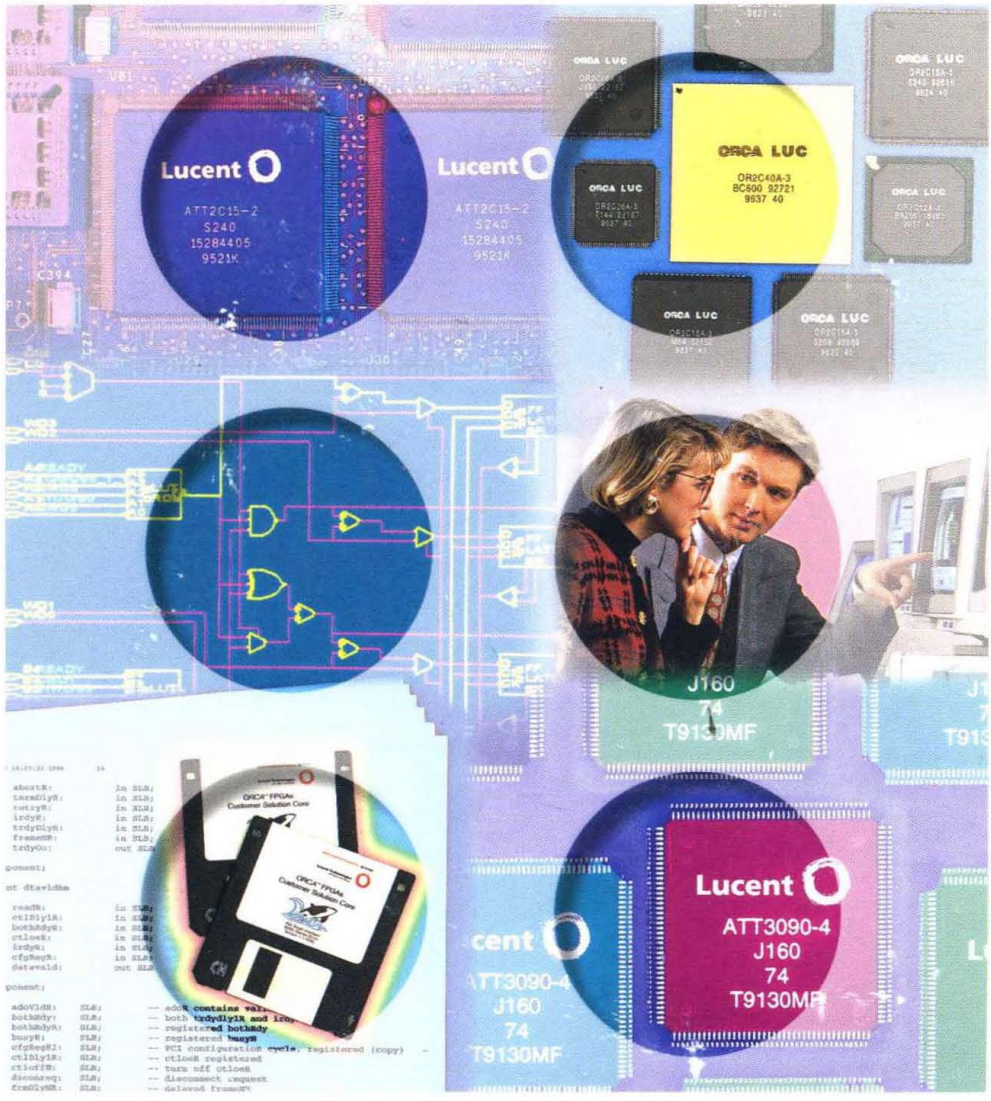
Data Book
October 1996

microelectronics group

Lucent Technologies
Bell Labs Innovations

Field-Programmable Gate Arrays



Data Book
October 1996

Field-Programmable Gate Arrays

microelectronics group

Lucent Technologies
Bell Labs Innovations

FPGA Data Book

October 1996

A Word About Trademarks. . . .

The following trademarks owned by Lucent Technologies are used in this document:

ORCA™ *SCUBA*™ *TAPDANCE*® *WE*®

The following trademarks owned by other entities are used in this document:

Advantest is a registered trademark of Advantest Corporation.

Cadence is a registered trademark of Cadence Design Systems, Inc.

CORE, Exemplar Logic, and *Galileo* are trademarks of Exemplar Logic, Inc.

HP is a registered trademark of Hewlett-Packard Company.

IBM is a registered trademark of International Business Machines Corporation.

IEEE is a registered trademark of The Institute of Electrical and Electronics Engineers, Inc.

Intel is a registered trademark of Intel Corporation.

Logic Innovations is a registered trademark of Logic Innovations, Inc.

Logic Modeling is a registered trademark of Logic Modeling Corporation.

MS-DOS, Microsoft, and *Windows* are registered trademarks of Microsoft Corporation.

Mentor Graphics is a registered trademark and *Autologic, Design Architect,* and *QuickSim* are trademarks of Mentor Graphics Corporation.

Model Technology is a registered trademark and *V-System/PLUS* is a trademark of Model Technology, Inc.

Mosaic and *Navigator* are trademarks of Netscape Communications Corporation.

MOTIVE is a registered trademark of Quad Design Technologies, Inc.

Motorola is a registered trademark of Motorola, Inc.

NEC is a trademark of NEC Electronics, Inc.

Open Look is a registered trademark of Novell, Inc. in the United States and other countries, licensed exclusively through X/Open Company, Ltd.

OSF/MOTIF is a registered trademark of Open Software Foundation.

PALASM is a registered trademark of Monolithic Memories, Inc.

Pentium and *Pentium Pro* are trademarks of Intel Corporation.

PKZIP and *PKUNZIP* are registered trademarks of PKWARE, Inc.

PREP is a trademark of Programmable Electronics Performance Corporation.

SmartModel is a registered trademark of Logic Modeling Group of Synopsys, Inc.

All *SPARC* trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the United States and other countries. Products bearing *SPARC* trademarks are based upon an architecture developed by Sun Microsystems, Inc.

Sun, Sun Microsystems, the *Sun* logo, *Sun Workstation, SunOS, Sun-4,* and *Solaris* are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Synopsys and *DesignWare* are registered trademarks and *Design Compiler, FPGA Compiler, HDL Compiler,* and *VHDL Compiler* are trademarks of Synopsys, Inc.

Synplicity is a registered trademark and *Synplify* is a trademark of Synplicity, Inc.

TEFLON is a registered trademark of DuPont Corporation.

Tektronix is a registered trademark of Tektronix, Inc.

UL is a registered trademark of Underwriters Laboratories, Inc.

UNIX is a registered trademark licensed exclusively through X/Open Company, Ltd.

Valid is a trademark of Valid Logic Systems.

Verilog and *Verilog-XL* are registered trademarks and *Veritime* is a trademark of Cadence Design Systems, Inc.

PROCapture, PROGen, PRODatapath, PROSeries, PROSim, PROSyn, PROWave, SpeedWave, ViewDatapath, ViewDraw, ViewGen, ViewLog, ViewSyn, ViewSynthesis, ViewTrace, ViewWave, and *WorkView Office* are trademarks and *PowerView, Viewlogic,* and *ViewSim* are registered trademarks of Viewlogic Systems, Inc.

XACT, Xilinx, XC1700, XC2700, XC3000, XC3090, XC3100, XC3700, and *XC4000* are registered trademarks and *EPIC, Foundry, Timing Wizard, TRACE, XC3020, XC3030, XC3042, XC3064, XC3120, XC3130, XC3142, XC3164,* and *XC3190* are trademarks of Xilinx, Inc.

X-Windows is a trademark of the Massachusetts Institute of Technology.

Lucent Technologies' 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 Lucent Technologies. 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.

Henry B. Schacht
Chairman of the Board and Chief Executive Officer

Contents

Table of Contents

Package Information ..... Chapter 5

Qualification Information ..... Chapter 6

Masked Array Conversion for *ORCA* ("MACO") ..... Chapter 7

Applications ..... Chapter 8

Technical Support ..... Chapter 9

Sales Offices ..... Chapter 10

Lucent Technologies
Bell Labs Innovations

General Information

About Lucent Technologies

In 1996, AT&T divested into three different entities: AT&T, NCR, and Lucent Technologies. Had Lucent Technologies been incorporated in 1995, it would have been a $21 billion company with more than 130,000 employees.

Today, just over a decade after entering the merchant semiconductor market, the Microelectronics Group of Lucent Technologies (Lucent), formerly known as AT&T Microelectronics, has emerged as a technology leader. In this short time, Lucent has become a world-class leader by providing systems and solutions for unique applications that enable our customers to deliver voice, data, image, and video communications at anytime, from anywhere.

Lucent Technologies is comprised of the following groups:

- **Bell Laboratories** is the research and development arm for the new company.
- **Business Communications Systems** develops, manufactures, markets, and services advanced communications products and systems for business customers.
- **Consumer Products** designs, manufactures, sells, services, and leases communications products for consumers.
- **Microelectronics** designs and manufactures high-performance integrated circuits, optoelectronic components, power systems, and printed-circuit boards for applications in the telecommunications and computing industries.
- **Multimedia Ventures and Technologies** provides advanced technology systems and support services for government and commercial customers, and manufactures and distributes network access products.
- **Network Systems** provides public networking systems and software to telecommunications providers and cable companies.

Lucent has been successfully growing both in revenue and number of customers served. This growth has been driven by providing innovative solutions that are built on the foundations of digital signal processing, wireless, networked computing, and communications technologies. Lucent has focused on using its high level of expertise in these technologies to provide solutions that enable our customers to succeed by making their product different.

Our customers are relying on our products to provide vital competitive advantages across a full range of the fastest growing applications: personal computers and workstations; wireless communications, such as cellular phones; voice/data/video switches; multimedia; consumer telephony products; and other high-volume electronic systems. As a result, there has been a steadily increasing demand for integrated circuits that has enabled Lucent to grow faster than the semiconductor industry average.

Unparalleled ASIC Solutions

Chances are very good that when you power up a workstation desktop or laptop PC, one or more of our integrated circuits or chip sets is serving you. Our integrated microperipherals are based on standard-cell architectures for powerful, cost-effective, application-specific integrated circuits (ASIC) solutions.

ASICs are a growth business. We are the largest standard-cell company in the world, in a market growing at a tremendous rate each year. Lucent's strengths and commitment to standard-cell ASICs translate well to our *ORCA* FPGAs. Three critical requirements of any standard-cell customer mirror those of sophisticated FPGA customers: world-class technical support, process technology that enables high-density and high-performance designs, and CAD tools that leverage that technology. In an effort to maintain our lead, Lucent Technologies is investing heavily in our core foundations: people, process technology, and CAD tools. This investment will ultimately benefit our customers by providing quick time-to-market, cost-effective ASIC solutions.

The Lucent FPGA Solution

Field-programmable gate arrays (FPGAs) are versatile logic ICs which can be programmed to perform the desired function by the customer. Compared to masked-programmable gate arrays, FPGAs allow designers to bring products to market in far less time. This time-to-market advantage has fueled a 53% annual growth rate for FPGAs in the last three years, making it the largest growing segment in all of the electronics industry.

In 1994 we introduced the world's highest-capacity FPGA with 26,000 usable gates. 1995 brought a 40,000 usable gate device: the *ORCA* ATT2C40—the highest density of any FPGA on the market. It rivals the speed and performance of gate arrays, while offering enhanced routability.

Lucent's complete *ORCA* Series FPGAs offers 3,500—40,000 usable gates and up to 480 user I/Os in 0.6 µm, 0.5 µm, and 0.35 µm CMOS technologies. *ORCA* FPGAs offer a real alternative to low-end gate arrays with their area-efficient architecture and 0.5 µm and 0.35 µm performance. The SRAM-based 0.6 µm and 0.55 µm ATT3000 Series FPGAs, which feature toggle rates up to 270 MHz, are a direct second source to the *Xilinx XC3000* FPGAs and are pin-for-pin and speed compatible with the *Xilinx XC3100* FPGAs.

In 1996 we are introducing two enhanced versions of the *ORCA* series of FPGAs: the OR2CxxA series (5 V operation), and the OR2TxxA series (3.3 V operation). In addition to being processed using advanced 0.35 µm processing, these devices also include many new features, such as synchronous single-port and dual-port RAM and enhanced multiplier support.

Both the *ORCA* and the ATT3000 Series FPGAs are supported by PC and workstation software from popular third-party software tool vendors, such as *Synopsys, Mentor Graphics, Viewlogic,* Cadence, *Exemplar Logic, Synplicity,* and many others. Lucent also supplies the map/place/route implementation software known as *ORCA* Foundry to support both series of devices.

Lucent's Commitment to FPGAs

Lucent Technologies has established itself as the high-performance, high-density leader in FPGAs. Several industry leaders currently use both the *ORCA* family and the ATT3000 family of FPGAs to accommodate increasingly sophisticated designs.

Whatever your needs, you can look to Lucent today as the vendor offering a total ASIC solution. Since Lucent offers FPGAs and mask-programmed gate arrays and is the largest supplier of standard-cell devices, FPGA designs that require a high volume of devices can be quickly and efficiently migrated to other, more cost-effective, technologies.

The Importance of Quality

We view quality from many different vantage points. It encompasses every facet of our business. From a design and manufacturing view, it starts with initial product concepts and extends all the way through product and technical support. This is where we build in the reliability, performance, and commitment that are evident long after the product is purchased. Quality service is the element we rely on to turn our corporate business relationships into successful, long-term personal endeavors.

Another measure of our success is the awards we receive. These achievements tell customers that we are operating at the highest standard levels in manufacturing, business management, and customer service.

Some of our recent award highlights:

- All of our manufacturing facilities have received ISO 9000 certification.
- Our Orlando facility received the prestigious Shingo Prize in manufacturing, as well as the Occupational Safety and Health Administration's "Star" designation for safety in the workplace.
- The Power Systems Group, a Lucent business unit, became the first American manufacturing company to win the widely esteemed Deming Prize.

The Deming Prize

Lucent Technologies Microelectronics Group's Power Systems, a Lucent business unit, became the first American manufacturing company to win the widely esteemed Deming Prize. The prize was established in 1951 by the Union of Japanese Scientists and Engineers and is regarded in industrial circles as the award with the most demanding and challenging criteria for measuring company performance. It was named for Dr. W. Edward Deming, the foremost promoter of quality control in the United States. Deming has been lauded for bringing (then) newly innovative statistical quality controls methods to post-World War II Japan's rebuilding efforts.

Leaders in Environmental Awareness

Every business should be responsible for setting high standards for environmental health. This attitude just makes good business sense, but it is also essential to ensure an environmentally sound future for our world. Lucent has often led the way in protecting the environment and establishing industrial environmental standards, both by example and by sharing our successes and their associated technologies with other companies worldwide. For example, we developed technologies that enabled us to eliminate all regulated Class I ozone-depleting substances from our manufacturing operations worldwide ahead of the mandated schedule. We are sharing this process with other industries who need this technology to meet the environmental standards set by the government.

Notes

Lucent Technologies
Bell Labs Innovations

Overview of Lucent Technologies FPGAs

Overview

Field-programmable gate arrays (FPGAs) have emerged as an attractive alternative to customized VLSI for implementing digital logic functions. Their user-programmable nature allows them to provide quick turnaround time of a design with low risk, allowing the designer to make logic changes at any time. As FPGA device densities and speeds continue to increase, many new applications that previously could only be implemented with mask-programmed devices, such as gate arrays and standard cells, can now be designed with FPGAs. Also, as the demand for the reprogrammable nature of FPGAs increases, many new types of systems that could not even be considered before can now be created using FPGAs.

Lucent Technologies has introduced two series of FPGAs: the *ORCA* Series and the ATT3000 Series. Lucent's advanced 0.6 µm, 0.55 µm, 0.5 µm, and 0.35 µm CMOS process technologies and innovative FPGA architectures make these devices some of the densest and fastest FPGAs available. Also, since all of these FPGAs use CMOS SRAM technology, their standby power consumption is very low.

This section describes all of the current FPGAs available from Lucent Technologies, along with information on the serial ROM that can be used to program the FPGAs. The tables below summarize the availability of all of these FPGA devices, as well as the packages and speed grades they are offered in.

Overview (continued)

Table 1. ATT3000 Package Matrix

Device	Speed	44-Pin PLCC M44	68-Pin PLCC M68	84-Pin PLCC M84	100-Pin QFP J100	100-Pin TQFP T100	132-Pin PPGA H132	144-Pin TQFP T144	160-Pin QFP J160	175-Pin PPGA H175	208-Pin SQFP S208
ATT3020	-70	—	CI	CI	CI	—	—	—	—	—	—
	-100	—	CI	CI	CI	—	—	—	—	—	—
	-125	—	CI	CI	CI	—	—	—	—	—	—
	-5	—	CI	CI	CI	—	—	—	—	—	—
	-4	—	C	C	C	—	—	—	—	—	—
	-3	—	C	C	C	—	—	—	—	—	—
ATT3030	-70	CI	CI	CI	CI	CI	—	—	—	—	—
	-100	CI	CI	CI	CI	CI	—	—	—	—	—
	-125	CI	CI	CI	CI	CI	—	—	—	—	—
	-5	CI	CI	CI	CI	CI	—	—	—	—	—
	-4	C	C	C	C	C	—	—	—	—	—
	-3	C	C	C	C	C	—	—	—	—	—
ATT3042	-70	—	—	CI	CI	CI	CI	CI	—	—	—
	-100	—	—	CI	CI	CI	CI	CI	—	—	—
	-125	—	—	CI	CI	CI	CI	CI	—	—	—
	-5	—	—	CI	CI	CI	CI	CI	—	—	—
	-4	—	—	C	C	C	C	C	—	—	—
	-3	—	—	C	C	C	C	C	—	—	—
ATT3064	-70	—	—	CI	—	CI	CI	CI	CI	—	—
	-100	—	—	CI	—	CI	CI	CI	CI	—	—
	-125	—	—	CI	—	CI	CI	CI	CI	—	—
	-5	—	—	CI	—	CI	CI	CI	CI	—	—
	-4	—	—	C	—	C	C	C	C	—	—
	-3	—	—	C	—	C	C	C	C	—	—
ATT3090	-70	—	—	CI	—	—	—	—	CI	CI	CI
	-100	—	—	CI	—	—	—	—	CI	CI	CI
	-125	—	—	CI	—	—	—	—	CI	CI	CI
	-5	—	—	CI	—	—	—	—	CI	CI	CI
	-4	—	—	C	—	—	—	—	C	C	C
	-3	—	—	C	—	—	—	—	C	C	C

Key: C = commercial, I = industrial.

Overview (continued)

Table 2. *ORCA* OR2CxxA/OR2TxxA Series Package Matrix

Packages	84-Pin PLCC	100-Pin TQFP	144-Pin TQFP	160-Pin QFP	208-Pin EIAJ SQFP/ SQFP2	240-Pin EIAJ SQFP/ SQFP2	256-Pin PBGA	304-Pin EIAJ SQFP/ SQFP2	352-Pin PBGA	432-Pin EBGA	600-Pin EBGA
	M84	T100	T144	J160	S208/ PS208	S240/ PS240	BA256	S304/ PS304	BA352	BC432	BC600
OR2C/2T04A	CI	CI	CI	CI	CI	—	—	—	—	—	—
OR2C/2T06A	CI	CI	CI	CI	CI	CI	CI	—	—	—	—
OR2C/2T08A	CI	—	—	CI	CI	CI	CI	—	—	—	—
OR2C/2T10A	CI	—	—	CI	CI	CI	CI	—	CI	—	—
OR2C/2T12A	CI	—	—	—	CI	CI	CI	CI	CI	—	—
OR2C/2T15A	CI	—	—	—	CI	CI	CI	CI	CI	CI	—
OR2C/2T26A	—	—	—	—	CI	CI	—	CI	CI	CI	CI
OR2C/2T40A	—	—	—	—	CI	CI	—	CI	—	CI	CI

Key: C = commercial, I = industrial.

Notes:
The package options with the SQFP/SQFP2 designation in the table above use the SQFP package for all densities up to and including the OR2C/2T15A, while the OR2C/2T26A and the OR2C/2T40A use the SQFP2. Availability of the OR2C15A in the 208-pin and 240-pin SQFP2 is to be announced.

The OR2TxxA series is not offered in the 304-pin SQFP/SQFP2 packages.

Table 3. *ORCA* ATT2Cxx Series Package Matrix

Packages	84-Pin PLCC	100-Pin TQFP	144-Pin TQFP	160-Pin QFP	208-Pin EIAJ SQFP/ SQFP-PQ2	240-Pin EIAJ SQFP/ SQFP-PQ2	256-Pin PBGA	304-Pin EIAJ SQFP/ SQFP-PQ2	364-Pin CPGA	428-Pin CPGA
	M84	T100	T144	J160	S208/ PS208	S240/ PS240	B256	S304/ PS304	R364	R428
ATT2C04	CI	CI	CI	CI	CI	—	—	—	—	—
ATT2C06	CI	CI	CI	CI	CI	CI	—	—	—	—
ATT2C08	CI	—	—	CI	CI	CI	CI	CI	—	—
ATT2C10	CI	—	—	CI	CI	CI	CI	CI	—	—
ATT2C12	—	—	—	—	CI	CI	CI	CI	CI	—
ATT2C15	—	—	—	—	CI	CI	—	CI	CI	—
ATT2C26	—	—	—	—	CI	CI	—	CI	—	CI
ATT2C40	—	—	—	—	CI	CI	—	CI	—	CI

Key: C = commercial, I = industrial.

Note: The package options with the SQFP/SQFP-PQ2 designation in the table above use the SQFP package for all densities up to and including the ATT2C15, while the ATT2C26 uses the SQFP-PQ2 package (chip-up orientation), and the ATT2C40 uses the SQFP-PQ2 package (chip-down orientation).

Table 4. ATT1700A Serial ROM Package Matrix

Device	Size	8-Pin DIP	8-Pin SOIC	20-Pin PLCC
		P8	SO8	M20
ATT1736A	36,288	CI	CI	CI
ATT1765A	65,536	CI	CI	CI
ATT17128A	131,072	CI	CI	CI

Key: C = commercial, I = industrial.

Notes

Lucent Technologies
Bell Labs Innovations

ORCA OR2CxxA (5.0 V) and OR2TxxA (3.3 V) Series Series Field-Programmable Gate Arrays

Features

- High-performance, cost-effective, low-power 0.35 µm CMOS technology (four-input look-up table delay less than 2.1 ns with -4 speed grade, less than 1.7 ns with preliminary -5 speed grade)
- High density (up to 43,200 usable, logic-only gates; or 99,400 gates including RAM)
- Up to 480 user I/Os (OR2TxxA I/Os are 5 V tolerant to allow interconnection to both 3.3 V and 5 V devices, selectable on a per-pin basis)
- Four 16-bit look-up tables and four latches/flip-flops per PFU, nibble-oriented for implementing 4-, 8-, 16-, and/or 32-bit (or wider) bus structures
- Fast on-chip user SRAM has features to simplify RAM design and increase RAM speed:
  — Asynchronous single port: 64 bits/PFU
  — Synchronous single port: 64 bits/PFU
  — Synchronous dual port: 32 bits/PFU
- Improved ability to combine PFUs to create larger RAM structures using write-port enable
- Fast, dense multipliers can be created with the multiplier mode (4 x 1 multiplier/PFU):
  — 8 x 8 multiplier requires only 16 PFUs
  — 30% increase in speed
- Flip-flop/latch options to allow programmable priority of synchronous set/reset vs. clock enable
- Enhanced cascadable nibble-wide data path capabilities for comparators and multiplexers
- Innovative, abundant, and hierarchical nibble-oriented routing resources that allow automatic use of internal gates for all device densities without sacrificing performance
- Internal fast-carry for arithmetic functions
- Upward bit stream compatible from the *ORCA* ATT2Cxx/ATT2Txx series of devices
- TTL or CMOS input levels programmable per pin for the OR2CxxA (5.0 V) devices
- Individually programmable drive capability: 12 mA sink/6 mA source or 6 mA sink/3 mA source
- Built-in boundary scan (*IEEE*1149.1)
- Full PCI bus compliance
- Supported by industry-standard CAE tools for design entry, synthesis, and simulation with *ORCA* Foundry Development System support (for back-end implementation)

Table 1. *ORCA* OR2CxxA/OR2TxxA Series FPGAs

Device	Usable Gates*	Latches/FFs	Max User RAM Bits	User I/Os	Array Size
OR2C04A/OR2T04A	4,800—11,000	400	6,400	160	10 x 10
OR2C06A/OR2T06A	6,900—15,900	576	9,216	192	12 x 12
OR2C08A/OR2T08A	9,400—21,600	784	12,544	224	14 x 14
OR2C10A/OR2T10A	12,300—28,300	1024	16,384	256	16 x 16
OR2C12A/OR2T12A	15,600—35,800	1296	20,736	288	18 x 18
OR2C15A/OR2T15A	19,200—44,200	1600	25,600	320	20 x 20
OR2C26A/OR2T26A	27,600—63,600	2304	36,864	384	24 x 24
OR2C40A/OR2T40A	43,200—99,400	3600	57,600	480	30 x 30

* The first number in the usable gates column assumes 48 gates per PFU (12 gates per 4-input LUT/FF pair) for logic-only designs. The second number assumes 30% of a design is RAM. PFUs used as RAM are counted at 4 gates per bit, with each PFU capable of implementing a 16 x 4 RAM (or 256 gates) per PFU.

Table of Contents

Description

The *ORCA* OR2CxxA/OR2TxxA series of SRAM-based FPGAs are an enhanced version of the *ORCA* 2C/2T architecture. The latest *ORCA* series includes patented architectural enhancements that make functions faster and easier to design while conserving the use of PLCs and routing resources.

The OR2CxxA/OR2TxxA devices can be used as drop-in replacements for the ATT2Cxx/ATT2Txx series, respectively, and they are also bit stream compatible with each other. Both series of devices are implemented using two 0.35 µm processes: one is optimized for 5.0 V operation, and the other is optimized for 3.3 V operation to allow equivalent system speeds at less than half the power. The usable gate counts associated with each series are provided in Table 1. Both series are offered in a variety of packages, speed grades, and temperature ranges.

The *ORCA* series FPGA consists of two basic elements: programmable logic cells (PLCs) and programmable input/output cells (PICs). An array of PLCs is surrounded by PICs as shown in Figure 1. Each PLC contains a programmable function unit (PFU). The PLCs and PICs also contain routing resources and configuration RAM. All logic is done in the PFU. Each PFU contains four 16-bit look-up tables (LUTs) and four latches/flip-flops (FFs).

The PLC architecture provides a balanced mix of logic and routing which allows a higher utilized gate/PFU than alternative architectures. The routing resources carry logic signals between PFUs and I/O pads. The routing in the PLC is symmetrical about the horizontal and vertical axes. This improves routability by allowing a bus of signals to be routed into the PLC from any direction.

Some examples of the resources required and the performance that can be achieved using these devices are represented in Table 2.

The FPGA's functionality is determined by internal configuration RAM. The FPGA's internal initialization/configuration circuitry loads the configuration data at powerup or under system control. The RAM is loaded by using one of several configuration modes. The configuration data resides externally in an EEPROM, EPROM, or ROM on the circuit board, or any other storage media. Serial ROMs provide a simple, low pin count method for configuring FPGAs.

Table 2. *ORCA* OR2CxxA/OR2TxxA System Performance

Function	# PFUs	Speed Grade				Unit
		-2	-3	-4	-5	
16-bit loadable up/down counter	4	51	67	87	102	MHz
16-bit accumulator	4	51	67	87	102	MHz
8 x 8 parallel multiplier:						
— multiplier mode, unpipelined[1]	22	14	19	24	30	MHz
— ROM mode, unpipelined[2]	36	21	28	34	43	MHz
— multiplier mode, pipelined[3]	44	57	76	96	115	MHz
32 x 16 RAM:						
— single port (read and write/cycle)[4]	9	21	27	34	46	MHz
— single port[5]	9	31	43	65	76	MHz
— dual port[6]	16	39	53	84	95	MHz
36-bit parity check (internal)	4	13.9	11.0	9.1	7.3	ns
32-bit address decode (internal)	3.25	12.3	9.5	7.5	6.0	ns

1. Implemented using 4 x 1 multiplier mode (unpipelined), register-to-register, two 8-bit inputs, one 16-bit output.
2. Implemented using a 256 x 8 ROM (unpipelined), register-to-register, one 8-bit input, one fixed operand, one 8-bit output.
3. Implemented using 4 x 1 multiplier mode (fully pipelined), two 8-bit inputs, one 16-bit output (28 of 44 PFUs contain only pipelining registers.)
4. Implemented using 16 x 4 synchronous single-port RAM mode allowing both read and write per clock cycle, including write/read address multiplexer.
5. Implemented using 16 x 4 synchronous single-port RAM mode allowing either read or write per clock cycle, including write/read address multiplexer.
6. Implemented using 16 x 2 synchronous dual-port RAM mode.

Note: Shaded values are preliminary.

Description (continued)

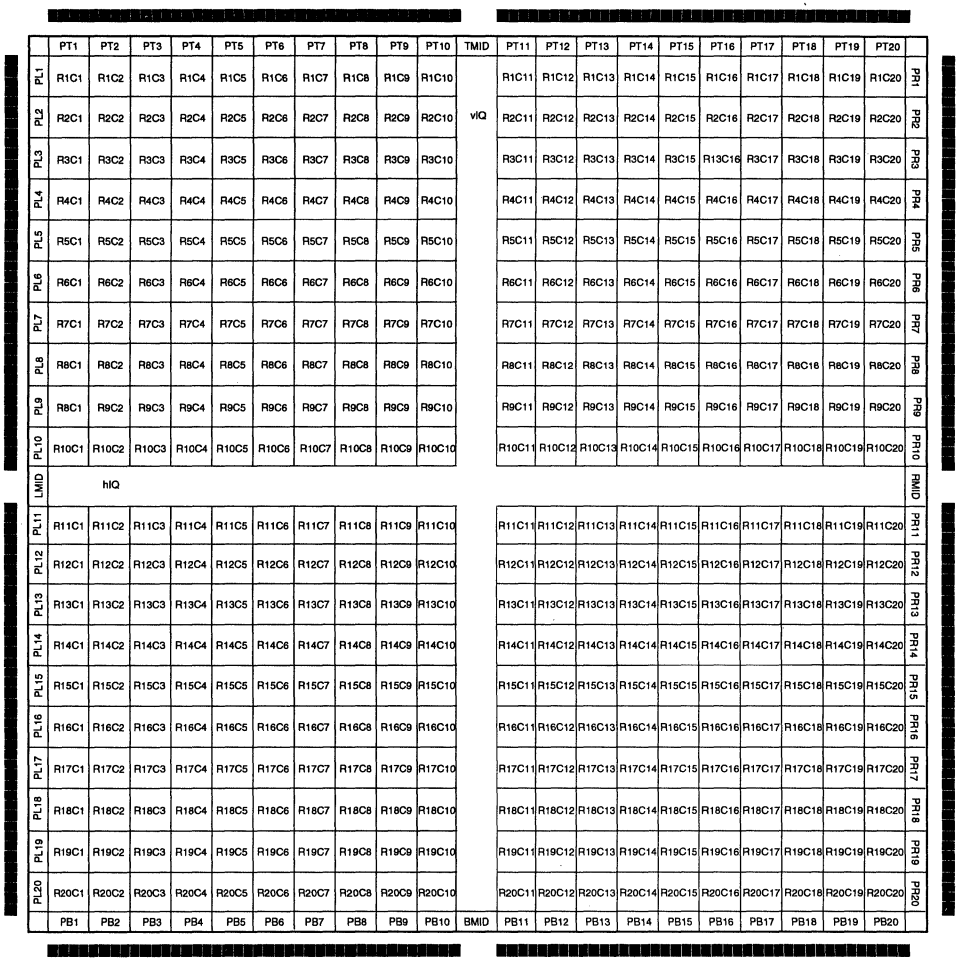


Figure 1. OR2C15A/OR2T15A Array

ORCA Foundry Development System Overview

The *ORCA* Foundry Development System interfaces to front-end design entry tools and provides the tools to produce a configured FPGA. In the design flow, the user defines the functionality of the FPGA at two points in the design flow: at design entry and at the bit stream generation stage.

Following design entry, the development system's map, place, and route tools translate the netlist into a routed FPGA. Its bit stream generator is then used to generate the configuration data which is loaded into the FPGA's internal configuration RAM. When using the bit stream generator, the user selects options that affect the functionality of the FPGA. Combined with the front-end tools, *ORCA* Foundry produces configuration data that implements the various logic and routing options discussed in this data sheet.

Architecture

The *ORCA* Series FPGA is comprised of two basic elements: PLCs and PICs. Figure 1 shows an array of programmable logic cells (PLCs) surrounded by programmable input/output cells (PICs). The OR2C/2T15A has PLCs arranged in an array of 20 rows and 20 columns. PICs are located on all four sides of the FPGA between the PLCs and the IC edge.

The location of a PLC is indicated by its row and column so that a PLC in the second row and third column is R2C3. PICs are indicated similarly, with PT (top) and PB (bottom) designating rows and PL (left) and PR (right) designating columns, followed by a number. The routing resources and configuration RAM are not shown, but the interquad routing blocks (hIQ, vIQ) present in the OR2CxxA/OR2TxxA series are shown.

Each PIC contains the necessary I/O buffers to interface to bond pads. The PICs also contain the routing resources needed to connect signals from the bond pads to/from PLCs. The PICs do not contain any user-accessible logic elements, such as flip-flops.

Combinatorial logic is done in look-up tables (LUTs) located in the PFU. The PFU can be used in different modes to meet different logic requirements. The LUT's configurable medium-/large-grain architecture can be used to implement from one to four combinatorial logic functions. The flexibility of the LUT to handle wide input functions as well as multiple smaller input functions maximizes the gate count/PFU.

The LUTs can be programmed to operate in one of three modes: combinatorial, ripple, or memory. In combinatorial mode, the LUTs can realize any four-, five-, or six-input logic functions. In ripple mode, the high-speed carry logic is used for arithmetic functions, the new multiplier function, or the enhanced data path functions. In memory mode, the LUTs can be used as a 16 x 4 read/write or read-only memory (asynchronous mode or the new synchronous mode) or a new 16 x 2 dual-port memory.

Programmable Logic Cells

The programmable logic cell (PLC) consists of a programmable function unit (PFU) and routing resources. All PLCs in the array are identical. The PFU, which contains four LUTs and four latches/FFs for logic implementation, is discussed in the next section.

Programmable Function Unit

The PFUs are used for logic. Each PFU has 19 external inputs and six outputs and can operate in several modes. The functionality of the inputs and outputs depends on the operating mode.

The PFU uses three input data buses (a[4:0], b[4:0], wd[3:0]), four control inputs (c0, ck, ce, lsr), and a carry input (cin); the last is used for fast arithmetic functions. There is a 5-bit output bus (o[4:0]) and a carry-out (cout).

Figure 2 and Figure 3 show high-level and detailed views of the ports in the PFU, respectively. The ports are referenced with a two- to four-character suffix to a PFU's location. As mentioned, there are two 5-bit input data buses (a[4:0] and b[4:0]) to the LUT, one 4-bit input data bus (wd[3:0]) to the latches/FFs, and an output data bus (o[4:0]).

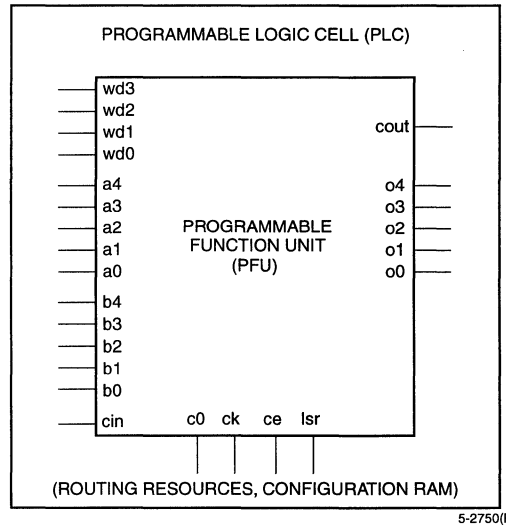
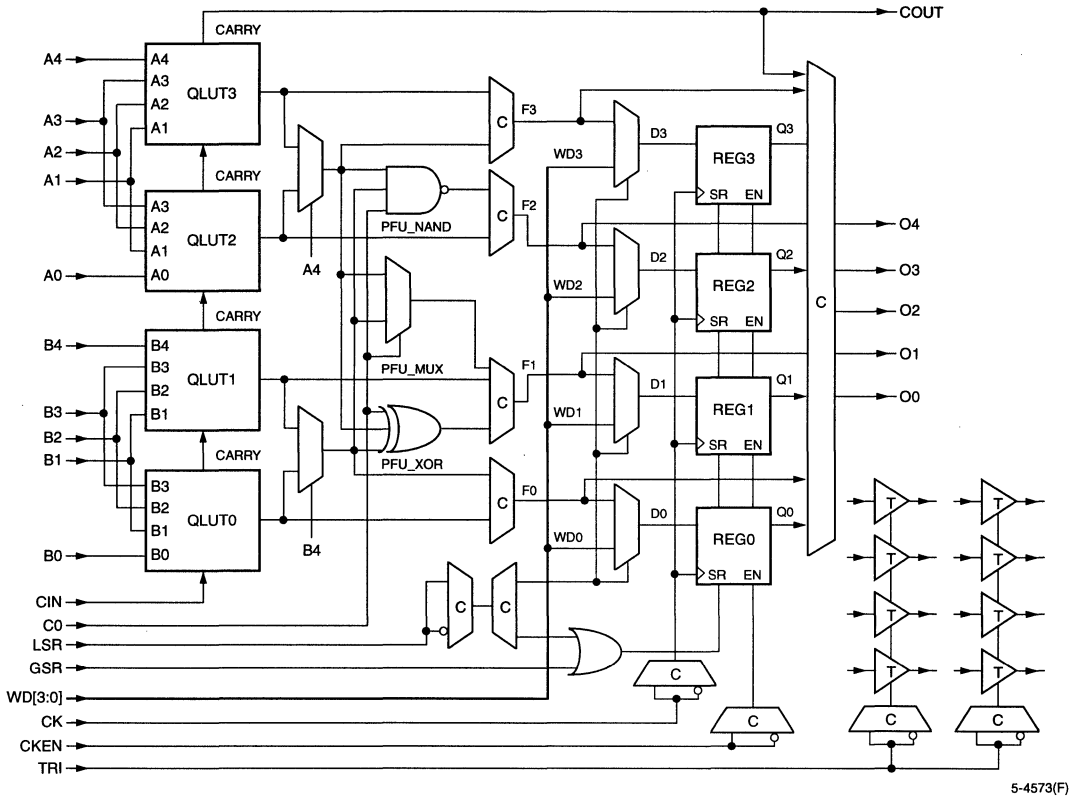


Figure 2. PFU Ports

Programmable Logic Cells (continued)



Key: C = controlled by configuration RAM.

Figure 3. Simplified PFU Diagram

Figure 3 shows the four latches/FFs (REG[3:0]) and the 64-bit look-up table (QLUT[3:0]) in the PFU. The PFU does combinatorial logic in the LUT and sequential logic in the latches/FFs. The LUT is static random access memory (SRAM) and can be used for read/write or read-only memory. The eight 3-state buffers found in each PLC are also shown, although they actually reside external to the PFU.

Each latch/FF can accept data from the LUT. Alternatively, the latches/FFs can accept direct data from wd[3:0], eliminating the LUT delay if no combinatorial function is needed. The LUT outputs can bypass the latches/FFs, which reduces the delay out of the PFU. It is possible to use the LUT and latches/FFs more or less independently. For example, the latches/FFs can be used as a 4-bit shift register, and the LUT can be used to detect when a register has a particular pattern in it.

Programmable Logic Cells (continued)

Table 3 lists the basic operating modes of the LUT. The operating mode affects the functionality of the PFU input and output ports and internal PFU routing. For example, in some operating modes, the wd[3:0] inputs are direct data inputs to the PFU latches/FFs. In the dual 16 x 2 memory mode, the same wd[3:0] inputs are used as a 4-bit data input bus into LUT memory.

The PFU is used in a variety of modes, as illustrated in Figures 4 through 11, and it is these specific modes that are most relevant to PFU functionality.

PFU Control Inputs

The four control inputs to the PFU are clock (ck), local set/reset (lsr), clock enable (ce), and c0. The ck, ce, and lsr inputs control the operation of all four latches in the PFU. An active-low global set/reset (gsrn) signal is also available to the latches/FFs in every PFU. Their operation is discussed briefly here, and in more detail in the Latches/Flip-Flops section. The polarity of the control inputs can be inverted.

The ck input is distributed to each PFU from a vertical or horizontal net. The ce input inhibits the latches/FFs from responding to data inputs. The ce input can be disabled, always enabling the clock. Each latch/FF can be independently programmed to be set or reset by the lsr and the global set/reset (gsrn) signals. Each PFU's lsr input can be configured as synchronous or asynchronous. The gsrn signal is always asynchronous. The lsr signal applies to all four latches/FFs in a PFU. The lsr input can be disabled (the default). The asynchronous set/reset is dominant over clocked inputs.

The c0 input is used as an input into the special PFU gates for wide functions in combinatorial logic mode. In the memory modes, this input is also used as the write-port enable input. The c0 input can be disabled (the default).

Look-Up Table Operating Modes

The look-up table (LUT) can be configured to operate in one of three general modes:

- Combinatorial logic mode
- Ripple mode
- Memory mode

The combinatorial logic mode uses a 64-bit look-up table to implement Boolean functions. The two 5-bit logic inputs, a[4:0] and b[4:0], and the c0 input are used as LUT inputs. The use of these ports changes based on the PFU operating mode.

The functionality of the LUT is determined by its operating mode. The entries in Table 2 show the basic modes of operation for combinatorial logic, ripple, and memory functions in the LUT. Depending on the operating mode, the LUT can be divided into sub-LUTs. The LUT is comprised of two 32-bit half look-up tables, HLUTA and HLUTB. Each half look-up table (HLUT) is comprised of two quarter look-up tables (QLUTs). HLUTA consists of QLUT2 and QLUT3, while HLUTB consists of QLUT0 and QLUT1. The outputs of QLUT0, QLUT1, QLUT2, and QLUT3 are f0, f1, f2, and f3, respectively.

Table 3. Look-Up Table Operating Modes

Mode	Function
F4A	Two functions of four inputs, some inputs shared (QLUT2/QLUT3)
F4B	Two functions of four inputs, some inputs shared (QLUT0/QLUT1)
F5A	One function of five inputs (HLUTA)
F5B	One function of five inputs (HLUTB)
R	4-bit ripple (LUT)
MA	16 x 2 Asynchronous memory (HLUTA)
MB	16 x 2 Asynchronous memory (HLUTB)
SSPM	16 x 4 synchronous single-port memory
SDPM	16 x 2 synchronous dual-port memory

For combinatorial logic, the LUT can be used to do any single function of six inputs, any two functions of five inputs, or four functions of four inputs (with some inputs shared), and three special functions based on the two five-input functions and c0.

