronics

Serial Pattern Detectors

Fixed Pattern Detector

This circuit compares a serial bit stream against a predetermined (configured) pattern. Two bits are compared in each ATT3000-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 ATT3000-series-CLB per pattern bit.

The output of the comparators are ANDed with 3-state buffers on a long line. The desired pattern is first shifted through the DIN input into the Y flip-flop, and then routed to the DIN input of the next CLB.

When the complete pattern has been shifted in, it is transferred with one clock pulse to the X flip-flops, using the lower half of the function generator. Data to be detected is then shifted in through the DIN input into the Y flip-flop, and from there to the DIN input of the next CLB. The upper half of the function generator compares the content of Qx and Qy, and indicates a match on the CLB output. For identity comparison, these outputs are ANDed through 3-state buffers driving a long line.

This circuit can also be used as a correlator, in which case the outputs must be summed in a Wallace-type adder.

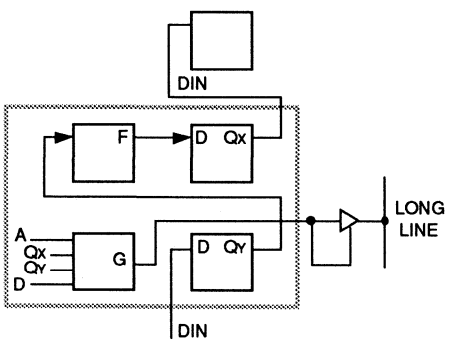


Figure 1. Fixed Pattern Detector

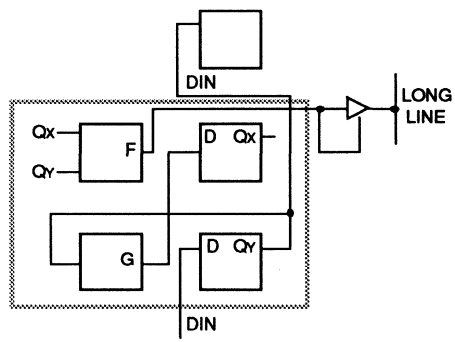


Figure 2. Serial Comparator Finds Pattern Match or Correlates Patterns

AT&T Microelectronics

Serial Code Conversion: Binary to BCD

Description

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

A binary-to-BCD converter requires three CLBs for every 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.

To remain a valid BCD number, a 4-bit number of 5 or greater must not just be shifted, but must be converted into the proper BCD representation of its doubled value. A one is shifted into the next higher decade, and the 5 is converted into a 0, a 6 into a 2, a 7 into a 4, an 8 into a 6, and 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.

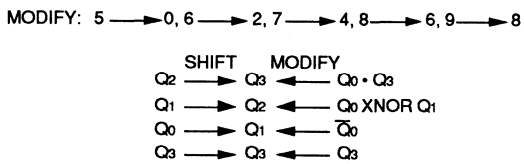


Figure 1. Binary to BCD Conversion Values

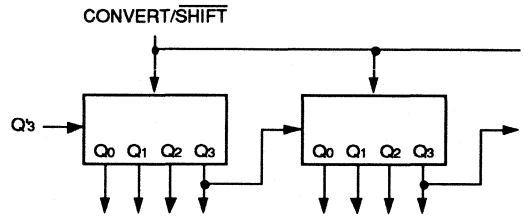


Figure 2. Binary to BCD (MSB First)

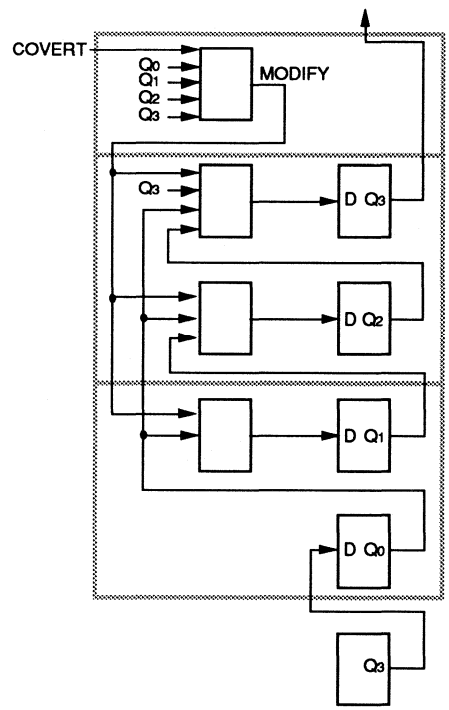


Figure 3. Binary to BCD Converter; Three CLBs per 4 Bits (MSB First)

AT&T Microelectronics

Serial Code Conversion: BCD to Binary

Description

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

A BCD-to-binary converter requires three CLBs per digit. BCD data is shifted in, least significant bit first. Once the complete BCD word has been shifted in, the conversion process begins, shifting out binary data, LSB first.

Each shift divides the content by two. When the LSB of a BCD digit is a one, shifting it one position down would give it a weight of 8 in the lower decade instead of the weight of 5 appropriate for a 10 divided by 2. A value of 3 is therefore subtracted from the content of the decade whenever a one is being shifted into it.

This design can be made smaller and faster by starting the conversion before the most significant BCD digit is being shifted in. Since these converters can be laid out with very short interconnect delays, they can operate at up to 60% of the specified toggle frequency, i.e., 42 MHz for the 70 MHz parts.

MODIFY: 0 → 5, 2 → 6, 4 → 7, 6 → 8, 8 → 9

SHIFT	MODIFY
$Q_1 \rightarrow Q_0$	$Q_0 \leftarrow \overline{Q_1}$
$Q_2 \rightarrow Q_1$	$Q_1 \leftarrow Q_1$ XOR Q_2
$Q_3 \rightarrow Q_2$	$Q_2 \leftarrow \overline{Q_3}$ AND ($\overline{Q_1}$ OR $\overline{Q_2}$)
$Q_0 \rightarrow Q_3$	$Q_3 \leftarrow Q_3$ OR ($Q_1 \cdot Q_2$)

Figure 1. BCD to Binary Conversion Values

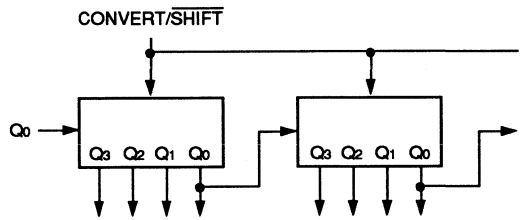


Figure 2. BCD to Binary (LSB First)

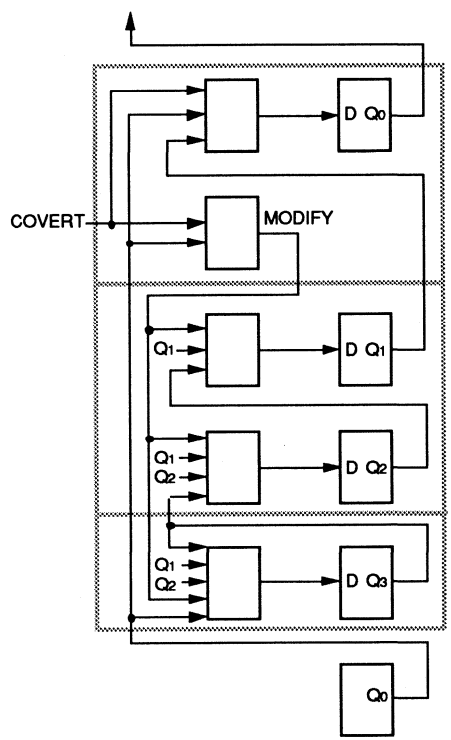


Figure 3. BCD to Binary Converter; Three CLBs per 4 Bits (LSB First)

Application Note
June 1992

AT&T Microelectronics

Corner Bender or 8-Bit Format Converter

Description

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 U.S. and Japan, A-law in the rest of the world using the CCITT standard.

These 8 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 8 bits representing one particular sample must be routed together. The telephone system thus uses a large number of serial-to-parallel and parallel-to-serial converters, all operating on 8-bit words, all running synchronously. Eight S-P converters with eight data inputs and eight data outputs can easily be combined in one package. Eight serial data streams are shifted in simultaneously. After eight clock pulses, the eight serial words can be shifted out in parallel, one word per clock pulse, and new serial bits can be shifted in simultaneously. It is interesting to note that the same circuit can also accept parallel words and shift them out in eight serial streams. The difference between S-P and P-S is not in the circuit, but in the mind of the beholder.

Such a corner bender is available as a standard part, the Plessey MJ 1410 8-Bit Format Converter. Its drawbacks are high power consumption (max 500 mW) and slow speed (2.4 MHz guaranteed worst case), a result of its NMOS heritage.

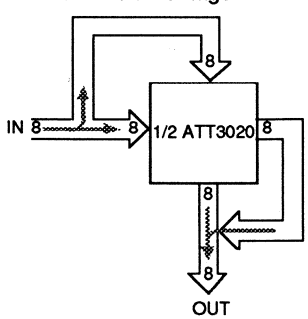


Figure 1. Corner Bender Converter

The FPGA implementation of a two-dimensional shift register is straightforward, as indicated in Figure 2.

A common clock drives all flip-flops, organized in an 8 x 8 array. In mode A, each flip-flop receives data from its left neighbor; in mode B, each flip-flop receives data from its neighbor above.

For the first eight clock pulses, the array is in mode A, receiving eight bit streams and right-shifting them into the array. For the next eight clock pulses, the array is in mode B, down-shifting the previously received 64 bits.

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

After eight clock pulses, the mode control is again changed to A and old data is shifted out on the right side while new data is shifted in from the left.

This design uses only 64 flip-flops and a mode-control signal derived from a divide-by-8 counter. The physical routing of the input signals can be done on-chip, but the eight bottom output pins can be externally combined with the eight righthand outputs in a wired OR.

The design fits exactly into one ATT2064 or into half of an ATT3020 and can run at up to 50 MHz.

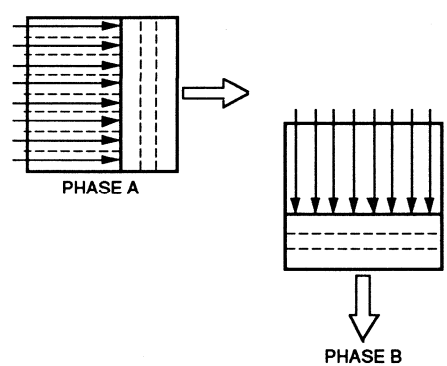


Figure 2. Two-Dimension Shift Register

Application Note
June 1992

AT&T Microelectronics

100 MHz Frequency Counter

Description

The block diagram below describes a complete 100 MHz frequency counter in an ATT3020 PC84.

A 32,768 kHz crystal oscillator generates a time base of two seconds. The frequency to be measured clocks an 8-digit BCD counter. At the end of the measuring period of two seconds, the counter content is transferred into four shift registers, and the counter is then reset before the beginning of the next measuring period. The shift register drives a seven-segment encoder that feeds into the LCD driving logic, which, in turn, drives seven 8-bit shift registers nestled in the IOBs.

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

The time base is generated by a 16-bit binary counter consisting of four asynchronously cascaded 4-bit synchronous counters. The control unit eliminates the clock ripple delay by resynchronizing 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 10 ns is achieved by using the divide-by-two flip-flop driven by the alternate clock buffer. This is the simplest and, therefore, fastest flip-flop on the device.

The whole frequency counter uses 60 of the 64 CLBs in an ATT3020:

Time Base	8 CLBs
BCD Counter	16 CLBs
Four Shift Registers	16 CLBs
Seven-Segment Encoder	4 CLBs
Leading Zero Suppressor	2 CLBs
Control	2 CLBs
Segment Conversion/ LCD Driving Logic	4 CLBs
Special Clock Generation	6 CLBs
Miscellaneous	2 CLBs

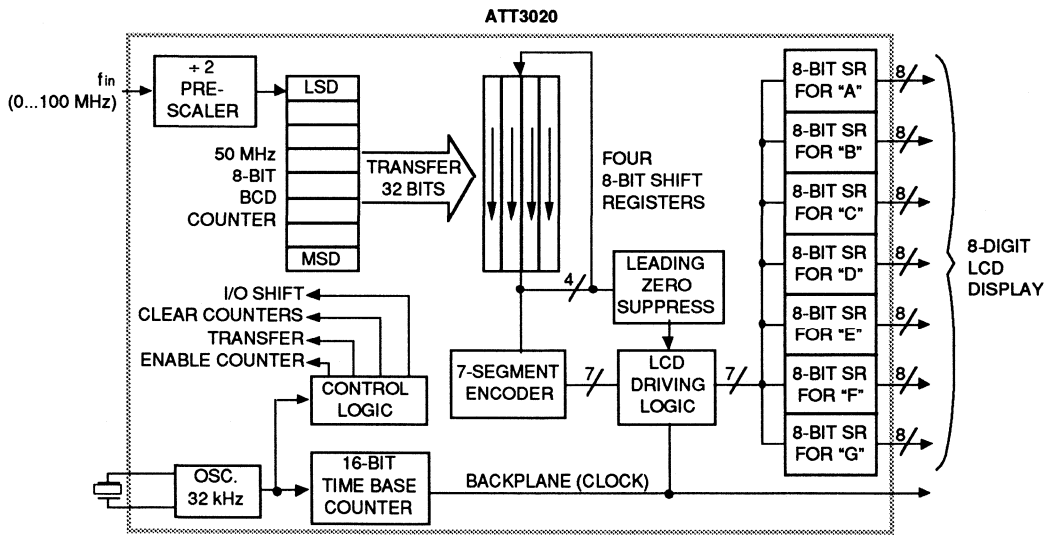


Figure 1. Block Diagram

Application Note
June 1992

AT&T Microelectronics

Megabit FIFO in Two Chips: One FPGA and One DRAM

Description

A bit-serial FIFO buffer is a general-purpose tool to relieve system bottlenecks, e.g., in LANs, in communications, and in the interface between computers and peripherals. Small FIFOs are usually designed as asynchronous shift registers, but a larger FIFO with more than 256 locations is better implemented as a controller plus a two-port RAM, or as a controller plus a single-port RAM, either SRAM or DRAM.

SRAMs are fast and easy to use, but at least four times more expensive than DRAMs of equivalent size. Dynamic RAMs offer low-cost data storage, but require complex timing and address multiplexing, which makes them unattractive in small designs. For FIFOs with more than 256K capacity, a DRAM offers the lowest-cost solution, if the controller can be implemented in a compact and cost-effective way. An ATT3020 FPGA can easily perform all of the control and addressing functions with many gates left over for additional features.

This FIFO DRAM controller consists of:

- An input/output buffer with synchronizing logic
- A 20-bit write pointer (counter)
- A 20-bit read pointer (counter)
- A 20-bit full/empty comparator
- A 4-to-1, 10-bit address multiplexer
- Control and arbitration logic

The write pointer defines the memory location where the incoming data is being written, and the read pointer defines the memory location where the next data can be read. The identity comparator signals when the FIFO is full or empty.

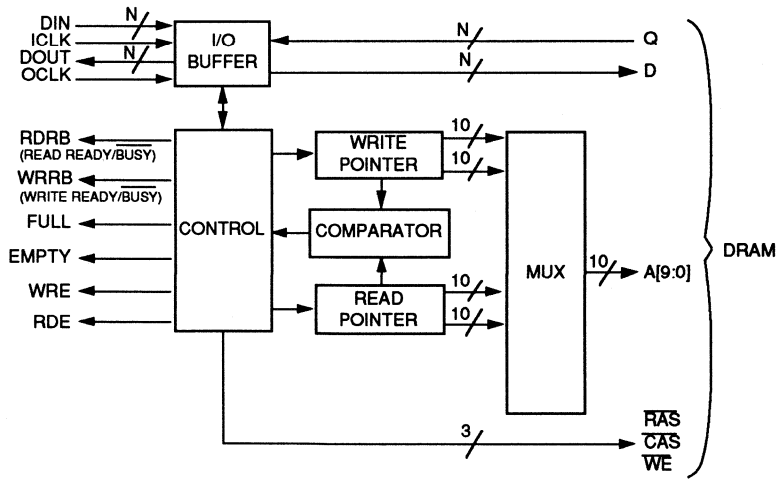


Figure 1. Megabit FIFO Controller and an ATT3020

Description (continued)

When the write and read pointers become identical as a result of a write operation, the FIFO is full, and further write operation must be prevented until data has been read out. When the two pointers become identical as a result of a read operation, the FIFO is empty and further read operation must be prevented until new data has been written in. With a single-port RAM, read and write operations must be inherently sequential, and there is no danger of confusing the full and empty state, a problem that has plagued some two-port designs.

A straightforward design would use synchronous binary counters for the two pointers, but it is far more efficient to use linear shift-register (LSR) counters. Such counters require far less logic and are faster since they avoid the carry propagation problems of binary counters. LSR counters have two peculiarities: they count in a pseudorandom 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 of these features are irrelevant, and the address sequence is arbitrary, provided both counter sequence identically.

This design fits two shift register counter bits in one ATT3000-series CLB, and the identity comparator uses the combinatorial portion of the same CLB.

The RAS/CAS multiplexing of the 20-bit address is performed without any logic by tapping every other bit of the shift register counter and using the ten 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 pseudorandom anyway, this is no problem. It's an elegant and efficient trick.)

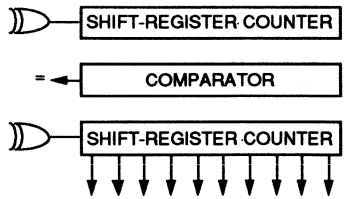


Figure 2. Shift-Register Counter and Free Row-Column MUX

The FIFO controller permits the user to perform totally asynchronous read and write operations, while it synchronizes communication with the DRAM. The design takes advantage of the DRAM internal refresh counter by using CAS-before-RAS refresh/address strobes.

Both 20-bit pointers, plus their 20-bit identity comparator, plus the row/column multiplexer, thus fit into only 20 CLBs, refresh timer and address multiplexer use another ten CLBs, and the data buffer plus control and arbitration logic take another 23 CLBs, for a total of 53 CLBs, an easy fit in an ATT3020.

This design can easily be modified for 256K DRAMs. Other variations are multiple parallel bits, e.g., byte-parallel operation, interrupt-driven control, multiplexed data for multiple parallel-bit storage, and byte parallel storage with bit-serial I/O. The latter case requires special attention when the FIFO is emptied after a noninteger number of bytes had been entered, requiring direct communication between the input serial-to-parallel converter and the output P/S converter.

This applications note shows that the ATT3020 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 Mbytes/s serially or 2 Mbytes/s byte-parallel.

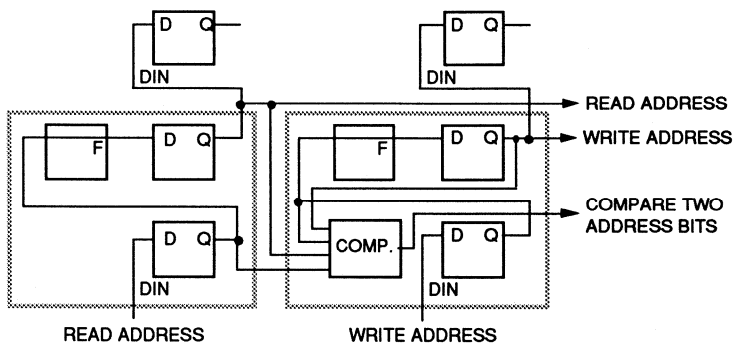


Figure 3. 2-Bit Slice of Two Counters and Comparator In Two CLBs

AT&T Microelectronics

State Machines

Description

State-machine design is a methodology that defines the contents of all flip-flops for any possible state of the design, and 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. FPGA devices with their abundance of flip-flops lend themselves well to state-machine designs.

Simple, Fast State Machines

Using the five-input function generator of the ATT3000-70 family devices as a 32-bit ROM, a state machine with up to 32 states without any conditional jumps uses only five CLBs and operates at up to 50 MHz.

The five registered CLB outputs drive the five function generator inputs of the five CLBs in parallel. This implements a fully programmable sequencer.

For a smaller number of states, some inputs can be used as conditional jump inputs. Encoding these condition codes may require an additional level of logic which reduces the maximum clock rate to 30 MHz.

Simple State Machine Runs at 30 MHz

This simple state machine uses only 11 CLBs. It has up to 16 states, and eight outputs, each decoding/encoding any combination of states. It performs a two-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 eight-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, or else go to 8.
- From state 7, if C = high, go to 3, or else stay in 7.
- From state 9, unconditionally go to 2.
- From state 6, execute the truth table below.

Table 1. Truth Table

AB	C = Low	C = High
00	12	0
10	13	1
01	14	2
11	15	3

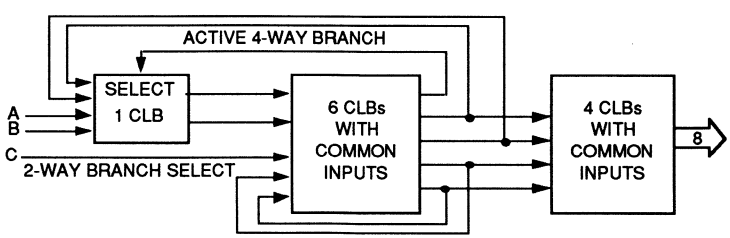


Figure 1. 30 MHz State Machine, 16 States, 2-Way/8-Way Branch, 8 Outputs

AT&T Microelectronics

Complex State Machine in One FPGA

Description

Simple and fast state machines can easily be implemented in an FPGA, as shown on the previous page. However, 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 FPGA configuration data.

Assume that an ATT3020 is configured in the master parallel mode, where it reads its configuration data out of a 256K (32K x 8) EPROM, starting at the top address location 7FFF (32K) through 77FF (about 30K). The remaining 94% of the EPROM can be used as a next-state look-up table with a capacity of 240 states.