Programmable Logic Cells (continued)

The LUT ripple mode operation offers standard arithmetic functions, such as 4-bit adders, subtractors, adder/subtractors, and counters. In the *ORCA* OR2CxxA/OR2TxxA series, there are two new ripple modes available. The first new mode is a 4 x 1 multiplier, and the second is a 4-bit comparator. These new modes offer the advantages of faster speeds as well as denser logic capabilities.

When the LUT is configured to operate in the memory mode, a 16 x 2 asynchronous memory fits into a HLUT. Both the MA and MB modes were available in previous *ORCA* architectures, and each mode can be configured in a HLUT separately. In the *ORCA* OR2CxxA/OR2TxxA series, there are two new memory modes available. The first is a 16 x 4 synchronous single-port memory (SSPM), and the second is a 16 x 2 synchronous dual-port memory (SDPM). These new modes offer easier implementation, faster speeds, denser RAMs, and a dual-port capability that wasn't previously offered as an option.

If the LUT is configured to operate in the ripple mode, it cannot be used for basic combinatorial logic or memory functions. In modes other than the ripple, SSPM, and SDPM modes, combinations of operating modes are possible. For example, the LUT can be configured as a 16 x 2 RAM in one HLUT and a five-input combinatorial logic function in the second HLUT. This can be done by configuring HLUTA in the MA mode and HLUTB in the F5B mode (or vice versa).

F4A/F4B Mode — Two Four-Input Functions

Each HLUT can be used to implement two four-input combinatorial functions, but the total number of inputs into each HLUT cannot exceed five. The two QLUTs within each HLUT share three inputs. In HLUTA, the a1, a2, and a3 inputs are shared by QLUT2 and QLUT3. Similarly, in HLUTB, the b1, b2, and b3 inputs are shared by QLUT0 and QLUT1. The four outputs are f0, f1, f2, and f3. The results can be routed to the d0, d1, d2, and d3 latch/FF inputs or as an output of the PFU. The use of the LUT for four functions of up to four inputs each is given in Figure 4.

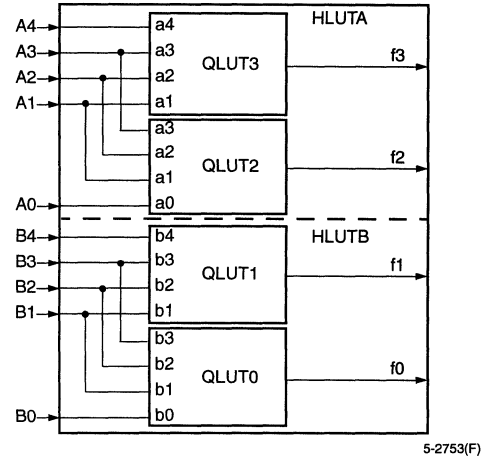


Figure 4. F4 Mode—Four Functions of Four Input Variables

F5A/F5B Mode—One Five-Input Variable Function

Each HLUT can be used to implement any five-input combinatorial function. The input ports are a[4:0] and b[4:0], and the output ports are f0 and f3. One five or less input function is input into a[4:0], and the second five or less input function is input into b[4:0]. The results are routed to the latch/FF d0 and latch/FF d3 inputs, or as a PFU output. The use of the LUT for two independent functions of up to five inputs is shown in Figure 5. In this case, the LUT is configured in the F5A and F5B modes. As a variation, the LUT can do one function of up to five input variables and two four-input functions using F5A and F4B modes or F4A and F5B modes.

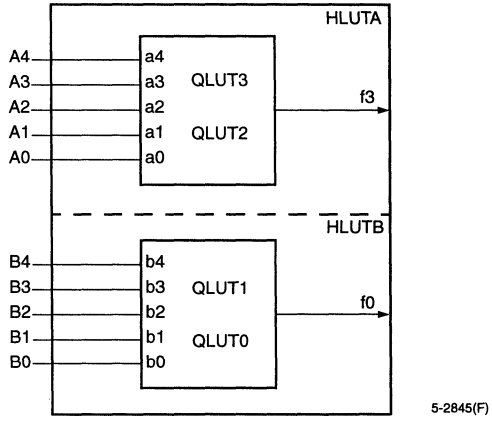


Figure 5. F5 Mode—Two Functions of Five Input Variables

Programmable Logic Cells (continued)

F5M and F5X Modes — Special Function Modes

The PFU contains logic to implement two special function modes which are variations on the F5 mode. As with the F5 mode, the LUT implements two independent five-input functions. Figure 6 and Figure 7 show the schematics for F5M and F5X modes, respectively. The F5X and F5M functions differ from the basic F5A/F5B functions in that there are three logic gates which have inputs from the two 5-input LUT outputs. In some cases, this can be used for faster and/or wider logic functions.

As can be seen, two of the three inputs into the NAND, XOR, and MUX gates, f0 and f3, are from the LUT. The third input is from the c0 input into PFU. Since the c0 input bypasses the LUTs, it has a much smaller delay through the PFU than for all other inputs into the special PFU gates. This allows multiple PFUs to be cascaded together while reducing the delay of the critical path through the PFUs. The output of the first special function (either XOR or MUX) is f1. Since the XOR and MUX share the f1 output, the F5X and F5M modes are mutually exclusive. The output of the NAND PFU gate is f2 and is always available in either mode.

To use either the F5M or F5X functions, the LUT must be in the F5A/F5B mode; i.e., only 5-input LUTs allowed. In both the F5X and F5M functions, the outputs of the five-input combinatorial functions, f0 and f3, are also usable simultaneously with the special PFU gate outputs.

The output of the MUX is:

$f1 = (HLUTA \,\&\, c0) + (HLUTB \,\&\, \overline{c0})$

$f1 = (f3 \,\&\, c0) + (f0 \,\&\, \overline{c0})$

The output of the exclusive OR is:

$f1 = HLUTA \oplus HLUTB \oplus c0$

$f1 = f3 \oplus f0 \oplus c0$

The output of the NAND is:

$f2 = \overline{HLUTA \,\&\, HLUTB \,\&\, c0}$

$f2 = \overline{f3 \,\&\, f0 \,\&\, c0}$

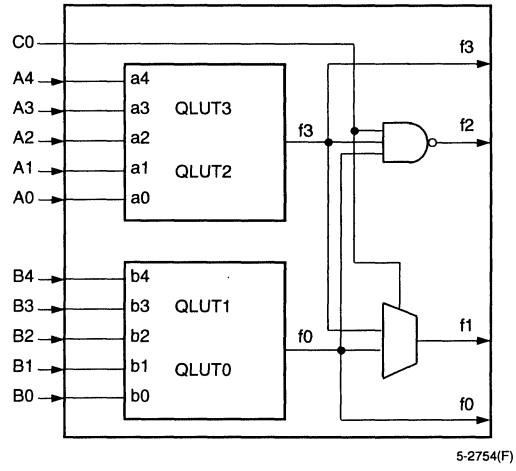


Figure 6. F5M Mode—Multiplexed Function of Two Independent Five-Input Variable Functions

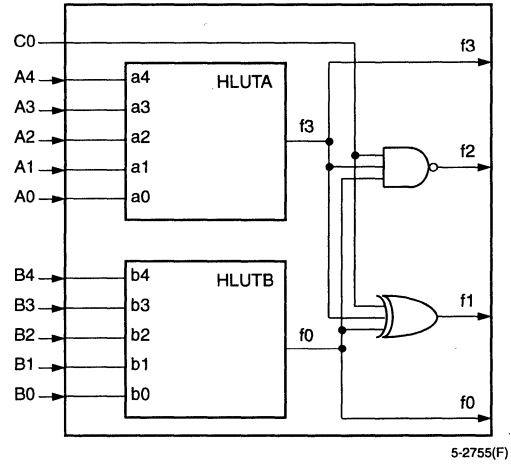


Figure 7. F5X Mode—Exclusive OR Function of Two Independent Five-Input Variable Functions

Programmable Logic Cells (continued)

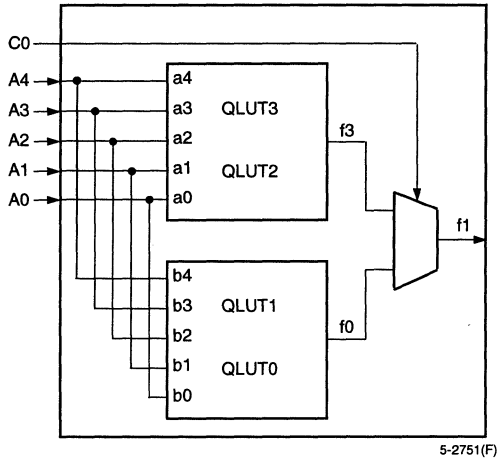


Figure 8. F5M Mode—One Six-Input Variable Function

F5M Mode — One Six-Input Variable Function

The LUT can be used to implement any function of six input variables. As shown in Figure 8, five input signals (A[4:0]) are routed into both the a[4:0] and b[4:0] ports, and the c0 port is used for the sixth input. The output port is f1.

Ripple Mode

The LUT can do nibble-wide ripple functions with high-speed carry logic. Each QLUT has a dedicated carry-out net to route the carry to/from the adjacent QLUT. Using the internal carry circuits, fast arithmetic and counter functions can be implemented in one PFU. Similarly, each PFU has carry-in (CIN) and carry-out (COUT) ports for fast-carry routing between adjacent PFUs.

The ripple mode is generally used in operations on two 4-bit buses. Each QLUT has two operands and a ripple (generally carry) input, and provides a result and ripple (generally carry) output. A single bit is rippled from the previous QLUT and is used as input into the current QLUT. For QLUT0, the ripple input is from the PFU CIN port. The CIN data can come from either the fast-carry routing or the PFU input b4, or it can be tied to logic 1 or logic 0.

The result output and ripple output are calculated by using generate/propagate circuitry. In ripple mode, the two operands are input into a[3:0] and b[3:0]. The four result bits, one per QLUT, are f[3:0] (see Figure 9). The ripple output from QLUT3 can be routed to dedicated carry-out circuitry into any of four adjacent PLCs, or it can be placed on the o4 PFU output, or both. This allows the PLCs to be cascaded in the ripple mode so that nibble-wide ripple functions can be expanded easily to any length.

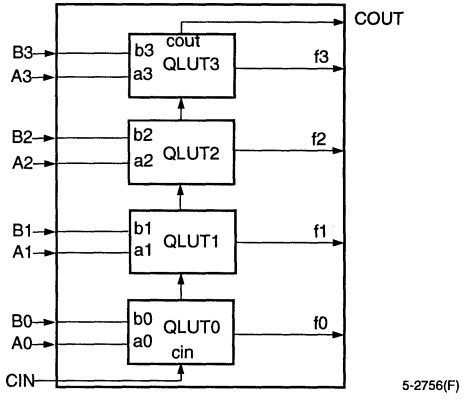


Figure 9. Ripple Mode

The ripple mode can be used in one of four submodes. The first of these is **adder/subtractor mode**. In this mode, each QLUT generates two separate outputs. One of the two outputs selects whether the carry-in is to be propagated to the carry-out of the current QLUT or if the carry-out needs to be generated. The result of this selection is placed on the carry-out signal, which is connected to the next QLUT or the COUT signal, if it is the last QLUT (QLUT3).

The other QLUT output creates the result bit for each QLUT that is connected to f[3:0]. If an adder/subtractor is needed, the control signal to select addition or subtraction is input on a4. The result bit is created in one-half of the QLUT from a single bit from each input bus, along with the ripple input bit. These inputs are also used to create the programmable propagate.

Programmable Logic Cells (continued)

The second submode is the **counter submode** (see Figure 10). The present count is supplied to input a[3:0], and then output f[3:0] will either be incremented by one for an up counter or decremented by one for a down counter. If an up counter or down counter is needed, the control signal to select the direction (up or down) is input on a4. Generally, the latches/FFs in the same PFU are used to hold the present count value.

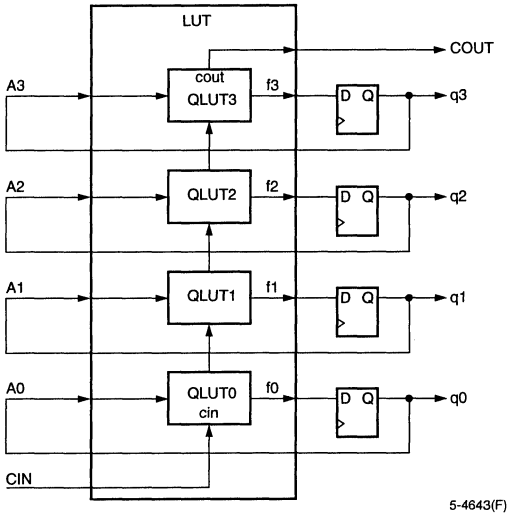


Figure 10. Counter Submode with Flip-Flops

In the third submode, **multiplier submode**, a single PFU can effect a 4 x 1 bit multiply and sum with a partial product (see Figure 11). The multiplier bit is input at a4, and the multiplicand bits are input at b[3:0], where b3 is the most significant bit (MSB). a[3:0] contains the partial product (or other input to be summed) from a previous stage. If a4 is logical 1, the multiplicand is added to the partial product. If a4 is logical zero, zero is added to the partial product, which is the same as passing the partial product. CIN can hold the carry-in from the less significant PFUs if the multiplicand is wider than 4 bits, and COUT holds any carry-out from the addition, which may then be used as part of the product or routed to another PFU in multiplier mode for multiplicand width expansion.

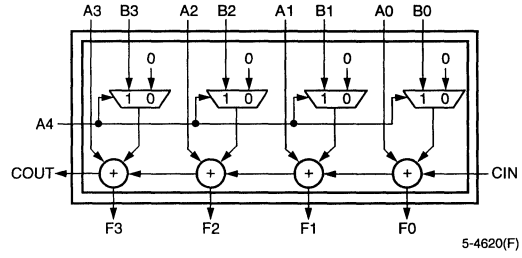


Figure 11. Multiplier Submode

Ripple mode's fourth submode features **equality comparators**, where one 4-bit bus is input on a[3:0], another 4-bit bus is input on b[3:0], and the carry-in is tied to 0 inside the PFU. The carry-out (≠) signal will be 0 if A = B or will be 1 if A ≠ B. If larger than 4 bits, the carry-out (≠) signal can be cascaded using fast-carry logic to the carry-in of any adjacent PFU. Comparators for greater than or equal or less than (>, =, <) continue to be supported using the ripple mode subtractor. The use of this submode could be shown using Figure 9 with CIN tied to 0.

Programmable Logic Cells (continued)

Asynchronous Memory Modes—MA and MB

The LUT in the PFU can be configured as either read/write or read-only memory. A read/write address (a[3:0],b[3:0]), write data (wd[1:0], wd[3:2]), and two write-enable (WE) ports are used for memory. In asynchronous memory mode, each HLUT can be used as a 16 x 2 memory. Each HLUT is configured independently, allowing functions such as a 16 x 2 memory in one HLUT and a logic function of five input variables or less in the other HLUT.

Figure 12 illustrates the use of the LUT for a 16 x 4 memory. When the LUTs are used as memory, there are independent address, input data, and output data buses. If the LUT is used as a 16 x 4 read/write memory, the a[3:0] and b[3:0] ports are address inputs (A[3:0]). The a4 and b4 ports are write-enable (WE) signals. The wd[3:0] inputs are the data inputs. The f[3:0] data outputs can be routed out on the o[4:0] PFU outputs or to the latch/FF d[3:0] inputs.

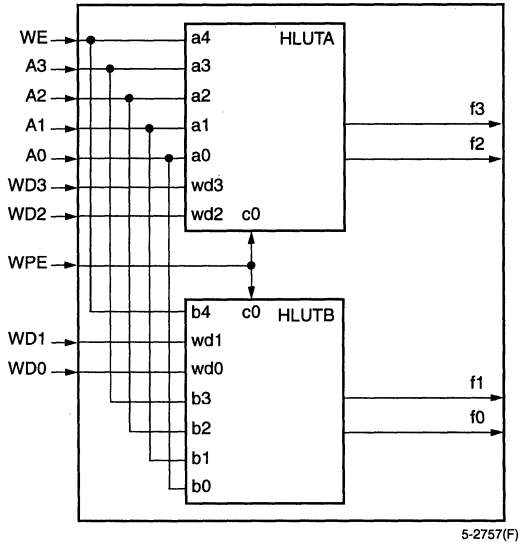


Figure 12. MA/MB Mode—16 x 4 RAM

To increase memory word depth above 16 (e.g., 32 x 4), two or more PLCs can be used. The address and write data inputs for the two or more PLCs are tied together (bit by bit), and the data outputs are routed through the four 3-statable BIDIs available in each PFU and are then tied together (bit by bit).

The control signal of the 3-statable BIDIs, called a RAM bank-enable, is created from a decode of upper address bits. The RAM bank-enable is then used to enable 4 bits of data from a PLC onto the read data bus.

The *ORCA* OR2CxxA/OR2TxxA series also has a new AND function available for each PFU in RAM mode. The inputs to this function are the write-enable (WE) signal and the write-port enable (WPE) signal. The write-enable signal is a4 for HLUTA and b4 for HLUTB, while the other input into the AND gates for both HLUTs is the write-port enable, input on c0 or CIN. Generally, the WPE input is driven by the same RAM bank-enable signal that controls the BIDIs in each PFU.

The selection as to which RAM bank to write data into does not require the use of LUTs from other PFUs, as in previous *ORCA* architectures. This reduces the number of PFUs required for RAMs larger than 16 words in depth. Note that if either HLUT is in MA/MB mode, then the same WPE is active for both HLUTs.

To increase the memory's word size (e.g., 16 x 8), two or more PLCs are used again. The address, write-enable, and write-port enable of the PLCs are tied together (bit by bit), and the data is different for each PLC. Increasing both the address locations and word size is done by using a combination of these two techniques.

The LUT can be used simultaneously for both memory and a combinatorial logic function. Figure 13 shows the use of a LUT implementing a 16 x 2 RAM (HLUTA) and any function of up to five input variables (HLUTB).

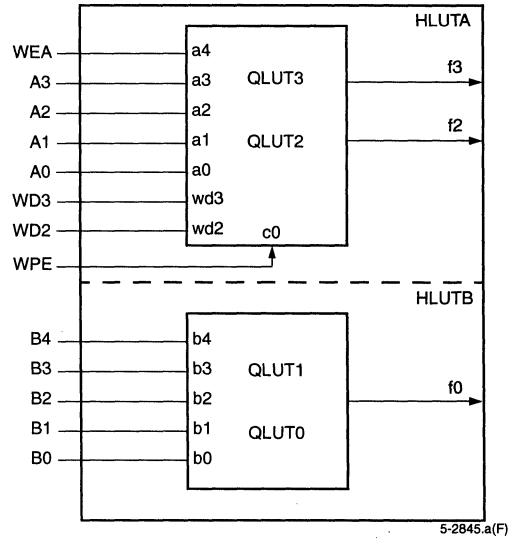


Figure 13. MA/F5 Mode—16 x 2 Memory and One Function of Five Input Variables

Programmable Logic Cells (continued)

Synchronous Memory Modes—SSPM and SDPM

The MA/MB asynchronous memory modes described previously allow the PFU to perform as a 16 x 4 (64 bits) single-port RAM. Synchronously writing to this RAM requires the write-enable control signal to be gated with the clock in another PFU to create a write pulse. To simplify this functionality, the OR2CxxA/OR2TxxA devices contain a **synchronous single-port memory** (SSPM) mode, where the generation of the write pulse is done in each PFU.

With SSPM mode, the entire LUT becomes a 16 x 4 RAM, as shown in Figure 14. In this mode, the input ports are write enable (WE), write-port enable (WPE), read/write address (A[3:0]), and write data (WD[3:0]). To synchronously write the RAM, WE (input into a4) and WPE (input into either c0 or cin) are latched and ANDed together. The result of this AND function is sent to a pulse generator in the LUT, which writes the RAM synchronous to the RAM clock. This RAM clock is the same one sent to the PFU latches/FFs; however, if necessary, it can be programmably inverted.

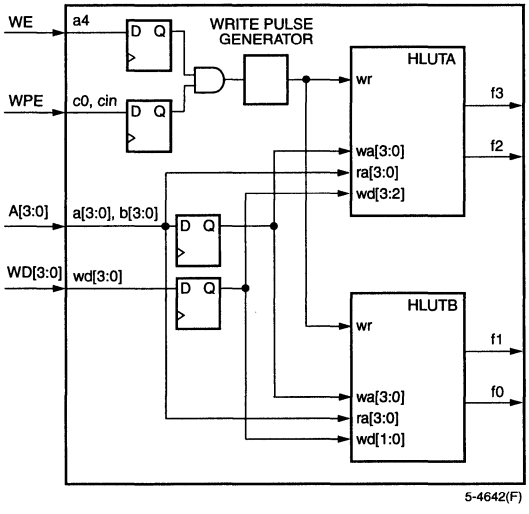


Figure 14. SSPM Mode—16 x 4 Synchronous Single-Port Memory

The write address (wa[3:0]) and write data (wd[3:0]) are also latched by the RAM clock in order to simplify the timing. Reading data from the RAM is done asynchronously, thus the read address (ra[3:0]) is not latched. The result from the read operation is placed on the LUT outputs (f[3:0]). The f[3:0] data outputs can be routed out of the PFU or sent to the latch/FF d[3:0] inputs.

There are two ways to use the latches/FFs in conjunction with the SSPM. If the phase of the latch/FF clock and the RAM clock are the same, only a read address or write address can be supplied to the RAM that meets the synchronous timing requirements of both the RAM clock and latch/FF clock. Therefore, either a write to the RAM or a read from the RAM can be done in each clock cycle, but not both. If the RAM clock is inverted from the latch/FF clock, then both a write to the RAM and a read from the RAM can occur in each clock cycle. This is done by adding an external write address/read address multiplexer as shown in Figure 15.

The write address is supplied on the phase of the clock that allows for setup to the RAM clock, and the read address is supplied on the phase of the clock that allows the read data to be setup to the latch/FF clock. If a higher-speed RAM is required that allows both a read and write in each clock cycle, the synchronous dual-port memory mode (SDPM) can be used, since it does not require the use of an external multiplexer.

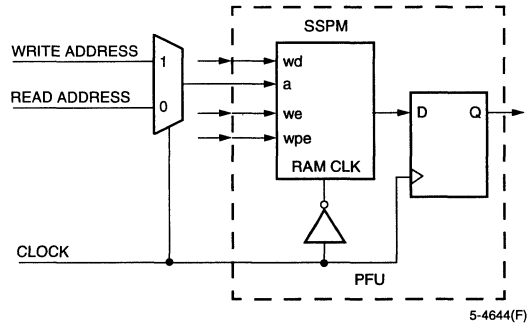
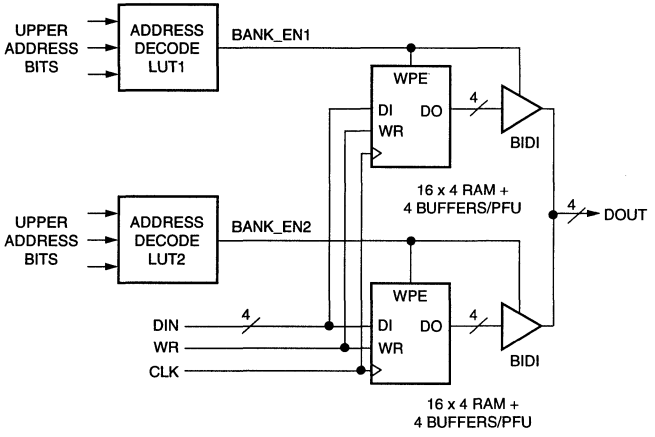


Figure 15. SSPM with Read/Write per Clock Cycle

Programmable Logic Cells (continued)



Note: The lower address bits are not shown.

Figure 16. Synchronous RAM with Write-Port Enable (WPE)

To increase memory word depth above 16 (e.g., 32 x 4), two or more PLCs can be used. The address and write data inputs for the two or more PLCs are tied together (bit by bit), and the data outputs are routed through the four 3-statable BIDIs available in each PFU. The BIDI outputs are then tied together (bit by bit), as seen in Figure 16.

The control signals of the 3-statable BIDIs, called RAM bank-enable (BANK_EN1 and BANK_EN2), are created from a decode of upper address bits. The RAM bank-enable is then used to enable 4 bits of data from a PLC onto the read data (DOUT) bus.

The OR2CxxA/OR2TxxA series now has a new AND function available for each PFU in RAM mode. The inputs to this function are the write-enable (WE) signal and the write-port enable (WPE) signal. The write-enable signal is input on a4, while the write-port enable is input on c0 or cin. Generally, the WPE input is driven by the same RAM bank-enable signal that controls the BIDIs in each PFU.

The selection as to which RAM bank to write data into does not require the use of LUTs from other PFUs, as in previous *ORCA* architectures. This reduces the number of PFUs required for RAMs larger than 16 words in depth.

A special use of this method can be to increase word depth to 32 words. Since both the WPE input into the RAM and the 3-state input into the BIDI can be inverted, a decode of the one upper address bit is not required. Instead, the bank-enable signal for both banks is tied to the upper address bit, with the WPE and 3-state inputs active-high for one bank and active-low for the other.

To increase the memory's word size (e.g., 16 x 8), two or more PLCs are used again. The address, write-enable, and write-port enable of the PLCs are tied together (bit by bit), and the data is different for each PLC. Increasing both the address locations and word size is accomplished by using a combination of these two techniques.

Programmable Logic Cells (continued)

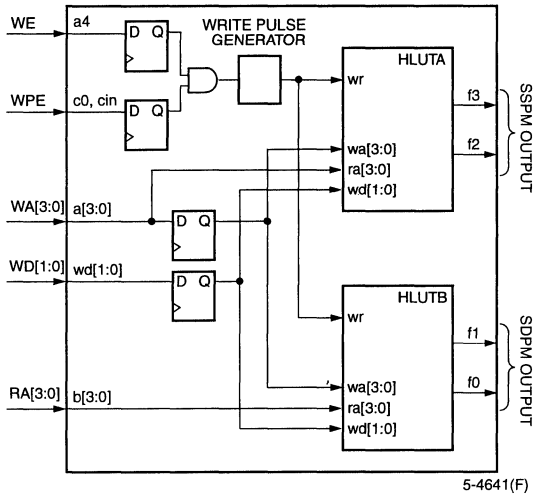


Figure 17. SDPM Mode—16 x 2 Synchronous Dual-Port Memory

The OR2CxxA/OR2TxxA devices have added a second synchronous memory mode known as the **synchronous dual-port memory** (SDPM) mode. This mode writes data into the memory synchronously in the same manner described previously for SSPM mode. The SDPM mode differs in that two separate 16 x 2 memories are created in each PFU that have the same WE, WPE, write data (wd[1:0]), and write address (wa[3:0]) inputs, as shown in Figure 17.

The outputs of HLUTA (f[3:2]) operate the same way they do in SSPM mode—the read address comes directly from the A[3:0] inputs used to create the latched write address. The outputs of HLUTB (f[1:0]) operate in a dual-port mode where the write address comes from the latched version of A[3:0], and the read address comes directly from RA[3:0], which is input on b[3:0].

Since external multiplexing of the write address and read address is not required, extremely fast RAMs can be created. New system applications that require an interface between two different asynchronous clocks can also be implemented using the SDPM mode. An example of this is accomplished by creating FIFOs where one clock controls the synchronous write of data into the FIFO, and the other clock controls the read address to allow reading of data at any time from the FIFO.

Latches/Flip-Flops

The four latches/FFs in the PFU can be used in a variety of configurations. In some cases, the configuration options apply to all four latches/FFs in the PFU. For other options, each latch/FF is independently programmable.

Table 4 summarizes these latch/FF options. The latches/FFs can be configured as either positive or negative level sensitive latches, or positive or negative edge-triggered flip-flops. All latches/FFs in a given PFU share the same clock, and the clock to these latches/FFs can be inverted. The input into each latch/FF is from either the corresponding QLUT output (f[3:0]) or the direct data input (wd[3:0]). For latches/FFs located in the two outer rings of PLCs, additional inputs are possible. These additional inputs are fast paths from I/O pads located in PICs in the same row or column as the PLCs. If the latch/FF is not located in the two outer rings of the PLCs, the latch/FF input can also be tied to logic 0, which is the default. The four latch/FF outputs, q[3:0], can be placed on the five PFU outputs, o[4:0].

Table 4. Configuration RAM Controlled Latch/Flip-Flop Operation

Function	Options
Functionality Common to All Latch/FFs in PFU	
LSR Operation	Asynchronous or Synchronous
Clock Polarity	Noninverted or Inverted
Front-End Select	Direct (wd[3:0]) or from LUT (f[3:0])
LSR Priority	Either lsr or ce has priority
Functionality Set Individually in Each Latch/FF in PFU	
Latch/FF Mode	Latch or Flip-Flop
Set/Reset Mode	Set or Reset

The four latches/FFs in a PFU share the clock (ck), clock enable (ce), and local set/reset (lsr) inputs. When ce is disabled, each latch/FF retains its previous value when clocked. Both the clock enable and lsr inputs can be inverted to be active-low.

Programmable Logic Cells (continued)

The set/reset operation of the latch/FF is controlled by two parameters: reset mode and set/reset value. When the global set/reset (gsrn) or local set/reset (lsr) are inactive, the storage element operates normally as a latch or FF. The reset mode is used to select a synchronous or asynchronous lsr operation. If synchronous, lsr is enabled only if clock enable (ce) is active. For the OR2CxxA/OR2TxxA series, a new option called the lsr priority allows the synchronous lsr to have priority over the ce input, thereby setting or resetting the FF independent of the state of ce. The clock enable is supported on FFs, not latches. The clock enable function is implemented by using a two-input multiplexer on the FF input, with one input being the previous state of the FF and the other input being the new data applied to the FF. The select of this two-input multiplexer is clock enable (ce), which selects either the new data or the previous state. When ce is inactive, the FF output does not change when the clock edge arrives.

The gsrn signal is only asynchronous, and it sets/resets all latches/FFs in the FPGA based upon the set/reset configuration bit for each latch/FF. The set/reset value determines whether gsrn and lsr are set or reset inputs. The set/reset value is independent for each latch/FF.

If the local set/reset is not needed, the latch/FF can be configured to have a data front-end select. Two data inputs are possible in the front-end select mode, with the lsr signal used to select which data input is used. The data input into each latch/FF is from the output of its associated QLUT f[3:0] or direct from wd[3:0], bypassing the LUT. In the front-end data select mode, both signals are available to the latches/FFs.

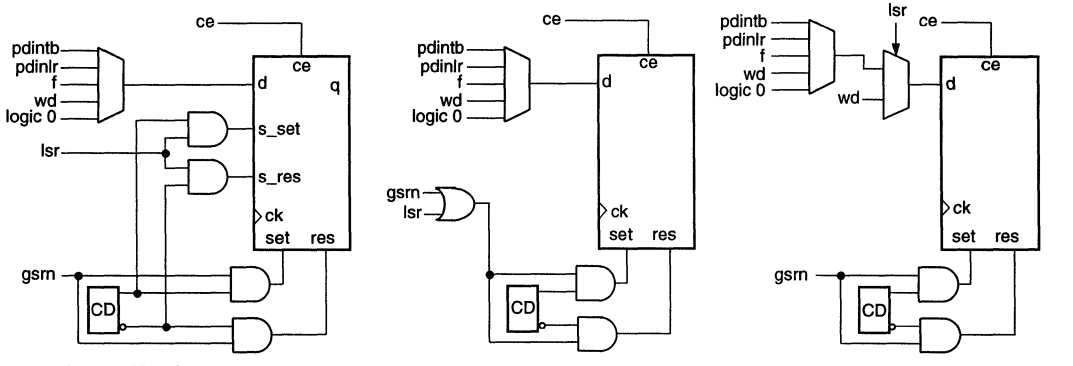
For PLCs that are in the two outside rows or columns of the array, the latch/FFs can have two inputs in addition to the f and wd inputs mentioned above. One input is from an I/O pad located at the PIC closest to either the left or right of the given PLC (if the PLC is in the left two columns or right two columns of the array). The other input is from an I/O pad located at the closest PIC either above or below the given PLC (if the PLC is in the top or the bottom two rows). It should be noted that both inputs are available for a 2 x 2 array of PLCs in each corner of the array. For the entire array of PLCs, if either or both of these inputs is unavailable, the latch/FF data input can be tied to a logic 0 instead (the default).

To speed up the interface between signals external to the FPGA and the latches/FFs, there are direct paths from latch/FF outputs to the I/O pads. This is done for each PLC that is adjacent to a PIC.

The latches/FFs can be configured in three modes:

1. Local synchronous set/reset: the input into the PFU's lsr port is used to synchronously set or reset each latch/FF.
2. Local asynchronous set/reset: the input into lsr asynchronously sets or resets each latch/FF.
3. Latch/FF with front-end select: the data select signal (actually lsr) selects the input into the latches/FFs between the LUT output and direct data in.

For all three modes, each latch/FF can be independently programmed as either set or reset. Each latch/FF in the PFU is independently configured to operate as either a latch or flip-flop. Figure 18 provides the logic functionality of the front-end select, global set/reset, and local set/reset operations.



Note: CD = configuration data.

Figure 18. Latch/FF Set/Reset Configurations

Programmable Logic Cells (continued)

PLC Routing Resources

Generally, the *ORCA* Foundry Development System is used to automatically route interconnections. Interactive routing with the *ORCA* Foundry design editor (EPIC) is also available for design optimization. To use EPIC for interactive layout, an understanding of the routing resources is needed and is provided in this section.

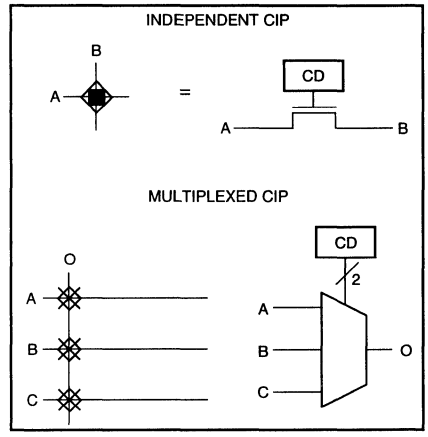
The routing resources consist of switching circuitry and metal interconnect segments. Generally, the metal lines which carry the signals are designated as routing nodes (R-nodes). The switching circuitry connects the routing nodes, providing one or more of three basic functions: signal switching, amplification, and isolation. A net running from a PFU or PIC output (source) to a PLC or PIC input (destination) consists of one or more R-nodes, connected by switching circuitry designated as configurable interconnect points (CIPs).

The following sections discuss PLC, PIC, and interquad routing resources. This section discusses the PLC switching circuitry, intra-PLC routing, inter-PLC routing, and clock distribution.

Configurable Interconnect Points

The process of connecting R-nodes uses three basic types of switching circuits: two types of configurable interconnect points (CIPs) and bidirectional buffers (BIDIs). The basic element in CIPs is one or more pass transistors, each controlled by a configuration RAM bit. The two types of CIPs are the mutually exclusive (or multiplexed) CIP and the independent CIP.

A mutually exclusive set of CIPs contains two or more CIPs, only one of which can be on at a time. An independent CIP has no such restrictions and can be on independent of the state of other CIPs. Figure 19 shows an example of both types of CIPs.



f.13(C)

Figure 19. Configurable Interconnect Point

3-Statable Bidirectional Buffers

Bidirectional buffers provide isolation as well as amplification for signals routed a long distance. Bidirectional buffers are also used to drive signals directly onto either vertical or horizontal xL and xH R-nodes (to be described later in the inter-PLC routing section). BIDIs are also used to indirectly route signals through the switching R-nodes. Any number from zero to eight BIDIs can be used in a given PLC.

The BIDIs in a PLC are divided into two nibble-wide sets of four (BIDI and BIDIH). Each of these sets has a separate BIDI controller that can have an application net connected to its TRI input, which is used to 3-state enable the BIDIs. Although only one application net can be connected to both BIDI controllers, the sense of this signal (active-high, active-low, or ignored) can be configured independently. Therefore, one set can be used for driving signals, the other set can be used to create 3-state buses, both sets can be used for 3-state buses, and so forth.

Programmable Logic Cells (continued)

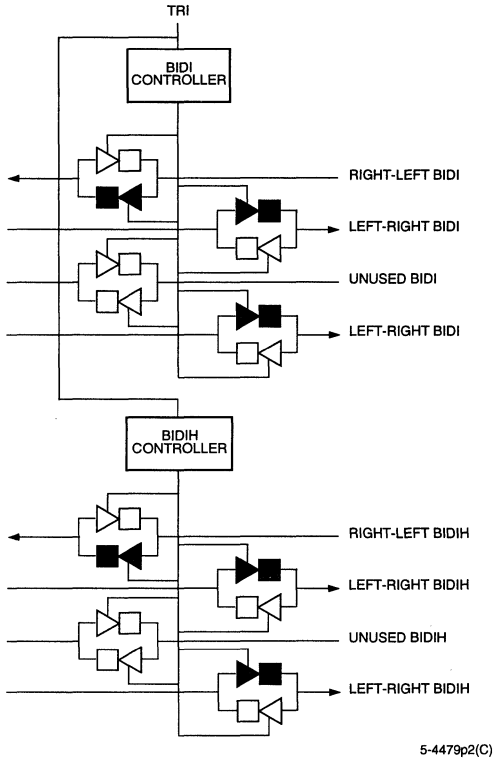


Figure 20. 3-Statable Bidirectional Buffers

Intra-PLC Routing

The function of the intra-PLC routing resources is to connect the PFU's input and output ports to the routing resources used for entry to and exit from the PLC. These are nets for providing PFU feedback, turning corners, or switching from one type of routing resource to another.

PFU Input and Output Ports. There are nineteen input ports to each PFU. The PFU input ports are labelled a[4:0], b[4:0], wd[3:0], c0, ck, lsr, cin, and ce. The six output ports are o[4:0] and cout. These ports correspond to those described in the PFU section.

Switching R-Nodes. There are four sets of switching R-nodes in each PLC, one in each corner. Each set consists of five switching elements, labelled sul[4:0], sur[4:0], sll[4:0], and slr[4:0], for the upper-left, upper-right, lower-left, and lower-right sections of the PFUs, respectively. The switching R-nodes connect to the PFU inputs and outputs as well as the BIDI and BIDIH R-nodes, to be described later. They also connect to both the horizontal and vertical x1 and x4 R-nodes (inter-PLC routing resources, described below) in their specific corner.

One of the four sets of switching R-nodes can be connected to a set of switching R-nodes in each of the four adjacent PLCs or PICs. This allows direct routing of up to five signals without using inter-PLC routing.

BIDI/BIDIH R-Nodes. There are two sets of bidirectional R-nodes in the PLC, each set consisting of four bidirectional buffers. They are designated BIDI and BIDIH and have similar functionality. The BIDI R-nodes are used in conjunction with the xL R-nodes, and the BIDIH R-nodes are used in conjunction with the xH R-nodes. Each side of the four BIDIs in the PLC is connected to a BIDI R-node on the left (BL[3:0]) and on the right (BR[3:0]). These R-nodes can be connected to the xL R-nodes through CIPs, with BL[3:0] connected to the vertical xL R-nodes and BR[3:0] connected to the horizontal xL R-nodes. Both BL[3:0] and BR[3:0] have CIPs which connect to the switching R-nodes.

Similarly, each side of the four BIDIHs is connected to a BIDIH R-node: BLH[3:0] on the left and BRH[3:0] on the right. These R-nodes can also be connected to the xH R-nodes through CIPs, with BLH[3:0] connected to the vertical xH R-nodes and BRH[3:0] connected to the horizontal xH R-nodes. Both BLH[3:0] and BRH[3:0] have CIPs which connect to the switching R-nodes.

CIPs are also provided to connect the BIDIH and BIDIL R-nodes together on each side of the BIDIs. For example, BLH3 can connect to BL3, while BRH3 can connect to BR3.

Programmable Logic Cells (continued)

Inter-PLC Routing Resources

The inter-PLC routing is used to route signals between PLCs. The R-nodes occur in groups of four, and differ in the numbers of PLCs spanned. The x1 R-nodes span one PLC, the x4 R-nodes span four PLCs, the xH R-nodes span one-half the width (height) of the PLC array, and the xL R-nodes span the width (height) of the PLC array. All types of R-nodes run in both horizontal and vertical directions.

Table 5 shows the groups of inter-PLC R-nodes in each PLC. In the table, there are two rows/columns each for x1 and x4 lines. In the design editor, the horizontal x1 and x4 R-nodes are located above and below the PFU. Similarly, the vertical segments are located on each side. The xL and xH R-nodes only run below and to the left of the PFU. The indexes specify individual R-nodes within a group. For example, the vx4[2] R-node runs vertically to the left of the PFU, spans four PLCs, and is the third line in the 4-bit wide bus.

Table 5. Inter-PLC Routing Resources

Horizontal R-Nodes	Vertical R-Nodes	Distance Spanned
hx1[3:0]	vx1[3:0]	One PLC
hx1[7:4]	vx1[7:4]	One PLC
hx4[3:0]	vx4[3:0]	Four PLCs
hx4[7:4]	vx4[7:4]	Four PLCs
hxL[3:0]	vxL[3:0]	PLC Array
hxH[3:0]	vxH[3:0]	1/2 PLC Array
ckl, ckr	ckt, ckb	PLC Array

Figure 21 shows the inter-PLC routing within one PLC. Figure 22 provides a global view of inter-PLC routing resources across multiple PLCs.

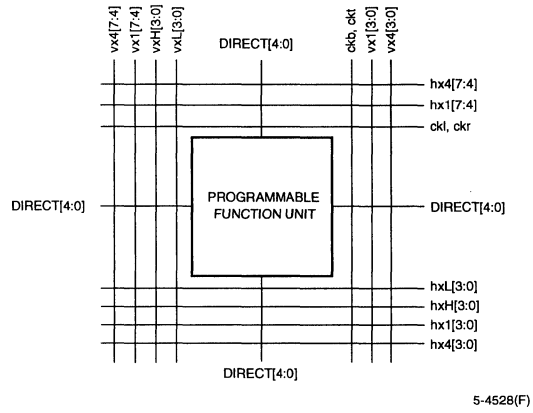


Figure 21. Single PLC View of Inter-PLC R-Nodes

x1 R-Nodes. There are a total of 16 x1 R-nodes per PLC: eight vertical and eight horizontal. Each of these is subdivided into nibble-wide buses: hx1[3:0], hx1[7:4], vx1[3:0], and vx1[7:4]. An x1 line is one PLC long. If a net is longer than one PLC, an x1 R-node can be lengthened to n times its length by turning on n – 1 CIPs. A signal is routed onto an x1 R-node via the switching R-nodes.

x4 R-Nodes. There are four sets of four x4 R-nodes, for a total of 16 x4 R-nodes per PLC. They are hx4[3:0], hx4[7:4], vx4[3:0], and vx4[7:4]. Each set of x4 R-nodes is twisted each time it passes through a PLC, and one of the four is broken with a CIP. This allows a signal to be routed for a length of four cells in any direction on a single line without additional CIPs. The x4 R-nodes can be used to route any nets that require minimum delay. A longer net is routed by connecting two x4 R-nodes together by a CIP. The x4 R-nodes are accessed via the switching R-nodes.

Programmable Logic Cells (continued)

xL R-Nodes. The long xL R-nodes run vertically and horizontally the height and width of the array, respectively. There are a total of eight xL R-nodes per PLC: four horizontal (hxL[3:0]) and four vertical (vxL[3:0]). Each PLC column has four xL lines, and each PLC row has four xL R-nodes. Each of the xL R-nodes connects to the two PICs at either end. The OR2C/2T12A, which consists of a 18 x 18 array of PLCs, contains 72 vxL and 72 hxL R-nodes. They are intended primarily for global signals which must travel long distances and require minimum delay and/or skew, such as clocks.

There are three methods for routing signals onto the xL R-nodes. In each PLC, there are two long-line drivers: one for a horizontal xL R-node, and one for a vertical xL R-node. Using the long-line drivers produces the least delay. The xL R-nodes can also be driven directly by PFU outputs using the BIDI R-nodes. In the third method, the xL R-nodes are accessed by the bidirectional buffers, again using the BIDI R-nodes.

xH R-Nodes. Four by half (xH) R-nodes run horizontally and four xH R-nodes run vertically in each row and column in the array. These R-nodes travel a distance of one-half the PLC array before being broken in the middle of the array, where they connect to the interquad block (discussed later). They also connect at the periphery of the FPGA to the PICs, like the xL R-nodes. The xH R-nodes do not twist like xL R-nodes, allowing nibble-wide buses to be routed easily.

Two of the three methods of routing signals onto the xL R-nodes can also be used for the xH R-nodes. A special xH line driver is not supplied for the xH R-nodes.

Clock R-Nodes. For a very fast and low-skew clock (or other global signal tree), clock R-nodes run the entire height and width of the PLC array. There are two horizontal clock R-nodes per PLC row (CKL, CKR) and two vertical clock R-nodes per PLC column (CKT, CKB). The source for these clock R-nodes can be any of the four I/O buffers in the PIC. The horizontal clock R-nodes in a row (CKL, CKR) are driven by the left and right PICs, respectively. The vertical clock R-nodes in a column (CKT, CKB) are driven by the top and bottom PICs, respectively.

The clock R-nodes are designed to be a clock spine. In each PLC, there is a fast connection available from the clock R-node to the long-line driver (described earlier). With this connection, one of the clock R-nodes in each PLC can be used to drive one of the four xL R-nodes perpendicular to it, which, in turn, creates a clock tree.

This feature is discussed in detail in the Clock Distribution Network section.

Minimizing Routing Delay

The CIP is an active element used to connect two R-nodes. As an active element, it adds significantly to the resistance and capacitance of a net, thus increasing the net's delay. The advantage of the x1 R-node over a x4 R-node is routing flexibility. A net from PLC db to PLC cb is easily routed by using x1 R-nodes. As more CIPs are added to a net, the delay increases. To increase speed, routes that are greater than two PLCs away are routed on the x4 R-nodes because a CIP is located only in every fourth PLC. A net that spans eight PLCs requires seven x1 R-nodes and six CIPs. Using x4 R-nodes, the same net uses two R-nodes and one CIP.

All routing resources in the PLC can carry 4-bit buses. In order for data to be used at a destination PLC that is in data path mode, the data must arrive unscrambled. For example, in data path operation, the least significant bit 0 must arrive at either a[0] or b[0]. If the bus is to be routed by using either x4 or xL R-nodes (both of which twist as they propagate), the bus must be placed on the appropriate lines at the source PLC so that the data arrives at the destination unscrambled. The switching R-nodes provide the most efficient means of connecting adjacent PLCs. Signals routed with these R-nodes have minimum propagation delay.

Programmable Logic Cells (continued)

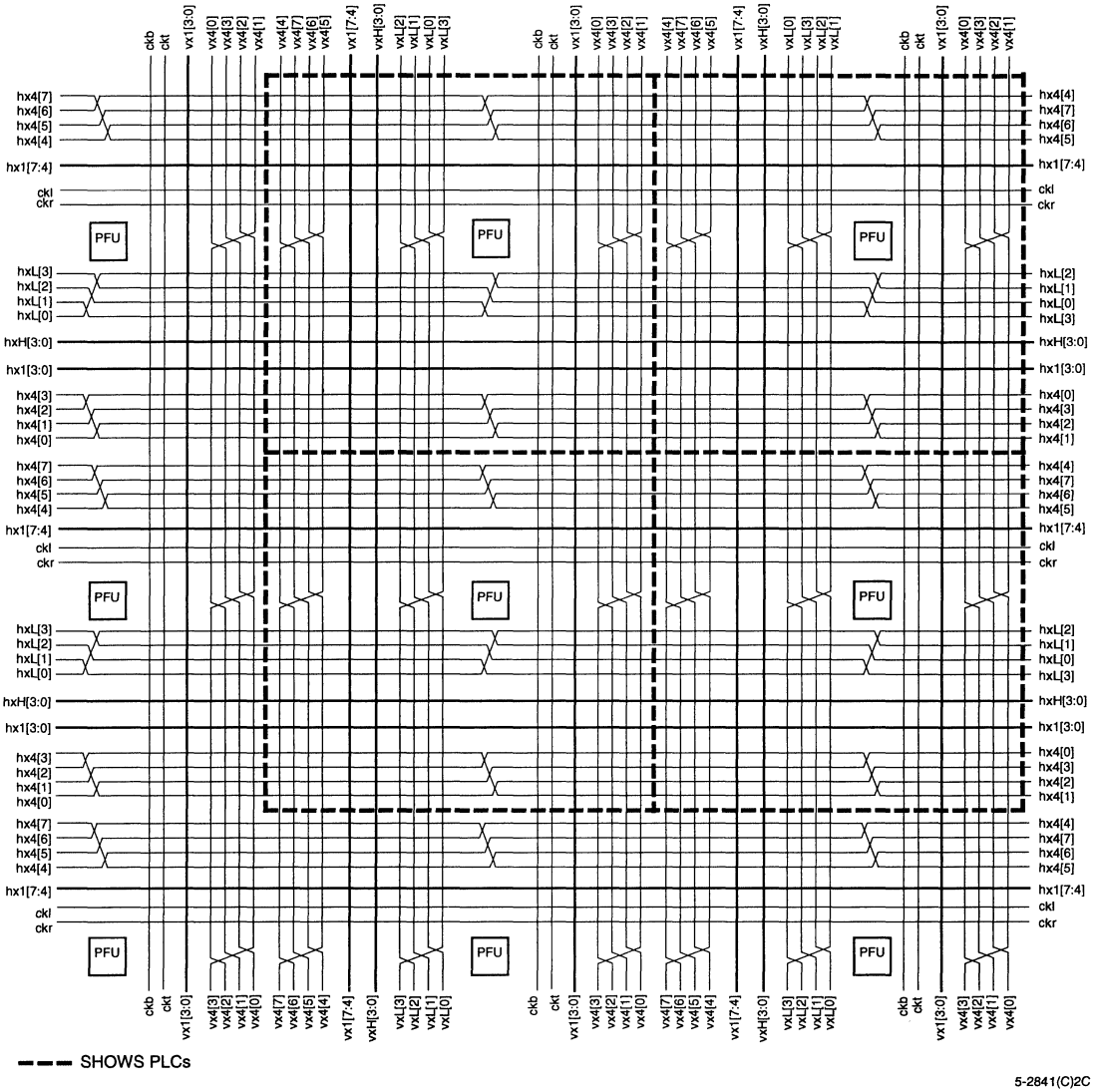


Figure 22. Multiple PLC View of Inter-PLC Routing

Programmable Logic Cells (continued)

PLC Architectural Description

Figure 23 is an architectural drawing of the PLC which reflects the PFU, the R-nodes, and the CIPs. A discussion of each of the letters in the drawing follows.

A. These are switching R-nodes which give the router flexibility. In general switching theory, the more levels of indirection there are in the routing, the more routable the network is. The switching R-nodes can also connect to adjacent PLCs.

The switching R-nodes provide direct connections to PLCs directly to the top, bottom, left, and right, without using other routing resources. The ability to disable this connection between PLCs is provided so that each side of these connections can be used exclusively as switching R-nodes in their respective PLC.

B. These CIPs connect the x1 routing. These are located in the middle of the PLC to allow the block to connect to either the left end of the horizontal x1 R-node from the right or the right end of the horizontal x1 R-node from the left, or both. By symmetry, the same principle is used in the vertical direction. The x1 lines are not twisted, making them suitable for data paths.

C. This set of CIPs is used to connect the x1 and x4 nets to the switching R-nodes or to other x1 and x4 nets. The CIPs on the major diagonal allow data to be transmitted from x1 nets to the switching R-nodes without being scrambled. The CIPs on the major diagonal also allow unscrambled data to be passed between the x1 and x4 nets.

In addition to the major diagonal CIPs for the x1 lines, other CIPs provide an alternative entry path into the PLC in case the first one is already used. The other CIPs are arrayed in two patterns, as shown. Both of these patterns start with the main diagonal, but the extra CIPs are arrayed on either a parallel diagonal shifted by one or shifted by two (modulo the size of the vertical bus (5)). This allows any four application nets incident to the PLC corner to be transferred to the five switching R-nodes in that corner. Many patterns of five nets can also be transferred.

D. The x4 R-nodes are twisted at each PLC. One of the four x4 lines is broken with a CIP, which allows a signal to be routed a distance of four PLCs in any direction on a single R-node without an intermediate CIP. The x4 R-nodes are less populated with CIPs than the x1 lines to increase their speed. A CIP can be enabled to extend an x4 R-node four more PLCs, and so on.

For example, if an application signal is routed onto hx4[4] in a PLC, it appears on hx4[5] in the PLC to the right. This signal step-up continues until it reaches hx4[7], two PLCs later. At this point, the user can break the connection or continue the signal for another four PLCs.

E. These symbols are bidirectional buffers (BIDIs). There are four BIDIs per PLC, and they provide signal amplification as needed to decrease signal delay. The BIDIs are also used to transmit signals on xL lines.

F. These are the BIDI and BIDIH controllers. The 3-state control signal can be disabled. They can be configured as active-high or active-low independently of each other.

G. This set of CIPs allows a BIDI to get or put a signal from one set of switching R-nodes on each side. The BIDIs can be accessed by the switching R-nodes. These CIPs allow a nibble of data to be routed though the BIDIs and continue to a subsequent block. They also provide an alternative routing resource to improve routability.

H. These CIPs are used to take data from/to the BIDIs to/from the xL R-nodes. These CIPs have been optimized to allow the BIDI buffers to drive the large load usually seen when using xL R-nodes.

I. Each latch/FF can accept data: from a LUT output; from a direct data input signal from general routing; or, as in the case of PLCs located in the two rows (columns) adjacent to PICs, directly from the pad. In addition, the LUT outputs can bypass the latches/FFs completely and output data on the general routing resources. The four inputs shown are used as the direct input to the latches/FFs from general routing resources. If the LUT is in memory mode, the four inputs wd[3:0] are the data input to the memory.

Programmable Logic Cells (continued)

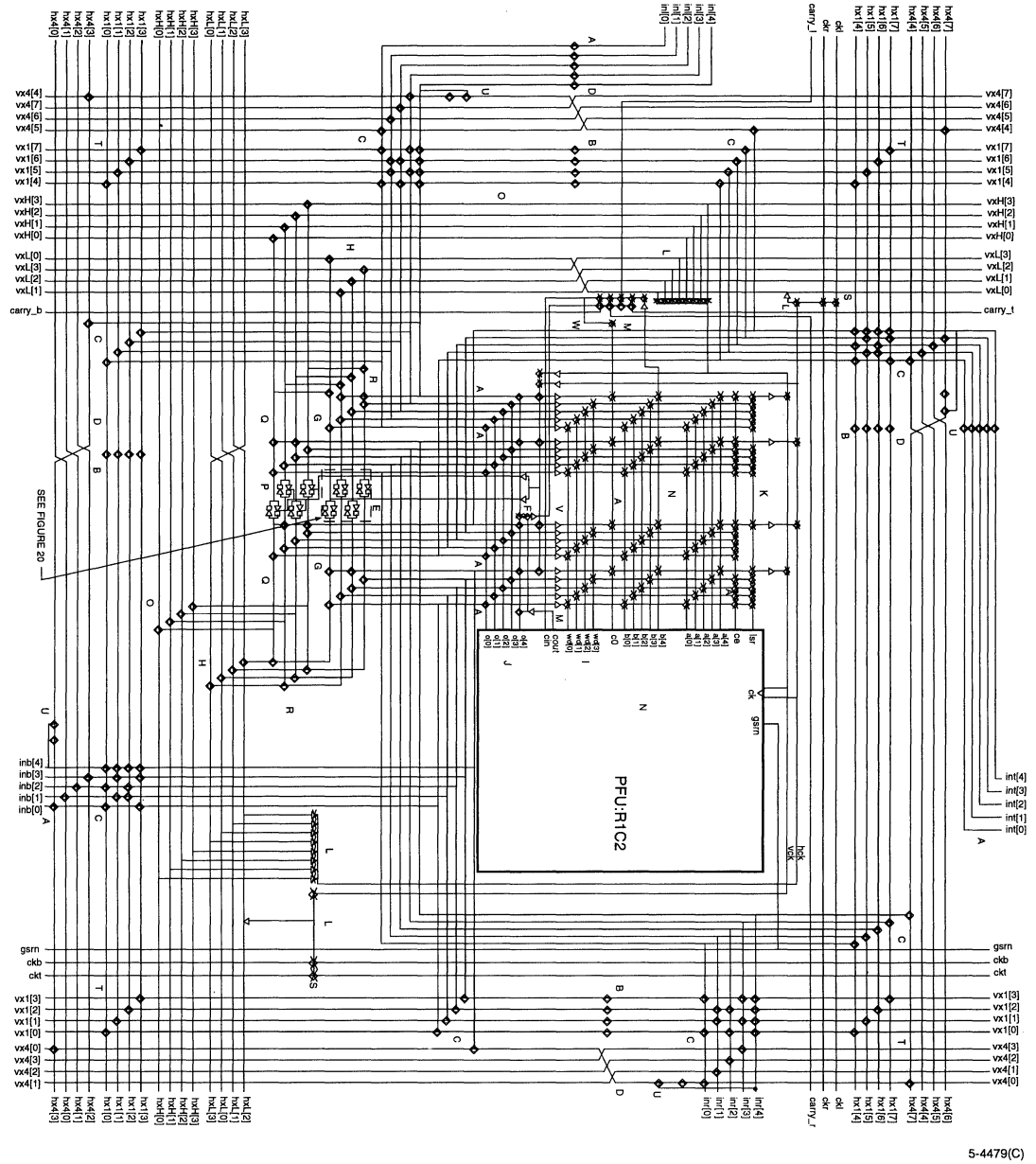


Figure 23. PLC Architecture

Programmable Logic Cells (continued)

J. Any five of the eight output signals can be routed out of the PLC. The eight signals are the four LUT outputs (f0, f1, f2, and f3) and the four latch/FF outputs (q0, q1, q2, and q3). This allows the user to access all four latch/FF outputs, read the present state and next state of a latch/FF, build a 4-bit shift register, etc. Each of the outputs can drive any number of the five PFU outputs. The speed of a signal can be increased by dividing its load among multiple PFU output drivers.

K. These lines deliver the auxiliary signals clock enable and set/reset to the latches/FFs. All four of the latches/FFs share these signals.

L. This is the clock input to the latches/FFs. Any of the horizontal and vertical xH or xL lines can drive the clock of the PLC latches/FFs. Long-line drivers are provided so that a PLC can drive one xL R-node in the horizontal direction and one xL R-node in the vertical direction. The xL lines in each direction exhibit the same properties as x4 lines, except there are no CIPs. The clock R-nodes (ckl, ckr, ckt, and ckb) and multiplexers/drivers are used to connect to the xL R-nodes for low-skew, low-delay global signals.

The long-lines run the length or width of the PLC array. They rotate to allow four PLCs in one row or column to generate four independent global signals. These lines do not have to be used for clock routing. Any highly used application net can use this resource, especially one requiring low skew.

M. These R-nodes are used to route the fast carry signal to/from the neighboring four PLCs. The carry-out (cout) of the PFU can also be routed out of the PFU onto the fifth output (o4). The carry-in (cin) signal can also be supplied by the b4 input to the PFU.

N. These are the 11 logic inputs to the LUT. The a[4:0] inputs are provided into HLUTA, and the b[4:0] inputs are provided into HLUTB. The c0 input bypasses the main LUT and is used in the pfumux, pfuxor, and pfunand functions (F5M, F5X modes). Since this input bypasses the LUT, it can be used as a fast path around the LUT, allowing the implementation of fast, wide combinatorial functions. The c0 input can be disabled or inverted.

O. The xH R-nodes run one-half the length (width) of the array before being broken by a CIP.

P. The BIDIHs are used to access the xH R-nodes.

Q. The BIDIH R-nodes are used to connect the BIDIHs to the xsw R-nodes, the xH R-nodes, or the BIDI R-nodes.

R. These CIPs connect the BIDI R-nodes and the BIDIH R-nodes.

S. These are clock R-nodes (ckt, ckb, ckl, and ckr) with the multiplexers and drivers to connect to the xL R-nodes.

T. These CIPs connect x1 R-nodes which cross in each corner to allow turns on the x1 R-nodes without using the xsw R-nodes.

U. These CIPs connect x4 R-nodes and xsw R-nodes, allowing nets that run a distance that is not divisible by four to be routed more efficiently.

V. This routing structure allows any PFU output, including LUT and latch/FF outputs, to be placed on o4 and be routed onto the fast carry routing.

W. This routing structure allows the fast carry routing to be routed onto the c0 PFU input.

Programmable Input/Output Cells

The programmable input/output cells (PICs) are located along the perimeter of the device. Each PIC interfaces to four bond pads and contains the necessary routing resources to provide an interface between I/O pads and the PLCs. Each PIC is composed of input buffers, output buffers, and routing resources as described below. Table 6 provides an overview of the programmable functions in an I/O cell. Figure 24A is a simplified diagram of the functionality of the OR2CxxA series I/O cells, while Figure 24B is a simplified functional diagram of the OR2TxxA series I/O cells.

Table 6. Input/Output Cell Options

Input	Option
Input Levels	TTL/CMOS (OR2CxxA only)
Input Speed	Fast/Delayed
Float Value	Pull-up/Pull-down/None
Direct-in to FF	Fast/Delayed
Output	**Option**
Output Drive	12 mA/6 mA or 6 mA/3 mA
Output Speed	Fast/Slewlim/Sinklim
Output Source	FF Direct-out/General Routing
Output Sense	Active-high/-low
3-State Sense	Active-high/-low (3-state)

Inputs

Each I/O can be configured to be either an input, an output, or bidirectional I/O. Inputs for the OR2CxxA can be configured as either TTL or CMOS compatible. The I/O for the OR2TxxA series devices are 5.0 V tolerant, and will be described in a later section of this data sheet. To allow zero hold time on PLC latches/FFs, the input signal can be delayed. Pull-up or pull-down resistors are available on inputs to minimize power consumption.

A fast path from the input buffer to the clock R-nodes is also provided. Any one of the four I/O pads on any PIC can be used to drive the clock R-node generated in that PIC.

To reduce the time required to input a signal into the FPGA, a dedicated path (pdin) from the I/O pads to the PFU flip-flops is provided. Like general input signals, this signal can be configured as normal or delayed. The delayed direct input can be selected independently from the delayed general input. If the fast clock routing is selected from a given I/O pad, then the direct input signal is automatically delayed, decreasing the delay of the fast clock.

Inputs should have transition times of less than 500 ns and should not be left floating. If an input can float, a pull-up or pull-down should be enabled. Floating inputs increase power consumption, produce oscillations, and increase system noise. The OR2CxxA inputs have a typical hysteresis of approximately 280 mV (200 mV for the OR2TxxA) to reduce sensitivity to input noise. The PIC contains input circuitry which provides protection against latch-up and electrostatic discharge.

Programmable Input/Output Cells

(continued)

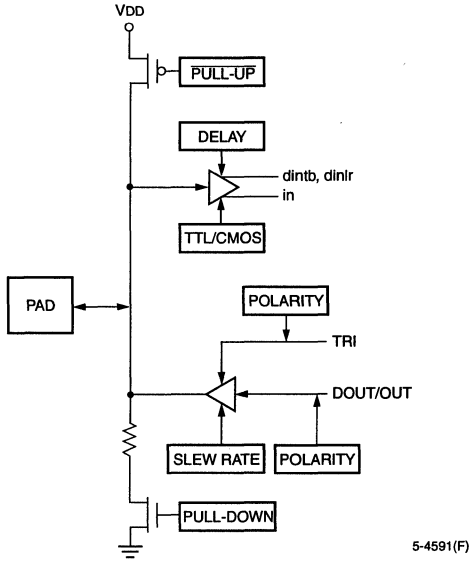


Figure 24A. Simplified Diagram of OR2CxxA Programmable I/O Cell

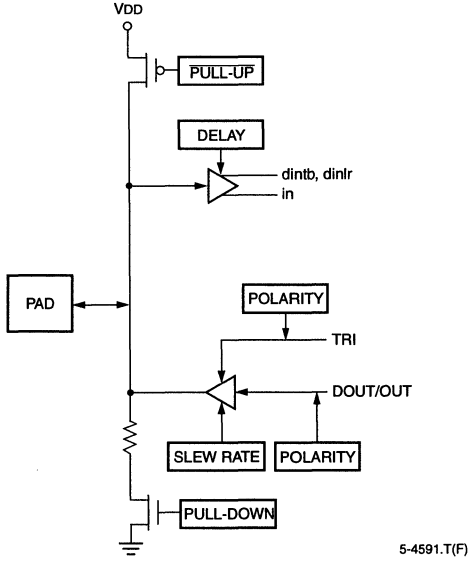


Figure 24B. Simplified Diagram of OR2TxxA Programmable I/O Cell (PIC)

Outputs

The PIC's output drivers have programmable drive capability and slew rates. Three propagation delays (fast, slewlim, sinklim) are available on output drivers. The sinklim mode has the longest propagation delay and is used to minimize system noise and minimize power consumption. The fast and slewlim modes allow critical timing to be met.

The drive current is 12 mA sink/6 mA source for the slewlim and fast output speed selections and 6 mA sink/3 mA source for the sinklim output. Two adjacent outputs can be interconnected to increase the output sink current to 24 mA.

All outputs that are not speed critical should be configured as sinklim to minimize power and noise. The number of outputs that switch simultaneously in the same direction should be limited to minimize ground bounce. To minimize ground bounce problems, locate heavily loaded output buffers near the ground pads. Ground bounce is generally a function of the driving circuits, traces on the PCB, and loads and is best determined with a circuit simulation.

Outputs can be inverted, and 3-state control signals can be active-high or active-low. An open-drain output may be obtained by using the same signal for driving the output and 3-state signal nets so that the buffer output is enabled only by a low. At powerup, the output drivers are in slewlim mode, and the input buffers are configured as TTL-level compatible with a pull-up. If an output is not to be driven in the selected configuration mode, it is 3-stated.

5 V Tolerant I/O

The I/O on the OR2TxxA series devices allow interconnection to both 3.3 V and 5 V device (selectable on a per-pin basis) by way of special $V_{DD}5$ pins that have been added to the OR2TxxA devices. If any I/O on the OR2TxxA device interface to a 5 V input, then all of the $V_{DD}5$ pins must be connected to the 5 V supply. If no pins on the device interface to a 5 V signal, then the $V_{DD}5$ pins must be connected to the 3.3 V supply.

If the $V_{DD}5$ pins are disconnected (i.e., they are floating), the device will not be damaged; however, the device may not operate properly until $V_{DD}5$ is returned to a proper voltage level. If the $V_{DD}5$ pins are then shorted to ground, a large current flow will develop, and the device may be damaged.

Programmable Input/Output Cells

(continued)

Regardless of the power supply the V_{DD5} pins are connected to (5 V or 3.3 V), the OR2TxxA devices will drive the pin to the 3.3 V levels when the output buffer is enabled. If the other device being driven by the OR2TxxA device has TTL-compatible inputs, then the device will not dissipate much input buffer power. This is because the OR2TxxA output is being driven to a higher level than the TTL level required. If the other device has a CMOS-compatible input, the amount of input buffer power will also be small. Both of these power values are dependent upon the input buffer characteristics of the other device when driven at the OR2TxxA output buffer voltage levels.

Global 3-State Functionality

To increase the testability of the *ORCA* Series FPGAs, the global 3-state function (ts_all) disables the device. The ts_all signal is driven from either an external pin or an internal signal. Before and during configuration, the ts_all signal is driven by the input pad $\overline{\text{RD\_CFGN}}$. After configuration, the ts_all signal can be disabled, driven from the $\overline{\text{RD\_CFGN}}$ input pad, or driven by a general routing signal in the upper-right corner. Before configuration, ts_all is active-low; after configuration, the sense of ts_all can be inverted.

The following occur when ts_all is activated:

1. All of the user I/O output buffers are 3-stated, the user I/O input buffers are pulled up (with the pull-down disabled), and the input buffers are configured with TTL input thresholds.
2. The TDO/RD_DATA output buffer is 3-stated.
3. The $\overline{\text{RD\_CFGN}}$, $\overline{\text{RESET}}$, and $\overline{\text{PRGM}}$ input buffers remain active with a pull-up.
4. The DONE output buffer is 3-stated and the input buffer is pulled-up.

PIC Routing Resources

The PIC routing is designed to route 4-bit wide buses efficiently. For example, any four consecutive I/O pads can have both their input and output signals routed into one PLC. Using only PIC routing, either the input or output data can be routed to/from a single PLC from/to any eight pads in a row, as in Figure 25.

The connections between PLCs and the I/O pad are provided by two basic types of routing resources. These are routing resources internal to the PIC and routing resources used for PIC-PLC connection. Figure 26 and Figure 27 show a high-level and detailed view of these routing resources, respectively.

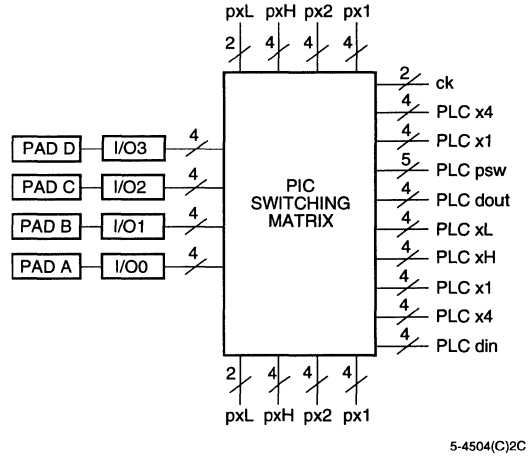


Figure 25. Simplified PIC Routing Diagram

Programmable Input/Output Cells

(continued)

The PIC's name is represented by a two-letter designation to indicate on which side of the device it is located followed by a number to indicate in which row or column it is located. The first letter, P, designates that the cell is a PIC and not a PLC. The second letter indicates the side of the array where the PIC is located. The four sides are left (L), right (R), top (T), and bottom (B). The individual I/O pad is indicated by a single letter (either A, B, C, or D) placed at the end of the PIC name. As an example, PL10A indicates a pad located on the left side of the array in the tenth row.

Each PIC has four pads and each pad can be configured as an input, an output (3-statable), a direct output, or a bidirectional I/O. When the pads are used as inputs, the external signals are provided to the internal circuitry at in[3:0]. When the pads are used to provide direct inputs to the latches/FFs, they are connected through din[3:0]. When the pads are used as outputs, the internal signals connect to the pads through out[3:0]. When the pads are used as direct outputs, the output from the latches/flip-flops in the PLCs to the PIC is designated dout[3:0]. When the outputs are 3-statable, the 3-state enable signals are ts[3:0].

Routing Resources Internal to the PIC

For inter-PIC routing, the PIC contains fourteen R-nodes used to route signals around the perimeter of the FPGA. Figure 25 shows these lines running vertically for a PIC located on the left side. Figure 26 shows the R-nodes running horizontally for a PIC located at the top of the FPGA.

pxL R-Nodes. Each PIC has two pxL R-nodes, labelled pxL[1:0]. Like the xL R-nodes of the PLC, the pxL R-nodes span the entire edge of the FPGA.

pxH R-Nodes. Each PIC has four pxH R-nodes, labelled pxH[3:0]. Like the xH R-nodes of the PLC, the pxH R-nodes span half the edge of the FPGA.

px2 R-Nodes. There are four px2 R-nodes in each PIC, labelled px2[3:0]. The px2 R-nodes pass through two adjacent PICs before being broken. These are used to route nets around the perimeter equally a distance of two or more PICs.

px1 R-Nodes. Each PIC has four px1 R-nodes, labelled px1[3:0]. The px1 R-nodes are one PIC long and are extended to adjacent PICs by enabling CIPs.

Programmable Input/Output Cells

(continued)

PIC Architectural Description

The PIC architecture given in Figure 26 is described using the following letter references. The figure depicts a PIC at the top of the array, so inter-PIC routing is horizontal and the indirect PIC-PLC routing is horizontal to vertical. In some cases, letters are provided in more than one location to indicate the path of an R-node.

A. As in the PLCs, the PIC contains a set of R-nodes which run the length (width) of the array. The pxL R-nodes connect in the corners of the array to other pxL R-nodes. The pxL R-nodes also connect to the PIC BIDI, PIC BIDIH, and LLDRV R-nodes. As in the PLC xL R-nodes, the pXH R-nodes twist as they propagate through the PICs.

B. As in the PLCs, the PIC contains a set of R-nodes which run one-half the length (width) of the array. The pxH R-nodes connect in the corners and in the middle of the array perimeter to other pxH R-nodes. The pxH R-nodes also connect to the PIC BIDI, PIC BIDIH, and LLDRV R-nodes. As in the PLC xH R-nodes, the pxH R-nodes do not twist as they propagate through the PICs.

C. The px2[3:0] R-nodes span a length of two PICs before intersecting with a CIP. The CIP allows the length of a path using px2 R-nodes to be extended two PICs.

D. The px1[3:0] R-nodes span a single PIC before intersecting with a CIP. The CIP allows the length of a path using px1 R-nodes to be extended by one PIC.

E. These are four dedicated direct output R-nodes connected to the output buffers. The dout[3:0] signals go directly from a PLC latch/FF to an output buffer, minimizing the latch/FF to pad propagation delay.

F. This is a direct path from the input pad to the PLC latch/flip-flops in the two rows (columns) adjacent to PICs. This input allows a reduced setup time. Direct inputs from the top and bottom PIC rows are pdintb[3:0]. Direct inputs from the left and right PIC columns are pdinlr[3:0].

G. The out[3:0], ts[3:0], and in[3:0] signals for each I/O pad can be routed directly to the adjacent PLC's switching R-nodes.

H. The four TRIDI buffers allow connections from the pads to the PLC xL R-nodes. The TRIDIs also allow connections between the PLC xL R-nodes and the pBIDI R-nodes, which are described in **J** below.

I. The four TRIDIH buffers allow connections from the pads to the PLC xH R-nodes. The TRIDIHs also allow connections between the PLC xH R-nodes and the pBIDIH R-nodes, which are described in **K** below.

J. The pBIDI R-nodes (bidi[3:0]) connect the pxL R-nodes, pxH R-nodes, and the px1 R-nodes. These are bidirectional in that the path can be from the pxL, pxH, or px1 R-nodes to the xL R-nodes, or from the xL R-nodes to the pxL, pxH, or px1 R-nodes.

K. The pBIDIH R-nodes (bidih[3:0]) connect the pxL R-nodes, pxH R-nodes, and the px1 R-nodes. These are bidirectional in that the path can be from the pxL, pxH, or px1 R-nodes to the xH R-nodes, or from the xH R-nodes to the pxL, pxH, or px1 R-nodes.

L. The llin[3:0] R-nodes provide a fast connection from the I/O pads to the xL and xH R-nodes.

M. This set of CIPs allows the eight x1 R-nodes (four on each side) of the PLC perpendicular to the PIC to be connected to either the px1 or px2 R-nodes in the PIC.

N. This set of CIPs allows the eight x4 R-nodes (four on each side) of the PLC perpendicular to the PIC to be connected to the px1 R-nodes. This allows fast access to/from the I/O pads from/to the PLCs.

O. All four of the PLC x4 R-nodes in a group connect to all four of the PLC x4 R-nodes in the adjacent PLC through a CIP. (This differs from the *ORCA* 1C Series in which two of the x4 R-nodes in adjacent PLCs are directly connected without any CIPs.)

P. The long-line driver (LLDRV) R-node can be driven by the xsw4 switching R-node of the adjacent PLC. To provide connectivity to the pads, the LLDRV R-node can also connect to any of the four pxH or to one of the pxL R-nodes. The 3-state enable (ts[i]) for all four I/O pads can be driven by xsw4, pxH, or pxL R-nodes.

Q. For fast clock routing, one of the four I/O pads in each PIC can be selected to be driven onto a dedicated clock R-node. The clock R-node spans the length (width) of the PLC array. This dedicated clock R-node is typically used as a clock spine. In the PLCs, the spine is connected to an xL R-node to provide a clock branch in the perpendicular direction. Since there is another clock R-node in the PIC on the opposite side of the array, only one of the I/O pads in a given row (column) can be used to generate a global signal in this manner, if all PLCs are driven by the signal.

Programmable Input/Output Cells (continued)

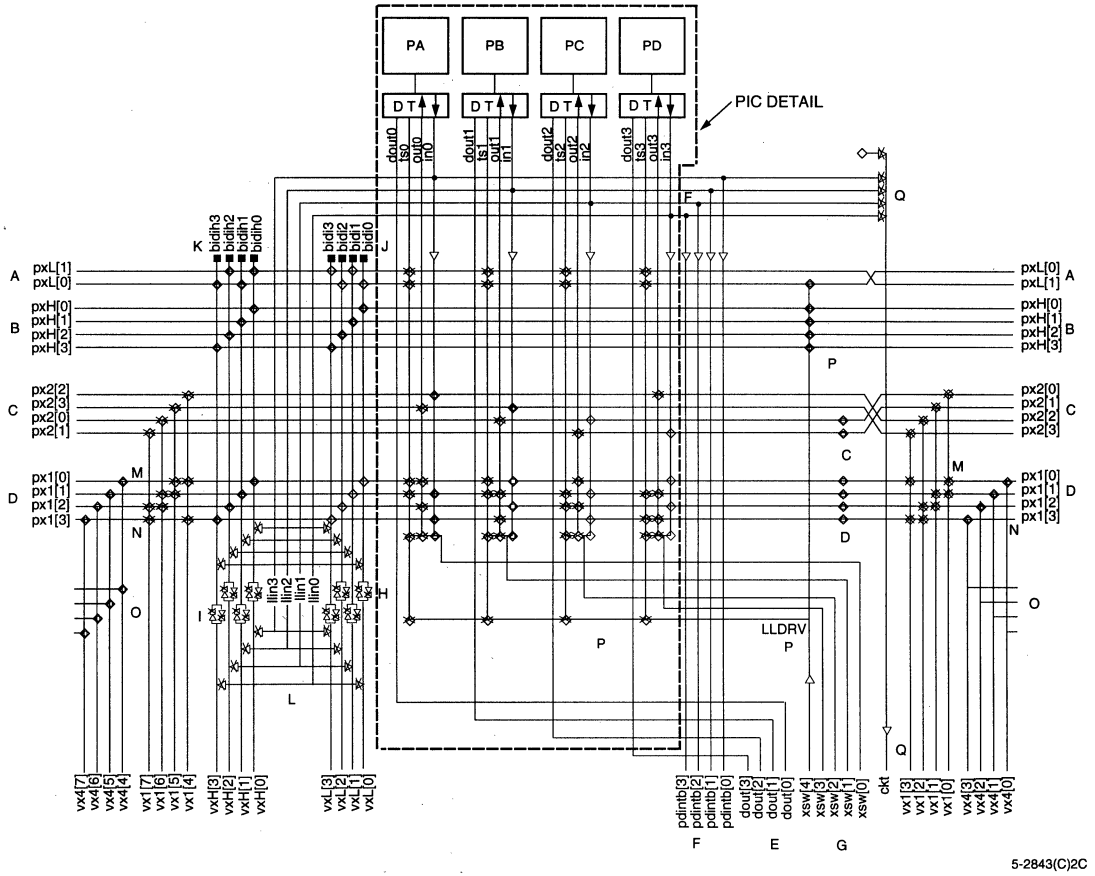


Figure 26. PIC Architecture

Programmable Input/Output Cells

(continued)

PLC-PIC Routing Resources

There is no direct connection between the inter-PIC R-nodes and the PLC R-nodes. All connections to/from the PLC must be done through the connecting R-nodes which are perpendicular to the R-nodes in the PIC. The use of perpendicular and parallel R-nodes will be clearer if the PLC and PIC architectures (Figure 23 and Figure 26) are placed side by side. Twenty-nine R-nodes in the PLC can be connected to the fifteen R-nodes in the PIC.

Multiple connections between the PIC px1 R-nodes and the PLC x1 R-nodes are available. These allow buses placed in any arbitrary order on the I/O pads to be unscrambled when placed on the PLC x1 R-nodes. Connections are also available between the PIC px2 R-nodes and the PLC x1 R-nodes.

There are eight tridirectional (four TRIDI/four TRIDIH) buffers in each PIC; they can do the following:

- Drive a signal from an I/O pad onto one of the adjacent PLC's xL or xH R-nodes
- Drive a signal from an I/O pad onto one of the two pxL or four pxH R-nodes in the PIC
- Drive a signal from the PLC xL or xH R-nodes onto one of the two pxL or four pxH R-nodes in the PIC
- Drive a signal from the PIC pxL or pxH R-nodes onto one of the PLC xL or xH R-nodes

Figure 27 shows paths to and from pads and the use of MUX CIPs to connect R-nodes. Detail A shows six MUX CIPs for the pad P0 used to construct the net for the 3-state signal. In the MUX CIP, one of six R-nodes is connected to an R-node to form the net. In this case, the ts0 signal can be driven by either of the two pxLs, px1[0], px1[1], xsw[0], or the lldrv R-nodes. Detail B shows the four MUX CIPs used to drive the P1 output. The source R-node for out1 is either xsw[1], px1[1], px1[3], or px2[2].

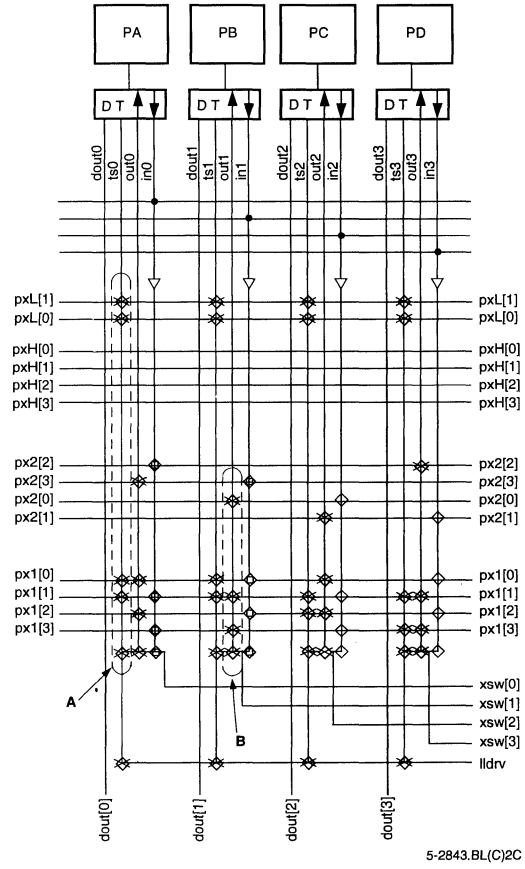


Figure 27. PIC Detail

Interquad Routing

In all the *ORCA* 2C Series devices, the PLC array is split into four equal quadrants. In between these quadrants, routing has been added to route signals between the quadrants, especially to the quadrant in the opposite corner. The two types of interquad blocks, vertical and horizontal, are pitch matched to PICs. Vertical interquad blocks (vIQ) run between quadrants on the left and right, while horizontal interquad blocks (hIQ) run between top and bottom quadrants. Since hIQ and vIQ blocks have the same logic, only the hIQ block is described below.

The interquad routing connects xL and xH R-nodes. It does not affect local routing (xsw, x1, x4, fast carry), so local routing is the same, whether PLC-PLC connections cross quadrants or not. There are no connections to the local R-nodes in the interquad blocks. Figure 28 presents a (not to scale) view of interquad routing.

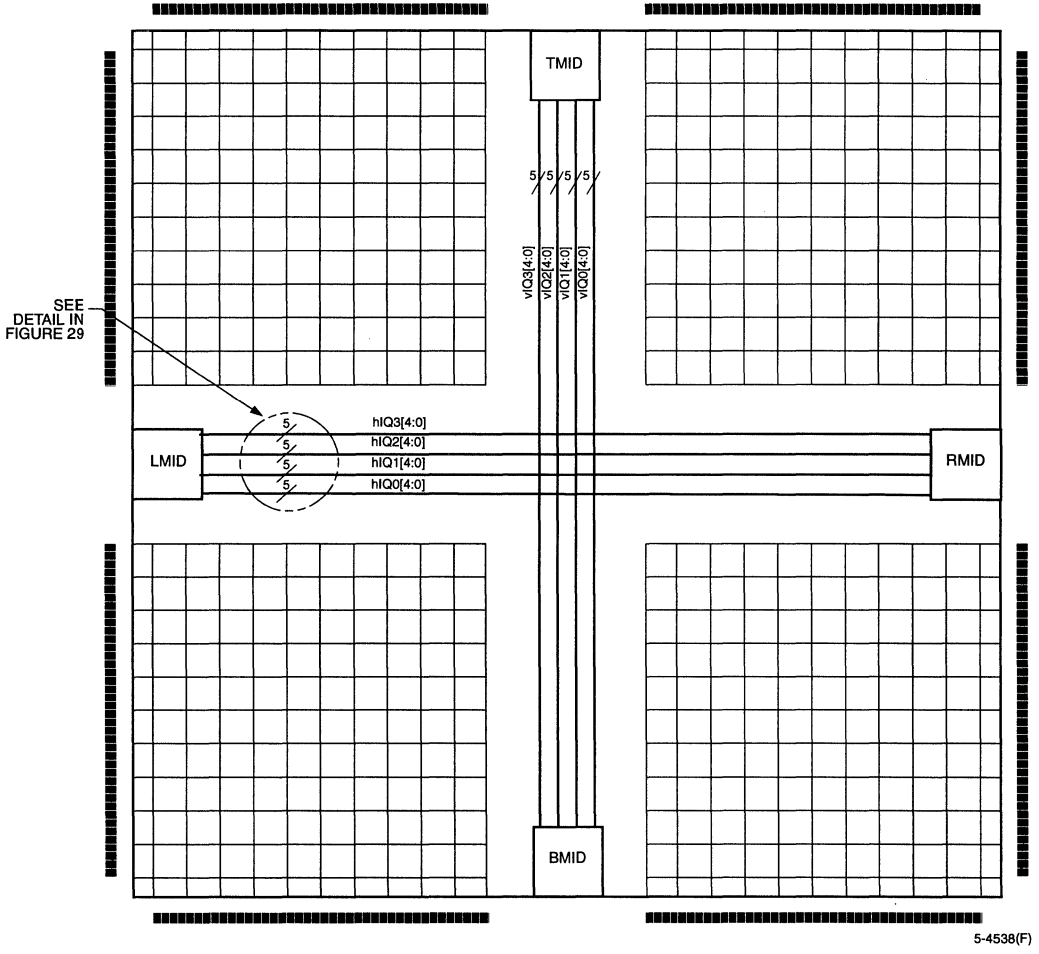


Figure 28. Interquad Routing

Interquad Routing (continued)

In the hIQ block in Figure 29, the xH R-nodes from one quadrant connect through a CIP to its counterpart in the opposite quadrant, creating a path that spans the PLC array. Since a passive CIP is used to connect the two xH R-nodes, a 3-state signal can be routed on the two xH R-nodes in the opposite quadrants, and then they can be connected through this CIP.