The state address is read out of the EPROM, and then manipulated (decoded, encoded, etc.) in the ATT3020. 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 FPGA 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 FPGA just IOBs; but it allows a number of states and a multiway branch complexity far in excess of any normal need. The user will usually reduce the multiway branch complexity by assigning identical values to many of the 128 possible next states.

The user has the logic resources of the FPGA available to add features like:

- State decoding/encoding
- Stack registers
- Loop counters
- More sophisticated branch logic, etc.

This design is straightforward, inexpensive, compact, and very flexible. Its speed is limited by the EPROM access time, which can be less than 100 ns. For higher speed—at a higher cost—the EPROM can be shadowed by fast SRAMs.

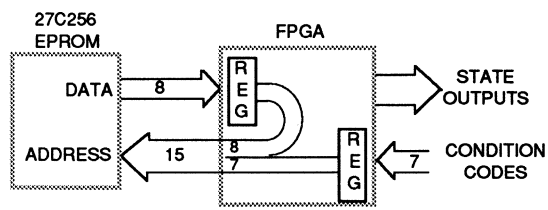


Figure 1. Complex State Machine In One FPGA

AT&T Microelectronics

PS/2 Micro Channel Interface

Description

IBM's general-purpose microcomputer, the *Personal System/2,* is available in several models, from the low-end Model 25 to the high-end Model 80. These third-generation PCs have several innovative features, including 3-1/2 inch floppy-disk drives, high-resolution VGA graphics, and a 20 MHz 80386 processor as the main engine for the Model 80. Among the most interesting features is the *Micro Channel* interface, the bus specification for the interface between the system and adapter cards. The *Micro Channel*'s streamlined characteristics and flexibility provide *PS/2* designers and users with many advantages over previous PC architectures.

One key aspect of this architecture is the ability to configure the system without the need for DIP switches on the bus adapter cards.

Defined with system configuration utilities, an add-on card's addressing and other optional configuration data are established and stored in CMOS battery-backed memory on the main board. Upon powerup, this information is loaded into programmable option select (POS) registers residing on the adapter cards.

Figure 1 indicates one way in which a logic cell array can be used for the POS-register section of a *Micro Channel* adapter card. The *Micro Channel* interface includes logic to decode the address, status, and control signals associated with the bus to identify the appropriate POS register to be accessed. These signals determine if the card is being addressed, and whether the current operation is a read or write.

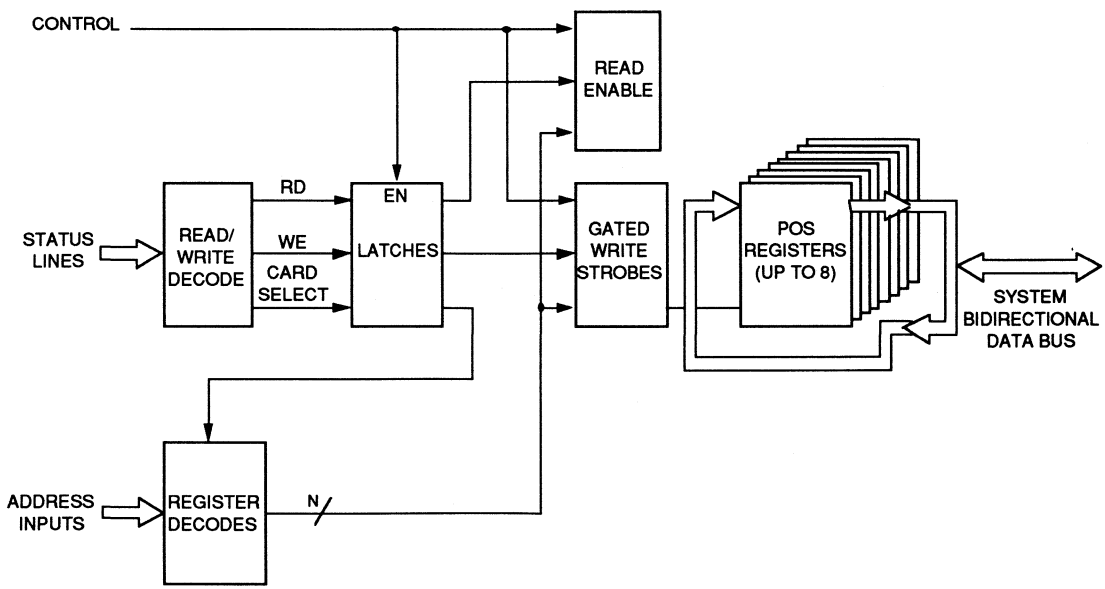


Figure 1. *Micro Channel* Interface Block Diagram

Description (continued)

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 FPGA devices.

Implementation of the POS registers, arbitration, logic, and control sections typically requires only 1/3 to 2/3 of a single ATT3020; the remainder of the FPGA is available for implementing the unique functionality of the specific adapter card.

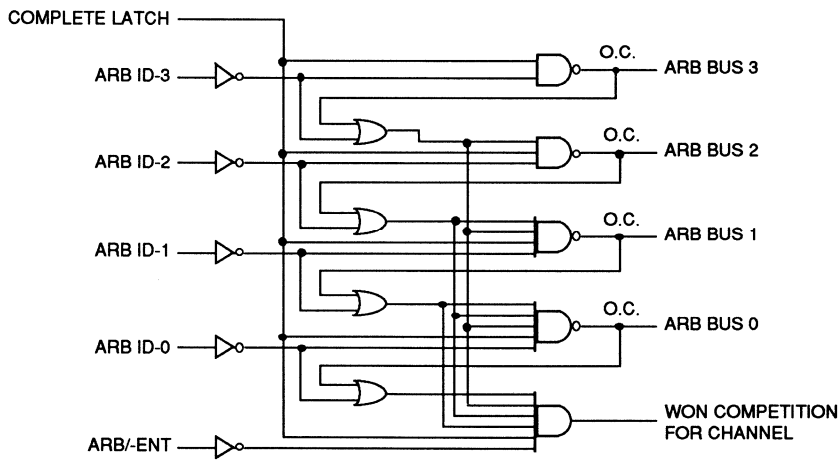


Figure 2. Local Arbitrator Logic

Application Note
June 1992

AT&T Microelectronics

DRAM Controller with Error Correction and Detection

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 ms 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.

Options For DRAM Controller Design (continued)

The need to match the DRAM controller to the specific requirements of the system has forced many engineers to consider two options for creating their own controllers: SSI/MSI packages or custom gate arrays. The use of SSI/MSI is low risk, but wastes space and power, while the use of the custom gate array provides a highly integrated solution, but at considerable risk and expense. Nonrecurring engineering costs (NRE), testing and simulation costs, inventory risk, and a long design cycle make the custom gate array option unattractive for most designs. Recent architectural advances in high-density field-programmable logic arrays have created a third option. AT&T's 3000 family of FPGAs brings unprecedented density to programmable logic, with devices containing as many as 9000 gates. The 3000-family architecture makes the devices particularly well-suited to memory-controller applications.

Implementing a DRAM Controller with a Field-Programmable Gate Array

There are several reasons why one would want to design a DRAM controller with a logic cell array. First, the true programmability of the FPGA device gives the designer the freedom to design the DRAM controller to the exact specifications of the memory system. There is no need for the external glue logic often necessary with standard solutions because any necessary design tweaking is implemented internally.

The FPGA 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 FPGA s makes it possible to implement DRAM control and error detection/correction in a single FPGA device. This is traditionally a two-chip solution using standard parts: a DRAM controller and a separate error correction and detection unit. It can, of course, be implemented in a single custom gate array, but again with the earlier caveats. Finally, the CMOS FPGA 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 FPGA . The example is an 8 MHz 8086-based system that directly addresses 1 Mbyte of memory comprising 44 256K DRAMs: 32 for data and 12 for the correction bits. A single FPGA device can serve as both the DRAM controller and the ECC, which performs single-bit error correction and double-bit error detection. There are several features of the 3000-family architecture that make this design possible. These include five input-configurable logic blocks (CLBs) with two storage elements, internal buses, and flexible input/output blocks (IOBs).

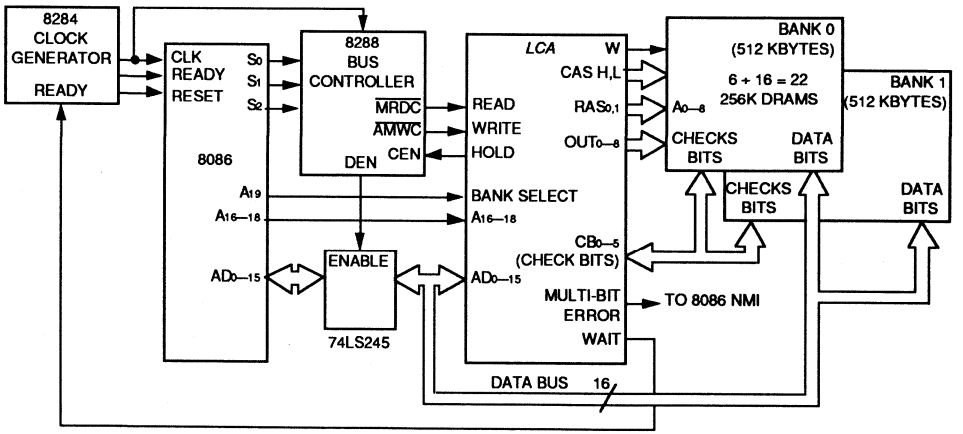


Figure 1. System Overview of DRAM Controller with Error Correction and Detection

Design Overview

The DRAM Controller/ECC uses a 16 MHz clock synchronized with the processor clock, and sits between the 8086 microprocessor with its 8288 bus controller and the system memory (Figure 1). The 8288 decodes the processor status lines (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 FPGAs.) The DRAM controller then performs the appropriate access issuing row address strobe (RAS), column address strobe (CAS), and write, if necessary. The error checker and corrector generates check bits on each write, and checks for and corrects errors on each read. The controller also signals the 8086 if the memory access requires a wait-state or if a non-correctable error is detected.

System Timing

Figures 2a—2c show the timing involved in some of the different memory cycles. The word write (Figure 2a) requires no wait-states as shown. The check bits from the ECC are written to memory along with the data. The read cycle (Figures 2b & 2c) requires a minimum of one wait-state. The insertion of a wait-state is unavoidable because of the time it takes the 120 ns DRAMs to output the data. If the ECC detects no errors in the data, the WAIT signal is released and the read operation is completed. If an error is detected, the insertion of two more wait-states is required to provide time to correct the error. The insertion of the two additional wait-states affects system performance, but this is the trade-off for having error correction, which avoids the fatal system errors that occur with parity-checking-only solutions.

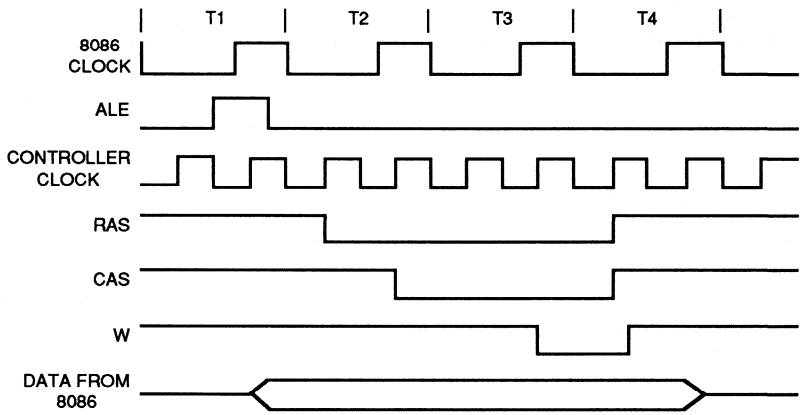


Figure 2A. Word Write Timing

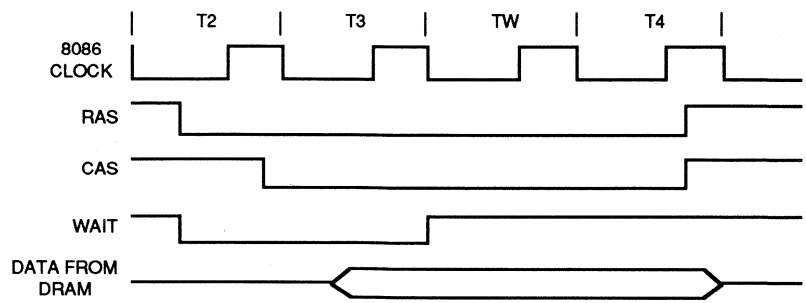


Figure 2B. Word Read Timing with Errors Detected

System Timing (continued)

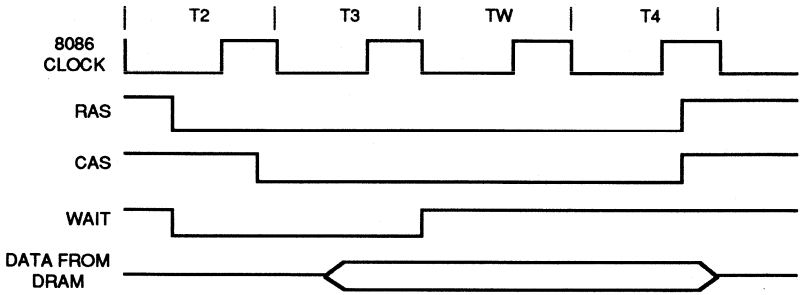


Figure 2C. Word Read Timing with No Errors Detected

Design Focus

The 3000-family architecture has a number of features that are essential to the DRAM controller design. The first such feature is the dual data input paths in the IOBs, one registered and one direct. This structure permits the address and data on a multiplexed bus to be latched from the same I/O pin. Figure 3 is a bit-sliced view of an IOB used to latch the multiplexed address/data bus. In this design, the address is latched into the IOB input flip-flop with the 8086 ALE. The data from the 8086 can enter the same input pin and go directly to the ECC circuit via the IOB direct input—there is no need for external latches.

Another feature of the IOBs is the output flip-flops with 3-state buffer enables. This feature permits bit-error correction using only one I/O pin. Figure 4 shows a bit-sliced view of how the ECC is accomplished. A memory read cycle provides the best example for showing the capabilities of the IOB structure. During a read, the IOB output is 3-stated, permitting the DRAM data on the data bus to enter the ECC via the IOB direct input. If the ECC detects a data bit error, it corrects the error and latches the corrected data word into the output flip-flops of the IOBs. The data bus is then 3-stated by turning off the DRAM outputs. The corrected word, latched in the outputs of the flip-flops, is then released onto the data bus by enabling the 3-state buffer. This permits the corrected data to be read by the 8086 at the same time it is being written back to the DRAM.

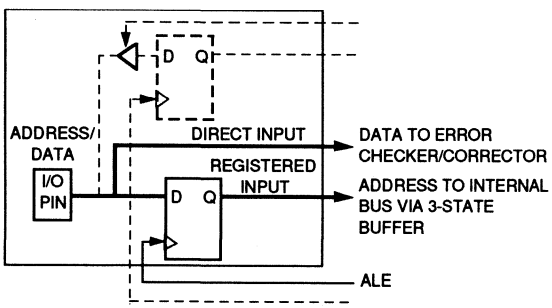


Figure 3. Address and Data Latching

Latching data off a multiplexed address and data bus: the input/output block configuration shown above illustrates how the direct and registered inputs in the IOBs can be used to latch a multiplexed address/data bus into the FPGA . The address is latched into the IOB flip-flop; the data flows directly into the ECC logic.

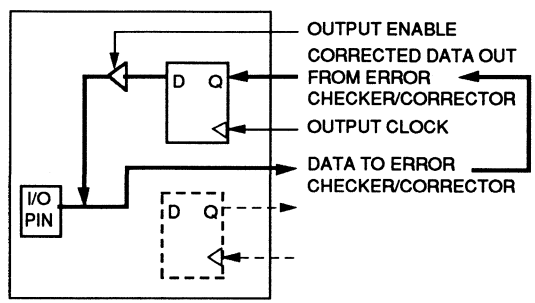


Figure 4. Data In and Out Through ECC

The data from the bus goes into the FPGA, where it is corrected in the ECC. The corrected data is then put back onto the bus via the IOB output flip-flop.

Design Focus (continued)

Figure 5 is a block level diagram of the DRAM controller and ECC that reside in the FPGA. A functional description of each block follows.

The refresh timer is driven by the 16 MHz clock to provide a signal that tells the DRAM controller that the memory needs refreshing. Each of the 256 rows of memory in this system must be refreshed every 4 ms. The controller attempts to refresh eight rows every 125 µs, so that all 256 rows are refreshed in 4 ms. The refreshing technique employed in this design is a unique combination of burst and hidden refreshing to show the flexibility of the FPGA-based solution. There is no need to force a system to conform to the constraints of an off-the-shelf part. The hidden refresh is performed when the 8086 is doing a read from or write to somewhere other than memory, like an I/O port. This involves giving the DRAM a refresh address from the refresh address counter via the address selector and a RAS pulse low from the timing generator. The burst refresh is performed only if it has not received its eight required refreshes during the 125 µs refresh period. When a burst refresh is required, the controller will isolate the memory from the 8086, insert wait-states, and provide the number of refreshes it needs to complete the eight refreshes required during the refresh period.

The timing generator, a state machine triggered by address latch enable (ALE) at the beginning of the processor cycle, produces all the timing required to perform the memory accesses and refreshes. The signals generated by this block include the row address and column address strobes (RAS and CAS), the WRITE signal, the WAIT-state signal for the processor, the HOLD signal that isolates the processor from the memory, the clock for the refresh address counter, and the select control for the address select.

The refresh address counter is an 8-bit counter that provides the 8-bit addresses necessary to refresh the DRAMs.

The address selector selects which address is sent to the DRAM. During a read or write cycle, the timing generator select control signal tells the address selector to select the DRAM row address, strobe it with the RAS, and then select the column address and strobe it with the CAS. During a refresh, the address selector selects the address from the refresh address selector and strobes it into the DRAM with RAS.

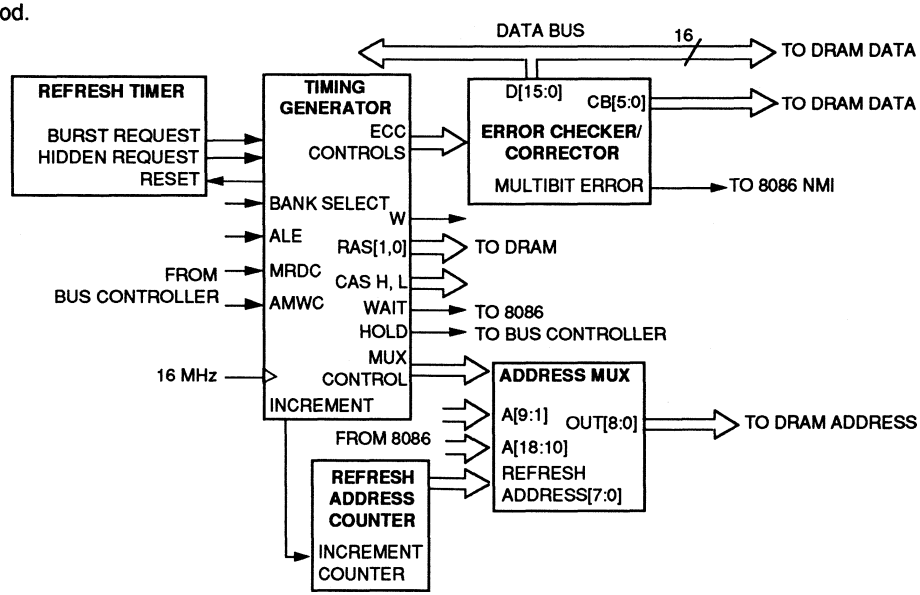


Figure 5. Block Diagram of the DRAM Controller Functions Implemented in the FPGA

Design Focus (continued)

During a write cycle, the error checker/corrector (ECC) generates six check bits using a modified Hamming code for each 16-bit data word and writes them to memory along with the data. Use of a modified Hamming code permits single-bit data correction and double-bit error detection. During a read cycle, the ECC compares the check bits read back from memory with new check bits generated from the data read back. If the comparison yields a correctable error, the ECC will correct it. If the error is not correctable, it will flag the NMI on the processor.

Perhaps the most important feature of the FPGA architecture for implementing a DRAM controller is its internal 3-state bus capability. The 34-state buffer enables onto the horizontal long lines allow the designer to implement an internal bus in the FPGA. 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 3-state enables, only one address is allowed onto the bus at a time.

This feature is essential to this design and has many other applications including performing wired-AND functions and address decoding.

Conclusion

Although the bottom-up design of a DRAM controller is a complex task, it is necessary in cases in which off-the-shelf controllers do not meet the requirements of the system. SSI/MSI and custom gate array solutions involve trade-offs and compromises. Designing a DRAM controller and ECC with an FPGA is a straightforward application and a good fit for the 3000- family architecture. The field-programmable gate array offers the flexibility necessary to match the many different memory systems, the integration desirable for board-level designs today, and the cost effectiveness required to make a competitive product.

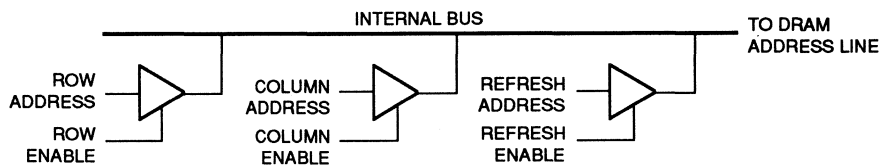


Figure 6. Address Multiplexing Using 3-State Enables onto Internal Buses

Section 6.