In the hIQ block, the 20 hIQ R-nodes span the array in a horizontal direction. The 20 hIQ R-nodes consist of four groups of five R-nodes each. To effectively route nibble-wide buses, each of these sets of five R-nodes can connect to only one of the bits of the nibble for both the xH and xL. For example, hIQ0 R-nodes can only connect to the xH0 and xL0 R-nodes, and the hIQ1 R-nodes can connect only to the xH1 and xL1 R-nodes, etc. Buffers are provided for routing signals from the xH and xL R-nodes onto the hIQ R-nodes and from the hIQ R-nodes onto the xH and xL R-nodes. Therefore, a connection from one quadrant to another can be made using only two xH R-nodes (one in each quadrant) and one interquad R-node.

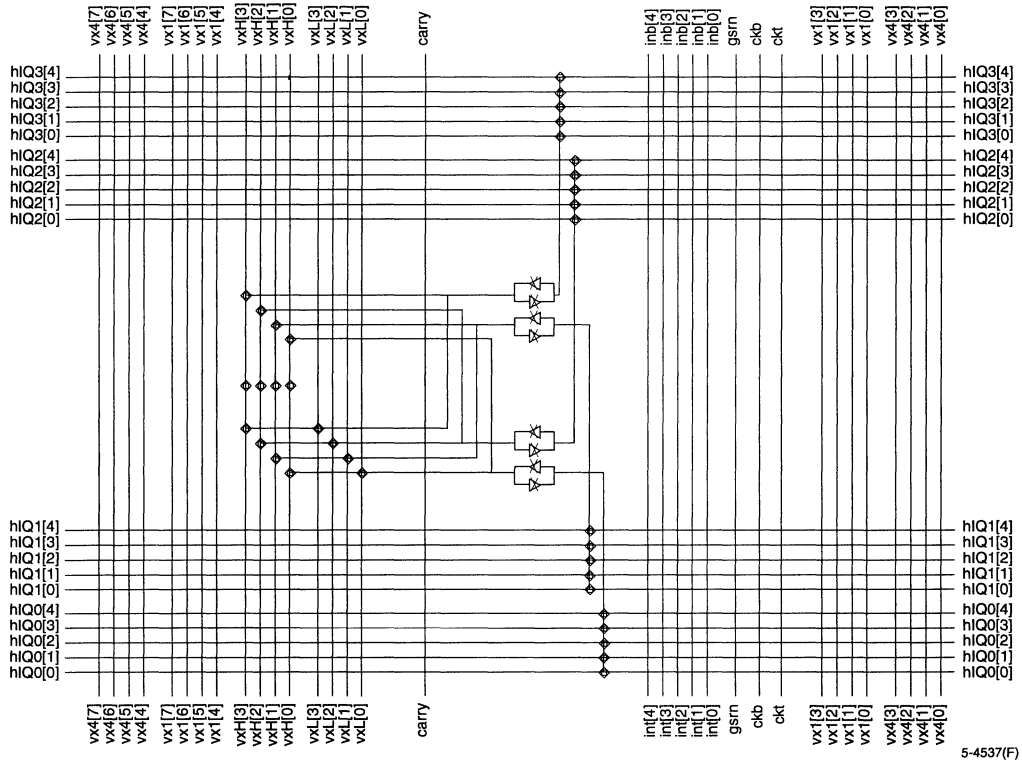


Figure 29. hIQ Block Detail

Interquad Routing (continued)

Subquad Routing

In the *ORCA* OR2C40A/OR2T40A, each quadrant of the device is split into smaller arrays of PLCs called subquads. Each of these subquads is made of a 4 x 4 array of PLCs (for a total of 16 per subquadrant), except at the outer edges of array, which have less than 16 PLCs per subquad. New routing resources, called subquad R-nodes, have been added between each adjacent pair of subquads to enhance the routability of the device. A portion of the center of the OR2C40A and OR2T40A array is shown in Figure 30, including the subquad blocks containing a 4 x 4 array of PLCs, the interquad routing R-nodes, and the subquad routing R-nodes.

All of the inter-PLC routing resources discussed previously continue to be routed between a PLC and its adjacent PLC, even if the two adjacent PLCs are in different subquad blocks. Since the PLC routing has not been modified for the OR2C40A/OR2T40A architectures, this means that all of the same routing connections are possible for these devices as for any other *ORCA* 2C series device. In this way, both the OR2C40A and OR2T40A are upwardly compatible when compared with the ATT2Cxx series devices. As the inter-PLC routing runs between subquad blocks, it crosses the new subquad R-nodes. When this happens, CIPs are used to connect the subquad R-nodes to the x4 and/or the xH R-nodes which lie along the other axis of the PLC array.

The x4 and xH R-nodes make the only connections to the subquad R-nodes; therefore, the array remains symmetrical and homogeneous. Since each subquad is made from a 4 x 4 array of PLCs, the distance between sets of subquad R-nodes is four PLCs, which is also the distance between the breaks of the x4 R-nodes. Therefore, each x4 R-node will cross exactly one set of subquad R-nodes. Since all x4 R-nodes make the same connections to the subquad R-nodes that they cross, all x4 R-nodes in the array have the same connectivity, and the symmetry of the routing is preserved. Since all xH R-nodes cross the same number of subquad blocks, the symmetry is maintained for the xH R-nodes as well.

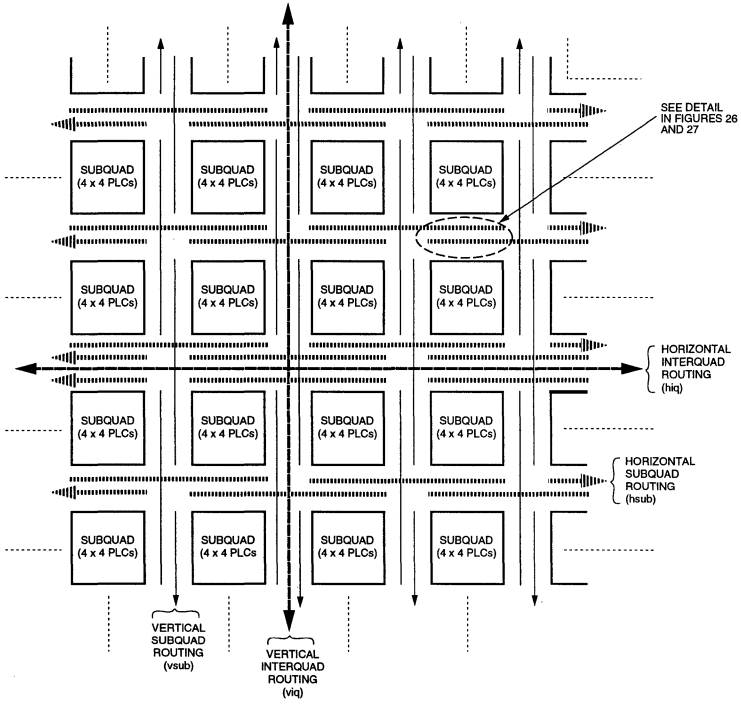


Figure 30. Subquad Blocks and Subquad Routing

Interquad Routing (continued)

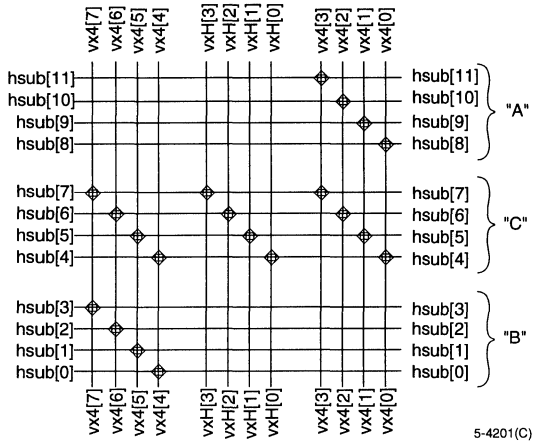


Figure 31. Horizontal Subquad Routing Connectivity

The new subquad R-nodes travel a length of eight PLCs (seven PLCs on the outside edge) before they are broken. Unlike other inter-PLC R-nodes, they cannot be connected end-to-end. As shown in Figure 30, some of the horizontal (vertical) subquad R-nodes have connectivity to the subquad to the left of (above) the current subquad, while others have connectivity to the subquad to the right (below). This allows connections to/from the current subquad from/to the PLCs in all subquads that surround it.

Between all subquads, including in the center of the array, there are three groups of subquad R-nodes where each group contains four R-nodes. Figure 31 shows the connectivity of these three groups of subquad R-nodes (hsub) to the vx4 and vxH R-nodes running between a vertical pair of PLCs. Between each vertical pair of subquad blocks, four of the blocks shown in Figure 31 are used, one for each pair of vertical PLCs.

The first two groups, depicted as A and B, have connectivity to only one of the two sets of x4 R-nodes between pairs of PLCs. Since they are very lightly loaded, they are very fast. The third group, C, connects to both groups of x4 R-nodes between pairs of PLCs, as well as all of the xH R-nodes between pairs of PLCs, providing high flexibility. The connectivity for the vertical subquad routing (vsub) is the same as described above for the horizontal subquad routing, when rotated onto the other axis.

At the center row and column of each quadrant, a fourth group of subquad R-nodes has been added. These subquad R-nodes only have connectivity to the xH R-nodes. The xH R-nodes are also broken at this point, which means that each xH R-node travels one-half of the quadrant (i.e., one-quarter of the device) before it is broken by a CIP. Since the xH R-nodes can be connected end-to-end, the resulting line can be either one-quarter, one-half, three-quarters, or the entire length of the array. The connectivity of the xH R-nodes and this fourth group of subquad R-nodes, indicated as D, are detailed in Figure 32. Again, the connectivity for the vertical subquad routing (vsub) is the same as the horizontal subquad routing, when rotated onto the other axis.

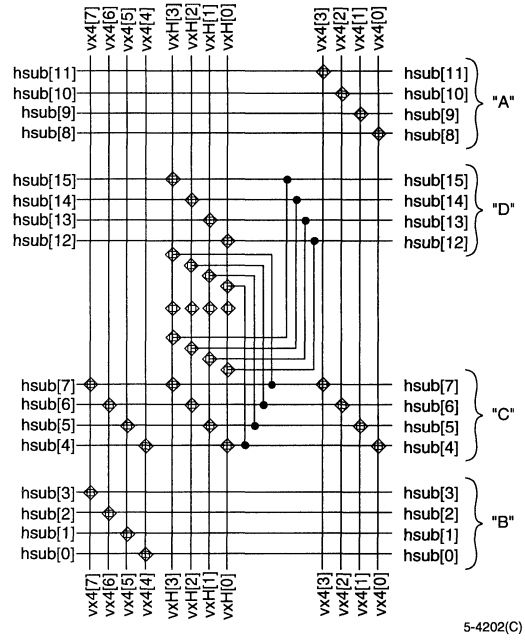


Figure 32. Horizontal Subquad Routing Connectivity (Half Quad)

Interquad Routing (continued)

PIC Interquad (MID) Routing

Between the PICs in each quadrant, there is also connectivity between the PIC routing and the interquad routing. These blocks are called LMID (left), TMID (top), RMID (right), and BMID (bottom). The TMID routing is shown in Figure 33. As with the hIQ and vIQ blocks, the only connectivity to the PIC routing is to the global pxH and pxL R-nodes.

The pxH R-nodes from the one quadrant can be connected through a CIP to its counterpart in the opposite quadrant, providing a path that spans the array of PICs. Since a passive CIP is used to connect the two pxH R-nodes, a 3-state signal can be routed on the two pxH R-nodes in the opposite quadrants, and then connected through this CIP. As with the hIQ and vIQ blocks, CIPs and buffers allow nibble-wide connections between the interquad R-nodes, the xH R-nodes, and the xL R-nodes.

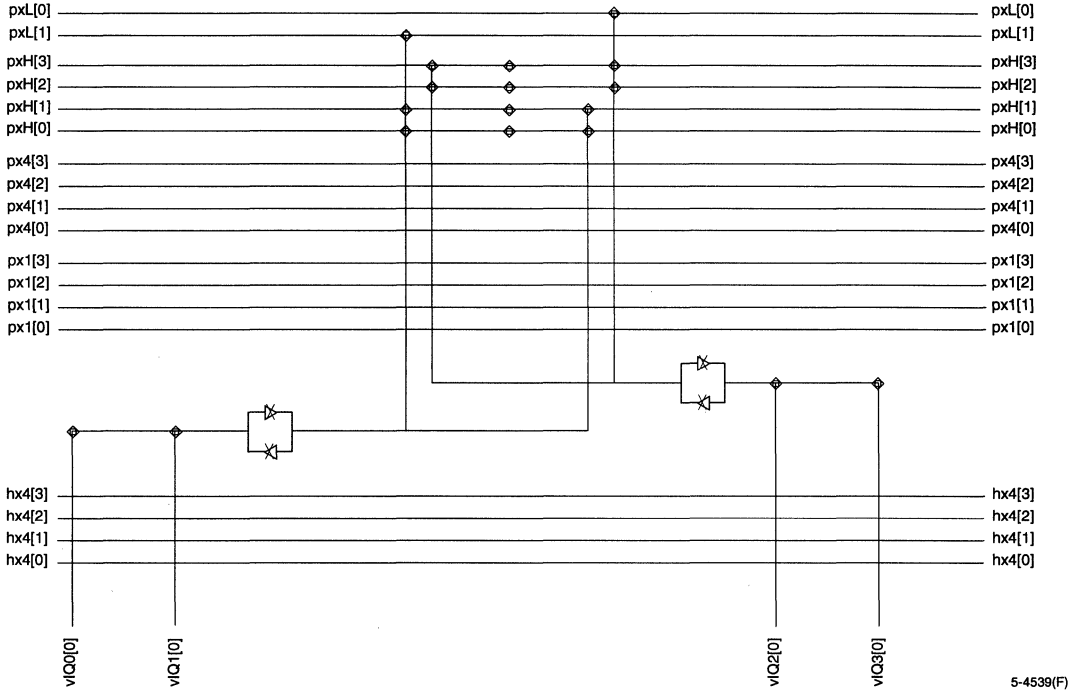


Figure 33. Top (TMID) Routing

Programmable Corner Cells

Programmable Routing

The programmable corner cell (PCC) contains the circuitry to connect the routing of the two PICs in each corner of the device. The PIC px1 and px2 R-nodes are directly connected together from one PIC to another. The PIC pxL R-nodes are connected from one block to another through tridirectional buffers. Four CIPs in each corner connect the four pxH R-nodes from each side of the device.

Special-Purpose Functions

In addition to routing functions, special-purpose functions are located in each FPGA corner. The upper-left PCC contains connections to the boundary-scan logic. The upper-right PCC contains connections to the readback logic and the connectivity to the global 3-state signal (ts_all). The lower-left PCC contains connections to the internal oscillator.

The lower-right PCC contains connections to the start-up and global reset logic. During configuration, the $\overline{\text{RESET}}$ input pad always initiates a configuration abort, as described in the FPGA States of Operation section. After configuration, the global set/reset signal (gsrn) can either be disabled (the default), directly connected to the $\overline{\text{RESET}}$ input pad, or sourced by a lower-right corner signal. If the $\overline{\text{RESET}}$ input pad is not used as a global reset after configuration, this pad can be used as a normal input pad. During start-up, the release of the global set/reset, the release of the I/Os, and the release of the external DONE signal can each be timed individually based upon the start-up clock. The start-up clock can come from CCLK or it can be routed into the start-up block using the lower-right corner routing resources. More details on start-up can be found in the FPGA States of Operation section.

Clock Distribution Network

The *ORCA* OR2CxxA/OR2TxxA Series clock distribution schemes use primary and secondary clocks. This provides the system designer with additional flexibility in assigning clock input pins.

One advantage is that board-level clock traces routed to the FPGA are shorter. On a PC board, the added length of high-speed clock traces routed to dedicated clock input pins can significantly increase the parasitic impedances. The primary advantage of the *ORCA* clock distribution is the availability of a large number of clocks, since all I/O pins are configurable as clocks.

Primary Clock

The primary clock distribution is shown in Figure 34. If the clock signal is from an I/O pad, it can be driven onto a clock R-node. The clock R-nodes do not provide clock signals directly to the PFU; they act as clock spines from which clocks are branched to xL R-nodes. The xL R-nodes then feed the clocks to PFUs. A multiplexer in each PLC is used to transition from the clock spine to the branch.

For a clock spine in the horizontal direction, the inputs into the multiplexer are the two R-nodes from the left and right PICs (ckl and ckr) and the local clock R-node from the perpendicular direction (hck). This signal is then buffered and driven onto one of the vertical xL R-nodes, forming the branches. The same structure is used for a clock spine in the vertical direction. In this case, the multiplexer selects from R-nodes from the top and bottom PICs (ckt, ckb, and vck) and drives the signal onto one of the horizontal xL R-nodes.

Figure 34 illustrates the distribution of the low-skew primary clock to a large number of loads using a main spine and branches. Each row (column) has two dedicated clock R-nodes originating from PICs on opposite sides of the array. The clock is input from the pads to the dedicated clock R-node ckt to form the clock spine (see Figure 34, Detail A). From the clock spine, net branches are routed using horizontal xL R-nodes and then PLC clock inputs are tapped from the xL R-nodes, as shown in Figure 34, Detail B.

Clock Distribution Network (continued)

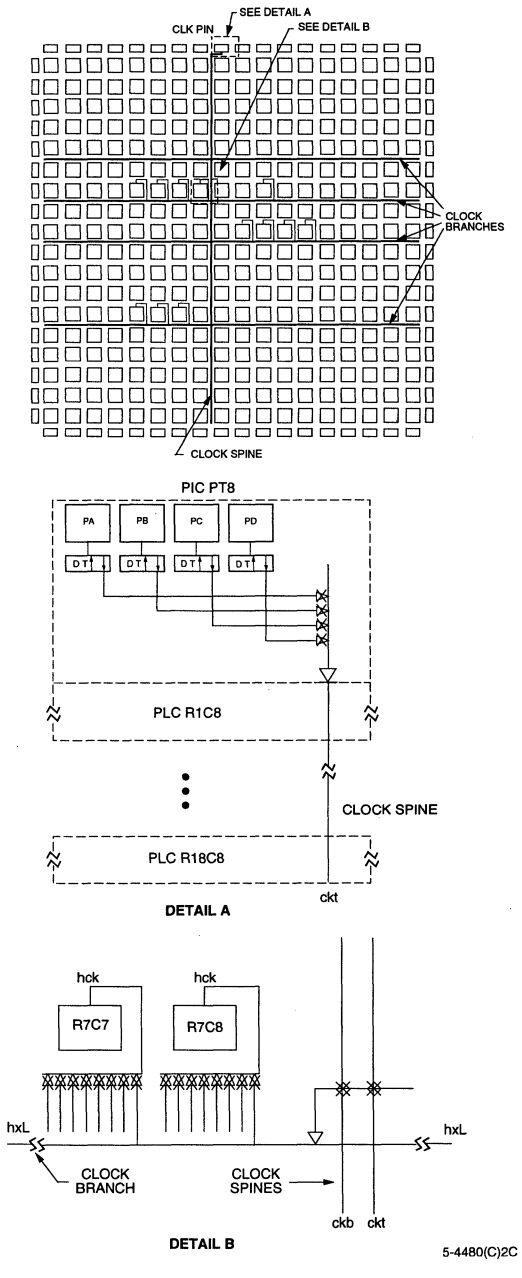


Figure 34. Primary Clock Distribution

Secondary Clock

There are times when a primary clock is either not available or not desired, and a secondary clock is needed. For example:

- Only one input pad per PIC can be placed on the clock routing. If a second input pad in a given PIC requires global signal routing, a secondary clock route must be used.
- Since there is only one branch driver in each PLC for either direction (vertical and horizontal), both clock R-nodes in a particular row or column (ckl and ckr, for example) cannot drive a branch. Therefore, two clocks should not be placed into I/O pads in PICs on the opposite sides of the same row or column if global clocks are to be used.
- Since the clock R-nodes can only be driven from input pads, internally generated clocks should use secondary clock routing.

Figure 35 illustrates the secondary clock distribution. If the clock signal originates from either the left or right side of the FPGA, it can be routed through the TRIDI buffers in the PIC onto one of the adjacent PLC's horizontal xL R-nodes. If the clock signal originates from the top or bottom of the FPGA, the vertical xL R-nodes are used for routing. In either case, an xL R-node is used as the clock spine. In the same manner, if a clock is only going to be used in one quadrant, the xH R-nodes can be used as a clock spine. The routing of the clock spine from the input pads to the vxL (vxH) using the BIDIs (BIDIHs) is shown in Figure 35, Detail A.

In each PLC, a low-skew connection through a long-line driver can be used to connect a horizontal xL R-node to a vertical xL R-node or vice versa. As shown in Figure 35, Detail B, this is used to route the branches from the clock spine. If the clock spine is a vertical xL R-node, then the branches are horizontal xL R-nodes and vice versa. The clock is then routed into each PLC from the xL R-node clock branches.

To minimize skew, the PLC clock input for all PLCs must be connected to the branch xL R-nodes, not the spine xL R-node. Even in PLCs where the clock is routed from the spine to the branches, the clock should be routed back into the PLC from the clock branch.

If the clock is to drive only a limited number of loads, the PFUs can be connected directly to the clock spine. In this case, all flip-flops driven by the clock must be located in the same row or column.

Clock Distribution Network (continued)

Alternatively, the clock can be routed from the spine to the branches by using the BIDIs instead of the long-line drivers. This results in added delay in the clock net, but the clock skew is approximately equal to the clock routed using the long-line drivers. This method can be used to create a clock that is used in only one quadrant. The xH R-nodes act as a clock spine, which is then routed to perpendicular xH R-nodes (the branches) using the BIDIHs.

Clock signals, such as the output of a counter, can also be generated in PLCs and routed onto an xL R-node, which then acts as a clock spine. Although the clock can be generated in any PLC, it is recommended that the clock be located as close to the center of the FPGA as possible to minimize clock skew.

Selecting Clock Input Pins

Any user I/O pin on an *ORCA* FPGA can be used as a very fast, low-skew clock input. Choosing the first clock pin is completely arbitrary, but using a pin that is near the center of an edge of the device (as shown in Figures 34 and 35) will provide the lowest skew clock network. The pin-to-pin timing numbers in the Timing Characteristics section of this data book assume that the clock pin is in one of the 4 PICs at the center of any side of the device.

Once the first clock pin has been chosen, there are only two sets of pins (within the center 4 PICs on each side of the device) that should not be chosen as the second clock pin: a pin from the same PIC, and/or a pin from the PIC on the exact opposite edge of the die (i.e., if a pin from a PIC on the top edge is chosen for the first clock, the same PIC on the bottom edge should not be chosen for the second clock).

These rules should be followed iteratively until a total of 8 clocks (or other global signals) have been selected: 4 from the left/right sides of the device, and 4 from the top/bottom sides of the device. If more than 8 clocks are needed, then select another pin outside the center 4 PICs to use primary-clock routing, use secondary clock routing for any pin, or use local clock routing.

For primary clock routing for one of the first 8 clocks that does not use a pin within the center 4 PICs, the pad names (see Pin Information) of the two clock pins on the top or bottom of the device **cannot** end with the same letter (e.g., PT14A and PB2A). The same rule applies to clock pins on the left or right side of the device.

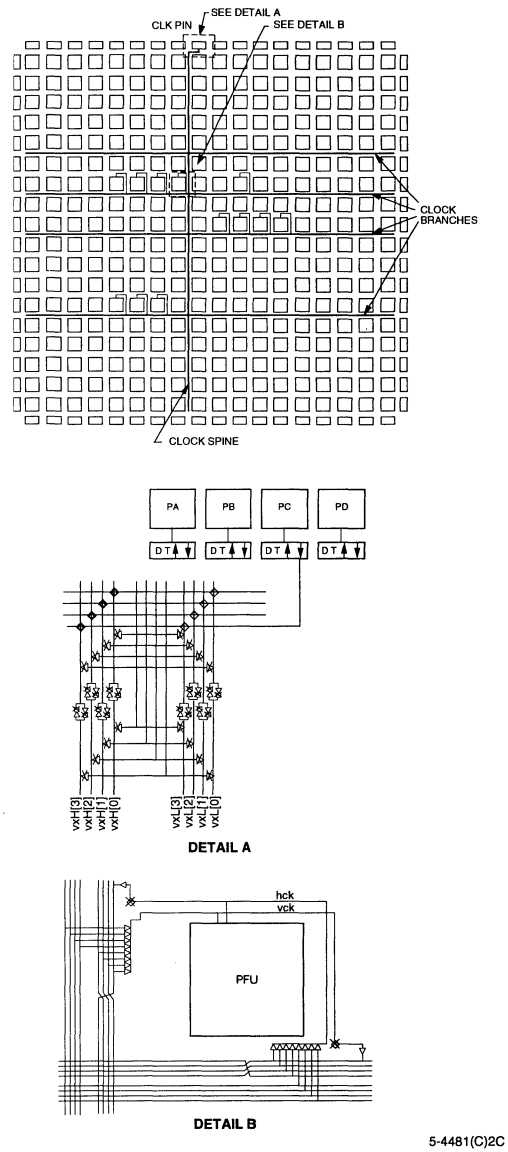


Figure 35. Secondary Clock Distribution

FPGA States of Operation

Prior to becoming operational, the FPGA goes through a sequence of states, including initialization, configuration, and start-up. Figure 36 outlines these three FPGA states.

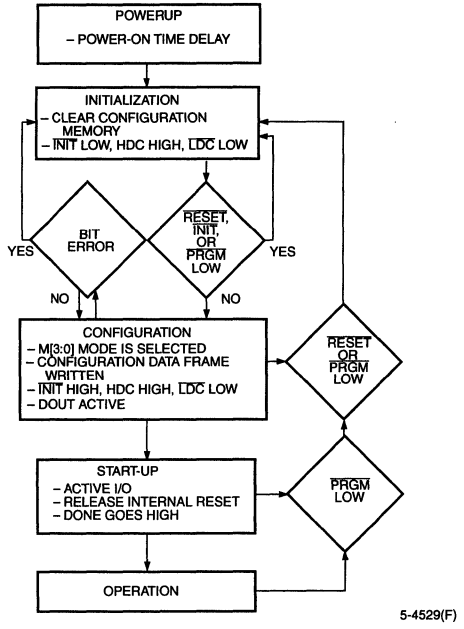


Figure 36. FPGA States of Operation

Initialization

Upon powerup, the device goes through an initialization process. First, an internal power-on-reset circuit is triggered when power is applied. When VDD reaches the voltage at which portions of the FPGA begin to operate (2.5 V to 3 V for the OR2CxxA, 2.2 V to 2.7 V for the OR2TxxA), the I/Os are configured based on the configuration mode, as determined by the mode select inputs M[2:0]. A time-out delay is initiated when VDD reaches between 3.0 V and 4.0 V (OR2CxxA) or 2.7 V to 3.0 V (OR2TxxA) to allow the power supply voltage to stabilize. The $\overline{\text{INIT}}$ and DONE outputs are low. At powerup, if VDD does not rise from 2.0 V to VDD in less than 25 ms, the user should delay configuration by inputting a low into $\overline{\text{INIT}}$, $\overline{\text{PRGM}}$, or $\overline{\text{RESET}}$ until VDD is greater than the recommended minimum operating voltage (4.75 V for OR2CxxA commercial devices and 3.0 V for OR2TxxA devices).

At the end of initialization, the default configuration option is that the configuration RAM is written to a low state. This prevents shorts prior to configuration. As a configuration option, after the first configuration (i.e., at reconfiguration), the user can reconfigure without clearing the internal configuration RAM first.

The active-low, open-drain initialization signal $\overline{\text{INIT}}$ is released and must be pulled high by an external resistor when initialization is complete. To synchronize the configuration of multiple FPGAs, one or more $\overline{\text{INIT}}$ pins should be wire-ANDed. If $\overline{\text{INIT}}$ is held low by one or more FPGAs or an external device, the FPGA remains in the initialization state. $\overline{\text{INIT}}$ can be used to signal that the FPGAs are not yet initialized. After $\overline{\text{INIT}}$ goes high for two internal clock cycles, the mode lines are sampled and the FPGA enters the configuration state.

The high during configuration (HDC), low during configuration ($\overline{\text{LDC}}$), and DONE signals are active outputs in the FPGA's initialization and configuration states. HDC, $\overline{\text{LDC}}$, and DONE can be used to provide control of external logic signals such as reset, bus enable, or PROM enable during configuration. For parallel master configuration modes, these signals provide PROM enable control and allow the data pins to be shared with user logic signals.

If configuration has begun, an assertion of $\overline{\text{RESET}}$ or $\overline{\text{PRGM}}$ initiates an abort, returning the FPGA to the initialization state. The $\overline{\text{PRGM}}$ and $\overline{\text{RESET}}$ pins must be pulled back high before the FPGA will enter the configuration state. During the start-up and operating states, only the assertion of $\overline{\text{PRGM}}$ causes a reconfiguration.

In the master configuration modes, the FPGA is the source of configuration clock (CCLK). In this mode, the initialization state is extended to ensure that, in daisy-chain operation, all daisy-chained slave devices are ready. Independent of differences in clock rates, master mode devices remain in the initialization state an additional six internal clock cycles after $\overline{\text{INIT}}$ goes high.

When configuration is initiated, a counter in the FPGA is set to 0 and begins to count configuration clock cycles applied to the FPGA. As each configuration data frame is supplied to the FPGA, it is internally assembled into data words. Each data word is loaded into the internal configuration memory. The configuration loading process is complete when the internal length count equals the loaded length count in the length count field, and the required end of configuration frame is written.

All OR2CxxA I/Os operate as TTL inputs during configuration (OR2TxxA I/Os are CMOS-only). All I/Os that are not used during the configuration process are 3-stated with internal pull-ups. During configuration, the PLC latch/FFs are held set/reset and the internal BIDI buffers are 3-stated. The TRIDIs in the PICs are not 3-stated. The combinatorial logic begins to function as the FPGA is configured. Figure 37 shows the general waveform of the initialization, configuration, and start-up states.

FPGA States of Operation (continued)

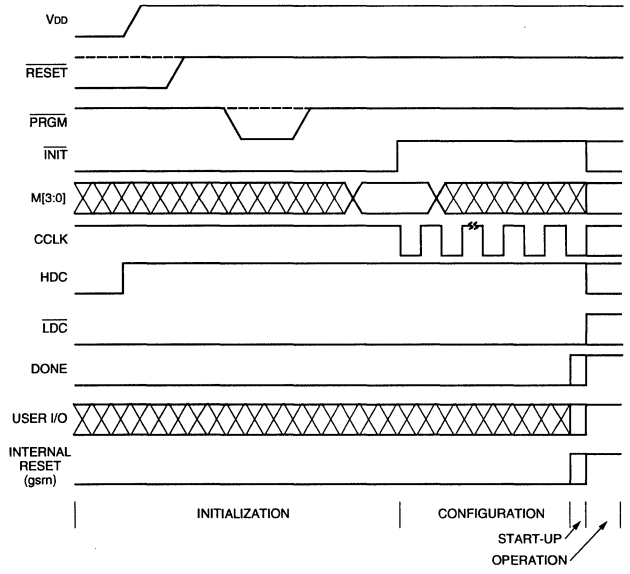


Figure 37. Initialization/Configuration/Start-Up Waveforms

Configuration

The *ORCA* Series FPGA functionality is determined by the state of internal configuration RAM. This configuration RAM can be loaded in a number of different modes. In these configuration modes, the FPGA can act as a master or a slave of other devices in the system. The decision as to which configuration mode to use is a system design issue. The next section discusses configuration in detail, including the configuration data format and the configuration modes used to load the configuration data in the FPGA.

Start-Up

After configuration, the FPGA enters the start-up phase. This phase is the transition between the configuration and operational states and begins when the number of CCLKs received after $\overline{\text{INIT}}$ goes high is equal to the value of the length count field in the configuration frame and when the end of configuration frame has been written. The system design issue in the start-up phase is to ensure the user I/Os become active without inadvertently activating devices in the system or causing bus contention. A second system design concern is the timing of the release of global set/reset of the PLC latches/FFs.

There are configuration options that control the relative timing of three events: DONE going high, release of the set/reset of internal FFs, and user I/Os becoming active. Figure 38 shows the start-up timing for both the *ORCA* and ATT3000 Series FPGAs. The system designer determines the relative timing of the I/Os becoming active, DONE going high, and the release of the set/reset of internal FFs. In the *ORCA* Series FPGA, the three events can occur in any arbitrary sequence. This means that they can occur before or after each other, or they can occur simultaneously.

There are four main start-up modes: CCLK_NOSYNC, CCLK_SYNC, UCLK_NOSYNC, and UCLK_SYNC. The only difference between the modes starting with CCLK and those starting with UCLK is that for the UCLK modes, a user clock must be supplied to the start-up logic. The timing of start-up events is then based upon this user clock, rather than CCLK. The difference between the SYNC and NOSYNC modes is that, for SYNC mode, the timing of two of the start-up events (release of the set/reset of internal FFs and the I/Os becoming active) is triggered by the rise of the external DONE pin followed by a variable number of rising clock edges (either CCLK or UCLK). For the NOSYNC mode, the timing of these two events is based only on either CCLK or UCLK.

FPGA States of Operation (continued)

DONE is an open drain bidirectional pin that may include an optional (enabled by default) pull-up resistor to accommodate wired ANDing. The open drain DONE signals from multiple FPGAs can be tied together (ANDed) with a pull-up (internal or external) and used as an active-high ready signal, an active-low PROM enable, or a reset to other portions of the system. When used in SYNC mode, these ANDed DONE pins can be used to synchronize the other two start-up events, since they can all be synchronized to the same external signal. This signal will not rise until all FPGAs release their DONE pins, allowing the signal to be pulled high.

The default for *ORCA* is the CCLK_SYNC synchronized start-up mode where DONE is released on the first CCLK rising edge, C1 (see Figure 38). Since this is a synchronized start-up mode, the open drain DONE signal can be held low externally to stop the occurrence of the other two start-up events. Once the DONE pin has been released and pulled up to a high level, the other two start-up events can be programmed individually to either happen immediately or after up to four rising edges of CCLK (Di, Di + 1, Di + 2, Di + 3, Di + 4). The default is for both events to happen immediately after DONE is released and pulled high.

A commonly used design technique is to release DONE one or more clock cycles before allowing the I/O to become active. This allows other configuration devices, such as PROMs, to be disconnected using the DONE signal so that there is no bus contention when the I/Os become active. In addition to controlling the FPGA during start-up, other start-up techniques that avoid contention include using isolation devices between the FPGA and other circuits in the system, reassigning I/O locations and maintaining I/Os as 3-stated outputs until contentions are resolved.

Each of these start-up options can be selected during bit stream generation in *ORCA* Foundry, using "Advanced Options". For more information, please see the *ORCA* Foundry documentation.

Reconfiguration

To reconfigure the FPGA when the device is operating in the system, a low pulse is input into $\overline{\text{PRGM}}$. The configuration data in the FPGA is cleared, and the I/Os not used for configuration are 3-stated. The FPGA then samples the mode select inputs and begins reconfiguration. When reconfiguration is complete, DONE is released, allowing it to be pulled high.

Partial Reconfiguration

All *ORCA* device families have been designed to allow a partial reconfiguration of the FPGA at any time. This is done by setting a bit stream option in the previous configuration sequence that tells the FPGA to not reset all of the configuration RAM during a reconfiguration. Then only the configuration frames that are to be modified need to be rewritten, thereby reducing the configuration time.

Other bit stream options are also available that allow one portion of the FPGA to remain in operation while a partial reconfiguration is being done. If this is done, the user must be careful to not cause contention between the two configurations (the bit stream resident in the FPGA and the partial reconfiguration bit stream) as the second reconfiguration bit stream is being loaded.

Other Configuration Options

Configuration options used during device start-up were previously discussed in the FPGA States of Operation section of this data sheet. There are many other configuration options available to the user that can be set during bit stream generation in *ORCA* Foundry. These include options to enable boundary scan, readback options, and options to control and use the internal oscillator after configuration.

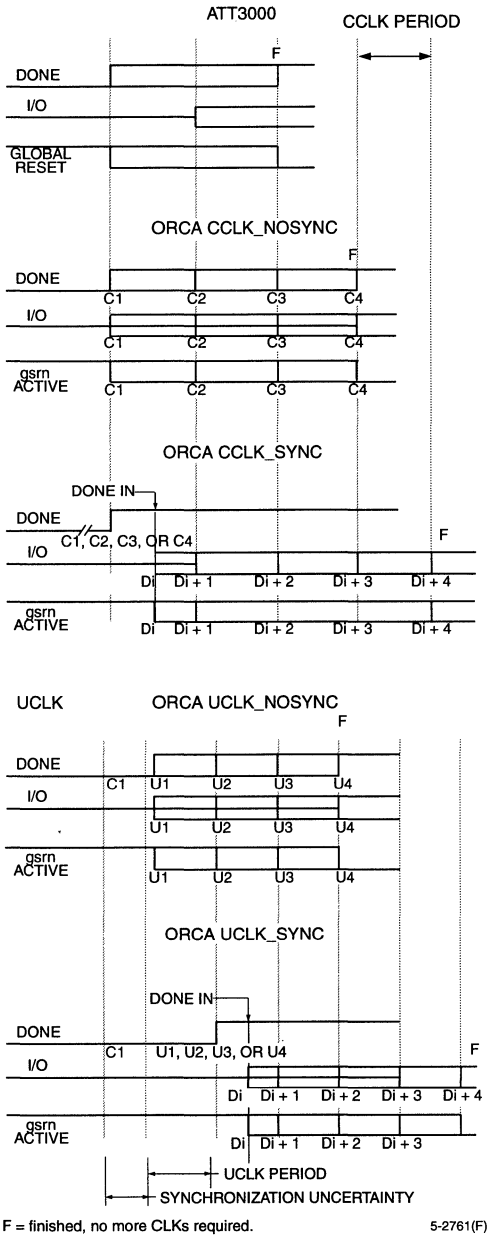
Other useful options that affect the next configuration (not the current configuration process) include options to disable the global set/reset during configuration, disable the 3-state of I/Os during configuration, and disable the reset of internal RAMs during configuration to allow for partial configurations (see above). For more information on how to set these and other configuration options, please see the *ORCA* Foundry documentation.

Configuration Data Format

The *ORCA* Foundry Development System interfaces with front-end design entry tools and provides the tools to produce a fully configured FPGA. This section discusses using the *ORCA* Foundry Development System to generate configuration RAM data and then provides the details of the configuration frame format.

The *ORCA* OR2CxxA/OR2TxxA series of FPGAs are enhanced versions of the *ORCA* ATT2Cxx/ATT2Txx architectures that provide upward bit stream compatibility for both series of devices as well as with each other.

Configuration Data Format (continued)



F = finished, no more CLKs required. 5-2761(F)

Figure 38. Start-Up Waveforms

Using *ORCA* Foundry to Generate Configuration RAM Data

The configuration data defines the I/O functionality, logic, and interconnections. The bit stream is generated by the development system. The bit stream created by the bit stream generation tool is a series of 1s and 0s used to write the FPGA configuration RAM. The bit stream can be loaded into the FPGA using one of the configuration modes discussed later. In the bit stream generator, the designer selects options which affect the FPGA's functionality. Using the output of the bit stream generator, circuit.bit, the development system's download tool can load the configuration data into the *ORCA* series FPGA evaluation board from a PC or workstation. Alternatively, a user can program a PROM (such as the ATT1700A Series Serial ROM or a standard EPROM) and load the FPGA from the PROM. The development system's PROM programming tool produces a file in .mks or .exo format.

Configuration Data Frame

A detailed description of the frame format is shown in Figure 39. The header frame begins with a series of 1s and a preamble of 0010, followed by a 24-bit length count field representing the total number of configuration clocks needed to complete the loading of the FPGAs. Following the header frame is an optional ID frame. This frame contains data used to determine if the bit stream is being loaded to the correct type of *ORCA* FPGA (i.e., a bit stream generated for an OR2C15A is being sent to an OR2C15A). Since the OR2CxxA devices are bit stream compatible with the ATT2Cxx, ATT2Txx, and OR2TxxA families, a bit stream from any of these devices will not cause an error when loaded into an OR2CxxA or OR2TxxA device. The ID frame has a secondary function of optionally enabling the parity checking logic for the rest of the data frames.

The configuration data frames follow. Each frame starts with a 0 start bit and ends with three or more 1 stop bits. Following each start bit are four control bits: a program bit, set to 1 if this is a data frame; a compress bit, set to 1 if this is a compressed frame; and the opar and epar parity bits (see Bit Stream Error Checking). An 11-bit address field that determines in which column the FPGA is to be written is followed by alignment and write control bits. For uncompressed frames, the data bits needed to write one column in the FPGA are next. For compressed frames, the data bits from the previous frame are sent to a different FPGA column, as specified by the new address bits; therefore, new data bits are not required. When configuration of the current FPGA is finished, an end-of-configuration frame (where the program bit is set to 0) is sent to the FPGA. The length and number of data frames and information on the PROM size for the OR2CxxA/OR2TxxA series FPGAs are given in Table 7.