Migration to Gate Arrays

Section 6. Migration to Gate Arrays

Table of Contents

Contents **Page**

Application Note
July 1992

AT&T Microelectronics

SoftPath FPGA to Gate Array Conversion (FPGA2GA)

Contents

Introduction

This application note introduces the field-programmable gate array (FPGA) to gate array conversion process. It is written for designers who are converting an existing FPGA to a gate array and for designers who anticipate that an FPGA design may migrate to a gate array.

The AT&T tool set used for design conversion is known as *SoftPath*. The focus of this application note is on the *SoftPath* tool FPGA2GA, which allows a designer to do a direct conversion from an AT&T (or *Xilinx*) FPGA to an AT&T gate array.

Tables 1 and 2 provide a list of the AT&T 3000 Series FPGA and AT&T 656 Series Gate Array families. The ATT3000 Series provides FPGAs ranging from 2000—9000 total gates while the ATT656 Series includes gate arrays ranging from 5K—177K gates. The two-level metal gate arrays are targeted for a conversion of one FPGA into one gate array.

The AT&T FPGA and gate array families are both manufactured using submicron, Si-gate CMOS technology. Both products are second-sourced and supported by a variety of AT&T and third-party design tools.

The CMOS process is also used for AT&T standard-cell ASICs. In addition to the tools discussed in this application note, AT&T offers a tool which converts from FPGAs to AT&T standard cells.

Table 1. AT&T FPGA Family

Part Number	Total Gates	Useable Gates	I/Os
ATT3020	2000	1600	64
ATT3030	3000	2400	80
ATT3042	4200	3360	96
ATT3064	6400	5120	120
ATT3090	9000	7200	144

Table 2. Subset of AT&T Gate Array Family

Part Number	Total Gates	Useable Gates	I/Os
ATT65630	5376	2420	84
ATT65636	8000	3600	100
ATT65640	11520	5180	120
ATT65646	16240	7300	140
ATT65650	21120	9500	160
ATT65654	30720	13800	192

Overview

How to Use This Application Note

This application note provides an overview of the process of converting from one FPGA to one gate array. The related set of documentation listed below (with the order number) focuses specifically on FPGA and gate array design concerns.

- *ATT656 Series CMOS Gate Arrays* Data Sheet (PN92-035GATE)
- *AT&T CMOS Gate Array Design Guide*
- *ATT656 Series CMOS Gate Arrays Cell Library*
- *ATT3000 Series Field Programmable Gate Arrays* Data Sheet (DS90-176CMOS)
- *ATT FPGAs Automatic CAE Tools Product Overview* (PN90-013CMOS)
- *ATT FPGA2GA User's Guide*
- *FPGA Migration*

Functional Description

FPGA2GA Overview

The FPGA2GA tool allows a designer to obtain the fast prototype time obtained from FPGAs and the low unit cost of gate arrays. It provides a direct conversion of an FPGA to a gate array to avoid board redesign. FPGA2GA automatically generates a functionally equivalent gate array netlist from an FPGA netlist. It is easy to use, requiring only a single line command. In the conversion process, gate count is minimized. The PWR/GND pins are mapped to footprint-compatible packages.

ASIC Design Methodology Economic Considerations

ASICs are currently developed using four design methodologies: PLDs, FPGAs, gate arrays, and standard cells. The design methodologies trade time to market, nonrecurring expense (NRE), and development risk for unit cost and speed. To minimize time-to-market and development cost and risk, a project may initially prototype using FPGAs and later migrate to gate arrays or standard cells. With the increasing effectiveness of technology migration tools, designers can determine the most efficient ASIC based on the phase of the product's life cycle.

The primary reason for converting from a single FPGA to a single gate array (or standard cell) is to attain a lower production unit cost. The break-even unit quantity for using an FPGA, gate array (GA), or standard-cell (SC) ASIC is determined from:

$$\text{units} * \text{unit\_cost}_{FPGA} = \text{NRE}_{GA} + \text{units} * \text{unit\_cost}_{GA} = \text{NRE}_{SC} + \text{units} * \text{unit\_cost}_{SC}$$

The break-even unit quantity between an FPGA and gate array, then, is:

$$\text{break-even units}_{GA} = \frac{\text{NRE}_{GA}}{\text{unit\_cost}_{FPGA} - \text{unit\_cost}_{GA}}$$

Similarly, the break-even quantity between FPGA and standard cell is:

$$\text{break-even units}_{SC} = \frac{\text{NRE}_{SC}}{\text{unit\_cost}_{FPGA} - \text{unit\_cost}_{SC}}$$

The cost of an engineer to convert the design needs to be factored into the NRE. This includes the labor to generate the test vectors and simulate the gate array. This cost is affected by the FPGA and gate array design tools available and the learning curve or experience of the designer. Other NRE factors include mask, wafer fab, and test program development costs.

Although the same AT&T and third-party software tools can be used to develop AT&T FPGAs and gate arrays (and standard cells), this often is not done. Most FPGAs have been developed using relatively inexpensive PC-based design tools. Gate arrays are almost always developed using more expensive engineering workstations.

As with standard gate array developments, AT&T can provide a turnkey conversion design service, or any level of support needed to aid the customer through the conversion process. The level of support required from the AT&T design center may depend on the availability of workstation and design tools used in the FPGA and gate array design tools.

An FPGA designer using PC-based design tools, but unfamiliar with engineering workstation base tools or *UNIX*, may need assistance from an AT&T design center. An FPGA designer who is also an experienced gate array designer may not need much assistance.

FPGA and Gate Array Architecture

Both FPGAs and gate arrays are logic arrays and are prediffused and inventoried in an advance (or completed) state of manufacturing. This is done in order to reduce or eliminate lead time and NRE.

The functional elements in the FPGA and gate array are connected differently, and it is useful to understand the differences in how the functional elements implement logic.

The FPGA consists of two major blocks shown in Figures 1 and 2: configurable logic blocks (CLBs) and input/output blocks (IOBs). The IOBs reside at the perimeter of the FPGA and provide an interface between the bond pads and the FPGA's internal logic. The internal logic is done by CLBs arranged in a matrix. For example, the 3090 consists of a 20 x 16 array of CLBs surrounded by 144 IOBs.

Functional Description (continued)

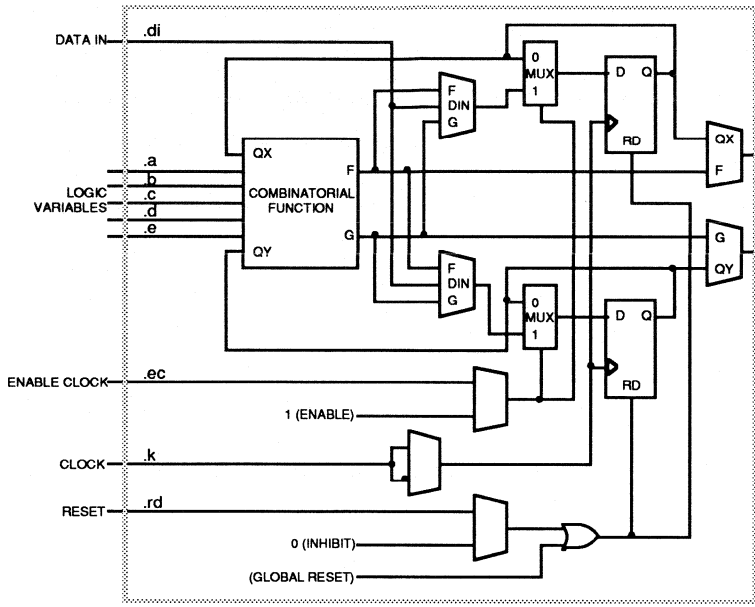


Figure 1. Configurable Logic Block Diagram

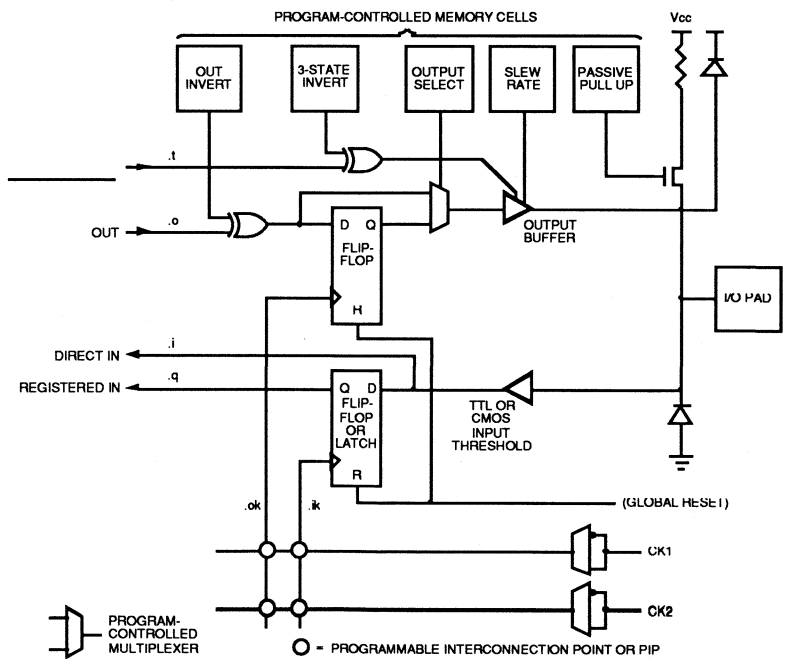


Figure 2. Input/Output Block Diagram

Functional Description (continued)

The interconnection of the functional blocks in the FPGA uses three types of metal interconnections: direct, general-purpose, and long-line interconnections. The routing paths are determined by configuring programmable interconnect points (PIPs) and switch matrices to make user-defined connections. There is static random access memory (SRAM) resident on the FPGA which controls the state of the switches (pass transistors) used in interconnections. This architecture allows the FPGA interconnections to be reconfigured by the user in minutes.

The gate array architecture consists of the unconnected transistor pairs shown in Figure 3. The logic is predominantly implemented using unconnected p-channel and n-channel devices. In addition to NAND/NOR based logic, the transistor pairs can be configured as RAM and/or ROM. In contrast with the FPGA logic blocks, in which much of the circuitry internal to CLBs is already connected, the gate array architecture provides a much lower level of granularity.

A lower level of granularity provides higher flexibility, but may require more routing resources to realize that flexibility. While the uncommitted devices in the gate array can be connected to realize almost any logic function, the flip-flop (FF) in the CLB is a dedicated FF function. If used, the FPGA FF is more area-efficient than an equivalent FF in the gate array flip-flop. If not used, the area is wasted.

In the FPGA, combinatorial logic is implemented in a RAM-based look-up table in each CLB. Each look-up table can implement any function of up to five variables, any two functions of up to four variables, and so on. The look-up table provides predictable delays for two-level functions of this width. If a combinatorial logic function is wider than five variables, or deeper than two levels, multiple CLBs are required. In this case, the speed is a strong function of the number of pass transistors and the length of metal interconnection between the CLBs.

In the direct conversion of one FPGA into a gate array, a two-level metal FPGA is converted into a two-level metal gate array. Higher-density gate arrays use three-level metal which generally achieves a percentage utilization which is 50% higher than two-level metal. Also, the ATT656 Series gate array supports single- and dual-ported RAM and ROMs.

A difference between FPGAs and gate arrays is that FPGA resident SRAM controls the FPGA's functionality. In a system, this configuration memory is loaded from memory resident on the printed-circuit board (PCB). Since the FPGA memory is SRAM, the FPGA needs to be reconfigured when power is lost. With a gate array, the metal connections are fixed, so there is no PCB memory requirement for this function.

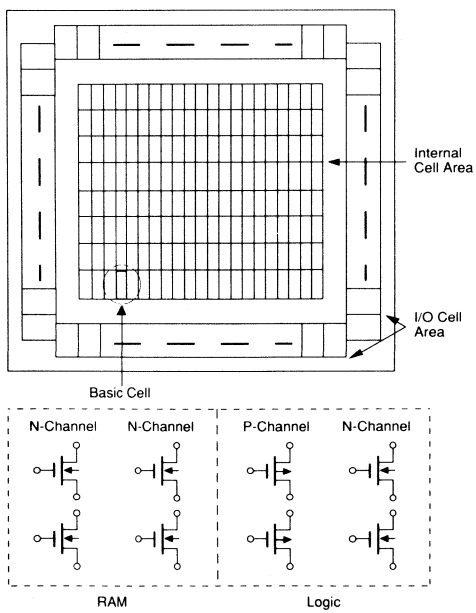


Figure 3. Gate Array Functional Element

Functional Description (continued)

Design Flow and Responsibilities

In converting from an FPGA to a gate array, the designer needs to do the following tasks:

- Select the appropriate gate array.
- Understand the differences between FPGA and gate array development tools.
- Use the gate array design tools to ensure correct operation of the gate array.
- Perform system analysis to make certain that the gate array does not cause problems when inserted in the FPGA socket.

There are several differences in FPGA and gate array product developments. In an FPGA product development, the design and design verification are done completely by the designer. The designer uses the AT&T development software to verify the FPGA operation in the system. This can be done in minutes/hours with minimal interaction with the FPGA supplier. Figure 4 shows the basic development tools used to design an ASIC using an FPGA.

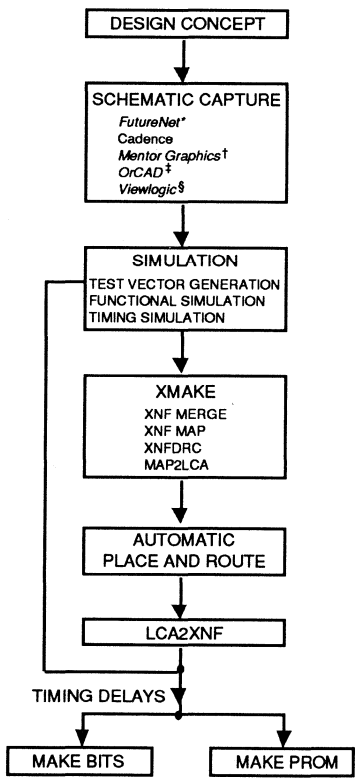


Figure 4. FPGA ASIC Development

In a gate array development, a sharing of responsibilities between the customer and AT&T is required to complete the gate array development. Because a gate array is not reconfigurable, design iterations are relatively expensive and time consuming when compared to FPGAs.

After the FPGA netlist is translated into a gate array netlist, the process of converting an FPGA to a gate array follows the basic gate array development flow. The designer generates test vectors, does a functional and timing simulation, and provides AT&T with a netlist. AT&T is responsible for the layout, layout verification, test program development, manufacturing, packaging, and testing of the gate array. The designer performs in-system verification prior to release of the gate array to full production.

Table 3 provides a detailed list of the sequence of tasks and responsibilities in the FPGA to gate array conversion.

The customer generates input stimulus waveforms for logic simulation. In this phase, tester requirements (discussed later in this application note) should be considered. The prelayout timing analysis consists of functional simulation under various environmental and processing conditions, assuming an autoroute factor to compensate for the routing delays. Optional audits can be run to determine if there are setup and hold violations, race conditions, and critical paths. The customer provides simulation output files for AT&T design review. This consists of input stimulus patterns and simulation results for best, typical, and worst-case environmental conditions.

After layout, AT&T runs layout vs. schematic (lvs), and layout parasitic extraction (LPE), and also provides the customer with the delay files for postlayout simulation. AT&T can run fault grading on the test patterns. In addition to pattern driven simulation, the customer can run static timing analysis to verify that critical paths are met. The customer then provides AT&T with the postlayout simulation files. AT&T extracts the test patterns from the simulation files, converts the test patterns to the format required by the automatic test equipment (ATE), and generates the test program.

Functional Description (continued)

Table 3. Schedule of Tasks and Responsibilities for FPGA2GA

Schedule of Tasks	Responsibility*
General Consultation	A
Feasibility Analysis	C
FPGA2GA Resynthesis	C
Electrical Rules Check	C
Develop Test Vectors	C
Functional Simulation	C
Timing Simulation	C
Submit Input Data	C
Review/Accept Input Data	A
Prelayout Design (floor plan, vector processing, and multiple delay at 10 MHz)	A
Place and Route	A
Postlayout Design (multiple delay with RC @ 10 MHz)	B
Resimulate with Annotated Wiring Delays (at system speed)	C
Customer Release 1	C
Mask Preparation	A
Design Review	B
Mask Generation	A
Wafer and Package Test Program Development	A
Fabricate, Package, Test, and Ship Prototypes (10 typical)	A
Customer Evaluation and Approval of Prototypes	C
Transfer to Manufacturing	A

* A = AT&T; C = Customer; B = Both.

Technology Resynthesis

SoftPath is an AT&T capability to resynthesize ASICs between FPGA, gate array, and standard-cell ASICs. AT&T supports a variety of third-party and AT&T proprietary tools aimed at resynthesizing and retargeting ASIC designs into FPGA, gate array, and standard-cell design methodologies. To avoid confusion in the terminology used by different CAE and ASIC vendors, the following defines the various translation functions.

Technology Mapping: Generic term covering all translations of designs from one ASIC to a second ASIC. The translation can be across design methodologies (FPGA, gate array, standard cell) and/or across processes/libraries. For example, this includes the mapping of a source standard-cell library to a target standard-cell library, such as translating a design from a 1.2 µm standard-cell library into a 0.9 µm standard-cell library. It also includes the translation functions done by *SoftPath*.

Merging: Integrating multiple logic blocks into a single, more highly integrated block, including: multiple logic blocks into one logic block, multiple PLDs into one FPGA, and multiple FPGAs into one gate array or standard-cell ASIC.

Fitting: Putting an ASIC design in the optimal size ASIC/package, such as putting a 4000 gate device in a 3064 132-pin QFP as opposed to a 172-pin 3090.

Partitioning: Dividing a large logic function into multiple ASICs, such as placing a 50K gate design into ten 3090s or a 500K gate design into five ATT65676 gate arrays.

Direct Conversion: Converting one FPGA into one gate array or one standard-cell ASIC, with identical logic, pinouts, and packages.

The tool used depends on a number of factors, including translation requirement, existing design environment, designer learning curve, and cost. If a designer is already using the *Synopsys Design Compiler*, *Viewlogic's* Retargeter, or *Mentor Graphics'* Autologic, the translation may be done using these tools with AT&T's libraries. If a designer is not using a third-party tool, AT&T's *SoftPath* tool set incorporates Bell Laboratories optimization technology, is learned in less than one hour, and costs significantly less than third-party translation tools. It may be appropriate for a designer to evaluate all of the tools and use the one which provides the best results.

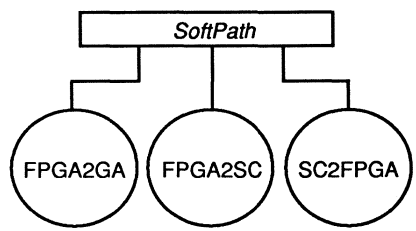


Figure 5. *SoftPath* Overview

As shown in Figure 5, the *SoftPath* tool set currently consists of three resynthesis programs. FPGA2GA resynthesizes an FPGA into a gate array. FPGA2SC resynthesizes an FPGA into a standard cell. SC2FPGA resynthesizes a standard-cell design gate array into an FPGA for prototype verification. The direct conversion of an FPGA to a gate array uses the same package to eliminate or minimize printed-circuit board (PCB) rework.

While this application note is focused on conversion of one FPGA into one gate array, *SoftPath* supports the merger of multiple FPGAs into a single gate array (although the process is not completely automated). This is important in systems where board space is a primary consideration.

Functional Description (continued)

To merge multiple FPGAs into a single gate array, the FPGA2GA tool is used to directly convert each FPGA netlist (xnf) into a gate array netlist (edif). The resulting netlists are then merged iteratively into a single gate array netlist. The I/Os are manually edited.

The FPGA2GA tool accepts an XNF file from a completed FPGA design and outputs a netlist which is used in layout and simulation of the gate array.

As shown in Figure 6, FPGA2GA generates an EDIF 2 0 0 netlist for use with third-party design tools. It can also generate lsi, pwc, and ddb design files for use with AT&T and NEC developed design tools. The different netlists are required because different gate array design tools require input in different formats. The lsi format is used in AT&T standard-cell designs.

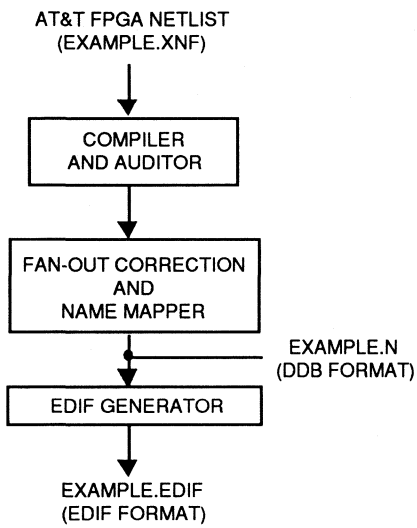


Figure 6. FPGA2GA Netlist Translation

To use the FPGA2GA tool, enter at the *UNIX* prompt:

fpga2ga <filename>

For the translation part of the resynthesis function done by the fpga2ga command, the .xnf file used must be the resulting output file from the lca2xnf conversion tools.

AT&T's *FPGA2GA User's Guide* provides a complete description of the installation and use of the FPGA2GA tool. Figures 7 and 8 show sample XNF input file and a resulting edif output file from running fpga2ga on the 16_addr design.

FPGA and Gate Array Design Environments

As migration between ASIC design methodologies becomes easier, the percentage of FPGA designs using engineering workstations is increasing.

For schematic capture and simulation of an FPGA, AT&T supports the third-party tools listed. Additionally, other design tools are provided and supported by third-party CAE vendors.

- Cadence
- *FutureNet*
- *Mentor Graphics*
- *OrCad*
- *Synopsys*
- *Viewlogic*

For schematic capture and simulation of AT&T 656 gate arrays, AT&T supports the AT&T and third-party tools listed.

- AT&T *Syscad*
- *Verilog*
- *Mentor Graphics*
- *Viewlogic*
- *Synopsys*

AT&T provides a design kit for the design system used. Each design kit contains a user manual and the AT&T gate array library. The design manual provides schematic capture, test vector generation, and simulation notes needed to design a gate array.

Gate Array Selection

Table 4 provides guidelines for the initial gate array selection, showing AT&T gate arrays compatible with currently available FPGA packages. This guide is based on FPGA and gate array utilization estimates and other considerations. The actual utilization varies as a function of the logic of the circuit. In some cases, FPGA designers intentionally underutilize the FPGA to allow them to recover from design mistakes or add features after the initial design. FPGAs have a higher pin/gate ratio than gate arrays, so it may be necessary to use a higher-density gate array to meet I/O requirements.

Functional Description (continued)

Table 4. Compatible ATT3000 Series FPGA and ATT656 Series Gate Array Packages

Package Type	ATT 3020	ATT 3030	ATT 3042	ATT 3064	ATT 3090
68 PLCC	65630	65636	—	—	—
84 PLCC	65636	65636	65636	65646	65650
100 QFP	65636	65640	65640	—	—
160 QFP	—	—	—	65654	65654

System Design Considerations

For the direct conversion of an FPGA into a gate array, the system needs to be evaluated to determine if the resulting gate array correctly emulates the FPGA. At start-up, the FPGA goes through three phases. The FPGA configuration and initialization process should be understood so that the gate array does not inadvertently affect system operation.

In this section, the system design concerns are addressed according to the phase of FPGA operation.

- Configuration
- Reset/initialization
- System I/O interface

Configuration

In addition to the SRAM resident on the FPGA, an FPGA requires that the system contain configuration memory to write to the FPGA's SRAM. This configuration memory is almost always nonvolatile memory resident on the printed-circuit board (PCB). The ATT1736 and ATT1765 Serial ROMs provide this function in a minimum PCB area.

With the gate array, the configuration memory is not needed and can be removed from the board. The FPGA contains logic and I/Os (some dedicated, some which become general purpose) used to load the FPGA configuration memory from the system memory. In evaluating the feasibility of technology migration, the designer needs to ensure that the input into these pins does not affect the operation of the gate array and that the gate array outputs do not adversely affect the operation of the ICs in the system which interfaces to the gate array.

The AT&T FPGAs have five configuration modes which are used in the loading of configuration memory. The configuration mode used is selected by M0, M1, and M2 inputs.

- Master serial
- Master parallel
- Peripheral
- Slave
- Daisy chain

Since there is no configuration memory in the gate array, the FPGA2GA tool does not generate the logic which supports the writing of the FPGA resident SRAM. Normally, the reduced power, size, and cost make elimination of this logic an advantage.

If all FPGAs in a system are directly converted into equivalent gate arrays, the configuration modes do not need to be supported. It may be necessary to retain the configuration logic in systems in which some, but not all, FPGAs are converted. For example, if the system uses the daisy-chain configuration mode, and upstream FPGAs are converted, downstream FPGAs still must be configured; so the configuration logic is needed in the gate array.

Reset

The reset input of the FPGA reinitializes the start of configuration if asserted prior to the completion of the configuration, and clears the IOB and CLB FFs if asserted after configuration. Only this last function is retained in the gate array.

During the configuration cycle, certain pins are active for configuration, after which they become standard I/O pins. These pins include dedicated and dual-purpose pins discussed below.

System Input/Output

The following pins are FPGA inputs which have no effect on the converted gate array.

PWDWN	RTRIG	$\overline{\text{CS0}}$ *
DIN	M0	$\overline{\text{S1}}$ *
M1	M2	$\overline{\text{CS2}}$ *
CCLK	WS	

* User I/O after configuration.

The following outputs are not provided by the target gate array. The designer should determine if the system needs the outputs.

RDATA	INIT\
CCLK	RCLK\
$\overline{\text{HDC}}$	RDY\BUSY\
$\overline{\text{LDC}}$	D0—D7
DONE	

Some of the FPGA output pins are active during configuration and become active after configuration. The DONE output becomes active one pin before or after the completion of the configuration phase. The designer must ensure that the system does not need this signal to start system operation. More generally, the designer must ensure that the target gate array does not cause any system interface timing problems due to faster I/Os. The FPGAs have a programmable slew rate output which provides one of two output transition times.

Functional Description (continued)

The faster transition time (5 ns) is used to improve critical timing while the slower transition time (30 ns) is used to reduce peak currents and system noise.

The FPGA $\overline{\text{PWRDWN}}$ input places the FPGA in a state in which the configuration memory content is maintained, but the logic functionality is indeterminate. Because a gate array is not reconfigurable, this input has no effect on the gate array.

FPGA and Gate Array Simulation

The types of simulation generally used are as follows:

- Functional simulation
- Static timing analysis

In a functional simulation, input test stimuli are generated to verify that the ASIC operates as intended. In a static timing analysis, critical paths are evaluated.

Because a functional test may not detect a faulty device, fault simulation is used. With a gate array, in-system verification is relatively expensive when compared to an FPGA, making simulation essential in gate array design.

Fault simulation is used to exercise as many of an IC's nodes as possible. This is to check for manufacturing defects which can affect a products' operation. The premise is that a functional simulation generally will not cover all possible states that an IC will operate in over its lifetime.

Fault grading is used to evaluate the ability of a test pattern set to screen manufacturing faults. Fault coverage is the percentage of node faults which are detected by a set of test patterns.

The simulation optimally uses worst-case process, temperature, and supply voltage conditions. The ASIC is first simulated functionally using unit delays. After AT&T provides the wiring delays from place and route, a timing simulation is done. The simulation should be done at actual system speed and at tester speed. AT&T's production test is done at 10 MHz.

Since the same input stimuli are used for simulation and test, tester characteristics need to be considered when generating the test vectors.

For ASIC design, AT&T supports a variety of simulation/test procedures, including ad hoc, full scan, partial scan, and built-in self-test (BIST). While these are neither needed nor used in the test of FPGAs, they are used in testing gate array and standard-cell ASICs. Ad hoc test techniques use functional patterns to obtain the required controllability/observability to determine that the ASIC functions correctly. Scan techniques are used to improve the testability of sequential logic.

The following metrics are used for the various test approaches.

- Test vector/test program development time
- Fault coverage
- Number of test vectors
- Die size/speed impact
- Tester dependence
- At-speed testing

Section 6 of the *AT&T CMOS Gate Array Design Guide* discusses the generation of test patterns. Test patterns are a series of ones and zeros which are provided into the gate array as input stimuli. One test vector is provided for each interval change.

Test patterns are used in both simulation and the functional test done by ATE. The number of test patterns should be the minimum number that tests all parts of the internal circuit, but under 64K so that production time is minimized. Fault detection can be increased by including test points within the circuit.

Analyzing Timing Differences Between the FPGA and Gate Array

The primary design concern in the FPGA to gate array conversion is to ensure that there are no internal race conditions which affect the gate array functionality. The article *FPGA Migration* discusses potential race conditions. The timing differences between the FPGA and gate array are largely due to the parasitic capacitance associated with the switching transistors and the metal interconnect or wiring. Because of the lower RC of the interconnection scheme, the gate array may be much faster than the FPGA. It is possible, however, for certain logic functions on the FPGA to be as fast or faster than the equivalent function on the gate array. This is because a look-up table can effectively implement certain logic, and the flip-flop in the CLB is a custom layout.

To evaluate the timing differences, a functional and timing simulation on both the FPGA and gate array compares the output for potential timing problems. Design goals vary, with some migration objectives having identical functionality at the same speed, some including enhanced speed, and some having enhanced functionality. Not all simulators support timing simulations with the routing delays back annotated. *Viewlogic* and *Mentor Graphics* provide this for both AT&T FPGAs and gate arrays.

In some cases, however, the FPGA will not have been simulated. FPGA development software allows FPGAs to be configured and verified in a system in hours, and as a result, designers do not always simulate. In other cases, the FPGA was developed on PC-based tools, and a different simulator is used for gate array design.

Functional Description (continued)

The *XACT* development system does not include a functional or timing simulator. To obtain timing data from *XACT*, the user specifies the source and destination nodes in a routed *LCA* by selecting the nodes. The development system provides the delay. Using the wiring delays determined by the *XACT* software, some design systems (*Mentor Graphics* and *Viewlogic*) back-annotate the delays for (*Quicksim* and *Viewsim*) timing reverification.

Testing FPGAs and Gate Arrays

The ATT3000 Series FPGAs are tested on the Advantest 3340 VLSI test system, using one million vectors for the 3042-3090, and 512K vectors for the 3020-3030. The ATT3000 Series *FPGA Test Methodology* application note describes the comprehensive testing done on FPGAs. The reconfigurability of FPGAs allows a test vector set which provides a very high fault coverage. The FPGA is, from a test view, a standard product with an extensive test program exercising all of the components on the FPGA.

In contrast, the gate array requires a set of test vectors specific to the gate array. The test program for a gate array is less extensive and more application-specific than the test program used for an FPGA. Using the test vectors, AT&T generates a test program specific to the gate array.

The standard AT&T production ac test is done at 10 MHz. Testing the gate array on ATE introduces physical problems, such as skew and initialization, which are not factors in simulation. These factors need to be considered when generating test vectors.

ATE Standard dc Tests

In addition to functional tests, the test program from the ATPG generates the following standard dc tests:

- Breakdown voltage
- Diode to V_{SS}
- Nominal input leakage
- Diode to V_{DD}
- Input noise immunity
- 3-state output leakage

Design for Test Recommendations

Design for test (DFT) procedures are used at the outset of a design in order to improve testability and minimize test problems. DFT is something of an afterthought for converting an existing FPGA into gate arrays. These recommendations are included here for FPGA designs which are anticipating a migration to a gate array. Please consult an AT&T design center for more comprehensive DFT procedures.

- Use synchronous logic. Asynchronous logic exhibits process-dependent race conditions which may not be seen in a deterministic simulation.
- Do not use unclocked feedback paths (such as ring oscillators).
- When using multiple clocks, design for worst-case skew between clocks.
- Avoid using large fan-outs on edge-sensitive cells. The maximum rise and fall time for internal input signals should be less than 500 ns.
- Initialize all storage elements at the beginning of simulation and test. With simulators, a generic set or reset command can be used for initialization. The initialization of an ASIC in a tester usually requires specific input stimuli to put the IC into a known state.
- Do not clock a cell and change data in the same test vectors.
- Use macrocells whose internal FFs can be controlled into known states upon powerup.
- Use dedicated test pins to monitor critical nodes. If pins are not available, use bidirectional buffers in place of input or output buffers. Unused registers and/or address bits can define test modes to improve the controllability and observability of internal nodes.
- Use test pins to exercise I/O buffers independently of internal logic.
- Partition large circuits into smaller more manageable sections. For example, divide lengthy counting sequences into smaller sections.

Product Brief
June 1992

AT&T Microelectronics

ATT656 Series CMOS Gate Arrays

Features

- 1.0 µm CMOS Si-gate triple-layer metal process technology; 0.75 µm effective channel length
- 270 ps typical gate delay (two-input NAND, fan-out = 1, V_{DD} = 5 V, T_A = 25 °C)
- Channelless architecture for maximum layout flexibility
- Ten array sizes from 5,000 to 177,000 available gates
- Workstation support includes the AT&T Design System, *Synopsys*, *Mentor Graphics*, *Verilog* , *Valid*, and *Viewlogic*.
- Design synthesis supported through *Synopsys*
- Typically over 60% utilization for triple-layer metal arrays; over 40% utilization for double-layer metal arrays
- Migration path offered between FPGAs, gate arrays, and standard cells
- Extensive library of macrocells, I/O buffers, macrofunctions, and standard memory configurations
- Up to 448 signal I/Os with choice of:
  — Input, output, or bidirectional buffers
  — Additional power pads
  — CMOS or TTL input levels
  — Schmitt trigger inputs
  — CMOS or TTL output levels
  — Buffer drive up to 12 mA
- Internal power dissipation <8 µW/gate/MHz
- Diversified package offering
- Licensed second source to *NEC*

Description

The ATT656 series of CMOS gate arrays combines the leading edge technology necessary for high-performance products with design capability, service, and high quality. This combined capability helps to increase both system performance and integration, while reducing the design cycle time.

The ATT656 series is manufactured using a 0.75 µm channel length Si-gate CMOS triple-layer metal technology. A sea of gates architecture is used on the ten base arrays which provide a range of 5,000 to 177,000 equivalent gates and up to 448 I/Os. The typical gate delay for a two-input NAND gate is 270 ps (5.0 V, 25 °C, nominal processing, fan-out of 1). The device performance and cell libraries are compatible with the *NEC* gate array families (CMOS-6, CMOS-6A).

AT&T provides design kits containing schematic libraries, simulation libraries, application software, and documentation. ATT656 series gate array design kits are available for the AT&T Design System, *Mentor Graphics, Synopsys, Valid, Viewlogic*, and *Verilog*.

The ATT656 series is also an integral part of AT&T's product migration capability which allows the gate arrays to be used as a prototyping vehicle for the AT&T standard-cell devices or as a cost reduction vehicle for FPGAs. This methodology also provides for fast prototyping of the ATT656 series gate arrays with FPGAs to drastically reduce a product's time to market.

AT&T's ASIC design centers, located throughout the world, provide local access to advanced technology and engineering support.

Description (continued)

Table 1. ATT656 Series Gate Array Features

Part Number		Available Gates	Estimated Usable Gates		I/O Pins
			50% Memory	All Logic	
2LM	ATT65630	5,000	3,000	2,000	84
	ATT65636	8,000	4,800	3,200	100
	ATT65640	11,000	6,600	4,400	120
	ATT65646	16,000	9,600	6,400	140
	ATT65650	21,000	12,600	8,400	160
	ATT65654	30,000	18,000	12,000	192
3LM	ATT65658	42,000	31,500	25,200	220
	ATT65664	72,000	54,000	43,200	288
	ATT65672	119,000	89,200	71,400	368
	ATT65676	177,000	132,700	106,200	448

Architecture

The ATT656 series gate arrays are divided into I/O and internal cell areas. The I/O cell area contains configurable buffers that isolate the internal cells from high-energy external signals. With over 140 I/O buffer options to choose from, designers can select drivers to meet performance, noise, and electromagnetic radiation requirements.

The internal cell area is an array of basic cells, each composed of two p-channel MOS transistors and two n-channel MOS transistors, as well as four additional n-channel MOS transistors for compact RAM design. These p-channel and n-channel transistors are sized to offer a superb ratio of speed to silicon area with an internal power dissipation of less than 8 μW/gate/MHz. The basic cells can be configured to implement any of the more than 300 block library elements, as well as ROMs, single-port RAMs, and dual-port RAMs.

Architecture (continued)

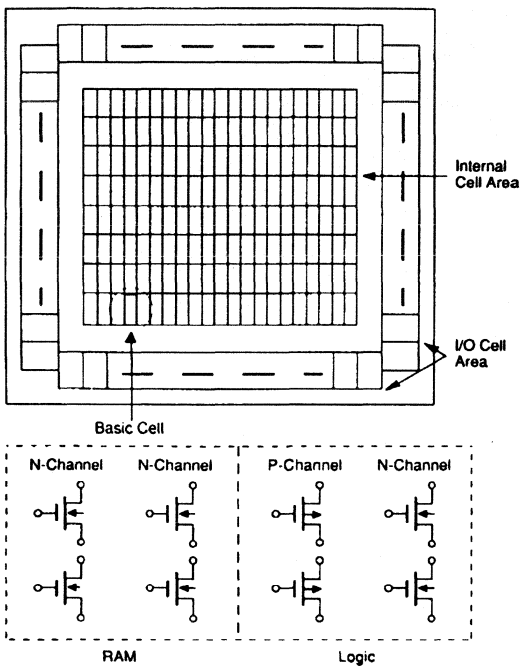


Figure 1. Chip Layout and Internal Cell Configuration

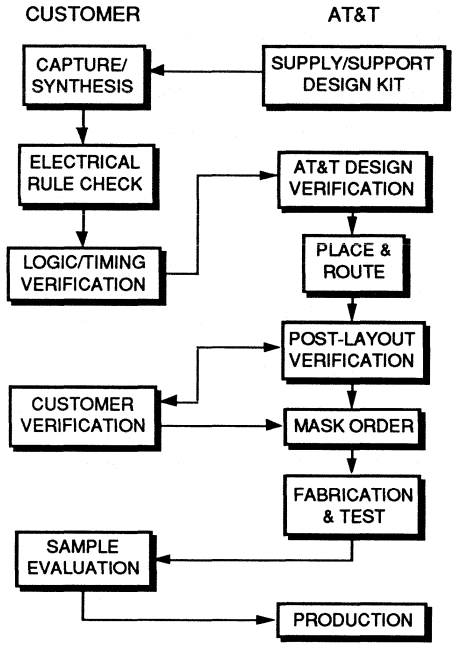


Figure 2. Gate Array Design Flow

Table 2. Gate Array Design Kits for Commercial CAD Systems

Design Kit	Schematic Capture	Logic Simulation	Timing Simulation	Electrical Rule Check	Synthesis
Mentor Graphics	x	x	x	x	
Valid	x	x	x	x	
Viewlogic	x	x	x	x	
Synopsys				x	x
Verilog		x	x		
AT&T Design System	x	x	x	x	

Architecture (continued)

Table 3. ATT656xx Gate Array Package Plan

Package	Pins	65630	65636	65640	65646	65650	65654	65658	65664	65672	65676
PLCC	68		X	X	X	X	X	X			
	84		X	X	X	X	X	X			
QFP	80	X	X	X	X	X	X				
(EIAJ)	100		X	X	X	X	X	X			
	120			X	X	X	X	X	X	X	
	160						X	X	X	X	
PQFP	84	X	X	X	X	X	X				
(JEDEC)	100		X	X	X	X	X	X			
	132					X	X	X	X		
	164						X	X	X	X	
SQFP	100		X	X	X	X	X	X			
	208								X	X	X
CPGA	72	*	*	*	*	*					
	132					*	*	*	*	*	
	176						*	*	*	*	*
	208							X	X	X	X
	280								X	X	X
	364									X	X
Butt Lead	288									*	*
CPGA	528										*

* These packages available by request.

Note: AT&T has begun qualification of the packages listed above and reserves the right to alter the package plan based on the results of qualification. All packages may not be immediately available. For current package availability, please contact your AT&T design center.

Table 4. Logic Cells/MSI/LSI Functions

Function	Type
Logic Gates	114
Flip-Flop/Latch	73
Parity Generator	2
Comparator	1
Multiplexer	6
Decoder	4
Counter	2
Single-Port RAM	25
Dual-Port RAM	25
ROM	23

Table 5. I/O Buffers

Function	Type
Input Buffer	18
Output Buffer	43
Bidirectional Buffer	75

Absolute Maximum Ratings

Stresses in excess of the Absolute Maximum Ratings can cause permanent damage to the device. These are absolute stress ratings only. Functional operation of the device is not implied at these or any other conditions in excess of those given in the operational sections of the data sheet. Exposure to Absolute Maximum Ratings for extended periods can adversely affect device reliability.

Parameter	Symbol	Min	Max	Unit
Power Supply Voltage	V_{DD}	–0.5	6.5	V
Input/Output Voltage	V_I/V_O	–0.5	V_{DD} + 0.5	V
Latch-up Current	I_{LATCH}	1.0	—	A
Operating Temperature	T_{OPT}	–40	85	°C
Storage Temperature	T_{stg}	–65	150	°C

Recommended Operating Conditions

Parameter	Symbol	CMOS Level		TTL Level		Unit
		Min	Max	Min	Max	
Power Supply Voltage	V_{DD}	4.5	5.5	4.75	5.25	V
Ambient Temperature	T_A	–40	85	0	70	°C
Low-level Input Voltage	V_{IL}	0	0.3 V_{DD}	0	0.8	V
High-level Input Voltage	V_{IH}	0.7 V_{DD}	V_{DD}	2.2	V_{DD}	V
Input Rise or Fall Time	t_R, t_F	0	200	0	200	ns
Input Rise or Fall Time, Schmitt	t_R, t_F	0	10	0	10	ms
Positive Schmitt Trigger Voltage	V_P	1.8	4.0	1.2	2.4	V
Negative Schmitt Trigger Voltage	V_N	0.6	3.1	0.6	1.8	V
Hysteresis	V_H	0.3	1.5	0.3	1.5	V

Electrical Characteristics

Table 6. Input/Output Capacitance

V_{DD} = V_I = 0 V; f = 1 MHz

Terminal	Symbol	Min	Typ	Max	Unit
Input	C_{IN}	—	10	25	pF
Output	C_{OUT}	—	10	25	pF
I/O	$C_{I/O}$	—	10	25	pF

Note: Values include package pin capacitance.