Configuration Data Format (continued)

Table 7. Configuration Frame Size

Devices	OR2C/ 2T04A	OR2C/ 2T06A	OR2C/ 2T08A	OR2C/ 2T10A	OR2C/ 2T12A	OR2C/ 2T15A	OR2C/ 2T26A	OR2C/ 2T40A
# of Frames	480	568	656	744	832	920	1096	1378
Data Bits/Frame	110	130	150	170	190	210	250	316
Configuration Data (# of frames x # of data bits/frame)	52,800	73,840	98,400	126,480	158,080	193,200	274,000	435,448
Maximum Total # Bits/Frame (align bits, 1 write bit, 8 stop bits)	136	160	176	200	216	240	280	344
Maximum Configuration Data (# bits x # of frames)	65,280	90,880	115,456	148,800	179,712	220,800	306,880	474,032
Maximum PROM Size (bits) (add 48-bit header, ID frame, and 40-bit end of configuration frame)	65,504	91,128	115,720	149,088	180,016	221,128	307,248	474,464

The data frames for all the OR2CxxA/OR2TxxA series devices are given in Table 8. An alignment field is required in the slave parallel mode for the uncompressed format. The alignment field (shown by [A]) is a series of 0s: five for the OR2C06A/OR2T06A, OR2C10A/OR2T10A, OR2C15A/OR2T15A, and OR2C26A/OR2T26A; three for the OR2C40A/OR2T40A; and one for the OR2C04A/OR2T04A, OR2C08A/OR2T08A, and OR2C12A/OR2T12A. The alignment field is not required in any other mode.

Table 8. Configuration Data Frames

OR2C04A/OR2T04A	
Uncompressed	010 opar epar [addr10:0] [A]1[Data109:0]111
Compressed	011 opar epar [addr10:0] 111
OR2C06A/OR2T06A	
Uncompressed	010 opar epar [addr10:0] [A]1[Data129:0]111
Compressed	011 opar epar [addr10:0] 111
OR2C08A/OR2T08A	
Uncompressed	010 opar epar [addr10:0] [A]1[Data149:0]111
Compressed	011 opar epar [addr10:0] 111
OR2C10A/OR2T10A	
Uncompressed	010 opar epar [addr10:0] [A]1[Data169:0]111
Compressed	011 opar epar [addr10:0] 111
OR2C12A/OR2T12A	
Uncompressed	010 opar epar [addr10:0] [A]1[Data189:0]111
Compressed	011 opar epar [addr10:0] 111
OR2C15A/OR2T15A	
Uncompressed	010 opar epar [addr10:0] [A]1[Data209:0]111
Compressed	011 opar epar [addr10:0] 111
OR2C26A/OR2T26A	
Uncompressed	010 opar epar [addr10:0] [A]1[Data249:0]111
Compressed	011 opar epar [addr10:0] 111
OR2C40A/OR2T40A	
Uncompressed	010 opar epar [addr10:0] [A]1[Data315:0]111
Compressed	011 opar epar [addr10:0] 111

Configuration Data Format (continued)

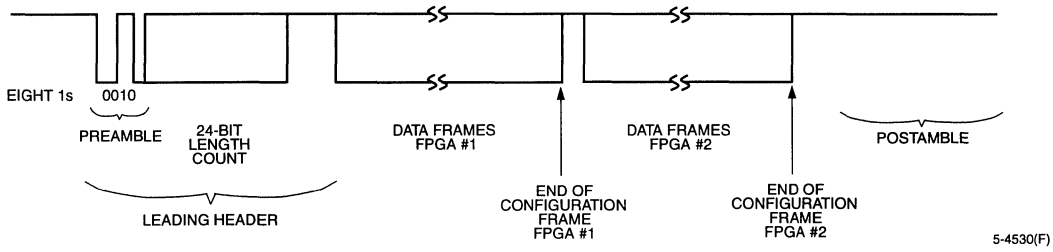


Figure 39. Serial Configuration Data Format

Header	11111111	Leading header — 4 bits minimum dummy bits
	0010	Preamble
	24-Bit Length Count	Configuration frame length
	1111	Trailing header — 4 bits minimum dummy bits
ID Frame (Optional)	0	Frame start
	P—1	Must be set to 1 to indicate data frame
	C—0	Must be set to 0 to indicate uncompressed
	Opar, Epar	Frame parity bits
	Addr[10:0] = 11111111111	ID frame address
	Prty_En	Set to 1 to enable parity
	Reserved [42:0]	Reserved bits set to 0
	ID	20-bit part ID
	111	Three or more stop bits (high) to separate frames
Configuration Data Frame (repeated for each data frame)	0	Frame start
	P—1 or 0	1 indicates data frame; 0 indicates all frames are written
	C—1 or 0	Uncompressed — 0 indicates data and address are supplied; Compressed — 1 indicates only address is supplied
	Opar, Epar	Frame parity bits
	Addr[10:0]	Column address in FPGA to be written
	A	Alignment bit (different number of 0s needed for each part)
	1	Write bit — used in uncompressed data frame
	Data Bits	Needed only in an uncompressed data frame
	.	.
	.	.
	111	One or more stop bits (high) to separate frames
End of Configuration	0010011111111111	16 bits—00 indicates all frames are written
Postamble	111111	Additional 1s

Note: For slave parallel mode, the byte containing the preamble must be 11110010. The number of leading header dummy bits must be (n * 8) + 4, where n is any nonnegative integer and the number of trailing dummy bits must be (n * 8), where n is any positive integer. The number of stop bits/frame for slave parallel mode must be (x * 8), where x is a positive integer. Note also that the bit stream generator tool supplies a bit stream which is compatible with all configuration modes, including slave parallel mode.

Figure 40. Configuration Frame Format and Contents

Bit Stream Error Checking

There are three different types of bit stream error checking performed in the *ORCA* OR2CxxA FPGAs: ID frame, frame alignment, and parity checking.

An optional ID data frame can be sent to a specified address in the FPGA. This ID frame contains a unique code for the part it was generated for which is compared within the FPGA. Any differences are flagged as an ID error. This frame is automatically created by the bit stream generation program in *ORCA* Foundry.

Every data frame in the FPGA begins with a start bit set to 0 and three or more stop bits set to 1. If any of the three previous bits were a 0 when a start bit is encountered, it is flagged as a frame alignment error.

Parity checking is also done on the FPGA for each frame, if it has been enabled by setting the prty_en bit to 1 in the ID frame. This is set by enabling the parity check option in the bit stream generation program of *ORCA* Foundry. Two parity bits, opar and epar, are used to check the parity of bits in alternating bit positions to even parity in each data frame. If an odd number of ones is found for either the even bits (starting with the start bit) or the odd bits (starting with the program bit), then a parity error is flagged.

When any of the three possible errors occur, the FPGA is forced into the INIT state, forcing $\overline{\text{INIT}}$ low. The FPGA will remain in this state until either the $\overline{\text{RESET}}$ or $\overline{\text{PRGM}}$ pins are asserted.

FPGA Configuration Modes

There are eight methods for configuring the FPGA. Seven of the configuration modes are selected on the M0, M1, and M2 inputs. The eighth configuration mode is accessed through the boundary-scan interface. A fourth input, M3, is used to select the frequency of the internal oscillator, which is the source for CCLK in some configuration modes. The nominal frequencies of the internal oscillator are 1.25 MHz and 10 MHz. The 1.25 MHz frequency is selected when the M3 input is unconnected or driven to a high state.

There are three basic FPGA configuration modes: master, slave, and peripheral. The configuration data can be transmitted to the FPGA serially or in parallel bytes. As a master, the FPGA provides the control signals out to strobe data in. As a slave device, a clock is generated externally and provided into CCLK. In the peripheral mode, the FPGA acts as a microprocessor peripheral. Table 9 lists the functions of the configuration mode pins.

Table 9. Configuration Modes

M2	M1	M0	CCLK	Configuration Mode	Data
0	0	0	Output	Master	Serial
0	0	1	Input	Slave Parallel	Parallel
0	1	0	Reserved		
0	1	1	Input	Sync Peripheral	Parallel
1	0	0	Output	Master (up)	Parallel
1	0	1	Output	Async Peripheral	Parallel
1	1	0	Output	Master (down)	Parallel
1	1	1	Input	Slave	Serial

Master Parallel Mode

The master parallel configuration mode is generally used to interface to industry-standard byte-wide memory, such as the 2764 and larger EPROMs. Figure 41 provides the connections for master parallel mode. The FPGA outputs an 18-bit address on A[17:0] to memory and reads one byte of configuration data on the rising edge of RCLK. The parallel bytes are internally serialized starting with the least significant bit, D0.

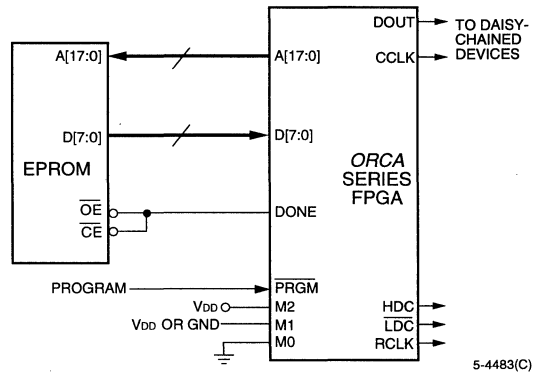


Figure 41. Master Parallel Configuration Schematic

There are two parallel master modes: master up and master down. In master up, the starting memory address is 00000 Hex and the FPGA increments the address for each byte loaded. In master down, the starting memory address is 3FFFF Hex and the FPGA decrements the address.

One master mode FPGA can interface to the memory and provide configuration data on DOUT to additional FPGAs in a daisy chain. The configuration data on DOUT is provided synchronously with the falling edge of CCLK. The frequency of the CCLK output is eight times that of RCLK.

FPGA Configuration Modes (continued)

Master Serial Mode

In the master serial mode, the FPGA loads the configuration data from an external serial ROM. The configuration data is either loaded automatically at start-up or on a $\overline{\text{PRGM}}$ command to reconfigure. The ATT1700 and ATT1700A Series can be used to configure the FPGA in the master serial mode. This provides a simple four-pin interface in an eight-pin package. The ATT1736, ATT1765, and ATT17128 serial ROMs store 32K, 64K, and 128K bits, respectively.

Configuration in the master serial mode can be done at powerup and/or upon a configure command. The system or the FPGA must activate the serial ROM's RESET/OE and $\overline{\text{CE}}$ inputs. At powerup, the FPGA and serial ROM each contain internal power-on reset circuitry that allows the FPGA to be configured without the system providing an external signal. The power-on reset circuitry causes the serial ROM's internal address pointer to be reset. After powerup, the FPGA automatically enters its initialization phase.

The serial ROM/FPGA interface used depends on such factors as the availability of a system reset pulse, availability of an intelligent host to generate a configure command, whether a single serial ROM is used or multiple serial ROMs are cascaded, whether the serial ROM contains a single or multiple configuration programs, etc. Because of differing system requirements and capabilities, a single FPGA/serial ROM interface is generally not appropriate for all applications.

Data is read in the FPGA sequentially from the serial ROM. The DATA output from the serial ROM is connected directly into the DIN input of the FPGA. The CCLK output from the FPGA is connected to the CLOCK input of the serial ROM. During the configuration process, CCLK clocks one data bit on each rising edge.

Since the data and clock are direct connects, the FPGA/serial ROM design task is to use the system or FPGA to enable the RESET/OE and $\overline{\text{CE}}$ of the serial ROM(s). There are several methods for enabling the serial ROM's RESET/OE and $\overline{\text{CE}}$ inputs. The serial ROM's RESET/OE is programmable to function with RESET active-high and $\overline{\text{OE}}$ active-low or $\overline{\text{RESET}}$ active-low and OE active-high.

In Figure 42, serial ROMs are cascaded to configure multiple daisy-chained FPGAs. The host generates a 500 ns low pulse into the FPGA's $\overline{\text{PRGM}}$ input. The FPGA's $\overline{\text{INIT}}$ input is connected to the serial ROMs' $\overline{\text{RESET}}$/OE input, which has been programmed to function with $\overline{\text{RESET}}$ active-low and OE active-high.

The FPGA DONE is routed to the $\overline{\text{CE}}$ pin. The low on DONE enables the serial ROMs. At the completion of configuration, the high on the FPGA's DONE disables the serial ROM.

Serial ROMs can also be cascaded to support the configuration of multiple FPGAs or to load a single FPGA when configuration data requirements exceed the capacity of a single serial ROM. After the last bit from the first serial ROM is read, the serial ROM outputs $\overline{\text{CEO}}$ low and 3-states the DATA output. The next serial ROM recognizes the low on $\overline{\text{CE}}$ input and outputs configuration data on the DATA output. After configuration is complete, the FPGA's DONE output into $\overline{\text{CE}}$ disables the serial ROMs.

This FPGA/serial ROM interface is not used in applications in which a serial ROM stores multiple configuration programs. In these applications, the next configuration program to be loaded is stored at the ROM location that follows the last address for the previous configuration program. The reason the interface in Figure 42 will not work in this application is that the low output on the $\overline{\text{INIT}}$ signal would reset the serial ROM address pointer, causing the first configuration to be reloaded.

In some applications, there can be contention on the FPGA's DIN pin. During configuration, DIN receives configuration data, and after configuration, it is a user I/O. If there is contention, an early DONE at start-up (selected in *ORCA* Foundry) may correct the problem. An alternative is to use $\overline{\text{LDC}}$ to drive the serial ROM's $\overline{\text{CE}}$ pin. In order to reduce noise, it is generally better to run the master serial configuration at 1.25 MHz (M3 pin tied high), rather than 10 MHz, if possible.

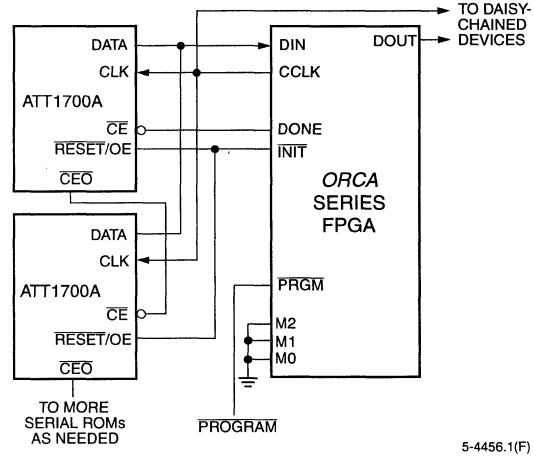


Figure 42. Master Serial Configuration Schematic

FPGA Configuration Modes (continued)

Asynchronous Peripheral Mode

Figure 43 shows the connections needed for the asynchronous peripheral mode. In this mode, the FPGA system interface is similar to that of a microprocessor-peripheral interface. The microprocessor generates the control signals to write an 8-bit byte into the FPGA. The FPGA control inputs include active-low $\overline{CS0}$ and active-high CS1 chip selects and a write $\overline{WR}$ input. The chip selects can be cycled or maintained at a static level during the configuration cycle. Each byte of data is written into the FPGA's D[7:0] input pins.

The FPGA provides a RDY/$\overline{BUSY}$ status output to indicate that another byte can be loaded. A low on RDY/$\overline{BUSY}$ indicates that the double-buffered hold/shift registers are not ready to receive data. The shortest time RDY/$\overline{BUSY}$ is low occurs when a byte is loaded into the hold register and the shift register is empty, in which case the byte is immediately transferred to the shift register. The longest time for RDY/$\overline{BUSY}$ to remain low occurs when a byte is loaded into the holding register and the shift register has just started shifting configuration data into configuration RAM. The RDY/$\overline{BUSY}$ status is also available on the D7 pin by enabling the chip selects, setting $\overline{WR}$ high, and setting $\overline{RD}$ low.

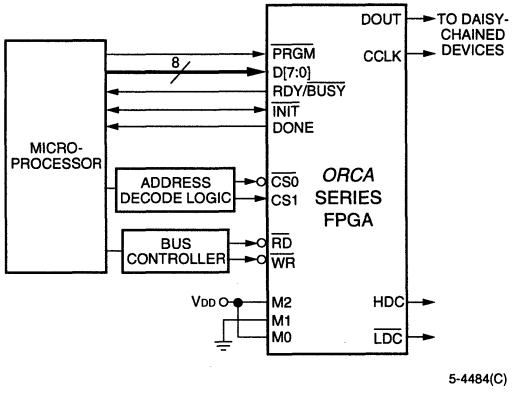


Figure 43. Asynchronous Peripheral Configuration Schematic

Synchronous Peripheral Mode

In the synchronous peripheral mode, byte-wide data is input into D[7:0] on the rising edge of the CCLK input. The first data byte is clocked in on the second CCLK after $\overline{INIT}$ goes high. Subsequent data bytes are clocked in on every eighth rising edge of CCLK. The RDY/$\overline{BUSY}$ signal is an output which acts as an acknowledge. RDY/$\overline{BUSY}$ goes high one CCLK after data is clocked and, after one CCLK cycle, returns low. The process repeats until all of the data is loaded into the FPGA. The data begins shifting on DOUT 1.5 cycles after it is loaded in parallel. It requires additional CCLKs after the last byte is loaded to complete the shifting. Figure 44 shows the connections for synchronous peripheral mode.

As with master modes, the peripheral modes can be used as the lead FPGA for a daisy chain of slave FPGAs.

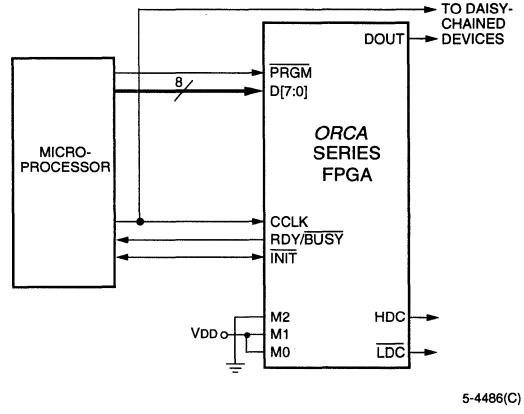


Figure 44. Synchronous Peripheral Configuration Schematic

FPGA Configuration Modes (continued)

Slave Serial Mode

The slave serial mode is primarily used when multiple FPGAs are configured in a daisy chain. The serial slave serial mode is also used on the FPGA evaluation board which interfaces to the download cable. A device in the slave serial mode can be used as the lead device in a daisy chain. Figure 45 shows the connections for the slave serial configuration mode.

The configuration data is provided into the FPGA's DIN input synchronous with the configuration clock CCLK input. After the FPGA has loaded its configuration data, it retransmits the incoming configuration data on DOUT. CCLK is routed into all slave serial mode devices in parallel.

Multiple slave FPGAs can be loaded with identical configurations simultaneously. This is done by loading the configuration data into the DIN inputs in parallel.

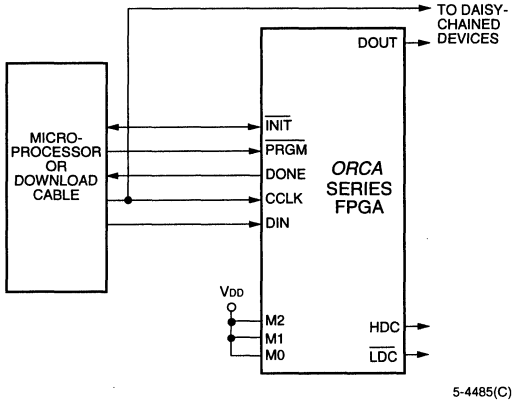


Figure 45. Slave Serial Configuration Schematic

Slave Parallel Mode

The slave parallel mode is essentially the same as the slave serial mode except that 8 bits of data are input on pins D[7:0] for each CCLK cycle. Due to 8 bits of data being input per CCLK cycle, the DOUT pin does not contain a valid bit stream for slave parallel mode. As a result, the lead device cannot be used in the slave parallel mode in a daisy-chain configuration.

Figure 46 is a schematic of the connections for the slave parallel configuration mode. $\overline{WR}$ and $\overline{CS0}$ are active-low chip select signals, and CS1 is an active-high chip select signal. These chip selects allow the user to configure multiple FPGAs in slave parallel mode using an 8-bit data bus common to all of the FPGAs. These chip selects can then be used to select the FPGA(s) to be configured with a given bit stream, but once an FPGA has been selected, it cannot be deselected until it has been completely programmed.

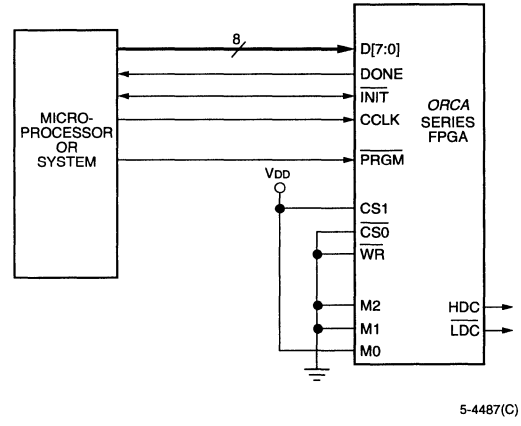


Figure 46. Slave Parallel Configuration Schematic

FPGA Configuration Modes (continued)

Daisy Chain

Multiple FPGAs can be configured by using a daisy chain of the FPGAs. Daisy chaining uses a lead FPGA and one or more FPGAs configured in slave serial mode. The lead FPGA can be configured in any mode except slave parallel mode. (Daisy chaining is not available with the boundary-scan ram_w instruction, discussed later.)

All daisy-chained FPGAs are connected in series. Each FPGA reads and shifts the preamble and length count in on positive CCLK and out on negative CCLK edges.

An upstream FPGA that has received the preamble and length count outputs a high on DOUT until it has received the appropriate number of data frames so that downstream FPGAs do not receive frame start bits (0s). After loading and retransmitting the preamble and length count to a daisy chain of slave devices, the lead device loads its configuration data frames. The loading of configuration data continues after the lead device has received its configuration data if its internal frame bit counter has not reached the length count. When the configuration RAM is full and the number of bits received is less than the length count field, the FPGA shifts any additional data out on DOUT.

The configuration data is read into DIN of slave devices on the positive edge of CCLK, and shifted out DOUT on the negative edge of CCLK. Figure 47 shows the connections for loading multiple FPGAs in a daisy-chain configuration.

The generation of CCLK for the daisy-chained devices which are in slave serial mode differs depending on the configuration mode of the lead device. A master parallel mode device uses its internal timing generator to produce an internal CCLK at eight times its memory address rate (RCLK). The asynchronous peripheral mode device outputs eight CCLKs for each write cycle. If the lead device is configured in either synchronous peripheral or a slave mode, CCLK is routed to the lead device and to all of the daisy-chained devices.

The development system can create a composite configuration bit stream for configuring daisy-chained FPGAs. The frame format is a preamble, a length count for the total bit stream, multiple concatenated data frames, an end-of-configuration frame per device, a postamble, and an additional fill bit per device in the serial chain.

As seen in Figure 47, the $\overline{\text{INIT}}$ pins for all of the FPGAs are connected together. This is required to guarantee that powerup and initialization will work correctly. In general, the DONE pins for all of the FPGAs are also connected together as shown to guarantee that all of the FPGAs enter the start-up state simultaneously. This may not be required, depending upon the start-up sequence desired.

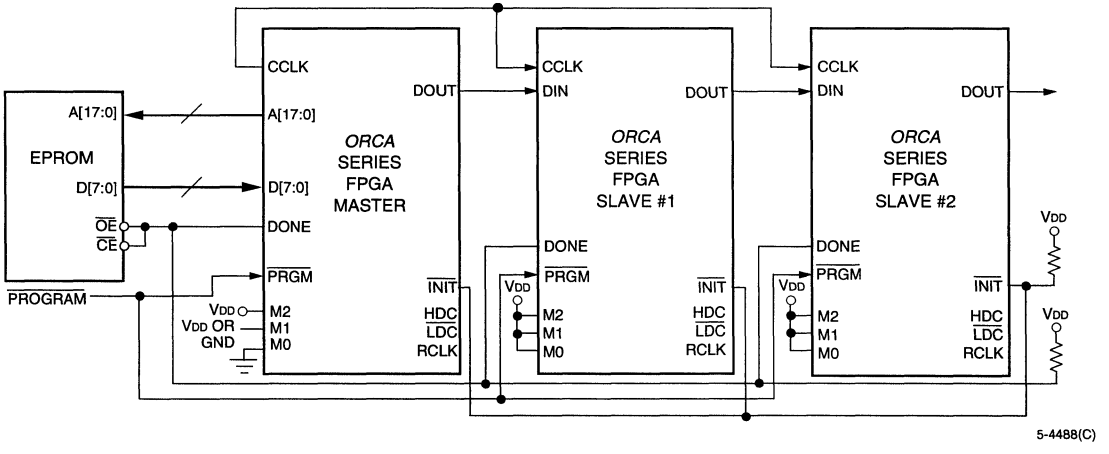


Figure 47. Daisy-Chain Configuration Schematic

Readback

Readback is used to read back the configuration data and, optionally, the state of the PFU outputs. A readback operation can be done while the FPGA is in normal system operation. The readback operation cannot be daisy chained. To use readback, the user selects options in the bit stream generator in the *ORCA* Foundry development system.

Table 10 provides readback options selected in the bit stream generator tool. The table provides the number of times that the configuration data can be read back. This is intended primarily to give the user control over the security of the FPGA's configuration program. The user can prohibit readback (0), allow a single readback (1), or allow unrestricted readback (U).

The pins used for readback are readback data (RD_DATA), read configuration ($\overline{\text{RD\_CFGN}}$), and configuration clock (CCLK). A readback operation is initiated by a high-to-low transition on $\overline{\text{RD\_CFGN}}$. The $\overline{\text{RD\_CFGN}}$ input must remain low during the readback operation. The readback operation can be restarted at frame 0 by setting the $\overline{\text{RD\_CFGN}}$ pin high, applying at least two rising edges of CCLK, and then applying $\overline{\text{RD\_CFGN}}$ low again. One bit of data is shifted out on RD_DATA on the rising edge of CCLK. The first start bit of the readback frame is transmitted out on the first rising edge of CCLK after $\overline{\text{RD\_CFGN}}$ is input low.

It should be noted that the RD_DATA output pin is also used as the dedicated boundary-scan output pin, TDO. If this pin is being used as TDO, the RD_DATA output from readback can be routed internally to any other pin desired. The $\overline{\text{RD\_CFGN}}$ input pin is also used to control the global 3-state (ts_all) function. Before and during configuration, the ts_all signal is always driven by the RD_CFGN input and readback is disabled. After configuration, the selection as to whether this input drives the readback or global 3-state function is determined by a set of bit stream options. If used as the $\overline{\text{RD\_CFGN}}$ input for readback, the internal ts_all input can be routed internally to be driven by any input pin.

The readback frame contains the configuration data and the state of the internal logic. During readback, the value of all five PFU outputs can be captured. The following options are allowed when doing a capture of the PFU outputs.

1. Do not capture data (the data written to the RAMs, usually 0, will be read back).
2. Capture data upon entering readback.
3. Capture data based upon a configurable signal internal to the FPGA. If this signal is tied to logic 0, capture RAMs are written continuously, which is equivalent to ATT3000 Series capture.
4. Capture data on either options 2 or 3 above.

The readback frame has a similar, but not identical, format to the configuration frame. This eases a bitwise comparison between the configuration and readback data. The readback data is not inverted. Every data frame has one low start bit and one high stop bit. The preamble, including the length count field, is not part of the readback frame. The readback frame contains states in locations not used in the configuration. These locations need to be masked out when comparing the configuration and readback frames. The development system optionally provides a readback bit steam to compare to readback data from the FPGA.

Table 10. Readback Options

Option	Function
0	Inhibit Readback
1	Allow One Readback Only
U	Allow Unrestricted Number of Readbacks

Boundary Scan

The increasing complexity of integrated circuits (ICs) and IC packages has increased the difficulty of testing printed-circuit boards (PCBs). To address this testing problem, the *IEEE* standard 1149.1 - 1990 (*IEEE* Standard Test Access Port and Boundary-Scan Architecture) is implemented in the *ORCA* series of FPGAs. It allows users to efficiently test the interconnection between integrated circuits on a PCB as well as test the integrated circuit itself. The *IEEE* 1149.1 standard is a well-defined protocol that ensures interoperability among boundary-scan (BSCAN) equipped devices from different vendors.

The *IEEE* 1149.1 standard defines a test access port (TAP) that consists of a four-pin interface with an optional reset pin for boundary-scan testing of integrated circuits in a system. The *ORCA* Series FPGA provides four interface pins: test data in (TDI), test mode select (TMS), test clock (TCK), and test data out (TDO). The $\overline{\text{PRGM}}$ pin used to reconfigure the device also resets the boundary-scan logic.

The user test host serially loads test commands and test data into the FPGA through these pins to drive outputs and examine inputs. In the configuration shown in Figure 48, where boundary scan is used to test ICs, test data is transmitted serially into TDI of the first BSCAN device (U1), through TDO/TDI connections between BSCAN devices (U2 and U3), and out TDO of the last BSCAN device (U4). In this configuration, the TMS and TCK signals are routed to all boundary-scan ICs in parallel so that all boundary-scan components operate in the same state. In other configurations, multiple scan paths are used instead of a single ring. When multiple scan paths are used, each ring is independently controlled by its own TMS and TCK signals.

Figure 49 provides a system interface for components used in the boundary-scan testing of PCBs. The three major components shown are the test host, boundary-scan support circuit, and the devices under test (DUTs). The DUTs shown here are *ORCA* Series FPGAs with dedicated boundary-scan circuitry. The test host is normally one of the following: automatic test equipment (ATE), a workstation, a PC, or a microprocessor.

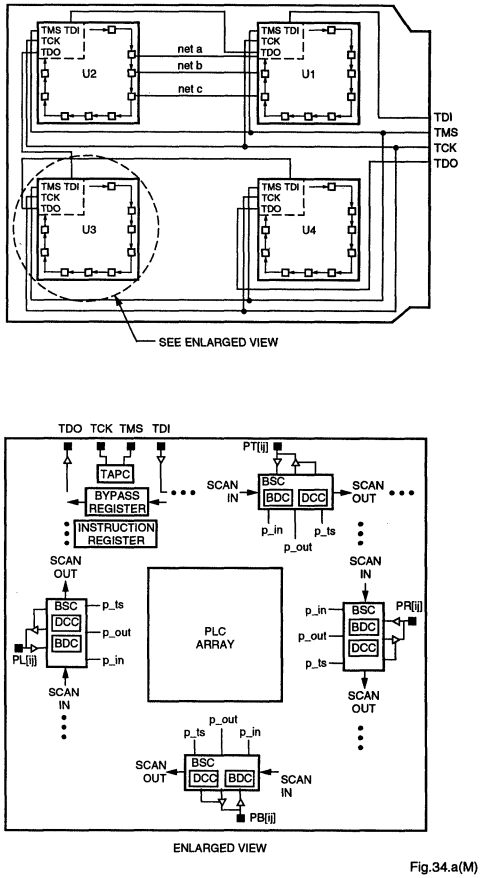


Fig.34.a(M)

Key: BSC = boundary-scan cell, BDC = bidirectional data cell, and DCC = data control cell.

Figure 48. Printed-Circuit Board with Boundary-Scan Circuitry

Boundary Scan (continued)

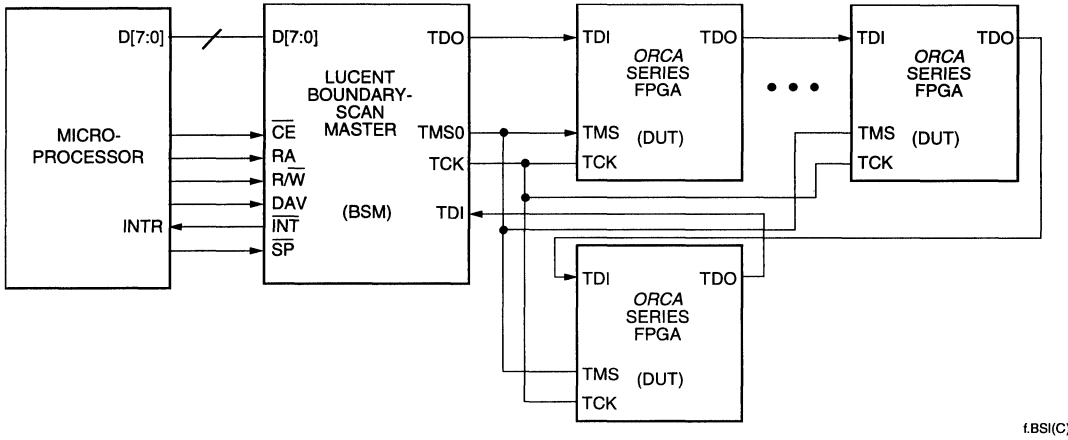


Figure 49. Boundary-Scan Interface

The boundary-scan support circuit shown in Figure 49 is the 497AA Boundary-Scan Master (BSM). The BSM off-loads tasks from the test host to increase test throughput. To interface between the test host and the DUTs, the BSM has a general microprocessor interface and provides parallel-to-serial/serial-to-parallel conversion, as well as three 8K data buffers. The BSM also increases test throughput with a dedicated automatic test pattern generator and with compression of the test response with a signature analysis register. The PC-based boundary-scan test card/software allows a user to quickly prototype a boundary-scan test setup.

Boundary-Scan Instructions

The *ORCA* Series boundary-scan circuitry is used for three mandatory *IEEE* 1149.1 tests (EXTEST, SAMPLE/PRELOAD, BYPASS) and four *ORCA*-defined instructions. The 3-bit wide instruction register supports the eight instructions listed in Table 11.

Table 11. Boundary-Scan Instructions

Code	Instruction
000	EXTEST
001	PLC Scan Ring 1
010	RAM Write (RAM_W)
011	Reserved
100	SAMPLE/PRELOAD
101	PLC Scan Ring 2
110	RAM Read (RAM_R)
111	BYPASS

The external test (EXTEST) instruction allows the interconnections between ICs in a system to be tested for opens and stuck-at faults. If an EXTEST instruction is performed for the system shown in Figure 48, the connections between U1 and U2 (shown by nets a, b, and c) can be tested by driving a value onto the given nets from one device and then determining whether the same value is seen at the other device. This is determined by shifting 2 bits of data for each pin (one for the output value and one for the 3-state value) through the BSR until each one aligns to the appropriate pin. Then, based upon the value of the 3-state signal, either the I/O pad is driven to the value given in the BSR, or the BSR is updated with the input value from the I/O pad, which allows it to be shifted out TDO.

The SAMPLE instruction is useful for system debugging and fault diagnosis by allowing the data at the FPGA's I/Os to be observed during normal operation. The data for all of the I/Os is captured simultaneously into the BSR, allowing them to be shifted-out TDO to the test host. Since each I/O buffer in the PICs is bidirectional, two pieces of data are captured for each I/O pad: the value at the I/O pad and the value of the 3-state control signal.

There are four *ORCA*-defined instructions. The PLC scan rings 1 and 2 (PSR1, PSR2) allow user-defined internal scan paths using the PLC latches/FFs. The RAM_Write Enable (RAM_W) instruction allows the user to serially configure the FPGA through TDI. The RAM_Read Enable (RAM_R) allows the user to read back RAM contents on TDO after configuration.

Boundary Scan (continued)

ORCA Boundary-Scan Circuitry

The *ORCA* Series boundary-scan circuitry includes a test access port controller (TAPC), instruction register (IR), boundary-scan register (BSR), and bypass register. It also includes circuitry to support the four predefined instructions.

Figure 50 shows a functional diagram of the boundary-scan circuitry that is implemented in the *ORCA* series. The input pins' (TMS, TCK, and TDI) locations vary depending on the part, and the output pin is the dedicated TDO/RD_DATA output pad. Test data in (TDI) is the serial input data. Test mode select (TMS) controls the boundary-scan test access port controller (TAPC). Test clock (TCK) is the test clock on the board.

The BSR is a series connection of boundary-scan cells (BSCs) around the periphery of the IC. Each I/O pad on the FPGA, except for CCLK, DONE, and the boundary-scan pins (TCK, TDI, TMS, and TDO), is included in the BSR. The first BSC in the BSR (connected to TDI) is located in the first PIC I/O pad on the left of the top side of the FPGA (PTA PIC). The BSR proceeds clockwise around the top, right, bottom, and left sides of the array. The last BSC in the BSR (connected to TDO) is located on the top of the left side of the array (PLA3).

The bypass instruction uses a single FF which resynchronizes test data that is not part of the current scan operation. In a bypass instruction, test data received on TDI is shifted out of the bypass register to TDO. Since the BSR (which requires a two FF delay for each pad) is bypassed, test throughput is increased when devices that are not part of a test operation are bypassed.

The boundary-scan logic is enabled before and during configuration. After configuration, a configuration option determines whether or not boundary-scan logic is used.

The 32-bit boundary-scan identification register contains the manufacturer's ID number, unique part number, and version, but is not implemented in the *ORCA* series of FPGAs. If boundary scan is not used, TMS, TDI, and TCK become user I/Os, and TDO is 3-stated or used in the readback operation.

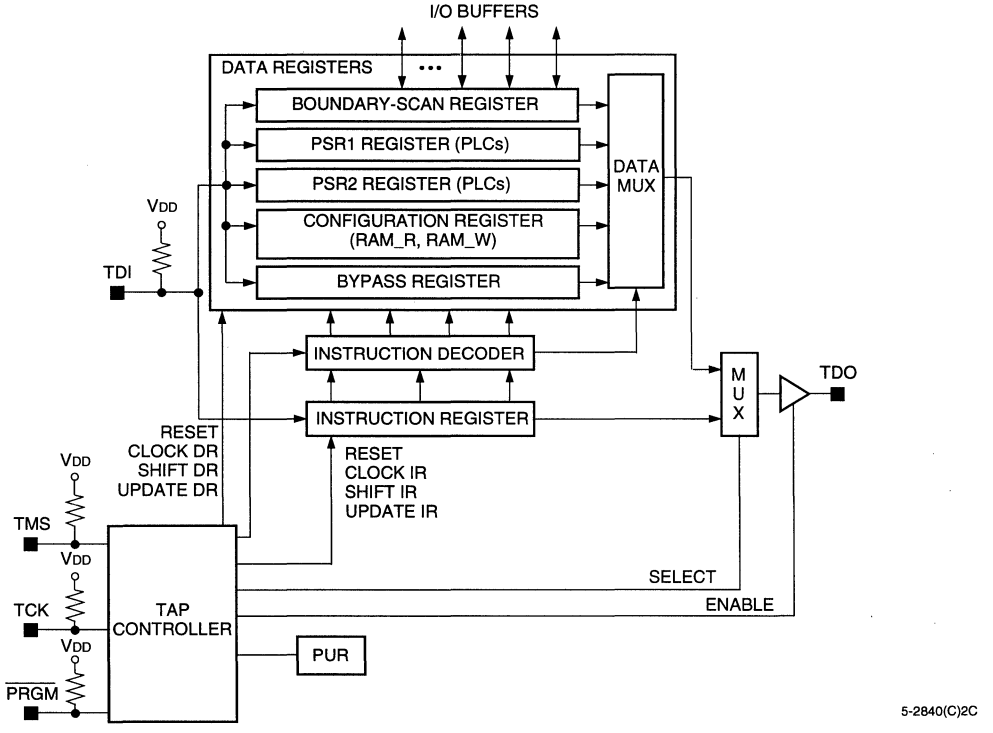


Figure 50. *ORCA* Series Boundary-Scan Circuitry Functional Diagram

Boundary Scan (continued)

ORCA Series TAP Controller (TAPC)

The *ORCA* Series TAP controller (TAPC) is a 1149.1 compatible test access port controller. The 16 JTAG state assignments from the *IEEE* 1149.1 specification are used. The TAPC is controlled by TCK and TMS. The TAPC states are used for loading the IR to allow three basic functions in testing: providing test stimuli (Update DR), test execution (Run-Test/Idle), and obtaining test responses (Capture DR). The TAPC allows the test host to shift in and out both instructions and test data/results. The inputs and outputs of the TAPC are provided in the table below. The outputs are primarily the control signals to the instruction register and the data register.

Table 12. TAP Controller Input/Outputs

Symbol	I/O	Function
TMS	I	Test Mode Select
TCK	I	Test Clock
PUR	I	Powerup Reset
$\overline{\text{PRGM}}$	I	BSCAN Reset
TRESET	O	Test Logic Reset
Select	O	Select IR (High); Select DR (Low)
Enable	O	Test Data Out Enable
Capture DR	O	Capture/Parallel Load DR
Capture IR	O	Capture/Parallel Load IR
Shift DR	O	Shift Data Register
Shift IR	O	Shift Instruction Register
Update DR	O	Update/Parallel Load DR
Update IR	O	Update/Parallel Load IR

The TAPC generates control signals which allow capture, shift, and update operations on the instruction and data registers. In the capture operation, data is loaded into the register. In the shift operation, the captured data is shifted out while new data is shifted in. In the update operation, either the instruction register is loaded for instruction decode, or the boundary-scan register is updated for control of outputs.

The test host generates a test by providing input into the *ORCA* Series TMS input synchronous with TCK. This sequences the TAPC through states in order to perform the desired function on the instruction register or a data register. Figure 51 provides a diagram of the state transitions for the TAPC. The next state is determined by the TMS input value.

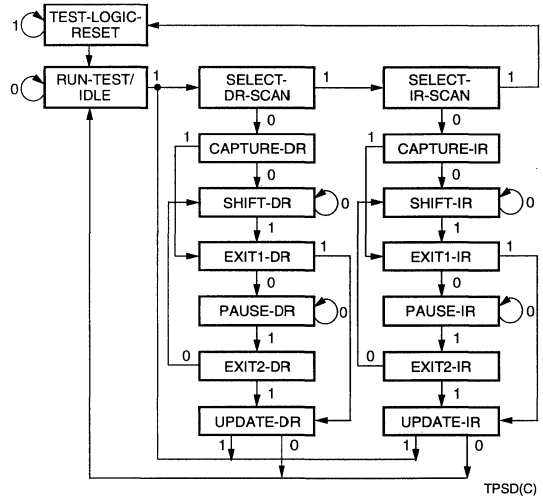


Figure 51. TAP Controller State Transition Diagram

Boundary Scan (continued)

Boundary-Scan Cells

Figure 52 is a diagram of the boundary-scan cell (BSC) in the *ORCA* series PICs. There are four BSCs in each PIC: one for each pad, except as noted above. The BSCs are connected serially to form the BSR. The BSC controls the functionality of the in, out, and 3-state signals for each pad.

The BSC allows the I/O to function in either the normal or test mode. Normal mode is defined as when an output buffer receives input from the PLC array and provides output at the pad or when an input buffer provides input from the pad to the PLC array. In the test mode, the BSC executes a boundary-scan operation, such as shifting in scan data from an upstream BSC in the BSR, providing test stimuli to the pad, capturing test data at the pad, etc.

The primary functions of the BSC are shifting scan data serially in the BSR and observing input (p_in), output (p_out), and 3-state (p_ts) signals at the pads. The BSC consists of two circuits: the bidirectional data cell is used to access the input and output data, and the direction control cell is used to access the 3-state value. Both cells consist of a flip-flop used to shift scan data which feeds a flip-flop to control the I/O buffer. The bidirectional data cell is connected serially to the direction control cell to form a boundary-scan shift register.