Table 7. Power Consumption

Parameter	Test Conditions	Max	Unit
Internal Cell	Fan-out = 3; L = 3 mm	8	µW/MHz
Input Block	Fan-out = 3; L = 3 mm	46	µW/MHz
Output Block	C_L = 15 pF	0.98	mW/MHz

Table 8. Memory Capability

Memory Type	Size	Cycle Time (typ)
Asynchronous Single-Port SRAM	32 x 16	20.3 ns
	64 x 16	20.8 ns
	128 x 16	23.1 ns
	256 x 16	27.4 ns
	512 x 8	29.8 ns
Asynchronous Dual-Port SRAM	32 x 16	20.2 ns
	64 x 16	20.8 ns
	128 x 16	23.1 ns
	256 x 16	27.5 ns
	512 x 8	29.9 ns
ROM	128 x 16	17.1 ns
	256 x 16	20.0 ns
	512 x 16	20.6 ns
	1K x 16	24.4 ns
	4K x 16	34.0 ns
	8K x 8	34.0 ns

Table 9. ac Characteristics

V_{DD} = 5 V ± 10%; T_A = –40 °C to +85 °C

Parameter	Symbol	Test Conditions	Min	Typ	Max	Unit
Toggle Frequency	f_{TOG}	D Flip-flop; Fan-out = 1	120	—	—	MHz
Delay Time Internal Gate	t_{PD}	Fan-out = 1; L = 0 mm	—	270	—	ps
		Fan-out = 3; L = 3 mm	—	700	—	ps
Delay Time Buffer	t_{PD}					
Input (FI01)		Fan-out = 1; L = 0 mm	—	0.8	—	ns
		Fan-out = 3; L = 3 mm	—	1.0	—	ns
Output (FO01)		C_L = 15 pF	—	2.0	—	ns
Output Rise Time	t_R	C_L = 15 pF	—	3.0	—	ns
Output Fall Time	t_F	C_L = 15 pF	—	2.0	—	ns

Electrical Characteristics (continued)

Table 10. dc Characteristics

V_{DD} = 5 V ± 10%; T_A = –40 °C to +85 °C

Parameter	Symbol	Test Conditions	Min	Typ	Max	Unit
Quiescent Current*	I_L	$V_I = V_{DD}$ or GND	—	0.1	400	µA
Input Leakage Current	I_I					
Regular		$V_I = V_{DD}$ or GND	—	10^{-5}	10	µA
50 kΩ Pull-up		V_I = GND	–40	–100	–270	µA
5 kΩ Pull-up		V_I = GND	–0.35	–1.0	–2.2	mA
50 kΩ Pull-down		$V_I = V_{DD}$	45	120	300	µA
Off-state Output Leakage Current	I_{OZ}	$V_O = V_{DD}$ or GND	—	—	10	µA
Input Clamp Voltage	V_{IC}	I_I = 18 mA	–1.2	—	—	V
Output Current Limit†	I_{OS}	V_O = 0 V	–250	—	—	mA
Low-level Output Current (CMOS)‡	I_{OL}					
3.0 mA		V_{OL} = 0.4 V	3.0	6.4	—	mA
6.0 mA		V_{OL} = 0.4 V	6.0	12.3	—	mA
9.0 mA		V_{OL} = 0.4 V	9.0	18.6	—	mA
12.0 mA		V_{OL} = 0.4 V	12.0	24.9	—	mA
High-level Output Current (CMOS)‡	I_{OH}					
1.5 mA		$V_{OH} = V_{DD} - 0.4$ V	–1.5	–3.0	—	mA
3.0 mA		$V_{OH} = V_{DD} - 0.4$ V	–3.0	–6.0	—	mA
4.5 mA		$V_{OH} = V_{DD} - 0.4$ V	–4.5	–9.0	—	mA
6.0 mA		$V_{OH} = V_{DD} - 0.4$ V	–6.0	–12.0	—	mA
Low-level Output Current (TTL)§	I_{OL}					
6.0 mA		V_{OL} = 0.4 V	6.0	11.8	—	mA
12.0 mA		V_{OL} = 0.4 V	12.0	23.4	—	mA
High-level Output Current (TTL)§	I_{OH}					
0.5 mA		V_{OH} = 2.4 V	–0.5	–1.1	—	mA
1.0 mA		V_{OH} = 2.4 V	–1.0	–2.1	—	mA
Low-level Output Voltage	V_{OL}	I_{OL} = 0 mA	—	—	0.1	V
High-level Output Voltage (CMOS)‡	V_{OH}	I_{OH} = 0 mA	V_{DD} – 0.1	—	—	V
High-level Output Voltage (TTL)§	V_{OH}	I_{OH} = 0 mA	2.6	3.4	—	V

* The maximum value reflects the use of pull-up/pull-down resistors and oscillator blocks.
† Rating is for only one output operating in this mode for less than 1 second.
‡ CMOS-level output buffer (V_{DD} = 5 V ± 10%, T_A = –40 °C to +85 °C).
§ TTL-level output buffer (V_{DD} = 5 V ± 5%, T_A = 0 °C to 70 °C).

Section 7.
Package Information

7. Package Information

Table of Contents

Contents **Page**

Application Note
June 1992

AT&T Microelectronics

AT&T FPGA Package Information

Introduction

Package information, including assembly characteristics and package thermal data, is presented here followed by individual package specifications.

Package Assembly Information

Package assembly information for the ATT3000 Series FPGA product family is summarized in Table 7-1 through Table 7-5.

Table 7-1. Assembly Information

Parameter	CPGA	PPGA	CQFP	PLCC	EIAJ-QFP
Assembly Location	AT&T Allentown, PA	AT&T Allentown, PA	AT&T Allentown, PA	AT&T Singapore	AT&T Singapore
Wafer Dicing Method	100% Saw Through	100% Saw Through	100% Saw Through	100% Saw Through	100% Saw Through
Package Material	Alumina	Epoxy/Glass Laminate Glass Transition Temp > 180 °C	Alumina	Plastic Sumikon 6300HD	Plastic Sumikon 6300HD

Table 7-2. Leadframe

Parameter	CPGA	PPGA	CQFP	PLCC	EIAJ-QFP
Material	*Kovar*	*Kovar*	*Kovar*	CDA 194 FH	KOBE KLF 125 FH
Pin Diameter	0.018 in.	0.018 in.	0.006 Thick	0.008 Thick	0.006 Thick
Under Plating	Nickel	—	Nickel	Nickel	Nickel
Outer Lead Finish	Gold	Solder	Gold	Solder	Solder

Table 7-3. Die Attach

Parameter	CPGA	PPGA	CQFP	PLCC	EIAJ-QFP
Material	Au Eutectic	Epoxy	Au Eutectic	Epoxy	Epoxy
Epoxy:					
Composition	—	Epoxy Resin	—	Epoxy Resin	Epoxy Resin
Filler Composition	—	Silver Filled	—	Silver Filled	Silver Filled
% Weight	—	70% + 10%, –5%	—	70% + 10%, –5%	70% + 10%, –5%
Supplier Name	—	Ablestik	—	Ablestik	Ablestik
Material ID	—	84—ILMIS	—	84—ILMIS	84—ILMIS
Method of Application	—	Dispensing	—	Dispensing	Dispensing

Package Assembly Information (continued)

Table 7-4. Wire Bond

Parameter	CPGA	PPGA	CQFP	PLCC	EIAJ-QFP
Method	Thermosonic	Thermosonic	Thermosonic	Thermosonic	Thermosonic
Wire Bond on Die	Ball	Ball	Ball	Ball	Ball
Wire Bond on Package	Wedge	Wedge	Wedge	Wedge	Wedge
Wire Material	Gold	Gold	Gold	Gold	Gold
Wire Diameter	0.001 in.	0.001 in.	0.001 in.	0.001 in.— 0.0013 in.	0.001 in.— 0.0013 in.
Ball Bond Diameter	3—4 mils	3—4 mils	3—4 mils	3—4 mils	3—4 mils

Table 7-5. Encapsulant

Parameter	CPGA	PPGA	CQFP	PLCC and EIAJ-PQFP
Encapsulation	Metal Lid	Epoxy Encapsulate	Metal Lid	Epoxy Novalac/Phenol Novalac Hardener
Supplier Name/Material ID Number	—	Dexter Hysol Epoxy, FP4322	—	Sumikon/ EME-6300HD
Flammability Rating	—	—	—	UL 94 V – 0 @ 1/8"
Filler Composition % Weight	—	Fused Silica 63% to 67%	—	Fused Silica 71% to 74%
Curing Agent	—	—	—	—
Curing Profile	—	3 hrs. at 170 °C	—	—
Alpha Particle Emission Rate	—	—	—	0.03
Glass Transition Temperature (ts)	—	160 °C	—	155 °C/min.
Braze Material	Gold/Tin	—	Gold/Tin	NA
Thermal Coefficient of Expansion	—	29×10^{-6} in./in./°C	—	Alpha1 = 19×10^{-6} Max Alpha1 = 75×10^{-6} Max

Package Thermal Resistance Guidelines

The package thermal resistances (θ_{JA}) provided in Table 7-6 are for natural convection and 300 fpm air cooling. These results are based upon available test data and analyses with a test package mounted on a double-sided FR-4 printed-wiring board having minimum copper metallization.

The customer's system printed-wiring board features (component density on the board, air flow, total power dissipation on the board, and the nearest neighbor device dissipation) may affect package thermal characteristics. Results may vary depending on the customer application.

Table 7-6. Package Thermal Resistance

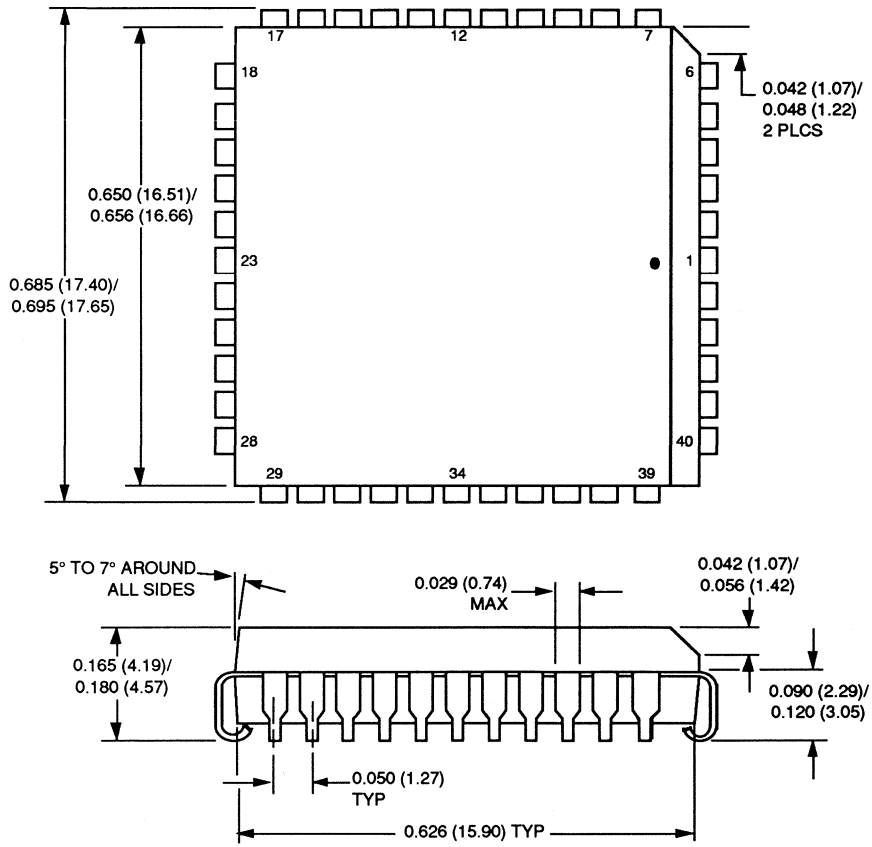
Device Type	Package Type	θ_{JA}	
		Natural Convection	300 fpm Air
ATT3020	68 PLCC	45	37
	84 PLCC	43	35
	84 CPGA	33	25
	EIAJ 100 QFP	85	66
	100 CQFP	46	37
ATT3030	44 PLCC	55	45
	68 PLCC	45	37
	84 PLCC	43	35
	84 CPGA	32	24
	EIAJ 100 QFP	85	66
ATT3042	84 PLCC	43	35
	84 CPGA	31	23
	EIAJ 100 QFP	85	66
	100 CQFP	44	35
	132 PPGA	22	17
	132 CPGA	22	18
ATT3064	84 PLCC	43	35
	132 PPGA	21	17
	132 CPGA	21	17
	EIAJ 160 QFP	38	29
ATT3090	EIAJ 160 QFP	38	29
	164 CQFP	28	22
	175 PPGA	22	17
	175 CPGA	19	15

Outline Diagrams

In this section, package dimensions are presented for the entire FPGA/Gate Array product line as well as for the three Serial E^2 ROM devices presented in Section 3.

44-Pin PLCC Package

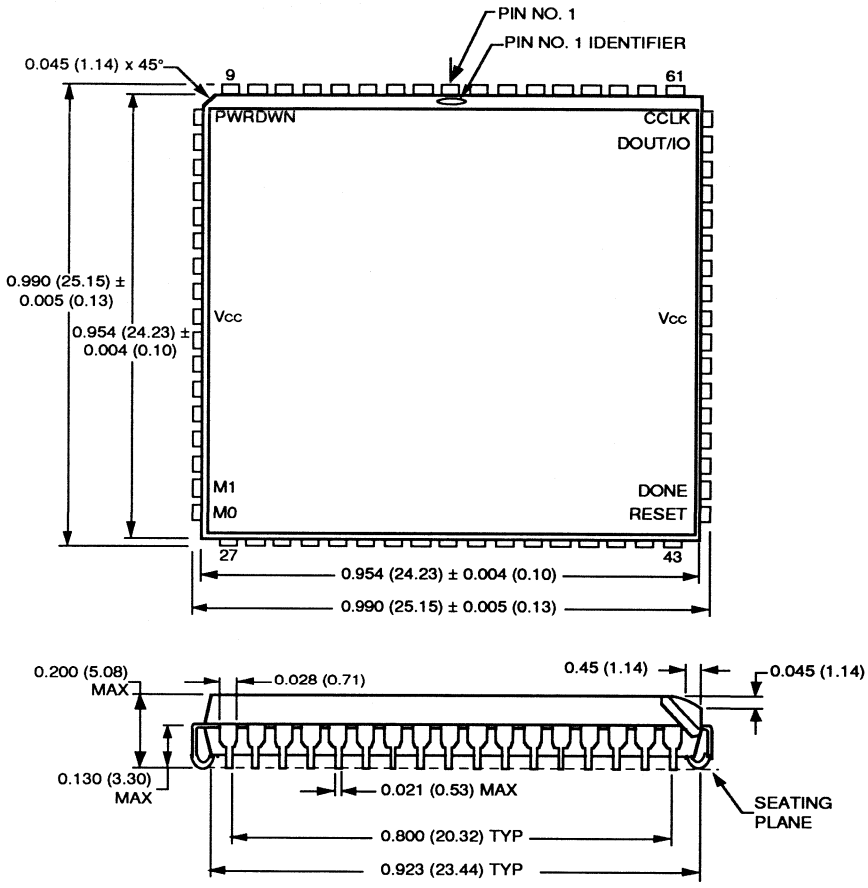
Dimensions are in inches and (millimeters).



Outline Diagrams (continued)

68-Pin PLCC Package

Dimensions are in inches and (millimeters).

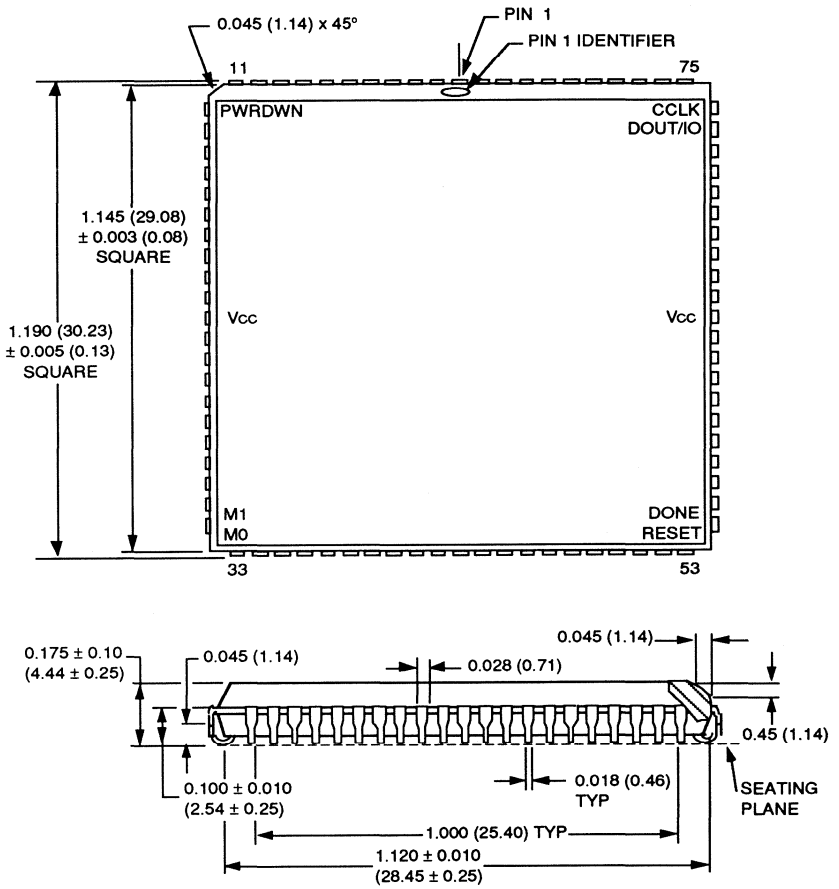


Notes:
Pin spacing 0.050 typical.
Lead coplanarity ± 0.0002 in.

Outline Diagrams (continued)

84-Pin PLCC Package

Dimensions are in inches and (millimeters).

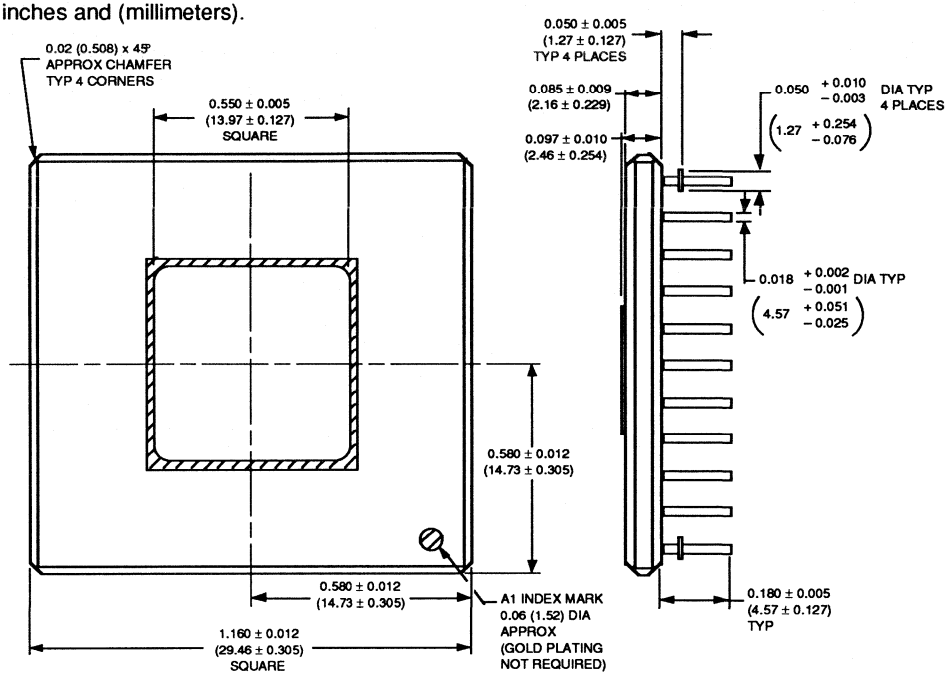


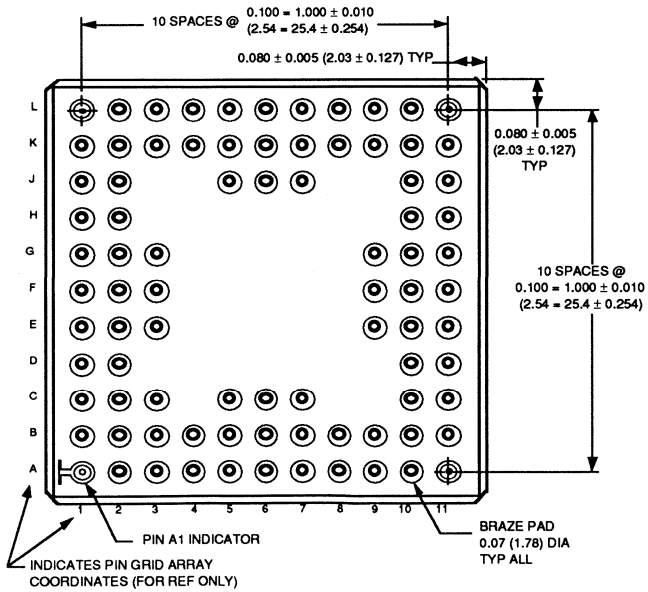
Note: Pin spacing 0.050 typical.

Outline Diagrams (continued)

84-Pin Ceramic PGA Package

Dimensions are in inches and (millimeters).

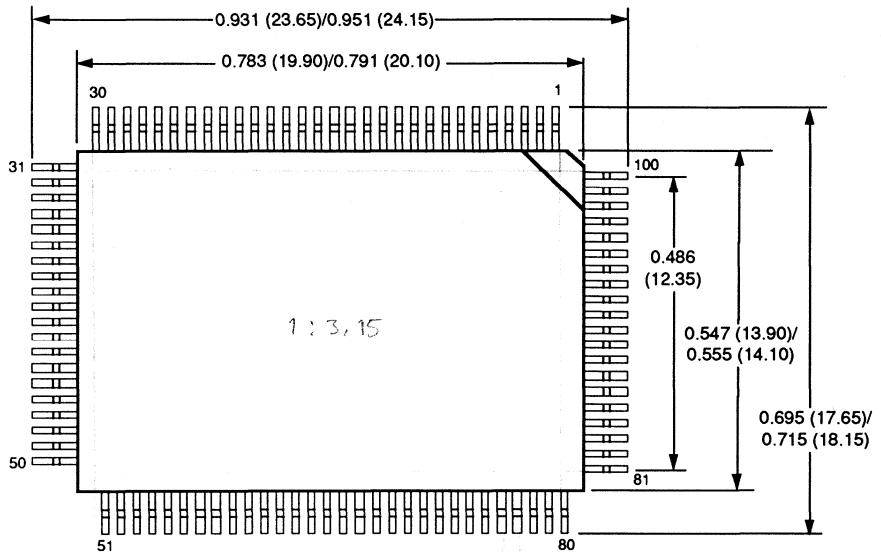


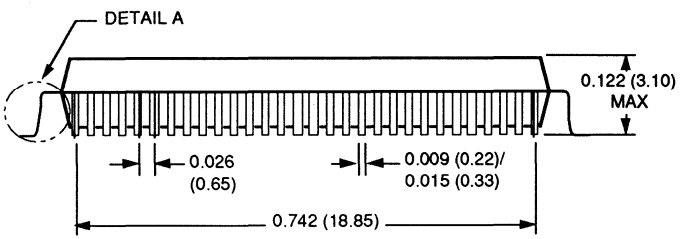


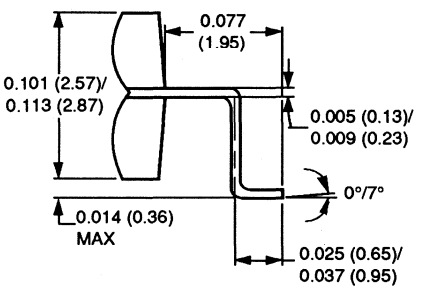
Outline Diagrams (continued)

100-Pin QFP Package

Dimensions are in inches and (millimeters).





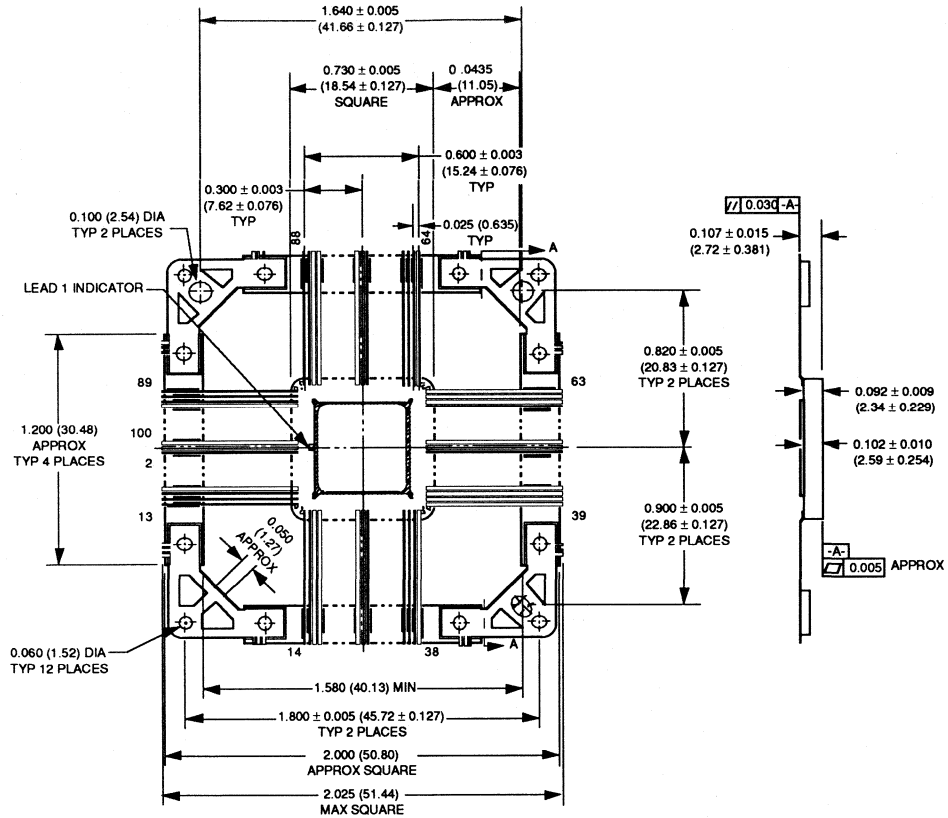


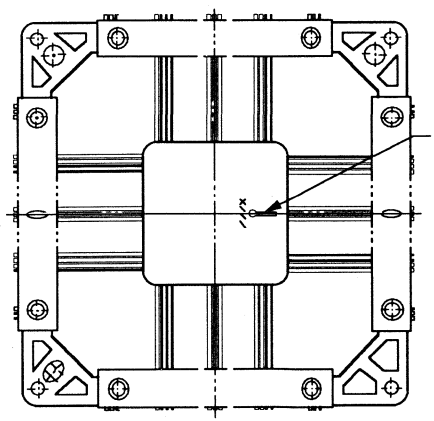
DETAIL A

Outline Diagrams (continued)

100-Pin CQFP Package

Dimensions are in inches and (millimeters).

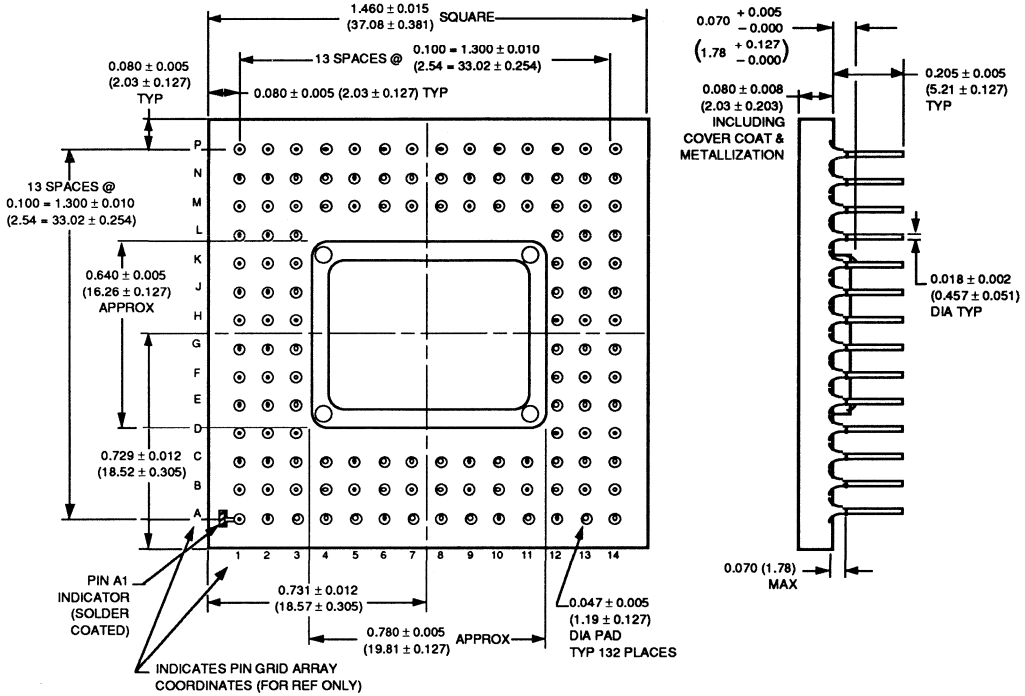


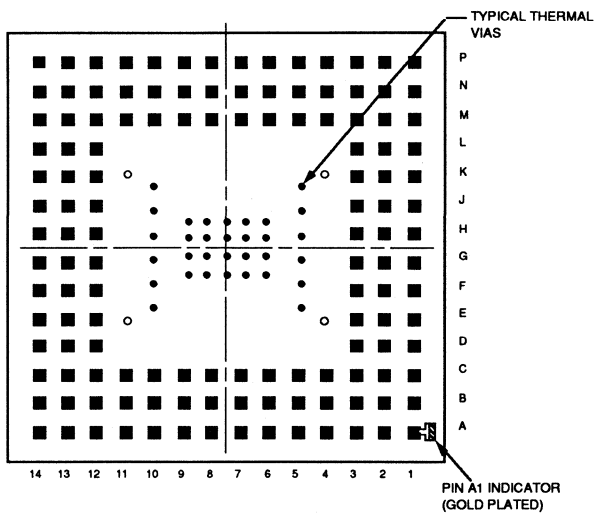


Outline Diagrams (continued)

132-Pin Plastic PGA Package

Dimensions are in inches and (millimeters).

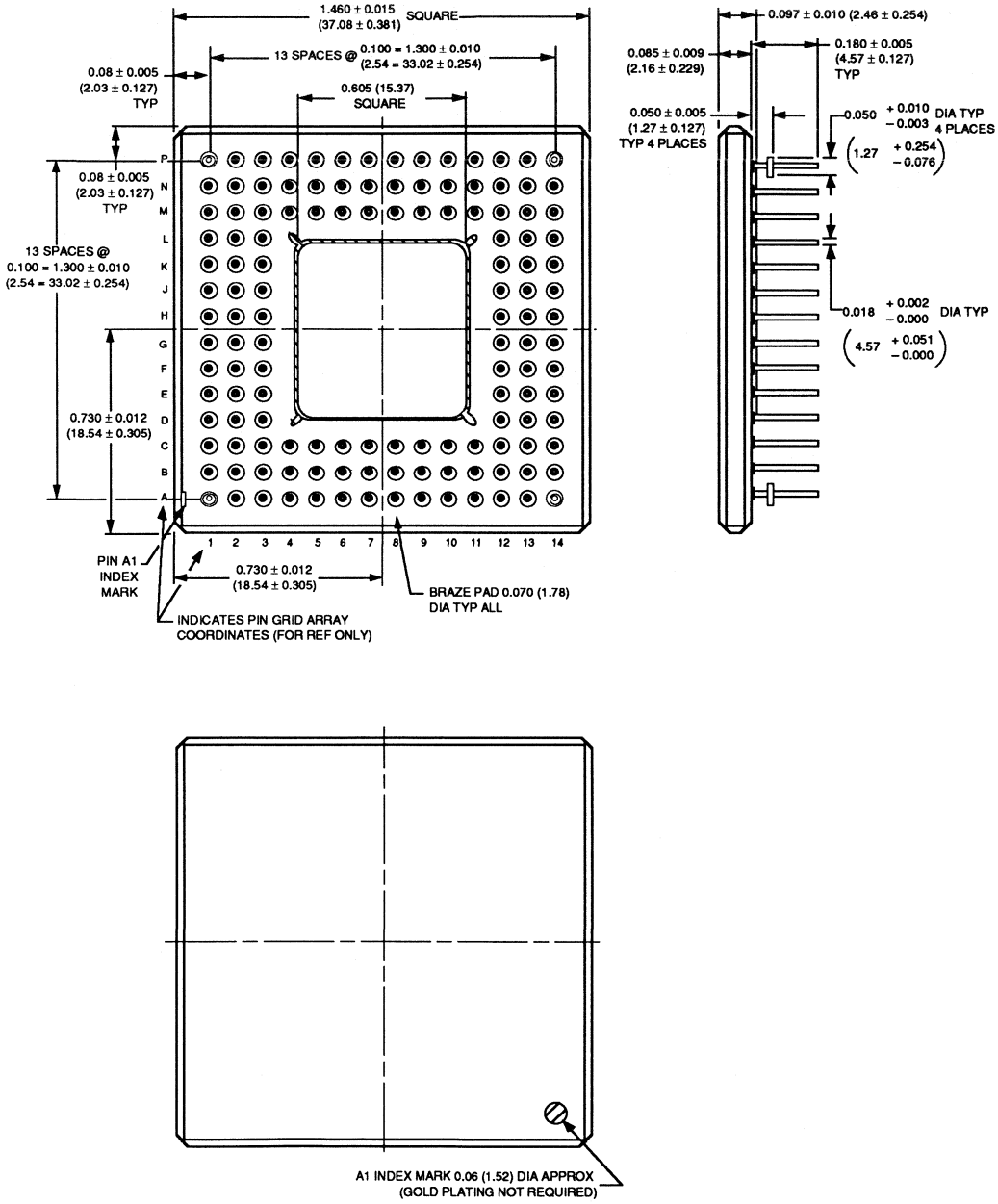




Outline Diagrams (continued)

132-Pin Ceramic PGA Package

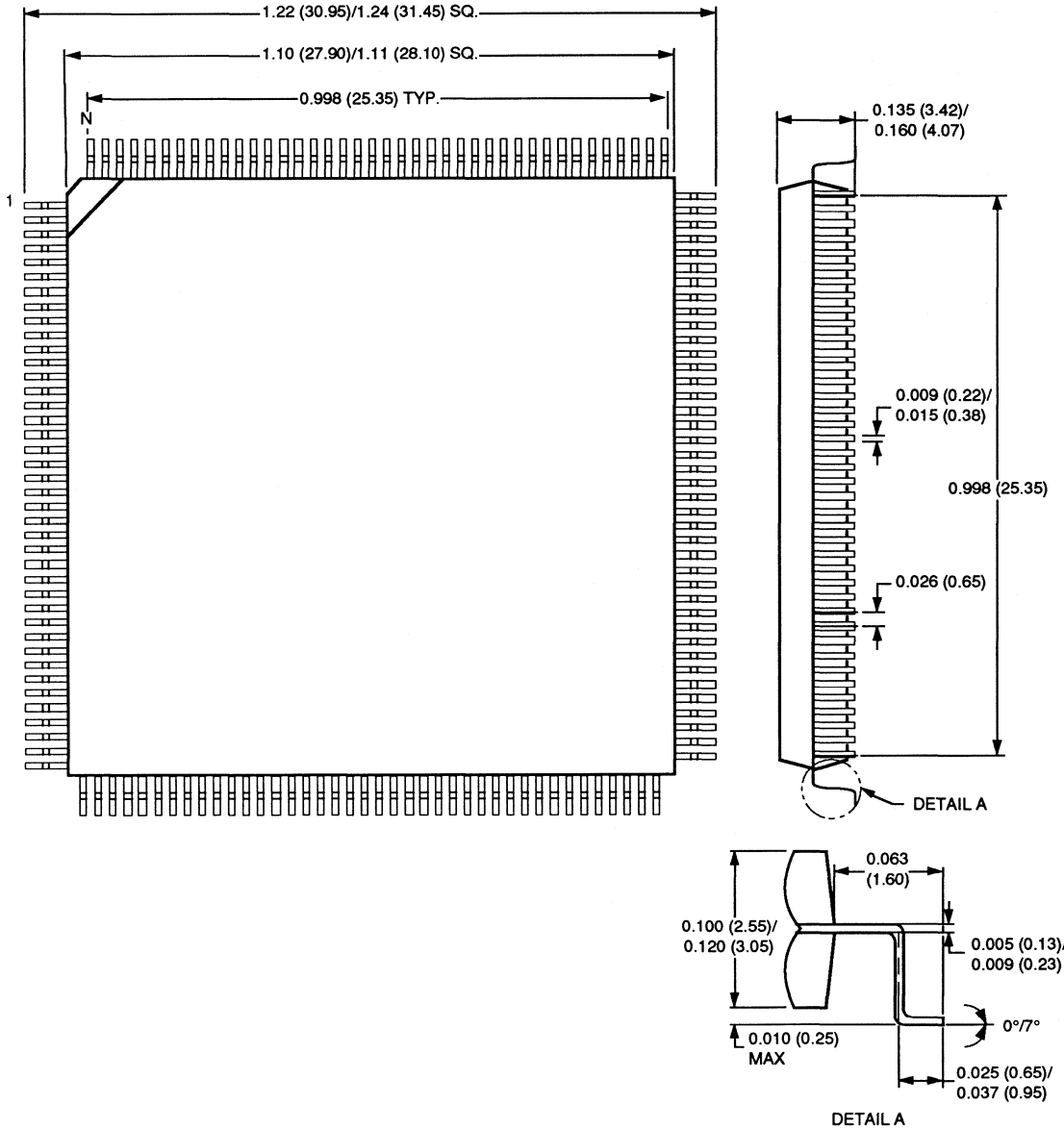
Dimensions are in inches and (millimeters).



Outline Diagrams (continued)

160-Pin QFP Package

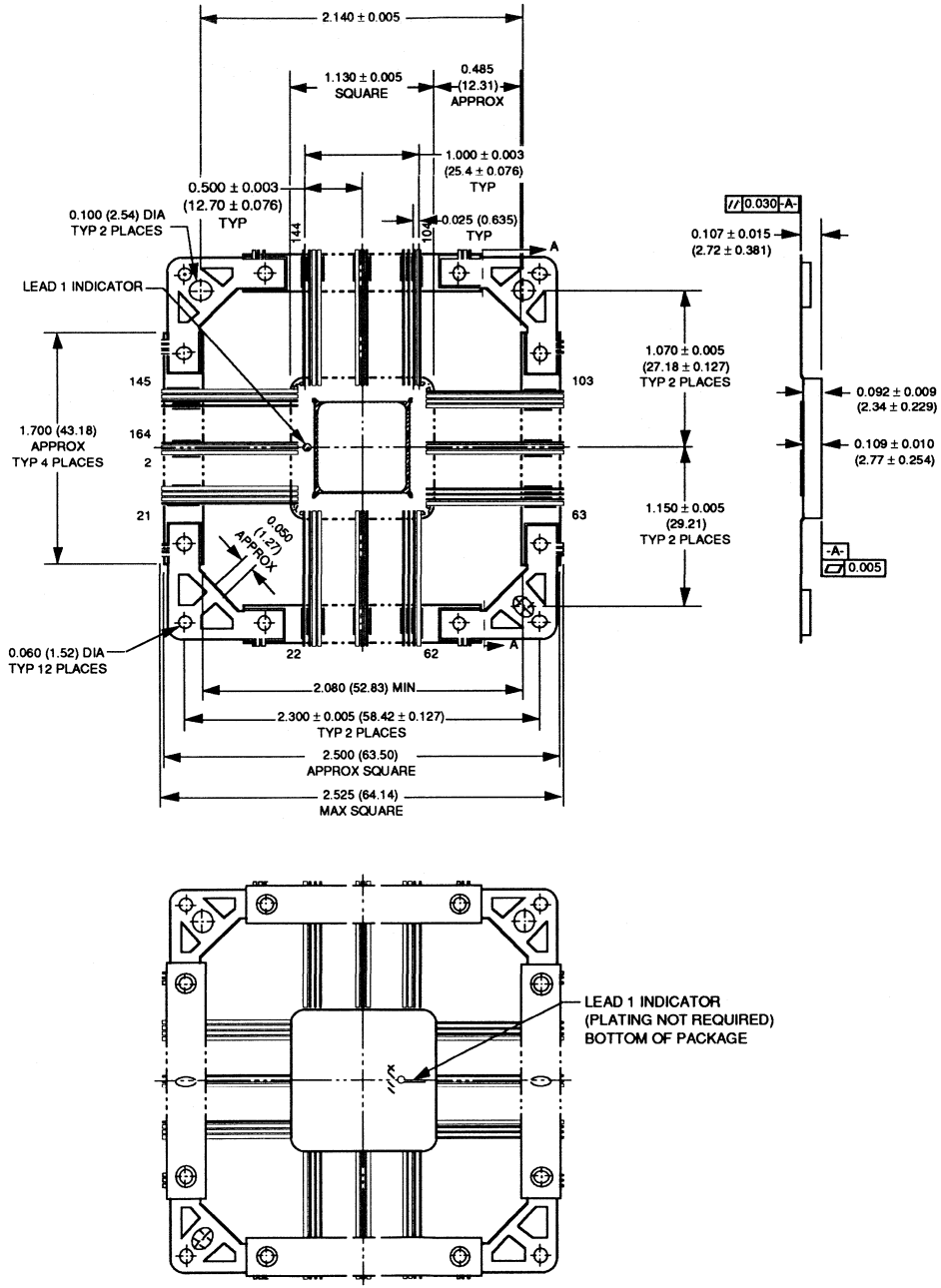
Dimensions are in inches and (millimeters).



Outline Diagrams (continued)

164-Pin CQFP Package

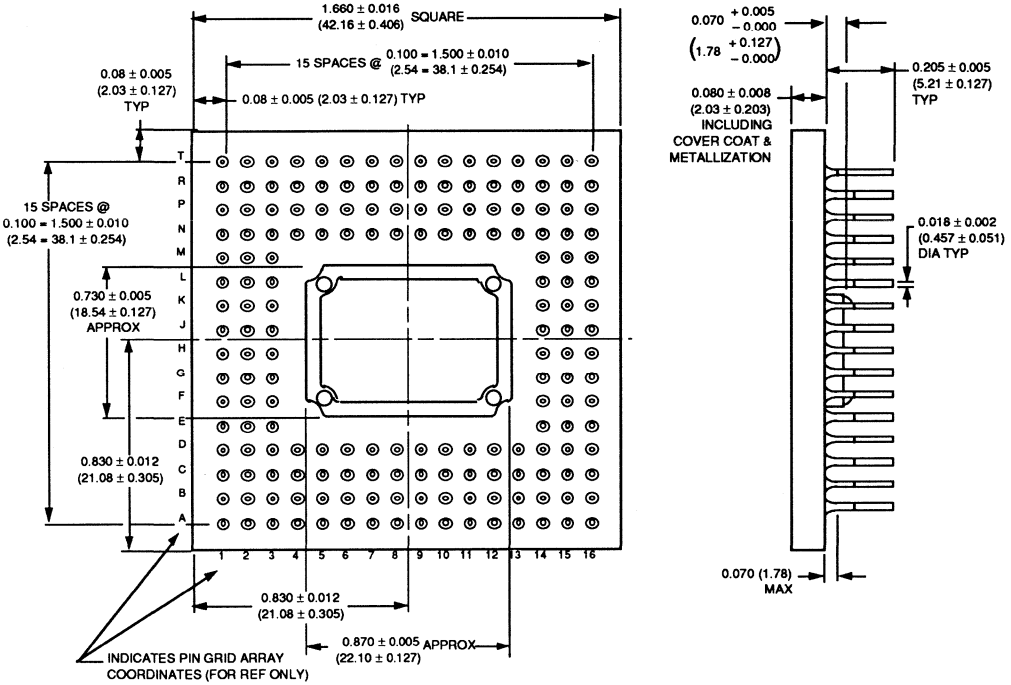
Dimensions are in inches and (millimeters).