The TAPC signals (capture, update, shiftn, treset, and TCK) and the MODE signal control the operation of the BSC. The bidirectional data cell is also controlled by the high out/low in (HOLI) signal generated by the direction control cell. When HOLI is low, the bidirectional data cell receives input buffer data into the BSC. When HOLI is high, the BSC is loaded with functional data from the PLC.

The MODE signal is generated from the decode of the instruction register. When the MODE signal is high (EXTEST), the scan data is propagated to the output buffer. When the MODE signal is low (BYPASS or SAMPLE), functional data from the FPGA's internal logic is propagated to the output buffer.

The boundary-scan description language (BSDL) is provided for each device in the *ORCA* series of FPGAs. The BSDL is generated from a device profile, pinout, and other boundary-scan information.

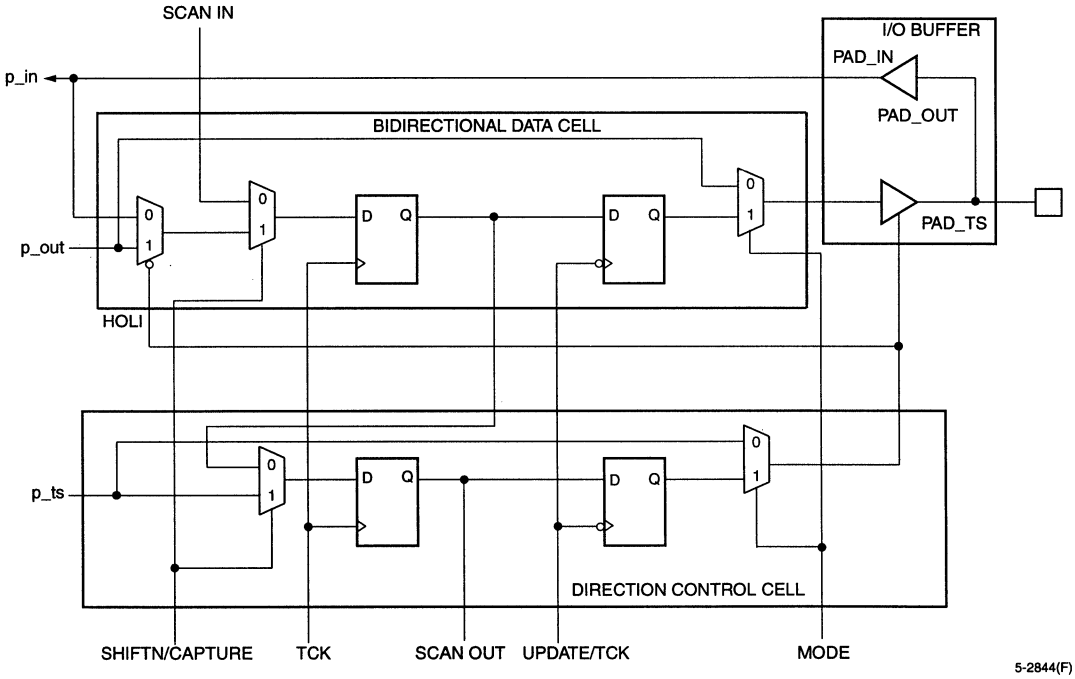


Figure 52. Boundary-Scan Cell

Boundary Scan (continued)

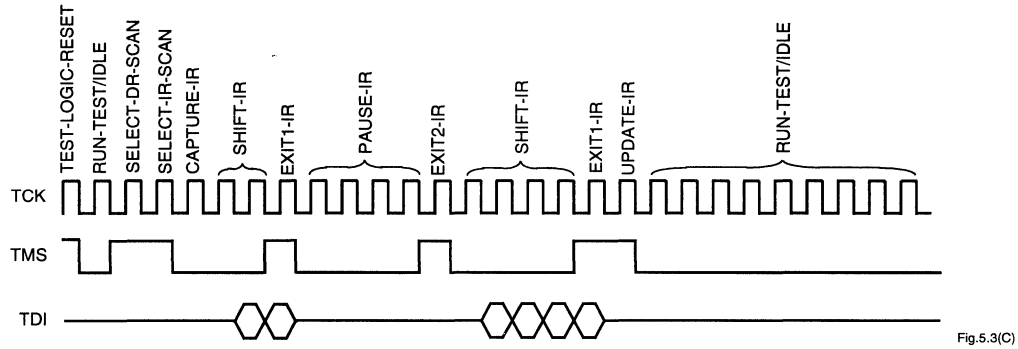


Figure 53. Instruction Register Scan Timing Diagram

Boundary-Scan Timing

To ensure race-free operation, data changes on specific clock edges. The TMS and TDI inputs are clocked in on the rising edge of TCK, while changes on TDO occur on the falling edge of TCK. In the execution of an EXTEST instruction, parallel data is output from the BSR to the FPGA pads on the falling edge of TCK. The maximum frequency allowed for TCK is 10 MHz.

Figure 53 shows timing waveforms for an instruction scan operation. The diagram shows the use of TMS to sequence the TAPC through states. The test host (or BSM) changes data on the falling edge of TCK, and it is clocked into the DUT on the rising edge.

ORCA Timing Characteristics

To define speed grades, the *ORCA* Series part number designation (see Table 52) uses a single-digit number to designate a speed grade. This number is not related to any single ac parameter. Higher numbers indicate a faster set of timing parameters. The actual speed sorting is based on testing the delay in a path consisting of an input buffer, combinatorial delay through all PLCs in a row, and an output buffer. Other tests are then done to verify other delay parameters, such as routing delays, setup times to FFs, etc.

The most accurate timing characteristics are reported by the timing analyzer in the *ORCA* Foundry Development System. A timing report provided by the development system after layout divides path delays into logic and routing delays. The timing analyzer can also provide logic delays prior to layout. While this allows routing budget estimates, there is wide variance in routing delays associated with different layouts.

The logic timing parameters noted in the Electrical Characteristics section of this data sheet are the same as those in the design tools. In the PFU timing given in Tables 31—53, symbol names are generally a concatenation of the PFU operating mode (as defined in Table 3) and the parameter type. The wildcard character (*) is used in symbol names to indicate that the parameter applies to any sub-LUT. The setup, hold, and propagation delay parameters, defined below, are designated in the symbol name by the SET, HLD, and DEL characters, respectively.

The values given for the parameters are the same as those used during production testing and speed binning of the devices. The junction temperature and supply voltage used to characterize the devices are listed in the delay tables. Actual delays at nominal temperature and voltage for best-case processes can be much better than the values given.

It should be noted that the junction temperature used in the tables is generally 85 °C. The junction temperature for the FPGA depends on the power dissipated by the device, the package thermal characteristics (Θ_{JA}), and the ambient temperature, as calculated in the following equation and as discussed further in the Package Thermal Characteristics section:

$$T_{Jmax} = T_{Amax} + (P \bullet \Theta_{JA})\ °C$$

Note: The user must determine this junction temperature to see if the delays from *ORCA* Foundry should be derated based on the following derating tables.

Tables 13A and 13B and provide approximate power supply and junction temperature derating for OR2CxxA commercial and industrial devices. Table 14 provides the same information for the OR2TxxA devices (both commercial and industrial). The delay values in this data sheet and reported by *ORCA* Foundry are shown as **1.00** in the tables. The method for determining the maximum junction temperature is defined in the Thermal Characteristics section. Taken cumulatively, the range of parameter values for best-case vs. worst-case processing, supply voltage, and junction temperature can approach 3 to 1.

Table 13A. Derating for Commercial Devices (OR2CxxA)

T_J (°C)	Power Supply Voltage		
	4.75 V	5.0 V	5.25 V
0	0.81	0.79	0.77
25	0.85	0.83	0.81
85	**1.00**	0.97	0.95
100	1.05	1.02	1.00
125	1.12	1.09	1.07

Table 13B. Derating for Industrial Devices (OR2CxxA)

T_J (°C)	Power Supply Voltage				
	4.5 V	4.75 V	5.0 V	5.25 V	5.5 V
–40	0.71	0.70	0.68	0.66	0.65
0	0.80	0.78	0.76	0.74	0.73
25	0.84	0.82	0.80	0.78	0.77
85	**1.00**	0.97	0.94	0.93	0.91
100	1.05	1.01	0.99	0.97	0.95
125	1.12	1.09	1.06	1.04	1.02

Table 14. Derating for Commercial/Industrial Devices (OR2TxxA)

T_J (°C)	Power Supply Voltage		
	3.0 V	3.3 V	3.6 V
–40	0.73	0.66	0.61
0	0.82	0.73	0.68
25	0.87	0.78	0.72
85	**1.00**	0.90	0.83
100	1.04	0.94	0.87
125	1.10	1.00	0.92

Note: The derating tables shown above are for a typical critical path that contains 33% logic delay and 66% routing delay. Since the routing delay derates at a higher rate than the logic delay, paths with more than 66% routing delay will derate at a higher rate than shown in the table. The approximate derating values vs. temperature are 0.26% per °C for logic delay and 0.45% per °C for routing delay. The approximate derating values vs. voltage are 0.13% per mV for both logic and routing delays at 25 °C.

ORCA Timing Characteristics
(continued)

In addition to supply voltage, process variation, and operating temperature, circuit and process improvements of the *ORCA* series FPGAs over time will result in significant improvement of the actual performance over those listed for a speed grade. Even though lower speed grades may still be available, the distribution of yield to timing parameters may be several speed bins higher than that designated on a product brand. Design practices need to consider best-case timing parameters (e.g., delays = 0), as well as worst-case timing.

The routing delays are a function of fan-out and the capacitance associated with the CIPs and metal interconnect in the path. The number of logic elements that can be driven (or fan-out) by PFUs is unlimited, although the delay to reach a valid logic level can exceed timing requirements. It is difficult to make accurate routing delay estimates prior to design compilation based on fan-out. This is because the CAE software may delete redundant logic inserted by the designer to reduce fan-out, and/or it may also automatically reduce fan-out by net splitting.

The waveform test points are given in the Measurement Conditions section of this data sheet. The timing parameters given in the electrical characteristics tables in this data sheet follow industry practices, and the values they reflect are described below.

Propagation Delay — the time between the specified reference points. The delays provided are the worst case of the tphh and tpll delays for noninverting functions, tplh and tphl for inverting functions, and tphz and tplz for 3-state enable.

Setup Time — the interval immediately preceding the transition of a clock or latch enable signal, during which the data must be stable to ensure it is recognized as the intended value.

Hold Time — the interval immediately following the transition of a clock or latch enable signal, during which the data must be held stable to ensure it is recognized as the intended value.

3-state Enable — the time from when a ts[3:0] signal becomes active and the output pad reaches the high-impedance state.

Estimating Power Dissipation

OR2CxxA

The total operating power dissipated is estimated by summing the standby (I_{DDSB}), internal, and external power dissipated. The internal and external power is the power consumed in the PLCs and PICs, respectively. In general, the standby power is small and may be neglected. The total operating power is as follows:

$$P_T = \Sigma P_{PLC} + \Sigma P_{PIC}$$

The internal operating power is made up of two parts: clock generation and PFU output power. The PFU output power can be estimated based upon the number of PFU outputs switching when driving an average fan-out of two:

$$P_{PFU} = 0.16 \text{ mW/MHz}$$

For each PFU output that switches, 0.16 mW/MHz needs to be multiplied times the frequency (in MHz) that the output switches. Generally, this can be estimated by using one-half the clock rate, multiplied by some activity factor; for example, 20%.

The power dissipated by the clock generation circuitry is based upon four parts: the fixed clock power, the power/clock branch row or column, the clock power dissipated in each PFU that uses this particular clock, and the power from the subset of those PFUs that is configured in either of the two synchronous modes (SSPM or SDPM). Therefore, the clock power can be calculated for the four parts using the following equations:

OR2C04A Clock Power

P = [0.62 mW/MHz
+ (0.22 mW/MHz – Branch) (# Branches)
+ (0.022 mW/MHz – PFU) (# PFUs)
+ (0.006 mW/MHz – SMEM_PFU)
(# SMEM_PFUs)] fCLK

For a quick estimate, the worst-case (typical circuit) OR2C04A Clock Power ≈ 3.9 mW/MHz.

OR2C06A Clock Power

P = [0.63 mW/MHz
+ (0.25 mW/MHz – Branch) (# Branches)
+ (0.022 mW/MHz – PFU) (# PFUs)
+ (0.006 mW/MHz – SMEM_PFU)
(# SMEM_PFUs)] fCLK

For a quick estimate, the worst-case (typical circuit) OR2C06A Clock Power ≈ 5.3 mW/MHz.

OR2C08A Clock Power

P = [0.65 mW/MHz
+ (0.29 mW/MHz – Branch) (# Branches)
+ (0.022 mW/MHz – PFU) (# PFUs)
+ (0.006 mW/MHz – SMEM_PFU)
(# SMEM_PFUs)] fCLK

For a quick estimate, the worst-case (typical circuit) OR2C08A Clock Power ≈ 6.6 mW/MHz.

OR2C10A Clock Power

P = [0.66 mW/MHz
+ (0.32 mW/MHz – Branch) (# Branches)
+ (0.022 mW/MHz – PFU) (# PFUs)
+ (0.006 mW/MHz – SMEM_PFU)
(# SMEM_PFUs)] fCLK

For a quick estimate, the worst-case (typical circuit) OR2C10A Clock Power ≈ 8.6 mW/MHz.

OR2C12A Clock Power

P = [0.68 mW/MHz
+ (0.35 mW/MHz – Branch) (# Branches)
+ (0.022 mW/MHz – PFU) (# PFUs)
+ (0.006 mW/MHz – SMEM_PFU)
(# SMEM_PFUs)] fCLK

For a quick estimate, the worst-case (typical circuit) OR2C12A Clock Power ≈ 10.5 mW/MHz.

OR2C15A Clock Power

P = [0.69 mW/MHz
+ (0.38 mW/MHz – Branch) (# Branches)
+ (0.022 mW/MHz – PFU) (# PFUs)
+ (0.006 mW/MHz – SMEM_PFU)
(# SMEM_PFUs)] fCLK

For a quick estimate, the worst-case (typical circuit) OR2C15A Clock Power ≈ 12.7 mW/MHz.

OR2C26A Clock Power

P = [0.73 mW/MHz
+ (0.44 mW/MHz – Branch) (# Branches)
+ (0.022 mW/MHz – PFU) (# PFUs)
+ (0.006 mW/MHz – SMEM_PFU)
(# SMEM_PFUs)] fCLK

For a quick estimate, the worst-case (typical circuit) OR2C26A Clock Power ≈ 17.8 mW/MHz.

Estimating Power Dissipation (continued)

OR2C40A Clock Power

P = [0.77 mW/MHz
+ (0.53 mW/MHz – Branch) (# Branches)
+ (0.022 mW/MHz – PFU) (# PFUs)
+ (0.006 mW/MHz – SMEM_PFU)
(# SMEM_PFUs)] fCLK

For a quick estimate, the worst-case (typical circuit) OR2C40A Clock Power ≈ 26.6 mW/MHz.

The power dissipated in a PIC is the sum of the power dissipated in the four I/Os in the PIC. This consists of power dissipated by inputs and ac power dissipated by outputs. The power dissipated in each I/O depends on whether it is configured as an input, output, or input/output. If an I/O is operating as an output, then there is a power dissipation component for P_{IN}, as well as P_{OUT}. This is because the output feeds back to the input.

The power dissipated by a TTL input buffer is estimated as:

$$P_{TTL} = 2.2 \text{ mW} + 0.17 \text{ mW/MHz}$$

The power dissipated by an input buffer is estimated as:

$$P_{CMOS} = 0.17 \text{ mW/MHz}$$

The ac power dissipation from an output or bidirectional is estimated by the following:

$$P_{OUT} = (C_L + 8.8 \text{ pF}) \times V_{DD}^2 \times F \text{ Watts}$$

where the unit for C_L is farads, and the unit for F is Hz.

As an example of estimating power dissipation, suppose that a fully utilized OR2C15A has an average of three outputs for each of the 400 PFUs, that all 20 clock branches are used, that 150 of the 400 PFUs have FFs clocked at 40 MHz (16 of which are operating in a synchronous memory mode), and that the PFU outputs have an average activity factor of 20%.

Twenty TTL-configured inputs, 20 CMOS-configured inputs, 32 outputs driving 30 pF loads, and 16 bidirectional I/Os driving 50 pF loads are also generated from the 40 MHz clock with an average activity factor of 20%. The worst-case (V_{DD} = 5.25 V) power dissipation is estimated as follows:

P_{PFU} = 400 x 3 (0.16 mW/MHz x 20 MHz x 20%)
= 768 mW

P_{CLK} = [0.69 mW/MHz + (0.38 mW/MHz – Branch)
(20 Branches)
+ (0.022 mW/MHz – PFU) (150 PFUs)
+ (0.006 mW/MHz – SMEM_PFU)
(16 SMEM_PFUs)] [40 MHz]
= 427 mW

P_{TTL} = 20 x [2.2 mW + (0.17 mW/MHz x 20 MHz
x 20%)]
= 57 mW

P_{CMOS} = 20 x [0.17 mW x 20 MHz x 20%]
= 13 mW

P_{OUT} = 30 x [(30 pF + 8.8 pF) x $(5.25)^2$ x 20 MHz
x 20%]
= 128 mW

P_{BID} = 16 x [(50 pF + 8.8 pF) x $(5.25)^2$ x 20 MHz
x 20%]
= 104 mW

TOTAL = 1.50 W

Estimating Power Dissipation (continued)

OR2TxxA

The total operating power dissipated is estimated by summing the standby (I_{DDSB}), internal, and external power dissipated. The internal and external power is the power consumed in the PLCs and PICs, respectively. In general, the standby power is small and may be neglected. The total operating power is as follows:

$$P_T = \Sigma P_{PLC} + \Sigma P_{PIC}$$

The internal operating power is made up of two parts: clock generation and PFU output power. The PFU output power can be estimated based upon the number of PFU outputs switching when driving an average fan-out of two:

$$P_{PFU} = 0.08 \text{ mW/MHz}$$

For each PFU output that switches, 0.08 mW/MHz needs to be multiplied times the frequency (in MHz) that the output switches. Generally, this can be estimated by using one-half the clock rate, multiplied by some activity factor; for example, 20%.

The power dissipated by the clock generation circuitry is based upon four parts: the fixed clock power, the power/clock branch row or column, the clock power dissipated in each PFU that uses this particular clock, and the power from the subset of those PFUs that is configured in either of the two synchronous modes (SSPM or SDPM). Therefore, the clock power can be calculated for the four parts using the following equations:

OR2T04A Clock Power

P = [0.29 mW/MHz
+ (0.10 mW/MHz – Branch) (# Branches)
+ (0.01 mW/MHz – PFU) (# PFUs)
+ (0.003 mW/MHz – SMEM_PFU)
(# SMEM_PFUs)] fCLK

For a quick estimate, the worst-case (typical circuit) OR2T04A Clock Power ≈ 1.8 mW/MHz.

OR2T06A Clock Power

P = [0.30 mW/MHz
+ (0.11 mW/MHz – Branch) (# Branches)
+ (0.01 mW/MHz – PFU) (# PFUs)
+ (0.003 mW/MHz – SMEM_PFU)
(# SMEM_PFUs)] fCLK

For a quick estimate, the worst-case (typical circuit) OR2T06A Clock Power ≈ 2.4 mW/MHz.

OR2T08A Clock Power

P = [0.31 mW/MHz
+ (0.12 mW/MHz – Branch) (# Branches)
+ (0.01 mW/MHz – PFU) (# PFUs)
+ (0.003 mW/MHz – SMEM_PFU)
(# SMEM_PFUs)] fCLK

For a quick estimate, the worst-case (typical circuit) OR2T08A Clock Power ≈ 3.2 mW/MHz.

OR2T10A Clock Power

P = [0.32 mW/MHz
+ (0.14 mW/MHz – Branch) (# Branches)
+ (0.01 mW/MHz – PFU) (# PFUs)
+ (0.003 mW/MHz – SMEM_PFU)
(# SMEM_PFUs)] fCLK

For a quick estimate, the worst-case (typical circuit) OR2T10A Clock Power ≈ 4.0 mW/MHz.

OR2T12A Clock Power

P = [0.33 mW/MHz
+ (0.15 mW/MHz – Branch) (# Branches)
+ (0.01 mW/MHz – PFU) (# PFUs)
+ (0.003 mW/MHz – SMEM_PFU)
(# SMEM_PFUs)] fCLK

For a quick estimate, the worst-case (typical circuit) OR2T12A Clock Power ≈ 4.9 mW/MHz.

OR2T15A Clock Power

P = [0.34 mW/MHz
+ (0.17 mW/MHz – Branch) (# Branches)
+ (0.01 mW/MHz – PFU) (# PFUs)
+ (0.003 mW/MHz – SMEM_PFU)
(# SMEM_PFUs)] fCLK

For a quick estimate, the worst-case (typical circuit) OR2T15A Clock Power ≈ 5.9 mW/MHz.

OR2T26A Clock Power

P = [0.35 mW/MHz
+ (0.19 mW/MHz – Branch) (# Branches)
+ (0.01 mW/MHz – PFU) (# PFUs)
+ (0.003 mW/MHz – SMEM_PFU)
(# SMEM_PFUs)] fCLK

For a quick estimate, the worst-case (typical circuit) OR2T26A Clock Power ≈ 8.3 mW/MHz.

Estimating Power Dissipation (continued)

OR2T40A Clock Power

P = [0.37 mW/MHz
+ (0.23 mW/MHz – Branch) (# Branches)
+ (0.01 mW/MHz – PFU) (# PFUs)
+ (0.003 mW/MHz – SMEM_PFU)
(# SMEM_PFUs)] fCLK

For a quick estimate, the worst-case (typical circuit) OR2T40A Clock Power ≈ 12.4 mW/MHz.

The power dissipated in a PIC is the sum of the power dissipated in the four I/Os in the PIC. This consists of power dissipated by inputs and ac power dissipated by outputs. The power dissipated in each I/O depends on whether it is configured as an input, output, or input/output. If an I/O is operating as an output, then there is a power dissipation component for P_{IN}, as well as P_{OUT}. This is because the output feeds back to the input.

The power dissipated by an input buffer ($V_{IH} = V_{DD} –$ 0.3 V or higher) is estimated as:

$$P_{IN} = 0.09 \text{ mW/MHz}$$

The 5 V tolerant input buffer feature dissipates additional dc power. The dc power, PTOL, is always dissipated for the OR2T15A, regardless of the number of 5 V tolerant input buffers used:

$$P_{TOL} = 15 \text{ mW}$$

The ac power dissipation from an output or bidirectional is estimated by the following:

$$P_{OUT} = (C_L + 8.8 \text{ pF}) \times V_{DD}^2 \times F \text{ Watts}$$

where the unit for C_L is farads, and the unit for F is Hz.

As an example of estimating power dissipation, suppose that a fully utilized OR2T15A has an average of three outputs for each of the 400 PFUs, that all 20 clock branches are used, that 150 of the 400 PFUs have FFs clocked at 40 MHz (16 of which are operating in a synchronous memory mode), and that the PFU outputs have an average activity factor of 20%.

Twenty inputs, 32 outputs driving 30 pF loads, and 16 bidirectional I/Os driving 50 pF loads are also generated from the 40 MHz clock with an average activity factor of 20%. The worst-case (V_{DD} = 3.6 V) power dissipation is estimated as follows:

P_{PFU} = 400 x 3 (0.08 mW/MHz x 20 MHz x 20%)
= 384 mW

P_{CLK} = [0.34 mW/MHz + (0.17 mW/MHz – Branch)
(20 Branches)
+ (0.01 mW/MHz – PFU) (150 PFUs)
+ (0.003 mW/MHz – SMEM_PFU)
(16 SMEM_PFUs)] [40 MHz]
= 212 mW

P_{IN} = 20 x [0.09 mW/MHz x 20 MHz x 20%]
= 7 mW

P_{TOL} = 15 mW

P_{OUT} = 30 x [(30 pF + 8.8 pF) x $(3.6)^2$ x 20 MHz
x 20%]
= 60 mW

P_{BID} = 16 x [(50 pF + 8.8 pF) x $(3.6)^2$ x 20 MHz
x 20%]
= 49 mW

TOTAL = 0.73 W

Pin Information

Table 15. Pin Descriptions

Symbol	I/O	Description
Dedicated Pins		
VDD	—	Positive power supply.
GND	—	Ground supply.
VDD5	—	5 V tolerant select. All VDD5 pins must be tied to either the 5 V power supply if 5 V tolerant I/O buffers are to be used, or to the 3.3 V power supply (VDD) if they are not.
$\overline{\text{RESET}}$	I	During configuration, $\overline{\text{RESET}}$ forces the restart of configuration and a pull-up is enabled. After configuration, $\overline{\text{RESET}}$ can be used as a general FPGA input or as a direct input, which causes all PLC latches/FFs to be asynchronously set/reset.
CCLK	I	In the master and asynchronous peripheral modes, CCLK is an output which strobes configuration data in. In the slave or synchronous peripheral mode, CCLK is input synchronous with the data on DIN or D[7:0].
DONE	I/O	DONE is a bidirectional pin with an optional pull-up resistor. As an active-high, open-drain output, a high-level on this signal indicates that configuration is complete. As an input, a low level on DONE delays FPGA start-up after configuration (see Note).
$\overline{\text{PRGM}}$	I	$\overline{\text{PRGM}}$ is an active-low input that forces the restart of configuration and resets the boundary-scan circuitry. This pin always has an active pull-up.
$\overline{\text{RD\_CFGN}}$	I	If readback is enabled, after configuration, a high-to-low transition on $\overline{\text{RD\_CFGN}}$ initiates a readback of configuration data, including PFU output states, starting with frame address 0. During configuration, this is an active-low input that activates the TS_ALL function and 3-states all the I/O. This same functionality can be selected after configuration as well. This pin always has an active pull-up.
RD_DATA/TDO	O	RD_DATA/TDO is a dual-function pin. If used for readback, RD_DATA provides configuration data out. If used in boundary scan, TDO is test data out.
Special-Purpose Pins		
RDY/$\overline{\text{BUSY}}$	O	During configuration in peripheral mode, RDY/$\overline{\text{BUSY}}$ indicates another byte can be written to the FPGA. If a read operation is done when the device is selected, the same status is also available on D7 in asynchronous peripheral mode. After configuration, the pin is a user-programmable I/O (see Note).
RCLK	O	During the master parallel configuration mode, RCLK is a read output signal to an external memory. This output is not normally used. After configuration, this pin is a user-programmable I/O pin (see Note).
DIN	I	During slave serial or master serial configuration modes, DIN accepts serial configuration data synchronous with CCLK. During parallel configuration modes, DIN is the D0 input. During configuration, a pull-up is enabled, and after configuration, this pin is a user-programmable I/O pin (see Note).
M0, M1, M2	I	M0—M2 are used to select the configuration mode. See Table 9 for the configuration modes. During configuration, a pull-up is enabled, and after configuration, the pins are user-programmable I/O (see Note).
M3	I	M3 is used to select the speed of the internal oscillator during configuration. When M3 is low, the oscillator frequency is 10 MHz. When M3 is high, the oscillator is 1.25 MHz. During configuration, a pull-up is enabled, and after configuration, this pin is a user-programmable I/O pin (see Note).

Note: The section FPGA States of Operation contains more information on how to control these signals during start-up. The timing of DONE release is controlled by one set of bit stream options, and the timing of the simultaneous release of all other configuration pins (and the activation of all user I/Os) is controlled by a second set of options.

Pin Information (continued)

Table 15. Pin Descriptions (continued)

Symbol	I/O	Description
Special-Purpose Pins (continued)		
TDI, TCK, TMS	I	If boundary scan is used, these pins are Test Data In, Test Clock, and Test Mode Select inputs. If boundary scan is not selected, all boundary-scan functions are inhibited once configuration is complete, and these pins are user-programmable I/O pins. Even if boundary scan is not used, either TCK or TMS must be held at logic 1 during configuration. Each pin has a pull-up enabled during configuration (see Note).
HDC	O	High During Configuration is output high until configuration is complete. It is used as a control output indicating that configuration is not complete. After configuration, this pin is a user-programmable I/O pin (see Note).
$\overline{\text{LDC}}$	O	$\overline{\text{Low During Configuration}}$ is output low until configuration is complete. It is used as a control output indicating that configuration is not complete. After configuration, this pin is a user-programmable I/O pin (see Note).
$\overline{\text{INIT}}$	I/O	$\overline{\text{INIT}}$ is a bidirectional signal before and during configuration. During configuration, a pull-up is enabled, but an external pull-up resistor is recommended. As an active-low open-drain output, $\overline{\text{INIT}}$ is held low during power stabilization and internal clearing of memory. As an active-low input, $\overline{\text{INIT}}$ holds the FPGA in the wait-state before the start of configuration. After configuration, the pin is a user-programmable I/O pin (see Note).
$\overline{\text{CS0}}$, CS1, $\overline{\text{WR}}$, $\overline{\text{RD}}$	I	$\overline{\text{CS0}}$, CS1, $\overline{\text{WR}}$, $\overline{\text{RD}}$ are used in the asynchronous peripheral configuration modes. The FPGA is selected when $\overline{\text{CS0}}$ is low and CS1 is high. When selected, a low on the write strobe, $\overline{\text{WR}}$, loads the data on D[7:0] inputs into an internal data buffer. $\overline{\text{WR}}$, $\overline{\text{CS0}}$, and CS1 are also used as chip selects in the slave parallel mode. A low on $\overline{\text{RD}}$ changes D7 into a status output. As a status indication, a high indicates ready and a low indicates busy. $\overline{\text{WR}}$ and $\overline{\text{RD}}$ should not be used simultaneously. If they are, the write strobe overrides. During configuration, a pull-up is enabled, and after configuration, the pins are user-programmable I/O pins (see Note).
A[17:0]	O	During master parallel configuration mode, A[17:0] address the configuration EPROM. During configuration, a pull-up is enabled, and after configuration, the pins are user-programmable I/O pins (see Note).
D[7:0]	I	During master parallel, peripheral, and slave parallel configuration modes, D[7:0] receive configuration data and each pin has a pull-up enabled. After configuration, the pins are user-programmable I/O pins (see Note).
DOUT	O	During configuration, DOUT is the serial data output that can drive the DIN of daisy-chained slave LCA devices. Data out on DOUT changes on the falling edge of CCLK. After configuration, DOUT is a user-programmable I/O pin (see Note).

Note: The section FPGA States of Operation contains more information on how to control these signals during start-up. The timing of DONE release is controlled by one set of bit stream options, and the timing of the simultaneous release of all other configuration pins (and the activation of all user I/Os) is controlled by a second set of options.

Pin Information (continued)

Package Compatibility

The package pinouts are consistent across *ORCA* Series FPGAs with the following exception: **user I/O pins that do not have any special functions will be converted to VDD5 pins for the OR2TxxA series.** If the designer does not use these pins for the OR2CxxA series, then pinout compatibility will be maintained between the *ORCA* OR2CxxA and OR2TxxA series of FPGAs. Note that they must be connected to a power supply for the OR2TxxA series.

Package pinouts being consistent across all *ORCA* Series FPGAs enables a designer to select a package based on I/O requirements and change the FPGA without laying out the printed-circuit board again. The change might be to a larger FPGA if additional functionality is needed, or it might be to a smaller FPGA to decrease unit cost.

Table 16A provides the number of user I/Os available for the *ORCA* OR2CxxA Series FPGAs for each available package, and Table 16B provides the number of user I/Os available in the *ORCA* OR2TxxA series. It should be noted that the number of user I/Os available for the OR2TxxA series is reduced from the equivalent OR2CxxA devices by the number of required VDD5 pins, as shown in Table 16B. The pins that are converted from user I/O to VDD5 are denoted as "I/O-VDD5" in the pin information tables (Tables 17 through 27). Each package has six dedicated configuration pins.

Tables 17—27 provide the package pin and pin function for the *ORCA* OR2CxxA/OR2TxxA series FPGAs and packages. The bond pad name is identified in the PIC nomenclature used in the *ORCA* Foundry design editor.

When the number of FPGA bond pads exceeds the number of package pins, bond pads are unused. When the number of package pins exceeds the number of bond pads, package pins are left unconnected (no connects). When a package pin is to be left as a no connect for a specific die, it is indicated as a note in the device pad column for the FPGA. The tables provide no information on unused pads.

Table 16A. *ORCA* OR2CxxA Series FPGA I/Os Summary

Device	84-Pin PLCC	100-Pin TQFP	144-Pin TQFP	160-Pin QFP	208-Pin SQFP/ SQFP-PQ2	240-Pin SQFP/ SQFP-PQ2	256-Pin PBGA	304-Pin SQFP/ SQFP-PQ2	352-Pin PBGA	432-Pin EBGA	600-Pin EBGA
OR2C04A											
User I/Os	64	77	114	130	160	—	—	—	—	—	—
VDD/VSS	14	17	24	24	31	—	—	—	—	—	—
OR2C06A											
User I/Os	64	77	114	130	171	192	192	—	—	—	—
VDD/VSS	14	17	24	24	31	42	26	—	—	—	—
OR2C08A											
User I/Os	64	—	—	130	171	192	221	—	—	—	—
VDD/VSS	14	—	—	24	31	40	26	—	—	—	—
OR2C10A											
User I/Os	64	—	—	130	171	192	221	—	256	—	—
VDD/VSS	14	—	—	24	31	40	26	—	48	—	—
OR2C12A											
User I/Os	64	—	—	—	171	192	223	252	288	—	—
VDD/VSS	14	—	—	—	31	42	26	46	48	—	—
OR2C15A											
User I/Os	64	—	—	—	171	192	223	252	298	320	—
VDD/VSS	14	—	—	—	31	42	26	46	48	84	—
OR2C26A											
User I/Os	—	—	—	—	171	192	—	252	298	342	384
VDD/VSS	—	—	—	—	31	42	—	46	48	84	140
OR2C40A											
User I/Os	—	—	—	—	171	192	—	252	—	342	454
VDD/VSS	—	—	—	—	31	42	—	46	—	84	140

Pin Information (continued)

Table 16B. *ORCA* OR2TxxA Series FPGA I/Os Summary

Device	84-Pin PLCC	100-Pin TQFP	144-Pin TQFP	160-Pin QFP	208-Pin SQFP/ SQFP-PQ2	240-Pin SQFP/ SQFP-PQ2	256-Pin PBGA	352-Pin PBGA	432-Pin EBGA	600-Pin EBGA
OR2T04A										
User I/Os	62	74	110	126	152	—	—	—	—	—
VDD/VSS	14	17	24	24	31	—	—	—	—	—
VDD5	2	3	4	4	8	—	—	—	—	—
OR2T06A										
User I/Os	62	74	110	126	163	184	182	—	—	—
VDD/VSS	14	17	24	24	31	42	26	—	—	—
VDD5	2	3	4	4	8	8	10	—	—	—
OR2T08A										
User I/Os	62	—	—	126	163	184	209	—	—	—
VDD/VSS	14	—	—	24	31	40	26	—	—	—
VDD5	2	—	—	4	8	8	12	—	—	—
OR2T10A										
User I/Os	62	—	—	126	163	184	209	244	—	—
VDD/VSS	14	—	—	24	31	40	26	48	—	—
VDD5	2	—	—	4	8	8	12	12	—	—
OR2T12A										
User I/Os	62	—	—	—	163	184	211	276	—	—
VDD/VSS	14	—	—	—	31	42	26	48	—	—
VDD5	2	—	—	—	8	8	12	12	—	—
OR2T15A										
User I/Os	62	—	—	—	163	184	211	286	307	—
VDD/VSS	14	—	—	—	31	42	26	48	84	—
VDD5	2	—	—	—	8	8	12	12	12	—
OR2T26A										
User I/Os	—	—	—	—	163	184	—	286	326	368
VDD/VSS	—	—	—	—	31	42	—	48	84	140
VDD5	—	—	—	—	8	8	—	12	16	16
OR2T40A										
User I/Os	—	—	—	—	163	184	—	—	326	438
VDD/VSS	—	—	—	—	31	42	—	—	84	140
VDD5	—	—	—	—	8	8	—	—	16	16

Pin Information (continued)

Table 17. OR2C/2T04A, OR2C/2T06A, OR2C/2T08A, OR2C/2T10A, OR2C/2T12A, and OR2C/2T15A 84-Pin PLCC Pinout

Pin	2C/2T04A Pad	2C/2T06A Pad	2C/2T08A Pad	2C/2T10A Pad	2C/2T12A Pad	2C/2T15A Pad	Function
1	VSS	VSS	VSS	VSS	VSS	VSS	VSS
2	PT5A	PT6A	PT7A	PT8A	PT9A	PT10A	I/O-D2
3	VSS	VSS	VSS	VSS	VSS	VSS	VSS
4	PT4D	PT5D	PT6D	PT7D	PT8D	PT9D	I/O-D1
5	PT4A	PT5A	PT6A	PT7A	PT8A	PT9A	I/O-D0/DIN
6	PT3A	PT4A	PT5A	PT6A	PT7A	PT8A	I/O-DOUT
7	PT2D	PT3D	PT4D	PT5D	PT6D	PT7D	I/O-VDD5
8	PT2A	PT3A	PT4A	PT4A	PT5A	PT6A	I/O-TDI
9	PT1D	PT2A	PT3A	PT3A	PT3A	PT4A	I/O-TMS
10	PT1A	PT1A	PT1A	PT1A	PT1A	PT1A	I/O-TCK
11	RD_DATA/TDO	RD_DATA/TDO	RD_DATA/TDO	RD_DATA/TDO	RD_DATA/TDO	RD_DATA/TDO	RD_DATA/TDO
12	VDD	VDD	VDD	VDD	VDD	VDD	VDD
13	VSS	VSS	VSS	VSS	VSS	VSS	VSS
14	PL1C	PL1A	PL2D	PL2D	PL2D	PL2D	I/O-A0
15	PL1A	PL2A	PL3A	PL3A	PL4A	PL5A	I/O-A1
16	PL2D	PL3D	PL4D	PL4A	PL5A	PL6A	I/O-A2
17	PL2A	PL3A	PL4A	PL5A	PL6A	PL7A	I/O-A3
18	PL3A	PL4A	PL5A	PL6A	PL7A	PL8A	I/O-A4
19	PL4D	PL5D	PL6D	PL7D	PL8D	PL9D	I/O-A5
20	PL4A	PL5A	PL6A	PL7A	PL8A	PL9A	I/O-A6
21	PL5A	PL6A	PL7A	PL8A	PL9A	PL10A	I/O-A7
22	VDD	VDD	VDD	VDD	VDD	VDD	VDD
23	PL6A	PL7A	PL8A	PL9A	PL10A	PL11A	I/O-A8
24	VSS	VSS	VSS	VSS	VSS	VSS	VSS
25	PL7D	PL8D	PL9D	PL10D	PL11D	PL12D	I/O-A9
26	PL7A	PL8A	PL9A	PL10A	PL11A	PL12A	I/O-A10
27	PL8A	PL9A	PL10A	PL11A	PL12A	PL13A	I/O-A11
28	PL9D	PL10D	PL11D	PL12D	PL13D	PL14D	I/O-A12
29	PL9A	PL10A	PL11A	PL13D	PL14B	PL15B	I/O-A13
30	PL10D	PL11A	PL12A	PL14C	PL16D	PL17D	I/O-A14
31	PL10A	PL12A	PL14A	PL16A	PL18A	PL20A	I/O-A15
32	CCLK	CCLK	CCLK	CCLK	CCLK	CCLK	CCLK
33	VDD	VDD	VDD	VDD	VDD	VDD	VDD
34	VSS	VSS	VSS	VSS	VSS	VSS	VSS
35	PB1A	PB1A	PB1A	PB1A	PB1A	PB1A	I/O-A16
36	PB1D	PB2A	PB3A	PB3B	PB3D	PB4D	I/O-A17
37	PB2A	PB3A	PB3D	PB4D	PB5B	PB6B	I/O
38	PB2D	PB3D	PB4D	PB5D	PB6D	PB7D	I/O
39	PB3A	PB4A	PB5A	PB6A	PB7A	PB8A	I/O
40	PB4A	PB5A	PB6A	PB7A	PB8A	PB9A	I/O
41	PB4D	PB5D	PB6D	PB7D	PB8D	PB9D	I/O
42	PB5A	PB6A	PB7A	PB8A	PB9A	PB10A	I/O
43	VSS	VSS	VSS	VSS	VSS	VSS	VSS

Note: The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

Pin Information (continued)

Table 17. OR2C/2T04A, OR2C/2T06A, OR2C/2T08A, OR2C/2T10A, OR2C/2T12A, and OR2C/2T15A 84-Pin PLCC Pinout (continued)

Pin	2C/2T04A Pad	2C/2T06A Pad	2C/2T08A Pad	2C/2T10A Pad	2C/2T12A Pad	2C/2T15A Pad	Function
44	PB6A	PB7A	PB8A	PB9A	PB10A	PB11A	I/O
45	VSS	VSS	VSS	VSS	VSS	VSS	VSS
46	PB7A	PB8A	PB9A	PB10A	PB11A	PB12A	I/O-VDD5
47	PB7D	PB8D	PB9D	PB10D	PB11D	PB12D	I/O
48	PB8A	PB9A	PB10A	PB11A	PB12A	PB13A	I/O-HDC
49	PB9A	PB10A	PB11A	PB12A	PB13A	PB14A	I/O-$\overline{\text{LDC}}$
50	PB9D	PB10D	PB11D	PB13A	PB13D	PB14D	I/O
51	PB10A	PB11A	PB12C	PB13D	PB15A	PB16A	I/O-$\overline{\text{INIT}}$
52	PB10D	PB12A	PB13D	PB15D	PB18D	PB20D	I/O
53	DONE	DONE	DONE	DONE	DONE	DONE	DONE
54	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$
55	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$
56	PR10A	PR12A	PR14A	PR16A	PR18A	PR20A	I/O-M0
57	PR10D	PR11A	PR12A	PR14A	PR16A	PR17A	I/O
58	PR9A	PR10A	PR11A	PR13B	PR15D	PR16D	I/O-M1
59	PR9D	PR10D	PR11D	PR12B	PR13A	PR14A	I/O
60	PR8A	PR9A	PR10A	PR11A	PR12A	PR13A	I/O-M2
61	PR7A	PR8A	PR9A	PR10A	PR11A	PR12A	I/O-M3
62	PR7D	PR8D	PR9D	PR10D	PR11D	PR12D	I/O
63	PR6A	PR7A	PR8D	PR9D	PR10A	PR11A	I/O
64	VDD	VDD	VDD	VDD	VDD	VDD	VDD
65	PR5A	PR6A	PR7A	PR8A	PR9A	PR10A	I/O
66	VSS	VSS	VSS	VSS	VSS	VSS	VSS
67	PR4A	PR5A	PR6A	PR7A	PR8A	PR9A	I/O
68	PR4D	PR5D	PR6D	PR7D	PR8D	PR9D	I/O
69	PR3A	PR4A	PR5A	PR6A	PR7A	PR8A	I/O-CS1
70	PR2A	PR3A	PR4A	PR5A	PR6A	PR7A	I/O-$\overline{\text{CS0}}$
71	PR2D	PR3D	PR4D	PR4D	PR5D	PR6D	I/O
72	PR1A	PR2A	PR3A	PR3A	PR4A	PR5A	I/O-$\overline{\text{RD}}$
73	PR1D	PR1A	PR2A	PR2A	PR2A	PR3A	I/O-$\overline{\text{WR}}$
74	$\overline{\text{RD\_CFGN}}$	$\overline{\text{RD\_CFGN}}$	$\overline{\text{RD\_CFGN}}$	$\overline{\text{RD\_CFGN}}$	$\overline{\text{RD\_CFGN}}$	$\overline{\text{RD\_CFGN}}$	$\overline{\text{RD\_CFGN}}$
75	VDD	VDD	VDD	VDD	VDD	VDD	VDD
76	VSS	VSS	VSS	VSS	VSS	VSS	VSS
77	PT10C	PT12A	PT13D	PT15D	PT17D	PT19A	I/O-RDY/RCLK
78	PT9D	PT11A	PT12C	PT13D	PT15D	PT16D	I/O-D7
79	PT9C	PT10D	PT11D	PT13A	PT14D	PT15D	I/O
80	PT9A	PT10A	PT11B	PT12B	PT13B	PT14B	I/O-D6
81	PT8A	PT9A	PT10A	PT11A	PT12A	PT13A	I/O-D5
82	PT7D	PT8D	PT9D	PT10D	PT11D	PT12D	I/O
83	PT7A	PT8A	PT9A	PT10A	PT11A	PT12A	I/O-D4
84	PT6A	PT7A	PT8A	PT9A	PT10A	PT11A	I/O-D3

Note: The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

Pin Information (continued)

Table 18. OR2C/2T04A and OR2C/2T06A 100-Pin TQFP Pinout

Pin	2C/2T04A Pad	2C/2T06A Pad	Function	Pin	2C/2T04A Pad	2C/2T06A Pad	Function
1	VDD	VDD	VDD	43	PB8C	PB9C	I/O
2	VSS	VSS	VSS	44	PB8D	PB9D	I/O
3	PL1C	PL1A	I/O-A0	45	PB9A	PB10A	I/O-$\overline{LDC}$
4	PL1A	PL2A	I/O-A1	46	PB9D	PB10D	I/O
5	PL2D	PL3D	I/O-A2	47	PB10A	PB11A	I/O-$\overline{INIT}$
6	PL2A	PL3A	I/O-A3	48	PB10D	PB12A	I/O
7	PL3D	PL4D	I/O	49	DONE	DONE	DONE
8	PL3A	PL4A	I/O-A4	50	VDD	VDD	VDD
9	PL4D	PL5D	I/O-A5	51	$\overline{RESET}$	$\overline{RESET}$	$\overline{RESET}$
10	PL4A	PL5A	I/O-A6	52	$\overline{PRGM}$	$\overline{PRGM}$	$\overline{PRGM}$
11	PL5D	PL6D	I/O	53	PR10A	PR12A	I/O-M0
12	PL5A	PL6A	I/O-A7	54	PR10D	PR11A	I/O
13	VDD	VDD	VDD	55	PR9A	PR10A	I/O-M1
14	PL6A	PL7A	I/O-A8	56	PR9D	PR10D	I/O
15	VSS	VSS	VSS	57	PR8A	PR9A	I/O-M2
16	PL7D	PL8D	I/O-A9	58	PR8D	PR9D	I/O
17	PL7A	PL8A	I/O-A10	59	PR7A	PR8A	I/O-M3
18	PL8A	PL9A	I/O-A11	60	PR7D	PR8D	I/O
19	PL9D	PL10D	I/O-A12	61	VSS	VSS	VSS
20	PL9C	PL10C	I/O	62	PR6A	PR7A	I/O
21	PL9A	PL10A	I/O-A13	63	VDD	VDD	VDD
22	PL10D	PL11A	I/O-A14	64	PR5A	PR6A	I/O
23	PL10A	PL12A	I/O-A15	65	VSS	VSS	VSS
24	VSS	VSS	VSS	66	PR4A	PR5A	I/O-VDD5
25	CCLK	CCLK	CCLK	67	PR4D	PR5D	I/O
26	VDD	VDD	VDD	68	PR3A	PR4A	I/O-CS1
27	VSS	VSS	VSS	69	PR3D	PR4D	I/O
28	PB1A	PB1A	I/O-A16	70	PR2A	PR3A	I/O-$\overline{CS0}$
29	PB1C	PB1D	I/O	71	PR2D	PR3D	I/O
30	PB1D	PB2A	I/O-A17	72	PR1A	PR2A	I/O-$\overline{RD}$
31	PB2A	PB3A	I/O	73	PR1C	PR2D	I/O
32	PB2D	PB3D	I/O	74	PR1D	PR1A	I/O-$\overline{WR}$
33	PB3A	PB4A	I/O	75	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$
34	PB4A	PB5A	I/O	76	VDD	VDD	VDD
35	PB4D	PB5D	I/O	77	VSS	VSS	VSS
36	PB5A	PB6A	I/O	78	PT10C	PT12A	I/O-RDY/RCLK
37	VSS	VSS	VSS	79	PT9D	PT11A	I/O-D7
38	PB6A	PB7A	I/O	80	PT9C	PT10D	I/O
39	VSS	VSS	VSS	81	PT9A	PT10A	I/O-D6
40	PB7A	PB8A	I/O-VDD5	82	PT8D	PT9D	I/O
41	PB7D	PB8D	I/O	83	PT8A	PT9A	I/O-D5
42	PB8A	PB9A	I/O-HDC	84	PT7D	PT8D	I/O

Note: The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

Pin Information (continued)

Table 18. OR2C/2T04A and OR2C/2T06A 100-Pin TQFP Pinout (continued)

Pin	2C/2T04A Pad	2C/2T06A Pad	Function	Pin	2C/2T04A Pad	2C/2T06A Pad	Function
85	PT7A	PT8A	I/O-D4	93	PT3D	PT4D	I/O
86	PT6D	PT7D	I/O	94	PT3A	PT4A	I/O-DOUT
87	PT6A	PT7A	I/O-D3	95	PT2D	PT3D	I/O-V_{DD5}
88	V_{SS}	V_{SS}	V_{SS}	96	PT2A	PT3A	I/O-TDI
89	PT5A	PT6A	I/O-D2	97	PT1D	PT2A	I/O-TMS
90	V_{SS}	V_{SS}	V_{SS}	98	PT1C	PT1D	I/O
91	PT4D	PT5D	I/O-D1	99	PT1A	PT1A	I/O-TCK
92	PT4A	PT5A	I/O-D0/DIN	100	RD_DATA/ TDO	RD_DATA/ TDO	RD_DATA/TDO

Note: The pins labeled "I/O-V_{DD5}" are user I/Os for the OR2CxxA series, but they are connected to V_{DD5} for the OR2TxxA series.

Pin Information (continued)

Table 19. OR2C/2T04A and OR2C/2T06A 144-Pin TQFP Pinout

Pin	2C/2T04A Pad	2C/2T06A Pad	Function	Pin	2C/2T04A Pad	2C/2T06A Pad	Function
1	VDD	VDD	VDD	43	PB2B	PB3B	I/O
2	VSS	VSS	VSS	44	PB2D	PB3D	I/O
3	PL1C	PL1A	I/O-A0	45	VDD	VDD	VDD
4	PL1B	PL2D	I/O	46	PB3A	PB4A	I/O
5	PL1A	PL2A	I/O-A1	47	PB3D	PB4D	I/O
6	PL2D	PL3D	I/O-A2	48	PB4A	PB5A	I/O
7	PL2A	PL3A	I/O-A3	49	PB4C	PB5C	I/O
8	PL3D	PL4D	I/O	50	PB4D	PB5D	I/O
9	PL3C	PL4C	I/O	51	PB5A	PB6A	I/O
10	PL3A	PL4A	I/O-A4	52	PB5C	PB6C	I/O
11	PL4D	PL5D	I/O-A5	53	PB5D	PB6D	I/O
12	PL4C	PL5C	I/O	54	VSS	VSS	VSS
13	PL4A	PL5A	I/O-A6	55	PB6A	PB7A	I/O
14	VSS	VSS	VSS	56	PB6C	PB7C	I/O
15	PL5D	PL6D	I/O	57	PB6D	PB7D	I/O
16	PL5C	PL6C	I/O	58	PB7A	PB8A	I/O-VDD5
17	PL5A	PL6A	I/O-A7	59	PB7D	PB8D	I/O
18	VDD	VDD	VDD	60	PB8A	PB9A	I/O-HDC
19	PL6D	PL7D	I/O	61	PB8C	PB9C	I/O
20	PL6C	PL7C	I/O-VDD5	62	PB8D	PB9D	I/O
21	PL6A	PL7A	I/O-A8	63	VDD	VDD	VDD
22	VSS	VSS	VSS	64	PB9A	PB10A	I/O-$\overline{\text{LDC}}$
23	PL7D	PL8D	I/O-A9	65	PB9C	PB10C	I/O
24	PL7A	PL8A	I/O-A10	66	PB9D	PB10D	I/O
25	PL8D	PL9D	I/O	67	PB10A	PB11A	I/O-$\overline{\text{INIT}}$
26	PL8C	PL9C	I/O	68	PB10C	PB11D	I/O
27	PL8A	PL9A	I/O-A11	69	PB10D	PB12A	I/O
28	PL9D	PL10D	I/O-A12	70	VSS	VSS	VSS
29	PL9C	PL10C	I/O	71	DONE	DONE	DONE
30	PL9A	PL10A	I/O-A13	72	VDD	VDD	VDD
31	PL10D	PL11A	I/O-A14	73	VSS	VSS	VSS
32	PL10C	PL12D	I/O	74	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$
33	PL10B	PL12B	I/O	75	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$
34	PL10A	PL12A	I/O-A15	76	PR10A	PR12A	I/O-M0
35	VSS	VSS	VSS	77	PR10B	PR12D	I/O
36	CCLK	CCLK	CCLK	78	PR10D	PR11A	I/O
37	VDD	VDD	VDD	79	PR9A	PR10A	I/O-M1
38	VSS	VSS	VSS	80	PR9C	PR10C	I/O
39	PB1A	PB1A	I/O-A16	81	PR9D	PR10D	I/O
40	PB1C	PB1D	I/O	82	PR8A	PR9A	I/O-M2
41	PB1D	PB2A	I/O-A17	83	PR8B	PR9B	I/O
42	PB2A	PB3A	I/O	84	PR8D	PR9D	I/O

Note: The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

Pin Information (continued)

Table 19. OR2C/2T04A and OR2C/2T06A 144-Pin TQFP Pinout (continued)

Pin	2C/2T04A Pad	2C/2T06A Pad	Function	Pin	2C/2T04A Pad	2C/2T06A Pad	Function
85	PR7A	PR8A	I/O-M3	115	PT9C	PT10D	I/O
86	PR7D	PR8D	I/O	116	PT9B	PT10C	I/O
87	VSS	VSS	VSS	117	PT9A	PT10A	I/O-D6
88	PR6A	PR7A	I/O	118	VDD	VDD	VDD
89	PR6C	PR7C	I/O	119	PT8D	PT9D	I/O
90	PR6D	PR7D	I/O	120	PT8A	PT9A	I/O-D5
91	VDD	VDD	VDD	121	PT7D	PT8D	I/O
92	PR5A	PR6A	I/O	122	PT7B	PT8B	I/O
93	PR5C	PR6C	I/O	123	PT7A	PT8A	I/O-D4
94	PR5D	PR6D	I/O	124	PT6D	PT7D	I/O
95	VSS	VSS	VSS	125	PT6C	PT7C	I/O
96	PR4A	PR5A	I/O-VDD5	126	PT6A	PT7A	I/O-D3
97	PR4C	PR5C	I/O	127	VSS	VSS	VSS
98	PR4D	PR5D	I/O	128	PT5D	PT6D	I/O
99	PR3A	PR4A	I/O-CS1	129	PT5C	PT6C	I/O
100	PR3D	PR4D	I/O	130	PT5A	PT6A	I/O-D2
101	PR2A	PR3A	I/O-$\overline{CS0}$	131	PT4D	PT5D	I/O-D1
102	PR2D	PR3D	I/O	132	PT4C	PT5C	I/O
103	PR1A	PR2A	I/O-$\overline{RD}$	133	PT4A	PT5A	I/O-D0/DIN
104	PR1B	PR2C	I/O	134	PT3D	PT4D	I/O
105	PR1C	PR2D	I/O	135	PT3A	PT4A	I/O-DOUT
106	PR1D	PR1A	I/O-$\overline{WR}$	136	VDD	VDD	VDD
107	VSS	VSS	VSS	137	PT2D	PT3D	I/O-VDD5
108	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	138	PT2C	PT3C	I/O
109	VDD	VDD	VDD	139	PT2A	PT3A	I/O-TDI
110	VSS	VSS	VSS	140	PT1D	PT2A	I/O-TMS
111	PT10D	PT12D	I/O	141	PT1C	PT1D	I/O
112	PT10C	PT12A	I/O-RDY/RCLK	142	PT1A	PT1A	I/O-TCK
113	PT10B	PT11D	I/O	143	VSS	VSS	VSS
114	PT9D	PT11A	I/O-D7	144	RD_DATA/ TDO	RD_DATA/ TDO	RD_DATA/TDO

Note: The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

Pin Information (continued)

Table 20. OR2C/2T04A, OR2C/2T06A, OR2C/2T08A, and OR2C/2T10A 160-Pin QFP Pinout

Pin	2C/2T04A Pad	2C/2T06A Pad	2C/2T08A Pad	2C/2T10A Pad	Function
1	VDD	VDD	VDD	VDD	VDD
2	VSS	VSS	VSS	VSS	VSS
3	PL1D	PL1D	PL1D	PL1D	I/O
4	PL1C	PL1A	PL2D	PL2D	I/O-A0
5	PL1B	PL2D	PL3D	PL3D	I/O
6	PL1A	PL2A	PL3A	PL3A	I/O-A1
7	PL2D	PL3D	PL4D	PL4A	I/O-A2
8	PL2C	PL3C	PL4C	PL5C	I/O
9	PL2A	PL3A	PL4A	PL5A	I/O-A3
10	PL3D	PL4D	PL5D	PL6D	I/O
11	PL3C	PL4C	PL5C	PL6C	I/O
12	PL3A	PL4A	PL5A	PL6A	I/O-A4
13	PL4D	PL5D	PL6D	PL7D	I/O-A5
14	PL4C	PL5C	PL6C	PL7C	I/O
15	PL4A	PL5A	PL6A	PL7A	I/O-A6
16	VSS	VSS	VSS	VSS	VSS
17	PL5D	PL6D	PL7D	PL8D	I/O
18	PL5C	PL6C	PL7C	PL8C	I/O
19	PL5A	PL6A	PL7A	PL8A	I/O-A7
20	VDD	VDD	VDD	VDD	VDD
21	PL6D	PL7D	PL8D	PL9D	I/O
22	PL6C	PL7C	PL8C	PL9C	I/O-VDD5
23	PL6A	PL7A	PL8A	PL9A	I/O-A8
24	VSS	VSS	VSS	VSS	VSS
25	PL7D	PL8D	PL9D	PL10D	I/O-A9
26	PL7B	PL8B	PL9B	PL10B	I/O
27	PL7A	PL8A	PL9A	PL10A	I/O-A10
28	PL8D	PL9D	PL10D	PL11D	I/O
29	PL8C	PL9C	PL10C	PL11C	I/O
30	PL8A	PL9A	PL10A	PL11A	I/O-A11
31	PL9D	PL10D	PL11D	PL12D	I/O-A12
32	PL9C	PL10C	PL11C	PL12C	I/O
33	PL9B	PL10B	PL11B	PL12B	I/O
34	PL9A	PL10A	PL11A	PL13D	I/O-A13
35	PL10D	PL11A	PL12A	PL14C	I/O-A14
36	PL10C	PL12D	PL13D	PL15D	I/O
37	PL10B	PL12B	PL14D	PL16D	I/O
38	PL10A	PL12A	PL14A	PL16A	I/O-A15
39	CCLK	CCLK	CCLK	CCLK	CCLK
40	VSS	VSS	VSS	VSS	VSS
41	VDD	VDD	VDD	VDD	VDD
42	VSS	VSS	VSS	VSS	VSS

Note: The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

Pin Information (continued)

Table 20. OR2C/2T04A, OR2C/2T06A, OR2C/2T08A, and OR2C/2T10A 160-Pin QFP Pinout (continued)

Pin	2C/2T04A Pad	2C/2T06A Pad	2C/2T08A Pad	2C/2T10A Pad	Function
43	PB1A	PB1A	PB1A	PB1A	I/O-A16
44	PB1B	PB1C	PB2A	PB2A	I/O
45	PB1C	PB1D	PB2D	PB2D	I/O
46	PB1D	PB2A	PB3A	PB3B	I/O-A17
47	PB2A	PB3A	PB3D	PB4D	I/O
48	PB2B	PB3B	PB4A	PB5A	I/O
49	PB2C	PB3C	PB4C	PB5C	I/O
50	PB2D	PB3D	PB4D	PB5D	I/O
51	VDD	VDD	VDD	VDD	VDD
52	PB3A	PB4A	PB5A	PB6A	I/O
53	PB3D	PB4D	PB5D	PB6D	I/O
54	PB4A	PB5A	PB6A	PB7A	I/O
55	PB4C	PB5C	PB6C	PB7C	I/O
56	PB4D	PB5D	PB6D	PB7D	I/O
57	PB5A	PB6A	PB7A	PB8A	I/O
58	PB5C	PB6C	PB7C	PB8C	I/O
59	PB5D	PB6D	PB7D	PB8D	I/O
60	VSS	VSS	VSS	VSS	VSS
61	PB6A	PB7A	PB8A	PB9A	I/O
62	PB6C	PB7C	PB8C	PB9C	I/O
63	PB6D	PB7D	PB8D	PB9D	I/O
64	PB7A	PB8A	PB9A	PB10A	I/O-VDD5
65	PB7D	PB8D	PB9D	PB10D	I/O
66	PB8A	PB9A	PB10A	PB11A	I/O-HDC
67	PB8C	PB9C	PB10C	PB11C	I/O
68	PB8D	PB9D	PB10D	PB11D	I/O
69	VDD	VDD	VDD	VDD	VDD
70	PB9A	PB10A	PB11A	PB12A	I/O-$\overline{\text{LDC}}$
71	PB9B	PB10B	PB11D	PB13A	I/O
72	PB9C	PB10C	PB12A	PB13B	I/O
73	PB9D	PB10D	PB12B	PB13C	I/O
74	PB10A	PB11A	PB12C	PB13D	I/O-$\overline{\text{INIT}}$
75	PB10B	PB11C	PB12D	PB14A	I/O
76	PB10C	PB11D	PB13D	PB15D	I/O
77	PB10D	PB12A	PB14D	PB16D	I/O
78	VSS	VSS	VSS	VSS	VSS
79	DONE	DONE	DONE	DONE	DONE
80	VDD	VDD	VDD	VDD	VDD
81	VSS	VSS	VSS	VSS	VSS
82	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$
83	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$

Note: The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

Pin Information (continued)

Table 20. OR2C/2T04A, OR2C/2T06A, OR2C/2T08A, and OR2C/2T10A 160-Pin QFP Pinout (continued)

Pin	2C/2T04A Pad	2C/2T06A Pad	2C/2T08A Pad	2C/2T10A Pad	Function
84	PR10A	PR12A	PR14A	PR16A	I/O-M0
85	PR10B	PR12D	PR13A	PR15A	I/O
86	PR10C	PR11A	PR13D	PR15D	I/O
87	PR10D	PR11B	PR12A	PR14A	I/O
88	PR9A	PR10A	PR11A	PR13B	I/O-M1
89	PR9B	PR10B	PR11B	PR13C	I/O
90	PR9C	PR10C	PR11C	PR12A	I/O
91	PR9D	PR10D	PR11D	PR12B	I/O
92	PR8A	PR9A	PR10A	PR11A	I/O-M2
93	PR8B	PR9B	PR10B	PR11B	I/O
94	PR8D	PR9D	PR10D	PR11D	I/O
95	PR7A	PR8A	PR9A	PR10A	I/O-M3
96	PR7D	PR8D	PR9D	PR10D	I/O
97	VSS	VSS	VSS	VSS	VSS
98	PR6A	PR7A	PR8A	PR9A	I/O
99	PR6C	PR7C	PR8C	PR9C	I/O
100	PR6D	PR7D	PR8D	PR9D	I/O
101	VDD	VDD	VDD	VDD	VDD
102	PR5A	PR6A	PR7A	PR8A	I/O
103	PR5C	PR6C	PR7C	PR8C	I/O
104	PR5D	PR6D	PR7D	PR8D	I/O
105	VSS	VSS	VSS	VSS	VSS
106	PR4A	PR5A	PR6A	PR7A	I/O-VDD5
107	PR4C	PR5C	PR6C	PR7C	I/O
108	PR4D	PR5D	PR6D	PR7D	I/O
109	PR3A	PR4A	PR5A	PR6A	I/O-CS1
110	PR3B	PR4B	PR5B	PR6B	I/O
111	PR3D	PR4D	PR5D	PR6D	I/O
112	PR2A	PR3A	PR4A	PR5A	I/O-$\overline{CS0}$
113	PR2C	PR3C	PR4B	PR4B	I/O
114	PR2D	PR3D	PR4D	PR4D	I/O
115	PR1A	PR2A	PR3A	PR3A	I/O-$\overline{RD}$
116	PR1B	PR2C	PR3C	PR3C	I/O
117	PR1C	PR2D	PR3D	PR3D	I/O
118	PR1D	PR1A	PR2A	PR2A	I/O-$\overline{WR}$
119	VSS	VSS	VSS	VSS	VSS
120	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$
121	VDD	VDD	VDD	VDD	VDD
122	VSS	VSS	VSS	VSS	VSS
123	PT10D	PT12D	PT14D	PT16D	I/O
124	PT10C	PT12A	PT13D	PT15D	I/O-RDY/RCLK

Note: The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

Pin Information (continued)

Table 20. OR2C/2T04A, OR2C/2T06A, OR2C/2T08A, and OR2C/2T10A 160-Pin QFP Pinout (continued)

Pin	2C/2T04A Pad	2C/2T06A Pad	2C/2T08A Pad	2C/2T10A Pad	Function
125	PT10B	PT11D	PT13A	PT15A	I/O
126	PT10A	PT11C	PT12D	PT14D	I/O
127	PT9D	PT11A	PT12C	PT13D	I/O-D7
128	PT9C	PT10D	PT12A	PT13B	I/O
129	PT9B	PT10C	PT11D	PT13A	I/O
130	PT9A	PT10A	PT11B	PT12B	I/O-D6
131	VDD	VDD	VDD	VDD	VDD
132	PT8D	PT9D	PT10D	PT11D	I/O
133	PT8A	PT9A	PT10A	PT11A	I/O-D5
134	PT7D	PT8D	PT9D	PT10D	I/O
135	PT7B	PT8B	PT9B	PT10B	I/O
136	PT7A	PT8A	PT9A	PT10A	I/O-D4
137	PT6D	PT7D	PT8D	PT9D	I/O
138	PT6C	PT7C	PT8C	PT9C	I/O
139	PT6A	PT7A	PT8A	PT9A	I/O-D3
140	VSS	VSS	VSS	VSS	VSS
141	PT5D	PT6D	PT7D	PT8D	I/O
142	PT5C	PT6C	PT7C	PT8C	I/O
143	PT5A	PT6A	PT7A	PT8A	I/O-D2
144	PT4D	PT5D	PT6D	PT7D	I/O-D1
145	PT4C	PT5C	PT6C	PT7C	I/O
146	PT4A	PT5A	PT6A	PT7A	I/O-D0/DIN
147	PT3D	PT4D	PT5D	PT6D	I/O
148	PT3C	PT4C	PT5C	PT6C	I/O
149	PT3A	PT4A	PT5A	PT6A	I/O-DOUT
150	VDD	VDD	VDD	VDD	VDD
151	PT2D	PT3D	PT4D	PT5D	I/O-VDD5
152	PT2C	PT3C	PT4C	PT5A	I/O
153	PT2B	PT3B	PT4B	PT4D	I/O
154	PT2A	PT3A	PT4A	PT4A	I/O-TDI
155	PT1D	PT2A	PT3A	PT3A	I/O-TMS
156	PT1C	PT1D	PT2A	PT2A	I/O
157	PT1B	PT1C	PT1D	PT1D	I/O
158	PT1A	PT1A	PT1A	PT1A	I/O-TCK
159	VSS	VSS	VSS	VSS	VSS
160	RD_DATA/TDO	RD_DATA/TDO	RD_DATA/TDO	RD_DATA/TDO	RD_DATA/TDO

Note: The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

Pin Information (continued)

Table 21. OR2C/2T04A, OR2C/2T06A, OR2C/2T08A, OR2C/2T10A, OR2C/2T12A, OR2C/2T15A, OR2C/2T26A, and OR2C/2T40A 208-Pin SQFP/SQFP2 Pinout

Pin	2C/2T04A Pad	2C/2T06A Pad	2C/2T08A Pad	2C/2T10A Pad	2C/2T12A Pad	2C/2T15A Pad	2C/2T26A Pad	2C/2T40A Pad	Function
1	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
2	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
3	PL1D	PL1D	PL1D	PL1D	PL1D	PL1D	PL1D	PL1D	I/O
4	PL1C	PL1A	PL2D	PL2D	PL2D	PL2D	PL2D	PL3D	I/O-A0
5	PL1B	PL2D	PL3D	PL3D	PL3D	PL4D	PL4D	PL5D	I/O-VDD5
6	See Note	PL2C	PL3C	PL3C	PL3A	PL4A	PL4A	PL6D	I/O
7	PL1A	PL2A	PL3A	PL3A	PL4A	PL5A	PL5A	PL8D	I/O-A1
8	PL2D	PL3D	PL4D	PL4A	PL5A	PL6A	PL6A	PL9A	I/O-A2
9	PL2C	PL3C	PL4C	PL5C	PL6D	PL7D	PL7D	PL10D	I/O
10	PL2B	PL3B	PL4B	PL5B	PL6B	PL7B	PL7B	PL10B	I/O
11	PL2A	PL3A	PL4A	PL5A	PL6A	PL7A	PL7A	PL10A	I/O-A3
12	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
13	PL3D	PL4D	PL5D	PL6D	PL7D	PL8D	PL8D	PL11D	I/O
14	PL3C	PL4C	PL5C	PL6C	PL7C	PL8C	PL8A	PL11A	I/O
15	PL3B	PL4B	PL5B	PL6B	PL7B	PL8B	PL9D	PL12D	I/O
16	PL3A	PL4A	PL5A	PL6A	PL7A	PL8A	PL9A	PL12A	I/O-A4
17	PL4D	PL5D	PL6D	PL7D	PL8D	PL9D	PL10D	PL13D	I/O-A5
18	PL4C	PL5C	PL6C	PL7C	PL8C	PL9C	PL10A	PL13A	I/O
19	PL4B	PL5B	PL6B	PL7B	PL8B	PL9B	PL11D	PL14D	I/O
20	PL4A	PL5A	PL6A	PL7A	PL8A	PL9A	PL11A	PL14A	I/O-A6
21	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
22	PL5D	PL6D	PL7D	PL8D	PL9D	PL10D	PL12D	PL15D	I/O
23	PL5C	PL6C	PL7C	PL8C	PL9C	PL10C	PL12C	PL15C	I/O
24	PL5B	PL6B	PL7B	PL8B	PL9B	PL10B	PL12B	PL15B	I/O
25	PL5A	PL6A	PL7A	PL8A	PL9A	PL10A	PL12A	PL15A	I/O-A7
26	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
27	PL6D	PL7D	PL8D	PL9D	PL10D	PL11D	PL13D	PL16D	I/O
28	PL6C	PL7C	PL8C	PL9C	PL10C	PL11C	PL13C	PL16C	I/O-VDD5
29	PL6B	PL7B	PL8B	PL9B	PL10B	PL11B	PL13B	PL16B	I/O
30	PL6A	PL7A	PL8A	PL9A	PL10A	PL11A	PL13A	PL16A	I/O-A8
31	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
32	PL7D	PL8D	PL9D	PL10D	PL11D	PL12D	PL14D	PL17D	I/O-A9
33	PL7C	PL8C	PL9C	PL10C	PL11C	PL12C	PL14A	PL17A	I/O
34	PL7B	PL8B	PL9B	PL10B	PL11B	PL12B	PL15D	PL18D	I/O
35	PL7A	PL8A	PL9A	PL10A	PL11A	PL12A	PL15A	PL18A	I/O-A10
36	PL8D	PL9D	PL10D	PL11D	PL12D	PL13D	PL16D	PL19D	I/O
37	PL8C	PL9C	PL10C	PL11C	PL12C	PL13C	PL16A	PL19A	I/O
38	PL8B	PL9B	PL10B	PL11B	PL12B	PL13B	PL17D	PL20D	I/O

Notes:
The OR2C04A and OR2T04A do not have bond pads connected to 208-pin SQFP package pin numbers 6, 45, 47, 56, 60, 102, 153, 154, 166, 201, and 203.

The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

Pin Information (continued)

Table 21. OR2C/2T04A, OR2C/2T06A, OR2C/2T08A, OR2C/2T10A, OR2C/2T12A, OR2C/2T15A, OR2C/2T26A, and OR2C/2T40A 208-Pin SQFP/SQFP2 Pinout (continued)

Pin	2C/2T04A Pad	2C/2T06A Pad	2C/2T08A Pad	2C/2T10A Pad	2C/2T12A Pad	2C/2T15A Pad	2C/2T26A Pad	2C/2T40A Pad	Function
39	PL8A	PL9A	PL10A	PL11A	PL12A	PL13A	PL17A	PL20A	I/O-A11
40	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
41	PL9D	PL10D	PL11D	PL12D	PL13D	PL14D	PL18D	PL21D	I/O-A12
42	PL9C	PL10C	PL11C	PL12C	PL13B	PL14B	PL18B	PL21B	I/O
43	PL9B	PL10B	PL11B	PL12B	PL14D	PL15D	PL19D	PL22D	I/O
44	PL9A	PL10A	PL11A	PL13D	PL14B	PL15B	PL19B	PL22B	I/O-A13
45	See Note	PL11D	PL12D	PL13B	PL15D	PL16D	PL20D	PL23D	I/O
46	PL10D	PL11A	PL12A	PL14C	PL16D	PL17D	PL21D	PL25A	I/O-A14
47	See Note	PL12D	PL13D	PL15D	PL17D	PL18D	PL22D	PL27D	I/O
48	PL10C	PL12C	PL13A	PL15A	PL17A	PL19D	PL23D	PL28D	I/O
49	PL10B	PL12B	PL14D	PL16D	PL18C	PL19A	PL23A	PL28A	I/O
50	PL10A	PL12A	PL14A	PL16A	PL18A	PL20A	PL24A	PL30A	I/O-A15
51	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
52	CCLK	CCLK	CCLK	CCLK	CCLK	CCLK	CCLK	CCLK	CCLK
53	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
54	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
55	PB1A	PB1A	PB1A	PB1A	PB1A	PB1A	PB1A	PB1A	I/O-A16
56	See Note	PB1B	PB1D	PB1D	PB1D	PB2A	PB2A	PB3A	I/O
57	PB1B	PB1C	PB2A	PB2A	PB2A	PB2D	PB2D	PB3D	I/O-VDD5
58	PB1C	PB1D	PB2D	PB2D	PB2D	PB3D	PB3D	PB4D	I/O
59	PB1D	PB2A	PB3A	PB3B	PB3D	PB4D	PB4D	PB5D	I/O-A17
60	See Note	PB2D	PB3D	PB4D	PB4D	PB5D	PB5D	PB6D	I/O
61	PB2A	PB3A	PB4A	PB5A	PB5B	PB6B	PB6B	PB7D	I/O
62	PB2B	PB3B	PB4B	PB5B	PB5D	PB6D	PB6D	PB8D	I/O
63	PB2C	PB3C	PB4C	PB5C	PB6B	PB7B	PB7B	PB9D	I/O
64	PB2D	PB3D	PB4D	PB5D	PB6D	PB7D	PB7D	PB10D	I/O
65	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
66	PB3A	PB4A	PB5A	PB6A	PB7A	PB8A	PB8A	PB11A	I/O
67	PB3B	PB4B	PB5B	PB6B	PB7B	PB8B	PB8D	PB11D	I/O
68	PB3C	PB4C	PB5C	PB6C	PB7C	PB8C	PB9A	PB12A	I/O
69	PB3D	PB4D	PB5D	PB6D	PB7D	PB8D	PB9D	PB12D	I/O
70	PB4A	PB5A	PB6A	PB7A	PB8A	PB9A	PB10A	PB13A	I/O
71	PB4B	PB5B	PB6B	PB7B	PB8B	PB9B	PB10D	PB13D	I/O
72	PB4C	PB5C	PB6C	PB7C	PB8C	PB9C	PB11A	PB14A	I/O
73	PB4D	PB5D	PB6D	PB7D	PB8D	PB9D	PB11D	PB14D	I/O
74	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS

Notes:
The OR2C04A and OR2T04A do not have bond pads connected to 208-pin SQFP package pin numbers 6, 45, 47, 56, 60, 102, 153, 154, 166, 201, and 203.

The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

Pin Information (continued)

Table 21. OR2C/2T04A, OR2C/2T06A, OR2C/2T08A, OR2C/2T10A, OR2C/2T12A, OR2C/2T15A, OR2C/2T26A, and OR2C/2T40A 208-Pin SQFP/SQFP2 Pinout (continued)

Pin	2C/2T04A Pad	2C/2T06A Pad	2C/2T08A Pad	2C/2T10A Pad	2C/2T12A Pad	2C/2T15A Pad	2C/2T26A Pad	2C/2T40A Pad	Function
75	PB5A	PB6A	PB7A	PB8A	PB9A	PB10A	PB12A	PB15A	I/O
76	PB5B	PB6B	PB7B	PB8B	PB9B	PB10B	PB12B	PB15B	I/O
77	PB5C	PB6C	PB7C	PB8C	PB9C	PB10C	PB12C	PB15C	I/O
78	PB5D	PB6D	PB7D	PB8D	PB9D	PB10D	PB12D	PB15D	I/O
79	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
80	PB6A	PB7A	PB8A	PB9A	PB10A	PB11A	PB13A	PB16A	I/O
81	PB6B	PB7B	PB8B	PB9B	PB10B	PB11B	PB13B	PB16B	I/O
82	PB6C	PB7C	PB8C	PB9C	PB10C	PB11C	PB13C	PB16C	I/O
83	PB6D	PB7D	PB8D	PB9D	PB10D	PB11D	PB13D	PB16D	I/O
84	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
85	PB7A	PB8A	PB9A	PB10A	PB11A	PB12A	PB14A	PB17A	I/O-VDD5
86	PB7B	PB8B	PB9B	PB10B	PB11B	PB12B	PB14D	PB17D	I/O
87	PB7C	PB8C	PB9C	PB10C	PB11C	PB12C	PB15A	PB18A	I/O
88	PB7D	PB8D	PB9D	PB10D	PB11D	PB12D	PB15D	PB18D	I/O
89	PB8A	PB9A	PB10A	PB11A	PB12A	PB13A	PB16A	PB19A	I/O-HDC
90	PB8B	PB9B	PB10B	PB11B	PB12B	PB13B	PB16D	PB19D	I/O
91	PB8C	PB9C	PB10C	PB11C	PB12C	PB13C	PB17A	PB20A	I/O
92	PB8D	PB9D	PB10D	PB11D	PB12D	PB13D	PB17D	PB20D	I/O
93	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
94	PB9A	PB10A	PB11A	PB12A	PB13A	PB14A	PB18A	PB21A	I/O-$\overline{\text{LDC}}$
95	PB9B	PB10B	PB11D	PB13A	PB13D	PB14D	PB18D	PB22D	I/O
96	PB9C	PB10C	PB12A	PB13B	PB14A	PB15A	PB19A	PB23A	I/O
97	PB9D	PB10D	PB12B	PB13C	PB14D	PB15D	PB19D	PB24D	I/O
98	PB10A	PB11A	PB12C	PB13D	PB15A	PB16A	PB20A	PB25A	I/O-$\overline{\text{INIT}}$
99	PB10B	PB11C	PB12D	PB14A	PB16A	PB17A	PB21A	PB26A	I/O
100	PB10C	PB11D	PB13A	PB15A	PB17A	PB18A	PB22A	PB27A	I/O
101	PB10D	PB12A	PB13D	PB15D	PB18A	PB19D	PB23D	PB28D	I/O
102	See Note	PB12D	PB14D	PB16D	PB18D	PB20D	PB24D	PB30D	I/O
103	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
104	DONE	DONE	DONE	DONE	DONE	DONE	DONE	DONE	DONE
105	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
106	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$
107	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$
108	PR10A	PR12A	PR14A	PR16A	PR18A	PR20A	PR24A	PR30A	I/O-M0
109	PR10B	PR12D	PR13A	PR15A	PR18D	PR19A	PR23A	PR28A	I/O
110	PR10C	PR11A	PR13D	PR15D	PR17B	PR18A	PR22A	PR27A	I/O

Notes:
The OR2C04A and OR2T04A do not have bond pads connected to 208-pin SQFP package pin numbers 6, 45, 47, 56, 60, 102, 153, 154, 166, 201, and 203.

The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

Pin Information (continued)

Table 21. OR2C/2T04A, OR2C/2T06A, OR2C/2T08A, OR2C/2T10A, OR2C/2T12A, OR2C/2T15A, OR2C/2T26A, and OR2C/2T40A 208-Pin SQFP/SQFP2 Pinout (continued)

Pin	2C/2T04A Pad	2C/2T06A Pad	2C/2T08A Pad	2C/2T10A Pad	2C/2T12A Pad	2C/2T15A Pad	2C/2T26A Pad	2C/2T40A Pad	Function
111	PR10D	PR11B	PR12A	PR14A	PR16A	PR17A	PR21A	PR26A	I/O
112	PR9A	PR10A	PR11A	PR13B	PR15D	PR16D	PR20D	PR23D	I/O-M1
113	PR9B	PR10B	PR11B	PR13C	PR14A	PR15A	PR19A	PR22A	I/O
114	PR9C	PR10C	PR11C	PR12A	PR14D	PR15D	PR19D	PR22D	I/O-VDD5
115	PR9D	PR10D	PR11D	PR12B	PR13A	PR14A	PR18A	PR21A	I/O
116	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
117	PR8A	PR9A	PR10A	PR11A	PR12A	PR13A	PR17A	PR20A	I/O-M2
118	PR8B	PR9B	PR10B	PR11B	PR12B	PR13B	PR17D	PR20D	I/O
119	PR8C	PR9C	PR10C	PR11C	PR12C	PR13C	PR16A	PR19A	I/O
120	PR8D	PR9D	PR10D	PR11D	PR12D	PR13D	PR16D	PR19D	I/O
121	PR7A	PR8A	PR9A	PR10A	PR11A	PR12A	PR15A	PR18A	I/O-M3
122	PR7B	PR8B	PR9B	PR10B	PR11B	PR12B	PR15D	PR18D	I/O
123	PR7C	PR8C	PR9C	PR10C	PR11C	PR12C	PR14A	PR17A	I/O
124	PR7D	PR8D	PR9D	PR10D	PR11D	PR12D	PR14D	PR17D	I/O
125	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
126	PR6A	PR7A	PR8A	PR9A	PR10A	PR11A	PR13A	PR16A	I/O
127	PR6B	PR7B	PR8B	PR9B	PR10B	PR11B	PR13B	PR16B	I/O
128	PR6C	PR7C	PR8C	PR9C	PR10C	PR11C	PR13C	PR16C	I/O
129	PR6D	PR7D	PR8D	PR9D	PR10D	PR11D	PR13D	PR16D	I/O
130	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
131	PR5A	PR6A	PR7A	PR8A	PR9A	PR10A	PR12A	PR15A	I/O
132	PR5B	PR6B	PR7B	PR8B	PR9B	PR10B	PR12B	PR15B	I/O
133	PR5C	PR6C	PR7C	PR8C	PR9C	PR10C	PR12C	PR15C	I/O
134	PR5D	PR6D	PR7D	PR8D	PR9D	PR10D	PR12D	PR15D	I/O
135	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
136	PR4A	PR5A	PR6A	PR7A	PR8A	PR9A	PR11A	PR14A	I/O-VDD5
137	PR4B	PR5B	PR6B	PR7B	PR8B	PR9B	PR11D	PR14D	I/O
138	PR4C	PR5C	PR6C	PR7C	PR8C	PR9C	PR10A	PR13A	I/O
139	PR4D	PR5D	PR6D	PR7D	PR8D	PR9D	PR10D	PR13D	I/O
140	PR3A	PR4A	PR5A	PR6A	PR7A	PR8A	PR9A	PR12A	I/O-CS1
141	PR3B	PR4B	PR5B	PR6B	PR7B	PR8B	PR9D	PR12D	I/O
142	PR3C	PR4C	PR5C	PR6C	PR7C	PR8C	PR8A	PR11A	I/O
143	PR3D	PR4D	PR5D	PR6D	PR7D	PR8D	PR8D	PR11D	I/O
144	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
145	PR2A	PR3A	PR4A	PR5A	PR6A	PR7A	PR7A	PR10A	I/O-$\overline{CS0}$
146	PR2B	PR3B	PR4B	PR4B	PR6B	PR7B	PR7B	PR10B	I/O

Notes:
The OR2C04A and OR2T04A do not have bond pads connected to 208-pin SQFP package pin numbers 6, 45, 47, 56, 60, 102, 153, 154, 166, 201, and 203.