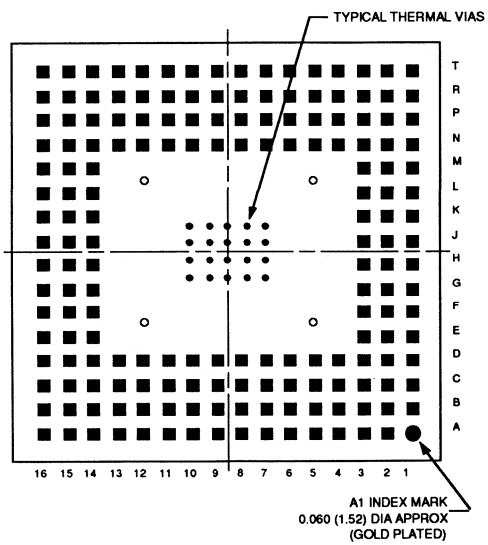


Outline Diagrams (continued)

175-Pin Plastic PGA Package

Dimensions are in inches and (millimeters).

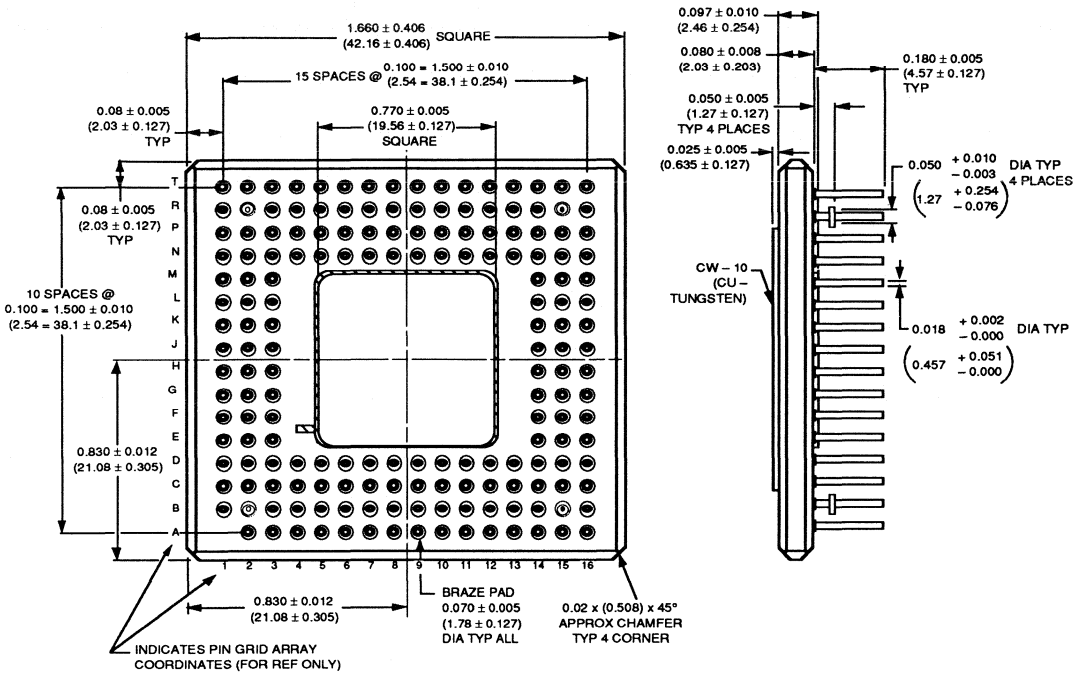


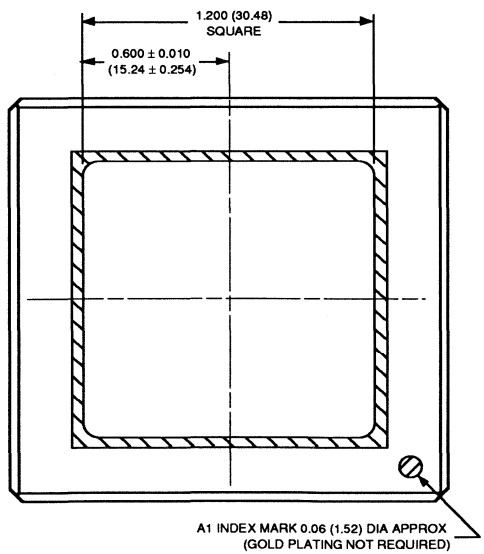


Outline Diagrams (continued)

175-Pin Ceramic PGA Package

Dimensions are in inches and (millimeters).

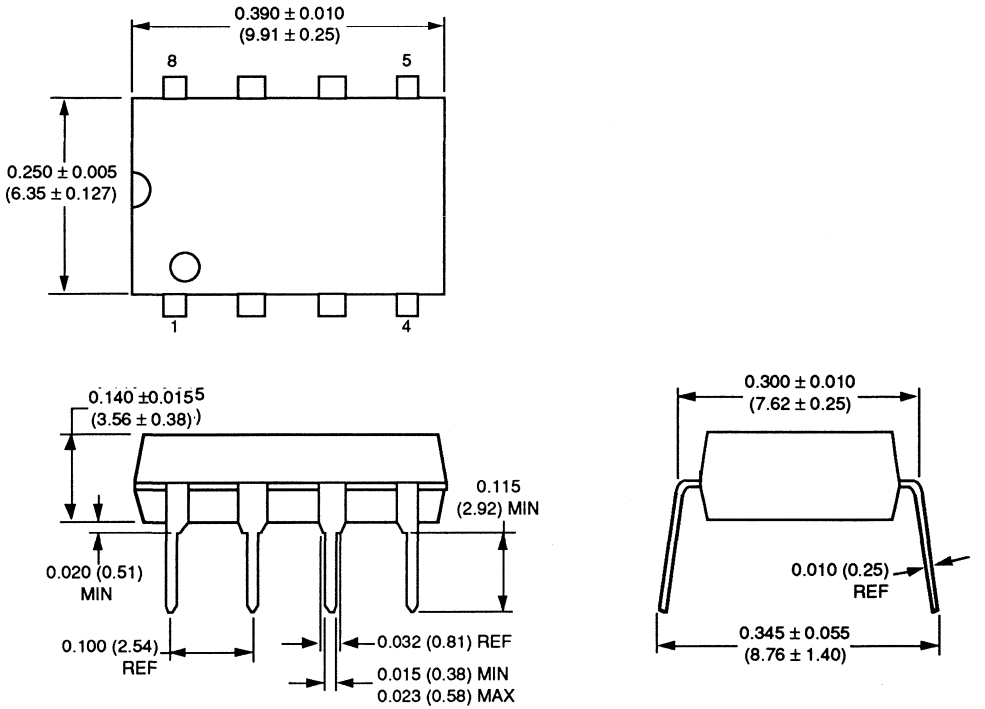




Outline Diagrams (continued)

8-Pin Plastic Package: ATT1736/1765 and ATT17128 Serial E² ROM

Dimensions are in inches and (millimeters).



AT&T Packing Methods

Dry Packing

Dry packing, which involves drying the product via a high-temperature bake and then placing it into a moisture barrier bag with desiccant, is required in order to keep our SMDs dry until the customer is ready to use them. The intent is to protect those SMDs that are suspected of being moisture sensitive from a phenomenon known as moisture-induced package cracking. This cracking, which can occur during the solder reflow process when a plastic SMD contains more than about 0.11 weight percent moisture, poses a potential reliability risk. The use of dry packing has become the norm for shipments of the larger SMDs, and is also widely used for shipments of devices with large die-to-package ratios. Devices can be dry packed in any of the packing options, including tubes, trays, or in tape and reel.

Tape and Reel

Tape-and-reel packing is an essential requirement for automatic board loading of PLCC and SOIC packages, with high (>5K) monthly usage rates (and is often desirable at lower rates). This packing method allows the customer to load their equipment (one time) with enough product for a full production run of boards, thereby eliminating the manual loading required when tubes are being used. We now have tape and reel packing capability at all of our assembly and test locations, and need to make our customers aware that we can provide this valuable packing option.

JEDEC Trays

The introduction of the EIAJ package type brought with it the need for tray packing, since, at the present time, trays are the only way to transport these fine-pitch SMDs. Like tape-and-reel packing, trays allow the end user to take full advantage of the automatic handling features which are a standard part of the newer pick-and-place board assembly machines. This reduction in operator handling translates into reduced cost and improved product quality. We had chosen to use the JEDEC thick tray design as our standard, but are now planning the introduction of the new JEDEC thin-style trays as an option. Both bakable and non-bakable trays are used, depending upon the particular application. These JEDEC-style trays are also being used for shipping of PQFP packages.

Valid Packing Options for MOS Devices in Plastic Packages

The following information is intended to document valid packing options so comcodes can be assigned correctly. Two specific rules that take precedence over all others are as follows:

- All internal plastic surface-mount devices must be dry packed. The following options are allowed:
  - — Tube dry pack (-D)
  - — Bakable tray dry pack (-DB)
  - — Tape-and-reel dry pack (-DT)
- Dry-packed product must not be shipped in non-bakable trays. Only bakable trays are allowed for dry packing because nonbakable trays are not ESD safe at the ultra low RH conditions that exist within the dry pack bags.

Other guidelines that are secondary to the above rules, with the options allowed, are as follows:

- EIAJ and SQFP devices can only be shipped in trays.
  - — Bakable tray standard packing (-BT)
  - — Bakable tray dry pack (-DB)
  - — Nonbakable tray standard packing (-NT)
- PQFP devices can be shipped in tubes or trays.
  - — Tube standard packing ( )
  - — Tube dry pack (-D)
  - — Bakable tray standard packing (-BT)
  - — Bakable tray dry pack (-DB)
  - — Nonbakable tray standard packing (-NT)
- SOIC and PLCC devices can be shipped in either tubes or in tape and reel.
  - — Tube standard packing ( )
  - — Tube dry pack (-D)
  - — Tape-and-reel standard packing (-TR)
  - — Tape-and-reel dry pack (-DT)
- DIP devices do not require dry packing and are only being shipped in plastic tubes.
  - — Tube standard packing ( )

The packing options described above have been developed through careful engineering evaluation. Any deviation from this list must be approved by the MOS packing engineer (M. B. Baillie).

Procedure for Changing Packing Options

Independent of the reason necessitating the change, all proposed changes must be approved by Production Control (PC).

The procedure for making changes to the packing option for products yet to be produced is relatively simple (since there is no Finished Goods Inventory or WIP to be considered). The process follows:

- Marketing requests that a change be made to the packing option for an existing comcode, through the Manufacturing Information Management (MIM), part of the AT&T Bell Laboratory DES Organization in Allentown.
- MIM runs the proposed change by PC for approval.
- Once the change has PC approval, MIM reissues the ADN with the change, and PC updates the PASS, EPPS, and ECSRA databases with the revised information.

If, on the other hand, product has already been made, the process is more complicated. This is because both WIP and finished goods must be taken into account, as well as the cost of converting these products to the desired packing format.

The procedure follows:

- Marketing requests that a change be made to the packing option for an existing comcode, through MIM.
- Production control and customer service jointly determine whether the existing comcode can be converted or whether a new comcode number is required. In either case, the following steps are required:
  - — Marketing and customer service will have to review the cost impact of any repacking of finished goods inventory.
  - — PC and PLPC will provide the WIP analysis and a conversion plan to marketing.
  - — Marketing will negotiate a final conversion plan with the customer (this may involve some extra charges).
  - — Marketing will arrange for any conversion of orders, and customer service will carry out the normal order management.

Any product being returned by the end customer for repacking must have marketing approval so that the customer is billed when applicable. The balance of the RMA procedure is unchanged.

Table 7-7. Packing Options

Package Type	Capacity/Carrier	Carrier No.	Total
44-Pin PLCC	27/Tube	PS-25388	96 Tubes per Bag
	500/Reel	6PZF04406	—
68-Pin PLCC	18/Tube	6ZPF06803	36 Tubes per Bag
	250/Reel	PS-25444	—
84-Pin PLCC	15/Tube	PS-25410	24 Tubes per Bag
	200/Reel	PS-25443	—
84-Pin Ceramic PGA	30/Chipboard Box	—	6 Boxes per Carton
100-Pin PQFP	19/Tube	6PZPF10035	36 Tubes per Bag
	55/JEDEC Tray	—	550 per Stack
132-Pin Plastic PGA	20/Chipboard Box	—	6 Boxes per Carton
132-Pin Ceramic PGA	20/Chipboard Box	—	6 Boxes per Carton
160-Pin QFP	24/Dry-packed, Bakable Trays	—	10 Trays per Box
164-Pin CQFP	4/JEDEC Tray	—	10 Trays per Box
175-Pin Plastic PGA	16/Chipboard Box	—	6 Boxes Per Carton
175-Pin CPGA	16/Chipboard Box	—	6 Boxes Per Carton
8-Pin Plastic Package	55/Tube	—	70 Tubes per Box

Section 8.
Sales Offices

8. Sales Offices

Table of Contents

AT&T Microelectronics

AT&T FPGA Data Book
Sales Offices

Introduction

For complete information and assistance in choosing the best AT&T FPGA/gate array devices for your applications, contact the product representative or office closest to you from the worldwide network of microelectronics component sales organizations of AT&T and its affiliates, featured in this section.

Customer Service 800 Numbers

Microelectronics (For customer inquiries only)	1-800-372-2447	FPGA Pager (12-Hour Coverage)	1-215-778-4100-X38907
DSP	1-800-346-3288	HPIC	1-800-325-HPIC
DBIC	1-800-ECL-ASIC	Memory	1-800-352-7768
FPGA Marketing	1-800-327-8742	Power	1-800-526-7819
FPGA Applications	1-800-327-8741		

Country Telephone Codes

For international direct calls from the United States, dial 011 before the number you are dialing. For international credit card or operator assisted calls from the United States, dial 01 before the number you are dialing.

Belgium	32	Hong Kong	852	Portugal	35	Switzerland	41
France	33	Italy	39	Singapore	65	South Korea	82
Finland	358	Japan	81	Spain	34	Taiwan	886
Germany	49	Netherlands	31	Sweden	46	United States	1
Great Britain	44						

OEM Sales Offices

World Headquarters

AT&T Microelectronics World Headquarters
2 Oak Way
Berkeley Heights, NJ 07922
(1) 908-771-2000 (phone)
(1) 908-771-3551 (FAX)

OEM Sales Offices (continued)

United States Regional OEM Sales Offices

Allentown
1247 S. Cedar Crest Blvd.
Dept. AL52060400
Allentown, PA 18103
(1) 215-770-2200 (phone)
(1) 215-770-2551 (FAX)

Atlanta
3295 River Exchange Dr.
Suite 350
Norcross, GA 30092
(1) 404-446-4700 (phone)
(1) 404-446-4728 (FAX)

Boston
Point West Place
111 Speen St.
Framingham, MA 01701
(1) 508-626-2161 (phone)
(1) 508-626-3710 (FAX)

Chicago
500 Park Boulevard
Suite 270
Itasca, IL 60143
(1) 708-290-3335 (phone)
(1) 708-290-3128 (FAX)

Dallas
222 West Las Colinas Blvd.
Suite 950
Irving, TX 75039
(1) 214-869-2040 (phone)
(1) 214-401-0390 (FAX)

Denver
7979 East Tufts Avenue
2nd Floor
Denver, CO 80237
(1) 303-290-3708 (phone)
(1) 303-290-4241 (FAX)

Los Angeles
6300 Gateway Drive
Cypress, CA 90630
(1) 714-220-6223 (phone)
(1) 714-220-6165 (FAX)

Minneapolis
1650 West 82nd Street
Suite 700
Bloomington, MN 55431
(1) 612-885-4300 (phone)
(1) 612-885-4331 (FAX)

Phoenix
1355 West University
Mesa, AZ 85201
(1) 602-844-6529 (phone)
(1) 602-844-6577 (FAX)

Portland
10220 S. W. Greenburg Road
Suite 520
Portland, OR. 97223
(1) 503-244-3883 (phone)
(1) 503-244-2315 (FAX)

Poughkeepsie
2 Jefferson Plaza
Poughkeepsie, NY 12601
(1) 914-485-7744 (phone)
(1) 914-485-7819 (FAX)

Raleigh
5400 Glenwood Avenue
Suite 412
Raleigh, NC 27612
(1) 919-881-8023 (phone)
(1) 919-881-8028 (FAX)

San Francisco
1090 E. Duane Avenue PC010
Sunnyvale, CA 94086
(1) 408-522-5555 (phone)
(1) 408-522-4219 (FAX)

International Regional OEM Sales Offices

Europe

Bracknell
AT&T Microelectronics
Powell Duffryn House
London Road
England
GB-Berks RG12 2AQ
(44) 344-487-111(phone)
(44) 344-48-57-35 (FAX)

Cheshire
AT&T Microelectronics
Cheshire House
164 Main Road
Goostrey
Cheshire
CW4 SJP
England
(44) 477-37179 (phone)
(44) 477-33449 (FAX)

Helsinki
AT&T Microelectronics
Kamreerintie 2B
SF-02770 Espoo
Finland
(358) 0-859-2955 (phone)
(358) 0-859-2466 (FAX)

OEM Sales Offices (continued)

International Regional OEM Sales Offices (continued)

Europe (continued)

Madrid
AT&T Microelectronics
Poligono Industrial de Tres Cantos
s/n (Zona Oeste)
E-28770 Colemenar Viejo
Spain
(34) 1-807-1700 (phone)
(34) 1-807-1699 (FAX)

Milan
AT&T Microelectronics
Viale Fulvio Testi 117
I-20092 Cinisello Balsamo
Italy
(39) 2-6601-1800 (phone)
(39) 2-6127-005 (FAX

Munich
AT&T Microelectronics
Bahnhofstrasse 27A
D-8043 Unterfoehring
Germany
(49) 89-95086-0 (phone)
(49) 89-95086-333 (FAX)

Paris
AT&T Microelectronics
Tour Horizon
52 Quai de Dion Bouton
France
F-92800 Puteaux
(33) 1-4776-4747 (phone)
(33) 1-4776-1840 (FAX)

Portugal
ATD Electronica L da
Rua Dr. Faria de Vasconcelos, 3
1900 Lisboa
Portugal
(35) 1-8472-200 (phone)
(35) 1-8472-197 (FAX)

Stockholm
AT&T Microelectronics
Klarabergsviadukten 70
Box 70363
S-107 24
Sweden
(46) 8700-5230 (phone)
(46) 8700-5263 (FAX)

Stuttgart
AT&T Microelectronics
Senefelderstrasse 8
D-7302 Ostfildern 1
Germany
(49) 711-44-20-50 (phone)
(49) 711-44-71-59 (FAX)

Far East

Korea-Seoul
AT&T Microelectronics
Samdo Building
9th Floor
1-170 Soonhwa-Dong, Choong ku
Seoul, 100-130
Korea
(822) 751-4155 (phone)
(822) 751-4165 (FAX)

Japan-Osaka
AT&T Microelectronics
2-27, Shiromi 1-Chome
Chuo-ku, Osaka 540 Japan
(81) 6-945-6515 (phone)
(81) 6-942-6886 (FAX)

Japan-Tokyo
AT&T Microelectronics
31-11, Yoyogi 1 - Chome
Shibuya-ku, Tokyo 151
(81) 3-5371-2700 (phone)
(81) 3-5371-3556 (FAX)

Singapore
AT&T Microelectronics
14 Science Park Drive
#03-02A/04 The Maxwell
Singapore 0511
(65) 778-8833 (phone)
(65) 777-7495 (FAX)

Taiwan-Taipei
AT&T Microelectronics
7/12/14th Floor, Overseas Trust Bldg.
249, Sec. 1, Tun Hwa South Road
Taipei, 106, Taiwan, R.O.C.
(886) 2-711-3091 (phone)
(886) 2-776-1366 (FAX)

Manufacturers Representatives

United States

Alabama
Prime Components, Inc.
4801 University Square, Suite 31
Huntsville, AL 35816
(1) 205-837-9668 (phone)
(1) 205-837-9687 (FAX)
(1) 602-991-0563 (FAX)

Arizona
Aztech Comp. Sales, Inc.
5230 North 75th St., Suite 1031
Scottsdale, AZ 85260
(1) 602-991-6300 (phone)
(1) 602-991-0563 (FAX)
attme!attmail!aztechcomp (E-mail)

California
Hadden & Asssociates
4710 Ruffner Streets, Suite H
San Diego, CA 92111
(1) 619-565-9444 (phone)
(1) 619-565-1802 (FAX)
attmail!hadden (E-mail)

Electec Sales, Inc.
4701 Patrick Henry Dr., Suite 2101
Santa Clara, CA 95054
(1) 408-496-0706 (phone)
(1) 408-727-9617 (FAX)
attme!attmail!esales (E-mail)

First Rep of Southern California
22024 Lassen Street, Suite 112
Chatsworth, CA 91311
(1) 818-718-6155 (phone)
(1) 818-718-1559 (FAX)
attmail!bakera (E-mail)

Colorado
Quorum 3
23 Inverness Way, Suite 120
Englewood, CO 80112
(1) 303-799-8100 (phone)
(1) 303-790-8686 (FAX)
attme!attmail!quorum3 (E-mail)

Connecticut
Scientific Components Inc. (SCI)
315 Highland, P.O. Box 160
Cheshire, CT 06410
(1) 203-272-2963 (phone)
(1) 203-271-3048 (FAX)

Florida
Cartwright & Bean, Inc.
1706 East Semoran Blvd., Suite 114
Apopka, FL 32703
(1) 407-889-9100 (phone)
(1) 407-889-2168 (FAX)
attme!attmail!cartwrightbe (E-mail)

Cartwright & Bean, Inc.
324 South Military Trail
Deerfield Beach, FL 33442
(1) 305-429-3400 (phone)
(1) 305-429-3402 (FAX)
attme!attmail!cartwrightbl (E-mail)

Manufacturers Representatives (continued)

United States (continued)

Georgia
Prime Components, Inc.
3008 Heritage Valley Court
Douglasville, GA 30135
(1) 404-942-5852 (phone)
(1) 404-489-0881 (FAX)
attme!attmail!primecomp2 (E-mail)

Illinois
Victory Sales Inc.
200 West Higgins Road, Suite 220
Schaumburg, IL 60195
(1) 708-490-0300 (phone)
(1) 708-490-1499 (FAX)
attme!attmail!victorysales (E-mail)

Indiana
Mohrfield Marketing, Inc.
4175 Millersville Road
Indianapolis, IN 46205
(1) 317-546-6969 (phone)
(1) 317-547-1729 (FAX)
attme!attmail!mohrfieldmar (E-mail)

Mohrfield Marketing, Inc.
9415 Teke Drive.
Leo, IN 46765
(1) 219-627-5355 (phone)
(1) 219-627-5355 (FAX)

Maryland
Advanced Technology Sales
100 West Road, Suite 412
Towson, MD 21204
(1) 301-296-9360 (phone)
(1) 301-296-9373 (FAX)
attme!attmail!advancedtech (E-mail)

Massachusetts
Anchor Engineering Corp.
11 Walkup Drive
Westborough, MA 01581
(1) 508-898-2724 (phone)
(1) 508-870-0573 (FAX)

Michigan
Trilogy Marketing
Incorporated
7 West Square Lake Road
Bloomfield Hills, MI 48302
(1) 313-335-1244 (phone)
(1) 313-335-0677 (FAX)
attme!attmail!trimarkinc (E-mail)

Manufacturers Representatives (continued)

United States (continued)

Minnesota
D.A. Case Associates
4660 West 77TH Street, Suite 250
Minneapolis, MN 55435
(1) 612-831-6777 (phone)
(1) 612-831-7076 (FAX)
attme!attmail!dacase (E-mail)

Missouri
Midtec Associates, Inc.
100 Progress Parkway
Suite 219
St. Louis, MO 63043
(1) 314-275-8666 (phone)
(1) 314-275-8859 (FAX)
Bo Bickley

Midtec Associates, Inc.
11900 W. 87th St. Pkwy
Suite 220
Geneva, KS 66215
(1) 913-541-0505 (phone)
(1) 913-541-1729 (FAX)
Matt Pansing

New Hampshire
Integrated Technologies
Incorporated (ITI)
182 Main Street, Suite 2
Salem, NH 03079
(1) 603-898-6333 (phone)
(1) 603-898-6895 (FAX)
attmail!judywhite (E-mail)

New Jersey
Technical Marketing
Group
175-3C Fairfield Avenue
West Caldwell, NJ 07006
(1) 201-226-3300 (phone)
(1) 201-226-9518 (FAX)
attme!attmail!techmarkgrp (E-mail)

Vantage Sales Co.
V-105 Executive Mews
1930 E. Marlton Pike
Cherry Hill, NJ 08003
(1) 609-424-6777 (phone)
(1) 609-424-8909 (FAX)
attme!attmail!vantagesales (E-mail)

New York
Entec
6037 Taft Road East, Suite 106
North Syracuse, NY 13212
(1) 315-458-7936 (phone)
(1) 315-452-5680 (FAX)
attme!attmail!entecgroup (E-mail)

Technical Marketing
Group
20 Broad Hollow Road
Melville, NY 11747
(1) 516-351-8833 (phone)
(1) 516-351-8667 (FAX)
attme!attmail!techmarkgrp1 (E-mail)

North Carolina
Prime Components, Inc.
6131 Falls of Neuse Road, Suite 107
Raleigh, NC 27609
(1) 919-850-9866 (phone)
(1) 919-850-9867 (FAX)
attme!attmail!primecomp (E-mail)

Prime Components, Inc.
5250-77 Center Drive, Suite 350
Charlotte, NC 28217
(1) 704-522-1150 (phone)
(1) 704-522-1151 (FAX)
attme!attmail!primecomp1 (E-mail)

Ohio
Midwest Marketing
Association
30 Marco Lane
Dayton, OH 45458
(1) 513-433-2511 (phone)
(1) 513-433-6853 (FAX)

Midwest Marketing
Association
5001 Mayfield Road, Suite 217
Lyndhurst, OH 44124
(1) 216-381-8575 (phone)
(1) 216-381-8857 (FAX)

Manufacturers Representatives (continued)

United States (continued)

Oregon
Advanced Technical Marketing
6700 S.W. 105th St., Suite 303
Beaverton, OR 97005
(1) 503-643-8307 (phone)
(1) 503-643-4364 (FAX)
attme!attmail!advancedtm1 (E-mail)

Pennsylvania
G&G Materials
237 Lancaster Avenue, Suite 103
Devon, PA 19333
(1) 215-293-0840 (phone)
(1) 215-293-0841 (FAX)

Puerto Rico
ETS, Inc.
P.O. Box 10758, Caparra Hts. Station
San Juan, Puerto Rico 00922
(1) 809-798-1300 (phone)
(1) 809-798-3661 (FAX)

Texas
Southern States
Marketing
1143 Rockingham, Suite 106
Richardson, TX 75080
(1) 214-238-7500 (phone)
(1) 214-231-7662 (FAX)
attme!attmail!ssmdal (E-mail)

Southern States
Marketing
10700 Richmond, Suite 243
Houston, TX 77042
(1) 713-789-2426 (phone)
(1) 713-789-3202 (FAX)
attme!attmail!lparks (E-mail)

Southern States
Marketing
400 E. Anderson Lane, Suite 610
Austin, TX 78752
(1) 512-835-5822 (phone)
(1) 512-835-1404 (FAX)
attme!attmail!dryndak (E-mail)

Utah
Anderson Associates
270 S. Main Street, Suite 108
Bountiful, UT 84010
(1) 801-292-8991 (phone)
(1) 801-298-1503 (FAX)
attme!attmail!andersonassc (E-mail)

Virginia
Advanced Technology Sales
406 Grinell Drive
Richmond, VA 23236
(1) 804-320-8756 (phone)
(1) 804-320-8761 (FAX)
attme!attmail!advtechsales (E-mail)

Washington
Advanced Technical Marketing
8521 154th Avenue NE
Redmond, WA 98052
(1) 206-869-7636 (phone)
(1) 206-869-9841 (FAX)
attme!attmail!advancedtm (E-mail)

Manufacturers Representatives (continued)

United States (continued)

Wisconsin
Victory Sales Inc.
N81 W12920 Leon Road, Suite 205
Menomonee Falls, WI 53501
(1) 414-255-6016 (phone)
(1) 414-255-6017 (FAX)

Canada

Ontario
J-Squared Technologies
300 March Road, Suite 401
Kanata, Ontario, Canada K2K2E2
(1) 613-592-9540 (phone)
(1) 613-831-0275 (FAX)

J-Squared Technologies
282 Belfield Road, Suite 201
Rexdale, Ontario, Canada M9W145
(1) 416-674-0927 (phone)
(1) 519-821-6053 (FAX)

Far East

Seraphim Engineering Co., LTD.
6th Floor, No. 34-1
Chiu Chuan Street
Taipei, Taiwan, R.O.C.
(886) 2-593-5195 (phone)
(886) 2-594-9424 (FAX)
James Lau

Exartech International Corp.
15F-4,6
Lane 144 Min Chuan West Road
Taipei, Taiwan, R.O.C.
(886) 2-537-2201-3 (phone)
(886) 2-542-2689 (FAX)
Michael Cheng

M-Systems
Room 301, Mirai Building
#706-4 Daelim 2 Dong
Youngdeungpo-ku
Seoul, South Korea
(82) 2-833-5548 (phone)
(82) 2-844-5568 (FAX)
Son-ung, OH

Distributors/Trading Companies

United States

Gus Lebkuecher, National Distribution Manager
1-908-771-2890 (phone)
1-908-771-4541 (FAX)

California
Almac Electronics Corporation (Hdqts.)
14360 SE Eastgate Way
Bellevue, WA 98007
(1) 206-643-9992 (phone)
(1) 206-643-9709 (FAX)
Keith Hanson

Distributors/Trading Companies (continued)

United States (continued)

LEX Electronics
90 East Tasman Drive
San Jose, CA 95134
(1) 408-432-7151 (phone)
(1) 408-943-0623 (FAX)
Ken Campbell (PM)

LEX Electronics
LA. County
26707 West Agoura Road
Calabasis, CA 91302
(1) 818-880-9686 (phone)
(1) 818-880-4687 (FAX)

Merit Electronics
2070 Ringwood Avenue
San Jose, CA 95131
(1) 408-434-0800 (phone)
(1) 408-434-0935 (FAX)
Nick Pipkin

Georgia
LEX Electronics
20 Technology Parkway, South
Suite 3275
Norcross, GA 30092
(1) 404-449-9170
(1) 404-242-6615

Illinois
LEX Electronics
945 North Plum Grove Road
Suite A-G
Schaumberg, IL 60173
(1) 708-517-6225 (phone)
(1) 708-330-3714 (FAX)
Steve Ryan

LEX Electronics
East Cromwell Road, Suite 100
Irvine, CA 92714
(1) 714-587-0404 (phone)
(1) 714-454-4206 (FAX)
Sandy Wood (PM)

Distributors/Trading Companies (continued)

United States (continued)

Massachusetts
LEX Electronics
301 Ballardvale Street
Wilmington, MA 01887
(1) 508-658-8796 (FAX)
John O'Day (PM)

New Jersey
LEX Electronics
1000 Briggs Road
Suite 200
Mt. Laurel, NJ 08054
(1) 609-727-7013 (phone)
(1) 609-273-7992 (FAX)
Rick Asher

New York
LEX Electronics Corporate
Jericho Turnpike
CB1032
Westbury, NY 11590
(1) 516-876-4201 (phone)
John Graffigna
Bernadette Zacor (Lightwave)

Texas
LEX Electronics
15167 Business Avenue
Dallas, TX 75244
(1) 214-247-6300 (phone)
(1) 214-247-0885 (FAX)
Bill Jefferson

Europe

AQL Electronica, s.a.
General Palanca, 26
28045 Madrid
Spain
(34) 1-467-75-12 (phone)
(34) 1-230-29-34 (FAX)
Juan Garcia

API Elektronik Vertriebs GmbH
Lorenz - Braren - Str. 32
D - 8062 Markt Indersdorf
Germany
(49) 8136-7092 (phone)
(49) 8136-7398 (FAX)
Gunter L. Anzer
Marianne Goettler
(DSP and FPGA Products Only)

Arcobel BV
Griekenweg 25
NL - 5342 PX Oss
The Netherlands
(32) 31-4120-30335 (phone)
(32) 31-4120-30635 (FAX)
Leo Verhoeckx
(FPGA Products Only)

Distributors/Trading Companies (continued)

Europe (continued)

Arrow
St. Martins Business Centre
Cambridge Road
Bedford
MK42 OHF.
England
(44) 234-272733 (phone)
(44) 234-214674 (FAX)
David Sprag
(FPGA Products Only)

ASTRONIC GmbH
Elektronik-Vertrieb
Gruenwalder Weg 30
D-8024 Deisenhofen
Germany
(49) 89-613-0303 (phone)
(49) 89-613-1668 (FAX)
Peter Riedl
(FPGA Products Only)

ATD Electronica LDA
Rua Dr. Faria de
Vasconcelos, 3-A
1900 Lisbon
Portugal
(351) 1-847-2200/01 (phone)
(351) 1-847-2197 (FAX)
Rafael Viela

Bytech Components Ltd.
12A Cedarwood
Chineham Business Park
Crockford Lane
Basingstoke
Hants
RG 24 OWD
England
(44) 256-707107 (phone)
(44) 256-707162 (FAX)
Robert McCarthy

C-88 AS
101 Kokkedal Industripark
Dk-2980 Kokkedal
Denmark
(45) 42-244-888 (phone)
(45) 42-244-889 (FAX)
(32) 2-2164606 (FAX)
Gert Rasmussen

Datadis SA
3 Bis, Rue Rene Cassin
B.P. 84-Z.I. De La Bonde
91303 Massy Cedex
France
(33) 1-69204141 (phone)
(33) 1-69204900 (FAX)
Daniel Louvet

INELCO Electronics
Avenue des Croix de Guerre 94
B-120 Brussels
Belgium
(32) 2-2442811 (phone)
(32) 2-2164606 (FAX)
Eddy Biesemans

Institut fur Kosmosforschung
Rudower Chaussee 5
0-1199 Berlin
Germany
(49) 372-161-231-9527 (phone)
(49) 372-161-231-9527 (FAX)
Dr. Thomas Forste
Jens Norden

ITT Bauelemente
ELKOSE GmbH -
Elektronik Vertrieb
Bahnhofstr. 44
D-7141 Moeglingen
Germany
(49) 7141-487288 (phone)
(49) 7141-487282 (FAX)
Alfred Buck
(LW Products Only)

Jermyn GmbH
Im Dachsstueck 9
D-6250 Limburg
Germany
(49) 6431-5080 (phone)
(49) 6431-508289 (FAX)
Jochem Pelzer

Lasi Electronica S.P.A
Viale Fulvio Testi, 280
20126 Milano
Italy
(39) 2-66101370 (phone)
(39) 2-66101385 (FAX)
Luciano Sandrini

Macro
Burnham Lane
Slough
SL1 6LN
England
(44) 628-604383 (phone)
(44) 628-666873 (FAX)
Mark Pyne
(FPGA Products Only)

Morgenstierne & Co. A/S
P.O. Box 15 Bogerud
0621 Oslo 6
Norway
(47) 228-9490 (phone)
(47) 228-9494 (FAX)
Jon Arne Kjaerstad

Naxab
P.O. Box 4115,
Hemvaernsgaten 11
S-171 04 Solna
Sweden
(46) 8-985140 (phone)
(46) 8-7634451 (FAX)
Jan Oernjaeger

Distributors/Trading Companies (continued)

Europe (continued)

Optilas
Mill Court
Wolverton Mill
Milton Keynes
MK12 5RE
England
(44) 908-221123 (phone)
(44) 908-221110 (FAX)
Chris Gladding
(Lightwave Products Only)

OY SW-Instruments AB
PL 36, 02211
Espoo
Finland
(358) 90-804-1041 (phone)
(358) 90-881-911 (FAX)
Hannu Kurkinen
(FPGA Products Only)

Powerline Elect. LTD.
5 Nimrod Way
Reading
Berks
RG2 OEB
England
(44) 734-868567 (phone)
(44) 734-755172 (FAX)
Bryn Morgan
(Power Products Only)

Sasco Vertrieb von Elektronischen Bauelementen GmbH
Hermann - Oberth - Str. 16
D - 8011 Putzbrunn
Germany
(49) 89-4611248 (phone)
(49) 89-4611270 (FAX)
Dr. Sentimenti

SMD Electronics Ltd.
P.O. Box 3075/53 Herzl Street
Israel - Netanya 42130
Israel
(972) 2-53-344754 (phone)
(972) 2-53-620935 (FAX)
Isaac Gilmor

Stolz AG
Taefernstr. 15
CH - 5405 Baden - Daettwil
Switzerland
(41) 5684-9000 (phone)
(41) 5683-1963 (FAX)
Andre Leutwiler

Far East

Dia Semicon Systems Inc.
Flowerhill-Shinmachi
23-9, Shinmachi 1-chome
Setagaya-ku, Tokyo 154
Japan
(81) 3-3439-1600 (phone)
(81) 3-3439-1601 (FAX)

JER Corporation
Ohmori Mitsubishi Bldg.
3-10, Sannoh 2-chome
Ohta-ku, Tokyo 143
Japan
(81) 3-3777-4111 (phone)
(81) 3-3773-3333 (FAX)

Kanematsu Electronics
Components Corporation
Shin-Ohsaki Kangyo Bldg. 11F
6-4, Ohsaki 1-chome
Shinagawa-ku, Tokyo 141
Japan
(81) 3-3779-7811 (phone)
(81) 3-3779-7811 (FAX)

TOMEN Electronics Corp.
1-1, Uchisaiwai-cho 20chome
Chiyoda-ku, Tokyo 100
Japan
(81) 3-3506-3657 (phone)
(81) 3-3506-3497 (FAX)

Toyo Comm. Equipment
Co., LTD.
20-4, Nishi-Shimbashi 3-chome
Minato-ku, Tokyo 105
Japan
(81) 3-3459-7300 (phone)
(81) 3-3436-1434 (FAX)

Note: For all **specific** price and delivery information, contact your local sales office.

Product Management Offices

Allentown (AL)
555 Union Boulevard
Allentown, PA 18103

Columbus (CB)
Columbus Works
6200 East Broad Street
Columbus, OH 43213

Greensboro (GC)
Guilford Center
PO Box 25000
W1E57
Greensboro, NC 27420

Holmdel (HO)
Crawfords Corner
Holmdel, NJ 07733

Indian Hill West (IW)
1100 East Warrenville Road
Naperville, IL 60566

Merrimack Valley (MV)
Merrimack Valley Works
1600 Osgood Street
North Andover, MA 01845

Oklahoma City (OC)
Oklahoma City Works
7725 W. Reno Avenue
Oklahoma City, OK 73125

Whippany (WH)
Whippany Road
Whippany, NJ 07981

Berkeley Heights II (BK2)
Two Oak Way
Berkeley Heights, NJ 07922

Denver (DR)
11900 N. Pecos
Denver, CO 80234

Heritage Park (INH)
6612 East 75th Street
P.O. Box 1008
Indianapolis, IN 46206

Indian Hill South (IX)
1200 East Warrenville Road
Naperville, IL 60566

Little Rock (TY)
7600 Interstate 30
Little Rock, AR 72209

Middletown (MT)
200 Laurel Avenue
Middletown, NJ 07748

Shreveport (SP)
9630 Interport Drive
P.O. Box 311
Shreveport, LA 71130-1111

Section 9.
Ordering Information

Section 9. Ordering Information

Table Of Contents

Contents **Page**

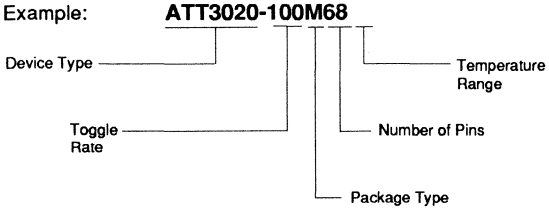
AT&T Microelectronics

AT&T FPGA Ordering Information

Ordering Information

FPGA Ordering Information

Example: **ATT3020-100M68**



ATT3020, 100 MHz, 68-lead PLCC, Commercial Temperature

Note: For availability of device types or packaging options, please contact your AT&T Sales Representative or an authorized distributor.

For more package information, including outline drawings, dimensions, assembly, and thermal data, see Section 7.

FPGA TEMPERATURE OPTIONS

Symbol	Description	Temperature
(Blank)	Commercial	0 °C to 70 °C
I	Industrial	–40 °C to +85 °C
M	Military	–50 °C to +125 °C

FPGA PACKAGE OPTIONS

Symbol	Description
H	Plastic Pin Grid Array
J	EIAJ Plastic Quad Flat Package
M	Plastic Leaded Chip Carrier
N	JEDEC Ceramic Quad Flat Package
R	Ceramic Pin Grid Array
Q	Square Plastic Quad Flat Package

		44 Pin	68 Pin	84 Pin		100 Pin		132 Pin		160 Pin	164 Pin	175 Pin		208 Pin
		PLCC	PLCC	PLCC	Ceramic PGA	EIAJ QFP	Ceramic QFP	Plastic PGA	Ceramic PGA	EIAJ QFP	Ceramic QFP	Plastic PGA	Ceramic PGA	Plastic SQFP
		M44	M68	M84	R84	J100	N100	H132	R132	J160	N164	H175	R175	Q208
ATT3020	–70		CI	CI	CI	CI	C							
	–100		CI	CI	CI	CI	C							
	–125		CI	CI	CI	CI	CI							
	–150		C	C	C	C	C							
ATT3030	–70	CI	CI	CI	CI	CI								
	–100	CI	CI	CI	CI	CI								
	–125	CI	CI	CI	CI	CI								
	–150	C	C	C	C	C								
ATT3042	–70			CI	CI	CI	C	CI	CI					
	–100			CI	CI	CI	C	CI	CI					
	–125			CI	CI	CI	CI	CI	CI					
	–150			C	C	C	C	C	C					
ATT3064	–70			CI				CI	CI	CI				
	–100			CI				CI	CI	CI				
	–125			CI				CI	CI	CI				
	–150			C				C	C	C				
ATT3090	–70			CI						CI	CI	CI	CI	CI
	–100			CI						CI	CI	CI	CI	CI
	–125			CI						CI	CI	CI	CI	CI
	–150			C						C	C	C	C	C

Notes:
C indicates commercial temperature option.
I indicates industrial temperature option.

Ordering Information (continued)

Serial PROM and E² ROM Ordering Information

Device	Part Number	Comcode
ATT1736	ATT1736-P8	106517790
ATT1765	ATT1765-P8	106517808
ATT17128	ATT17128-P8	106517816

Device	Part Number	Comcode
ATT1736F	ATT1736F-P8	106766017
ATT1765F	ATT1765F-P8	106766009
ATT17128F	ATT17128F-P8	106765993