The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

Pin Information (continued)

Table 21. OR2C/2T04A, OR2C/2T06A, OR2C/2T08A, OR2C/2T10A, OR2C/2T12A, OR2C/2T15A, OR2C/2T26A, and OR2C/2T40A 208-Pin SQFP/SQFP2 Pinout (continued)

Pin	2C/2T04A Pad	2C/2T06A Pad	2C/2T08A Pad	2C/2T10A Pad	2C/2T12A Pad	2C/2T15A Pad	2C/2T26A Pad	2C/2T40A Pad	Function
147	PR2C	PR3C	PR4C	PR4C	PR5B	PR6B	PR6B	PR9B	I/O
148	PR2D	PR3D	PR4D	PR4D	PR5D	PR6D	PR6D	PR9D	I/O
149	PR1A	PR2A	PR3A	PR3A	PR4A	PR5A	PR5A	PR8A	I/O-$\overline{RD}$
150	PR1B	PR2C	PR3C	PR3C	PR4D	PR5D	PR5D	PR6A	I/O
151	PR1C	PR2D	PR3D	PR3D	PR3A	PR4A	PR4A	PR5A	I/O
152	PR1D	PR1A	PR2A	PR2A	PR2A	PR3A	PR3A	PR4A	I/O-$\overline{WR}$
153	See Note	PR1C	PR2D	PR2D	PR2C	PR2A	PR2A	PR3A	I/O
154	See Note	PR1D	PR1A	PR1A	PR1A	PR1A	PR1A	PR2A	I/O
155	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
156	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$
157	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
158	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
159	PT10D	PT12D	PT14D	PT16D	PT18D	PT20D	PT24D	PT30D	I/O
160	PT10C	PT12A	PT13D	PT15D	PT17D	PT19A	PT23A	PT28A	I/O-RDY/RCLK
161	PT10B	PT11D	PT13A	PT15A	PT16D	PT17D	PT21D	PT26D	I/O
162	PT10A	PT11C	PT12D	PT14D	PT16A	PT17A	PT21A	PT26A	I/O
163	PT9D	PT11A	PT12C	PT13D	PT15D	PT16D	PT20D	PT25D	I/O-D7
164	PT9C	PT10D	PT12A	PT13B	PT14D	PT15D	PT19D	PT24D	I/O-VDD5
165	PT9B	PT10C	PT11D	PT13A	PT14A	PT15A	PT19A	PT23D	I/O
166	See Note	PT10B	PT11C	PT12D	PT13D	PT14D	PT18D	PT22D	I/O
167	PT9A	PT10A	PT11B	PT12B	PT13B	PT14B	PT18B	PT21D	I/O-D6
168	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
169	PT8D	PT9D	PT10D	PT11D	PT12D	PT13D	PT17D	PT20D	I/O
170	PT8C	PT9C	PT10C	PT11C	PT12C	PT13C	PT17A	PT20A	I/O
171	PT8B	PT9B	PT10B	PT11B	PT12B	PT13B	PT16D	PT19D	I/O
172	PT8A	PT9A	PT10A	PT11A	PT12A	PT13A	PT16A	PT19A	I/O-D5
173	PT7D	PT8D	PT9D	PT10D	PT11D	PT12D	PT15D	PT18D	I/O
174	PT7C	PT8C	PT9C	PT10C	PT11C	PT12C	PT15A	PT18A	I/O
175	PT7B	PT8B	PT9B	PT10B	PT11B	PT12B	PT14D	PT17D	I/O
176	PT7A	PT8A	PT9A	PT10A	PT11A	PT12A	PT14A	PT17A	I/O-D4
177	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
178	PT6D	PT7D	PT8D	PT9D	PT10D	PT11D	PT13D	PT16D	I/O
179	PT6C	PT7C	PT8C	PT9C	PT10C	PT11C	PT13C	PT16C	I/O
180	PT6B	PT7B	PT8B	PT9B	PT10B	PT11B	PT13B	PT16B	I/O
181	PT6A	PT7A	PT8A	PT9A	PT10A	PT11A	PT13A	PT16A	I/O-D3
182	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS

Notes:
The OR2C04A and OR2T04A do not have bond pads connected to 208-pin SQFP package pin numbers 6, 45, 47, 56, 60, 102, 153, 154, 166, 201, and 203.

The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

Pin Information (continued)

Table 21. OR2C/2T04A, OR2C/2T06A, OR2C/2T08A, OR2C/2T10A, OR2C/2T12A, OR2C/2T15A, OR2C/2T26A, and OR2C/2T40A 208-Pin SQFP/SQFP2 Pinout (continued)

Pin	2C/2T04A Pad	2C/2T06A Pad	2C/2T08A Pad	2C/2T10A Pad	2C/2T12A Pad	2C/2T15A Pad	2C/2T26A Pad	2C/2T40A Pad	Function
183	PT5D	PT6D	PT7D	PT8D	PT9D	PT10D	PT12D	PT15D	I/O
184	PT5C	PT6C	PT7C	PT8C	PT9C	PT10C	PT12C	PT15C	I/O
185	PT5B	PT6B	PT7B	PT8B	PT9B	PT10B	PT12B	PT15B	I/O-VDD5
186	PT5A	PT6A	PT7A	PT8A	PT9A	PT10A	PT12A	PT15A	I/O-D2
187	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
188	PT4D	PT5D	PT6D	PT7D	PT8D	PT9D	PT11D	PT14D	I/O-D1
189	PT4C	PT5C	PT6C	PT7C	PT8C	PT9C	PT11A	PT14A	I/O
190	PT4B	PT5B	PT6B	PT7B	PT8B	PT9B	PT10D	PT13D	I/O
191	PT4A	PT5A	PT6A	PT7A	PT8A	PT9A	PT10A	PT13A	I/O-D0/DIN
192	PT3D	PT4D	PT5D	PT6D	PT7D	PT8D	PT9D	PT12D	I/O
193	PT3C	PT4C	PT5C	PT6C	PT7C	PT8C	PT9A	PT12A	I/O
194	PT3B	PT4B	PT5B	PT6B	PT7B	PT8B	PT8D	PT11D	I/O
195	PT3A	PT4A	PT5A	PT6A	PT7A	PT8A	PT8A	PT11A	I/O-DOUT
196	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
197	PT2D	PT3D	PT4D	PT5D	PT6D	PT7D	PT7D	PT10D	I/O
198	PT2C	PT3C	PT4C	PT5A	PT6A	PT7A	PT7A	PT9A	I/O
199	PT2B	PT3B	PT4B	PT4D	PT5C	PT6C	PT6C	PT8A	I/O
200	PT2A	PT3A	PT4A	PT4A	PT5A	PT6A	PT6A	PT7A	I/O-TDI
201	See Note	PT2D	PT3D	PT3D	PT4A	PT5A	PT5A	PT6A	I/O
202	PT1D	PT2A	PT3A	PT3A	PT3A	PT4A	PT4A	PT5A	I/O-TMS
203	See Note	PT1D	PT2D	PT2D	PT2C	PT3A	PT3A	PT4A	I/O
204	PT1C	PT1C	PT2A	PT2A	PT2A	PT2A	PT2A	PT3A	I/O
205	PT1B	PT1B	PT1D	PT1D	PT1D	PT1D	PT1D	PT2D	I/O
206	PT1A	PT1A	PT1A	PT1A	PT1A	PT1A	PT1A	PT1A	I/O-TCK
207	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
208	RD_DATA/ TDO	RD_DATA/ TDO	RD_DATA/ TDO	RD_DATA/ TDO	RD_DATA/ TDO	RD_DATA/ TDO	RD_DATA/ TDO	RD_DATA/ TDO	RD_DATA/TDO

Notes:
The OR2C04A and OR2T04A do not have bond pads connected to 208-pin SQFP package pin numbers 6, 45, 47, 56, 60, 102, 153, 154, 166, 201, and 203.

The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

Pin Information (continued)

Table 22. OR2C/2T06A, OR2C/2T08A, OR2C/2T10A, OR2C/2T12A, OR2C/2T15A, OR2C/2T26A, and OR2C/2T40A 240-Pin SQFP/SQFP2 Pinout

Pin	2C/2T06A Pad	2C/2T08A Pad	2C/2T10A Pad	2C/2T12A Pad	2C/2T15A Pad	2C/2T26A Pad	2C/2T40A Pad	Function
1	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
2	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
3	PL1D	PL1D	PL1D	PL1D	PL1D	PL1D	PL1D	I/O
4	PL1C	PL1B	PL1B	PL1C	PL1C	PL1C	PL1A	I/O
5	PL1B	PL1A	PL1A	PL1B	PL1B	PL1B	PL2D	I/O
6	PL1A	PL2D	PL2D	PL2D	PL2D	PL2D	PL3D	I/O-A0
7	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
8	PL2D	PL3D	PL3D	PL3D	PL4D	PL4D	PL5D	I/O-VDD5
9	PL2C	PL3C	PL3C	PL3A	PL4A	PL4A	PL6D	I/O
10	PL2B	PL3B	PL3B	PL4D	PL5D	PL5D	PL7D	I/O
11	PL2A	PL3A	PL3A	PL4A	PL5A	PL5A	PL8D	I/O-A1
12	PL3D	PL4D	PL4A	PL5A	PL6A	PL6A	PL9A	I/O-A2
13	PL3C	PL4C	PL5C	PL6D	PL7D	PL7D	PL10D	I/O
14	PL3B	PL4B	PL5B	PL6B	PL7B	PL7B	PL10B	I/O
15	PL3A	PL4A	PL5A	PL6A	PL7A	PL7A	PL10A	I/O-A3
16	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
17	PL4D	PL5D	PL6D	PL7D	PL8D	PL8D	PL11D	I/O
18	PL4C	PL5C	PL6C	PL7C	PL8C	PL8A	PL11A	I/O
19	PL4B	PL5B	PL6B	PL7B	PL8B	PL9D	PL12D	I/O
20	PL4A	PL5A	PL6A	PL7A	PL8A	PL9A	PL12A	I/O-A4
21	PL5D	PL6D	PL7D	PL8D	PL9D	PL10D	PL13D	I/O-A5
22	PL5C	PL6C	PL7C	PL8C	PL9C	PL10A	PL13A	I/O
23	PL5B	PL6B	PL7B	PL8B	PL9B	PL11D	PL14D	I/O
24	PL5A	PL6A	PL7A	PL8A	PL9A	PL11A	PL14A	I/O-A6
25	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
26	PL6D	PL7D	PL8D	PL9D	PL10D	PL12D	PL15D	I/O
27	PL6C	PL7C	PL8C	PL9C	PL10C	PL12C	PL15C	I/O
28	PL6B	PL7B	PL8B	PL9B	PL10B	PL12B	PL15B	I/O
29	PL6A	PL7A	PL8A	PL9A	PL10A	PL12A	PL15A	I/O-A7
30	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
31	PL7D	PL8D	PL9D	PL10D	PL11D	PL13D	PL16D	I/O
32	PL7C	PL8C	PL9C	PL10C	PL11C	PL13C	PL16C	I/O-VDD5
33	PL7B	PL8B	PL9B	PL10B	PL11B	PL13B	PL16B	I/O
34	PL7A	PL8A	PL9A	PL10A	PL11A	PL13A	PL16A	I/O-A8
35	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
36	PL8D	PL9D	PL10D	PL11D	PL12D	PL14D	PL17D	I/O-A9
37	PL8C	PL9C	PL10C	PL11C	PL12C	PL14A	PL17A	I/O
38	PL8B	PL9B	PL10B	PL11B	PL12B	PL15D	PL18D	I/O
39	PL8A	PL9A	PL10A	PL11A	PL12A	PL15A	PL18A	I/O-A10
40	PL9D	PL10D	PL11D	PL12D	PL13D	PL16D	PL19D	I/O
41	PL9C	PL10C	PL11C	PL12C	PL13C	PL16A	PL19A	I/O

Notes:
The OR2C/2T08A and OR2C/2T10A do not have bond pads connected to 240-pin SQFP package pin numbers 113 and 188.

The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

Pin Information (continued)

Table 22. OR2C/2T06A, OR2C/2T08A, OR2C/2T10A, OR2C/2T12A, OR2C/2T15A, OR2C/2T26A, and OR2C/2T40A 240-Pin SQFP/SQFP2 Pinout (continued)

Pin	2C/2T06A Pad	2C/2T08A Pad	2C/2T10A Pad	2C/2T12A Pad	2C/2T15A Pad	2C/2T26A Pad	2C/2T40A Pad	Function
42	PL9B	PL10B	PL11B	PL12B	PL13B	PL17D	PL20D	I/O
43	PL9A	PL10A	PL11A	PL12A	PL13A	PL17A	PL20A	I/O-A11
44	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
45	PL10D	PL11D	PL12D	PL13D	PL14D	PL18D	PL21D	I/O-A12
46	PL10C	PL11C	PL12C	PL13B	PL14B	PL18B	PL21B	I/O
47	PL10B	PL11B	PL12B	PL14D	PL15D	PL19D	PL22D	I/O
48	PL10A	PL11A	PL13D	PL14B	PL15B	PL19B	PL22B	I/O-A13
49	PL11D	PL12D	PL13B	PL14A	PL15A	PL19A	PL22A	I/O
50	PL11C	PL12C	PL13A	PL15D	PL16D	PL20D	PL23D	I/O
51	PL11B	PL12B	PL14D	PL15B	PL16B	PL20B	PL24D	I/O
52	PL11A	PL12A	PL14C	PL16D	PL17D	PL21D	PL25A	I/O-A14
53	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
54	PL12D	PL13D	PL15D	PL17D	PL18D	PL22D	PL27D	I/O
55	PL12C	PL13A	PL15A	PL17A	PL19D	PL23D	PL28D	I/O
56	PL12B	PL14D	PL16D	PL18C	PL19A	PL23A	PL28A	I/O
57	PL12A	PL14A	PL16A	PL18A	PL20A	PL24A	PL30A	I/O-A15
58	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
59	CCLK	CCLK	CCLK	CCLK	CCLK	CCLK	CCLK	CCLK
60	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
61	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
62	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
63	PB1A	PB1A	PB1A	PB1A	PB1A	PB1A	PB1A	I/O-A16
64	PB1B	PB1D	PB1D	PB1D	PB2A	PB2A	PB3A	I/O
65	PB1C	PB2A	PB2A	PB2A	PB2D	PB2D	PB3D	I/O-VDD5
66	PB1D	PB2D	PB2D	PB2D	PB3D	PB3D	PB4D	I/O
67	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
68	PB2A	PB3A	PB3B	PB3D	PB4D	PB4D	PB5D	I/O-A17
69	PB2B	PB3B	PB4B	PB4D	PB5D	PB5D	PB6D	I/O
70	PB2C	PB3C	PB4C	PB5A	PB6A	PB6A	PB7A	I/O
71	PB2D	PB3D	PB4D	PB5B	PB6B	PB6B	PB7D	I/O
72	PB3A	PB4A	PB5A	PB5D	PB6D	PB6D	PB8D	I/O
73	PB3B	PB4B	PB5B	PB6A	PB7A	PB7A	PB9A	I/O
74	PB3C	PB4C	PB5C	PB6B	PB7B	PB7B	PB9D	I/O
75	PB3D	PB4D	PB5D	PB6D	PB7D	PB7D	PB10D	I/O
76	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
77	PB4A	PB5A	PB6A	PB7A	PB8A	PB8A	PB11A	I/O
78	PB4B	PB5B	PB6B	PB7B	PB8B	PB8D	PB11D	I/O
79	PB4C	PB5C	PB6C	PB7C	PB8C	PB9A	PB12A	I/O
80	PB4D	PB5D	PB6D	PB7D	PB8D	PB9D	PB12D	I/O
81	PB5A	PB6A	PB7A	PB8A	PB9A	PB10A	PB13A	I/O
82	PB5B	PB6B	PB7B	PB8B	PB9B	PB10D	PB13D	I/O

Notes:
The OR2C/2T08A and OR2C/2T10A do not have bond pads connected to 240-pin SQFP package pin numbers 113 and 188.

The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

Pin Information (continued)

Table 22. OR2C/2T06A, OR2C/2T08A, OR2C/2T10A, OR2C/2T12A, OR2C/2T15A, OR2C/2T26A, and OR2C/2T40A 240-Pin SQFP/SQFP2 Pinout (continued)

Pin	2C/2T06A Pad	2C/2T08A Pad	2C/2T10A Pad	2C/2T12A Pad	2C/2T15A Pad	2C/2T26A Pad	2C/2T40A Pad	Function
83	PB5C	PB6C	PB7C	PB8C	PB9C	PB11A	PB14A	I/O
84	PB5D	PB6D	PB7D	PB8D	PB9D	PB11D	PB14D	I/O
85	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
86	PB6A	PB7A	PB8A	PB9A	PB10A	PB12A	PB15A	I/O
87	PB6B	PB7B	PB8B	PB9B	PB10B	PB12B	PB15B	I/O
88	PB6C	PB7C	PB8C	PB9C	PB10C	PB12C	PB15C	I/O
89	PB6D	PB7D	PB8D	PB9D	PB10D	PB12D	PB15D	I/O
90	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
91	PB7A	PB8A	PB9A	PB10A	PB11A	PB13A	PB16A	I/O
92	PB7B	PB8B	PB9B	PB10B	PB11B	PB13B	PB16B	I/O
93	PB7C	PB8C	PB9C	PB10C	PB11C	PB13C	PB16C	I/O
94	PB7D	PB8D	PB9D	PB10D	PB11D	PB13D	PB16D	I/O
95	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
96	PB8A	PB9A	PB10A	PB11A	PB12A	PB14A	PB17A	I/O-VDD5
97	PB8B	PB9B	PB10B	PB11B	PB12B	PB14D	PB17D	I/O
98	PB8C	PB9C	PB10C	PB11C	PB12C	PB15A	PB18A	I/O
99	PB8D	PB9D	PB10D	PB11D	PB12D	PB15D	PB18D	I/O
100	PB9A	PB10A	PB11A	PB12A	PB13A	PB16A	PB19A	I/O-HDC
101	PB9B	PB10B	PB11B	PB12B	PB13B	PB16D	PB19D	I/O
102	PB9C	PB10C	PB11C	PB12C	PB13C	PB17A	PB20A	I/O
103	PB9D	PB10D	PB11D	PB12D	PB13D	PB17D	PB20D	I/O
104	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
105	PB10A	PB11A	PB12A	PB13A	PB14A	PB18A	PB21A	I/O-$\overline{\text{LDC}}$
106	PB10B	PB11D	PB13A	PB13D	PB14D	PB18D	PB22D	I/O
107	PB10C	PB12A	PB13B	PB14A	PB15A	PB19A	PB23A	I/O
108	PB10D	PB12B	PB13C	PB14D	PB15D	PB19D	PB24D	I/O
109	PB11A	PB12C	PB13D	PB15A	PB16A	PB20A	PB25A	I/O-$\overline{\text{INIT}}$
110	PB11B	PB12D	PB14A	PB15D	PB16D	PB20D	PB25D	I/O
111	PB11C	PB13A	PB15A	PB16A	PB17A	PB21A	PB26A	I/O
112	PB11D	PB13B	PB15B	PB16D	PB17D	PB21D	PB26D	I/O
113	VSS	See Note	See Note	VSS	VSS	VSS	VSS	VSS
114	PB12A	PB13D	PB15D	PB17A	PB18A	PB22A	PB27A	I/O
115	PB12B	PB14A	PB16A	PB17D	PB19A	PB23A	PB28A	I/O
116	PB12C	PB14B	PB16B	PB18A	PB19D	PB23D	PB28D	I/O
117	PB12D	PB14D	PB16D	PB18D	PB20D	PB24D	PB30D	I/O
118	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
119	DONE	DONE	DONE	DONE	DONE	DONE	DONE	DONE
120	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
121	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
122	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$
123	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$

Notes:
The OR2C/2T08A and OR2C/2T10A do not have bond pads connected to 240-pin SQFP package pin numbers 113 and 188.

The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

Pin Information (continued)

Table 22. OR2C/2T06A, OR2C/2T08A, OR2C/2T10A, OR2C/2T12A, OR2C/2T15A, OR2C/2T26A, and OR2C/2T40A 240-Pin SQFP/SQFP2 Pinout (continued)

Pin	2C/2T06A Pad	2C/2T08A Pad	2C/2T10A Pad	2C/2T12A Pad	2C/2T15A Pad	2C/2T26A Pad	2C/2T40A Pad	Function
124	PR12A	PR14A	PR16A	PR18A	PR20A	PR24A	PR30A	I/O-M0
125	PR12B	PR14D	PR16D	PR18C	PR20D	PR24D	PR29D	I/O
126	PR12C	PR13A	PR15A	PR18D	PR19A	PR23A	PR28A	I/O
127	PR12D	PR13D	PR15D	PR17B	PR18A	PR22A	PR27A	I/O
128	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
129	PR11A	PR12A	PR14A	PR16A	PR17A	PR21A	PR26A	I/O
130	PR11B	PR12B	PR14C	PR16D	PR17D	PR21D	PR25A	I/O
131	PR11C	PR12C	PR14D	PR15A	PR16A	PR20A	PR24A	I/O
132	PR11D	PR12D	PR13A	PR15C	PR16C	PR20C	PR24D	I/O
133	PR10A	PR11A	PR13B	PR15D	PR16D	PR20D	PR23D	I/O-M1
134	PR10B	PR11B	PR13C	PR14A	PR15A	PR19A	PR22A	I/O
135	PR10C	PR11C	PR12A	PR14D	PR15D	PR19D	PR22D	I/O-VDD5
136	PR10D	PR11D	PR12B	PR13A	PR14A	PR18A	PR21A	I/O
137	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
138	PR9A	PR10A	PR11A	PR12A	PR13A	PR17A	PR20A	I/O-M2
139	PR9B	PR10B	PR11B	PR12B	PR13B	PR17D	PR20D	I/O
140	PR9C	PR10C	PR11C	PR12C	PR13C	PR16A	PR19A	I/O
141	PR9D	PR10D	PR11D	PR12D	PR13D	PR16D	PR19D	I/O
142	PR8A	PR9A	PR10A	PR11A	PR12A	PR15A	PR18A	I/O-M3
143	PR8B	PR9B	PR10B	PR11B	PR12B	PR15D	PR18D	I/O
144	PR8C	PR9C	PR10C	PR11C	PR12C	PR14A	PR17A	I/O
145	PR8D	PR9D	PR10D	PR11D	PR12D	PR14D	PR17D	I/O
146	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
147	PR7A	PR8A	PR9A	PR10A	PR11A	PR13A	PR16A	I/O
148	PR7B	PR8B	PR9B	PR10B	PR11B	PR13B	PR16B	I/O
149	PR7C	PR8C	PR9C	PR10C	PR11C	PR13C	PR16C	I/O
150	PR7D	PR8D	PR9D	PR10D	PR11D	PR13D	PR16D	I/O
151	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
152	PR6A	PR7A	PR8A	PR9A	PR10A	PR12A	PR15A	I/O
153	PR6B	PR7B	PR8B	PR9B	PR10B	PR12B	PR15B	I/O
154	PR6C	PR7C	PR8C	PR9C	PR10C	PR12C	PR15C	I/O
155	PR6D	PR7D	PR8D	PR9D	PR10D	PR12D	PR15D	I/O
156	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
157	PR5A	PR6A	PR7A	PR8A	PR9A	PR11A	PR14A	I/O-VDD5
158	PR5B	PR6B	PR7B	PR8B	PR9B	PR11D	PR14D	I/O
159	PR5C	PR6C	PR7C	PR8C	PR9C	PR10A	PR13A	I/O
160	PR5D	PR6D	PR7D	PR8D	PR9D	PR10D	PR13D	I/O
161	PR4A	PR5A	PR6A	PR7A	PR8A	PR9A	PR12A	I/O-CS1
162	PR4B	PR5B	PR6B	PR7B	PR8B	PR9D	PR12D	I/O
163	PR4C	PR5C	PR6C	PR7C	PR8C	PR8A	PR11A	I/O
164	PR4D	PR5D	PR6D	PR7D	PR8D	PR8D	PR11D	I/O

Notes:
The OR2C/2T08A and OR2C/2T10A do not have bond pads connected to 240-pin SQFP package pin numbers 113 and 188.

The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

Pin Information (continued)

Table 22. OR2C/2T06A, OR2C/2T08A, OR2C/2T10A, OR2C/2T12A, OR2C/2T15A, OR2C/2T26A, and OR2C/2T40A 240-Pin SQFP/SQFP2 Pinout (continued)

Pin	2C/2T06A Pad	2C/2T08A Pad	2C/2T10A Pad	2C/2T12A Pad	2C/2T15A Pad	2C/2T26A Pad	2C/2T40A Pad	Function
165	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
166	PR3A	PR4A	PR5A	PR6A	PR7A	PR7A	PR10A	I/O-$\overline{CS0}$
167	PR3B	PR4B	PR4B	PR6B	PR7B	PR7B	PR10B	I/O
168	PR3C	PR4C	PR4C	PR5B	PR6B	PR6B	PR9B	I/O
169	PR3D	PR4D	PR4D	PR5D	PR6D	PR6D	PR9D	I/O
170	PR2A	PR3A	PR3A	PR4A	PR5A	PR5A	PR8A	I/O-$\overline{RD}$
171	PR2B	PR3B	PR3B	PR4B	PR5B	PR5B	PR7A	I/O
172	PR2C	PR3C	PR3C	PR4D	PR5D	PR5D	PR6A	I/O
173	PR2D	PR3D	PR3D	PR3A	PR4A	PR4A	PR5A	I/O
174	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
175	PR1A	PR2A	PR2A	PR2A	PR3A	PR3A	PR4A	I/O-$\overline{WR}$
176	PR1B	PR2D	PR2D	PR2C	PR2A	PR2A	PR3A	I/O
177	PR1C	PR1A	PR1A	PR1A	PR1A	PR1A	PR2A	I/O
178	PR1D	PR1D	PR1D	PR1D	PR1D	PR1D	PR1D	I/O
179	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
180	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$
181	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
182	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
183	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
184	PT12D	PT14D	PT16D	PT18D	PT20D	PT24D	PT30D	I/O
185	PT12C	PT14C	PT16C	PT18B	PT20A	PT24A	PT29A	I/O
186	PT12B	PT14A	PT16A	PT18A	PT19D	PT23D	PT28D	I/O
187	PT12A	PT13D	PT15D	PT17D	PT19A	PT23A	PT28A	I/O-RDY/RCLK
188	VSS	See Note	See Note	VSS	VSS	VSS	VSS	VSS
189	PT11D	PT13B	PT15B	PT16D	PT17D	PT21D	PT26D	I/O
190	PT11C	PT13A	PT15A	PT16C	PT17C	PT21C	PT26C	I/O
191	PT11B	PT12D	PT14D	PT16A	PT17A	PT21A	PT26A	I/O
192	PT11A	PT12C	PT13D	PT15D	PT16D	PT20D	PT25D	I/O-D7
193	PT10D	PT12A	PT13B	PT14D	PT15D	PT19D	PT24D	I/O-VDD5
194	PT10C	PT11D	PT13A	PT14A	PT15A	PT19A	PT23D	I/O
195	PT10B	PT11C	PT12D	PT13D	PT14D	PT18D	PT22D	I/O
196	PT10A	PT11B	PT12B	PT13B	PT14B	PT18B	PT21D	I/O-D6
197	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
198	PT9D	PT10D	PT11D	PT12D	PT13D	PT17D	PT20D	I/O
199	PT9C	PT10C	PT11C	PT12C	PT13C	PT17A	PT20A	I/O
200	PT9B	PT10B	PT11B	PT12B	PT13B	PT16D	PT19D	I/O
201	PT9A	PT10A	PT11A	PT12A	PT13A	PT16A	PT19A	I/O-D5
202	PT8D	PT9D	PT10D	PT11D	PT12D	PT15D	PT18D	I/O
203	PT8C	PT9C	PT10C	PT11C	PT12C	PT15A	PT18A	I/O
204	PT8B	PT9B	PT10B	PT11B	PT12B	PT14D	PT17D	I/O
205	PT8A	PT9A	PT10A	PT11A	PT12A	PT14A	PT17A	I/O-D4

Notes:
The OR2C/2T08A and OR2C/2T10A do not have bond pads connected to 240-pin SQFP package pin numbers 113 and 188.

The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

Pin Information (continued)

Table 22. OR2C/2T06A, OR2C/2T08A, OR2C/2T10A, OR2C/2T12A, OR2C/2T15A, OR2C/2T26A, and OR2C/2T40A 240-Pin SQFP/SQFP2 Pinout (continued)

Pin	2C/2T06A Pad	2C/2T08A Pad	2C/2T10A Pad	2C/2T12A Pad	2C/2T15A Pad	2C/2T26A Pad	2C/2T40A Pad	Function
206	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
207	PT7D	PT8D	PT9D	PT10D	PT11D	PT13D	PT16D	I/O
208	PT7C	PT8C	PT9C	PT10C	PT11C	PT13C	PT16C	I/O
209	PT7B	PT8B	PT9B	PT10B	PT11B	PT13B	PT16B	I/O
210	PT7A	PT8A	PT9A	PT10A	PT11A	PT13A	PT16A	I/O-D3
211	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
212	PT6D	PT7D	PT8D	PT9D	PT10D	PT12D	PT15D	I/O
213	PT6C	PT7C	PT8C	PT9C	PT10C	PT12C	PT15C	I/O
214	PT6B	PT7B	PT8B	PT9B	PT10B	PT12B	PT15B	I/O-VDD5
215	PT6A	PT7A	PT8A	PT9A	PT10A	PT12A	PT15A	I/O-D2
216	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
217	PT5D	PT6D	PT7D	PT8D	PT9D	PT11D	PT14D	I/O-D1
218	PT5C	PT6C	PT7C	PT8C	PT9C	PT11A	PT14A	I/O
219	PT5B	PT6B	PT7B	PT8B	PT9B	PT10D	PT13D	I/O
220	PT5A	PT6A	PT7A	PT8A	PT9A	PT10A	PT13A	I/O-D0/DIN
221	PT4D	PT5D	PT6D	PT7D	PT8D	PT9D	PT12D	I/O
222	PT4C	PT5C	PT6C	PT7C	PT8C	PT9A	PT12A	I/O
223	PT4B	PT5B	PT6B	PT7B	PT8B	PT8D	PT11D	I/O
224	PT4A	PT5A	PT6A	PT7A	PT8A	PT8A	PT11A	I/O-DOUT
225	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
226	PT3D	PT4D	PT5D	PT6D	PT7D	PT7D	PT10D	I/O
227	PT3C	PT4C	PT5A	PT6A	PT7A	PT7A	PT9A	I/O
228	PT3B	PT4B	PT4D	PT5C	PT6C	PT6C	PT8A	I/O
229	PT3A	PT4A	PT4A	PT5A	PT6A	PT6A	PT7A	I/O-TDI
230	PT2D	PT3D	PT3D	PT4D	PT5D	PT5D	PT6D	I/O
231	PT2C	PT3C	PT3C	PT4A	PT5A	PT5A	PT6A	I/O
232	PT2B	PT3B	PT3B	PT3D	PT4D	PT4D	PT5D	I/O
233	PT2A	PT3A	PT3A	PT3A	PT4A	PT4A	PT5A	I/O-TMS
234	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
235	PT1D	PT2D	PT2D	PT2C	PT3A	PT3A	PT4A	I/O
236	PT1C	PT2A	PT2A	PT2A	PT2A	PT2A	PT3A	I/O
237	PT1B	PT1D	PT1D	PT1D	PT1D	PT1D	PT2D	I/O
238	PT1A	PT1A	PT1A	PT1A	PT1A	PT1A	PT1A	I/O-TCK
239	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
240	RD_DATA/ TDO	RD_DATA/ TDO	RD_DATA/ TDO	RD_DATA/ TDO	RD_DATA/ TDO	RD_DATA/ TDO	RD_DATA/ TDO	RD_DATA/TDO

Notes:
The OR2C/2T08A and OR2C/2T10A do not have bond pads connected to 240-pin SQFP package pin numbers 113 and 188.

The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

Pin Information (continued)

Table 23. OR2C/2T06A, OR2C/2T08A, OR2C/2T10A, OR2C/2T12A, and OR2C/2T15A 256-Pin PBGA Pinout

Pin	2C/2T06A Pad	2C/2T08A Pad	2C/2T10A Pad	2C/2T12A Pad	2C/2T15A Pad	Function
C2	PL1D	PL1D	PL1D	PL1D	PL1D	I/O
D2	PL1C	PL1B	PL1B	PL1C	PL1C	I/O
D3	PL1B	PL1A	PL1A	PL1B	PL1B	I/O
E4	PL1A	PL2D	PL2D	PL2D	PL2D	I/O-A0
C1	—	PL2C	PL2C	PL2C	PL2A	I/O
D1	—	PL2B	PL2B	PL2B	PL3D	I/O
E3	—	PL2A	PL2A	PL2A	PL3A	I/O
E2	PL2D	PL3D	PL3D	PL3D	PL4D	I/O-VDD5
E1	PL2C	PL3C	PL3C	PL3A	PL4A	I/O
F3	PL2B	PL3B	PL3B	PL4D	PL5D	I/O
G4	PL2A	PL3A	PL3A	PL4A	PL5A	I/O-A1
F2	—	—	PL4D	PL5D	PL6D	I/O
F1	PL3D	PL4D	PL4A	PL5A	PL6A	I/O-A2
G3	PL3C	PL4C	PL5C	PL6D	PL7D	I/O
G2	PL3B	PL4B	PL5B	PL6B	PL7B	I/O
G1	PL3A	PL4A	PL5A	PL6A	PL7A	I/O-A3
H3	PL4D	PL5D	PL6D	PL7D	PL8D	I/O
H2	PL4C	PL5C	PL6C	PL7C	PL8C	I/O
H1	PL4B	PL5B	PL6B	PL7B	PL8B	I/O
J4	PL4A	PL5A	PL6A	PL7A	PL8A	I/O-A4
J3	PL5D	PL6D	PL7D	PL8D	PL9D	I/O-A5
J2	PL5C	PL6C	PL7C	PL8C	PL9C	I/O
J1	PL5B	PL6B	PL7B	PL8B	PL9B	I/O
K2	PL5A	PL6A	PL7A	PL8A	PL9A	I/O-A6
K3	PL6D	PL7D	PL8D	PL9D	PL10D	I/O
K1	PL6C	PL7C	PL8C	PL9C	PL10C	I/O
L1	PL6B	PL7B	PL8B	PL9B	PL10B	I/O
L2	PL6A	PL7A	PL8A	PL9A	PL10A	I/O-A7
L3	PL7D	PL8D	PL9D	PL10D	PL11D	I/O
L4	PL7C	PL8C	PL9C	PL10C	PL11C	I/O-VDD5
M1	PL7B	PL8B	PL9B	PL10B	PL11B	I/O
M2	PL7A	PL8A	PL9A	PL10A	PL11A	I/O-A8
M3	PL8D	PL9D	PL10D	PL11D	PL12D	I/O-A9
M4	PL8C	PL9C	PL10C	PL11C	PL12C	I/O
N1	PL8B	PL9B	PL10B	PL11B	PL12B	I/O
N2	PL8A	PL9A	PL10A	PL11A	PL12A	I/O-A10
N3	PL9D	PL10D	PL11D	PL12D	PL13D	I/O
P1	PL9C	PL10C	PL11C	PL12C	PL13C	I/O
P2	PL9B	PL10B	PL11B	PL12B	PL13B	I/O

Notes:

The W3 pin on the 256-pin PBGA package is unconnected for all devices listed in this table.

The OR2C/2T08A do not have bond pads connected to the 256-pin PBGA package pins F2 and Y17.

The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

The pins labeled "VSS-ETC" are the 4 x 4 array of thermal balls located at the center of the package. The balls can be attached to the ground plane of the board for Enhanced Thermal Capability (see Table 28), or they can be left unconnected.

Pin Information (continued)

Table 23. OR2C/2T06A, OR2C/2T08A, OR2C/2T10A, OR2C/2T12A, and OR2C/2T15A 256-Pin PBGA Pinout (continued)

Pin	2C/2T06A Pad	2C/2T08A Pad	2C/2T10A Pad	2C/2T12A Pad	2C/2T15A Pad	Function
R1	PL9A	PL10A	PL11A	PL12A	PL13A	I/O-A11
P3	PL10D	PL11D	PL12D	PL13D	PL14D	I/O-A12
R2	PL10C	PL11C	PL12C	PL13B	PL14B	I/O
T1	PL10B	PL11B	PL12B	PL14D	PL15D	I/O
P4	PL10A	PL11A	PL13D	PL14B	PL15B	I/O-A13
R3	PL11D	PL12D	PL13B	PL14A	PL15A	I/O
T2	PL11C	PL12C	PL13A	PL15D	PL16D	I/O
U1	PL11B	PL12B	PL14D	PL15B	PL16B	I/O
T3	PL11A	PL12A	PL14C	PL16D	PL17D	I/O-A14
U2	—	PL13D	PL15D	PL17D	PL18D	I/O-VDD5
V1	PL12D	PL13C	PL15C	PL17C	PL18C	I/O
T4	PL12C	PL13B	PL15B	PL17B	PL18A	I/O
U3	PL12B	PL13A	PL15A	PL17A	PL19D	I/O
V2	—	PL14D	PL16D	PL18D	PL19C	I/O
W1	—	PL14C	PL16C	PL18C	PL19A	I/O
V3	—	PL14B	PL16B	PL18B	PL20D	I/O
W2	PL12A	PL14A	PL16A	PL18A	PL20A	I/O-A15
Y1	CCLK	CCLK	CCLK	CCLK	CCLK	CCLK
Y2	PB1A	PB1A	PB1A	PB1A	PB1A	I/O-A16
W4	—	PB1C	PB1C	PB1C	PB1D	I/O
V4	PB1B	PB1D	PB1D	PB1D	PB2A	I/O
U5	PB1C	PB2A	PB2A	PB2A	PB2D	I/O-VDD5
Y3	PB1D	PB2B	PB2B	PB2B	PB3A	I/O
Y4	—	PB2C	PB2C	PB2C	PB3C	I/O
V5	—	PB2D	PB2D	PB2D	PB3D	I/O
W5	PB2A	PB3A	PB3B	PB3D	PB4D	I/O-A17
Y5	PB2B	PB3B	PB4B	PB4D	PB5D	I/O
V6	PB2C	PB3C	PB4C	PB5A	PB6A	I/O
U7	PB2D	PB3D	PB4D	PB5B	PB6B	I/O
W6	PB3A	PB4A	PB5A	PB5D	PB6D	I/O
Y6	PB3B	PB4B	PB5B	PB6A	PB7A	I/O
V7	PB3C	PB4C	PB5C	PB6B	PB7B	I/O
W7	PB3D	PB4D	PB5D	PB6D	PB7D	I/O
Y7	PB4A	PB5A	PB6A	PB7A	PB8A	I/O
V8	PB4B	PB5B	PB6B	PB7B	PB8B	I/O
W8	PB4C	PB5C	PB6C	PB7C	PB8C	I/O
Y8	PB4D	PB5D	PB6D	PB7D	PB8D	I/O
U9	PB5A	PB6A	PB7A	PB8A	PB9A	I/O
V9	PB5B	PB6B	PB7B	PB8B	PB9B	I/O

Notes:

The W3 pin on the 256-pin PBGA package is unconnected for all devices listed in this table.

The OR2C/2T08A do not have bond pads connected to the 256-pin PBGA package pins F2 and Y17.

The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

The pins labeled "VSS-ETC" are the 4 x 4 array of thermal balls located at the center of the package. The balls can be attached to the ground plane of the board for Enhanced Thermal Capability (see Table 28), or they can be left unconnected.

Pin Information (continued)

Table 23. OR2C/2T06A, OR2C/2T08A, OR2C/2T10A, OR2C/2T12A, and OR2C/2T15A 256-Pin PBGA Pinout (continued)

Pin	2C/2T06A Pad	2C/2T08A Pad	2C/2T10A Pad	2C/2T12A Pad	2C/2T15A Pad	Function
W9	PB5C	PB6C	PB7C	PB8C	PB9C	I/O
Y9	PB5D	PB6D	PB7D	PB8D	PB9D	I/O
W10	PB6A	PB7A	PB8A	PB9A	PB10A	I/O
V10	PB6B	PB7B	PB8B	PB9B	PB10B	I/O
Y10	PB6C	PB7C	PB8C	PB9C	PB10C	I/O
Y11	PB6D	PB7D	PB8D	PB9D	PB10D	I/O
W11	PB7A	PB8A	PB9A	PB10A	PB11A	I/O
V11	PB7B	PB8B	PB9B	PB10B	PB11B	I/O
U11	PB7C	PB8C	PB9C	PB10C	PB11C	I/O
Y12	PB7D	PB8D	PB9D	PB10D	PB11D	I/O
W12	PB8A	PB9A	PB10A	PB11A	PB12A	I/O-$V_{DD}5$
V12	PB8B	PB9B	PB10B	PB11B	PB12B	I/O
U12	PB8C	PB9C	PB10C	PB11C	PB12C	I/O
Y13	PB8D	PB9D	PB10D	PB11D	PB12D	I/O
W13	PB9A	PB10A	PB11A	PB12A	PB13A	I/O-HDC
V13	PB9B	PB10B	PB11B	PB12B	PB13B	I/O
Y14	PB9C	PB10C	PB11C	PB12C	PB13C	I/O
W14	PB9D	PB10D	PB11D	PB12D	PB13D	I/O
Y15	PB10A	PB11A	PB12A	PB13A	PB14A	I/O-$\overline{LDC}$
V14	PB10B	PB11B	PB12C	PB13B	PB14B	I/O
W15	PB10C	PB11C	PB12D	PB13C	PB14C	I/O
Y16	PB10D	PB11D	PB13A	PB13D	PB14D	I/O
U14	—	PB12A	PB13B	PB14A	PB15A	I/O
V15	—	PB12B	PB13C	PB14D	PB15D	I/O
W16	PB11A	PB12C	PB13D	PB15A	PB16A	I/O-$\overline{INIT}$
Y17	—	—	PB14A	PB15D	PB16D	I/O
V16	—	PB12D	PB14B	PB16A	PB17A	I/O-$V_{DD}5$
W17	PB11B	PB13A	PB15A	PB16D	PB17D	I/O
Y18	PB11C	PB13B	PB15B	PB17A	PB18A	I/O
U16	PB11D	PB13C	PB15C	PB17C	PB18D	I/O
V17	PB12A	PB13D	PB15D	PB17D	PB19A	I/O
W18	PB12B	PB14A	PB16A	PB18A	PB19D	I/O
Y19	PB12C	PB14B	PB16B	PB18B	PB20A	I/O
V18	PB12D	PB14C	PB16C	PB18C	PB20B	I/O
W19	—	PB14D	PB16D	PB18D	PB20D	I/O
Y20	DONE	DONE	DONE	DONE	DONE	DONE
W20	$\overline{RESET}$	$\overline{RESET}$	$\overline{RESET}$	$\overline{RESET}$	$\overline{RESET}$	$\overline{RESET}$
V19	$\overline{PRGM}$	$\overline{PRGM}$	$\overline{PRGM}$	$\overline{PRGM}$	$\overline{PRGM}$	$\overline{PRGM}$
U19	PR12A	PR14A	PR16A	PR18A	PR20A	I/O-M0

Notes:

The W3 pin on the 256-pin PBGA package is unconnected for all devices listed in this table.

The OR2C/2T08A do not have bond pads connected to the 256-pin PBGA package pins F2 and Y17.

The pins labeled "I/O-$V_{DD}5$" are user I/Os for the OR2CxxA series, but they are connected to $V_{DD}5$ for the OR2TxxA series.

The pins labeled "V_{SS}-ETC" are the 4 x 4 array of thermal balls located at the center of the package. The balls can be attached to the ground plane of the board for Enhanced Thermal Capability (see Table 28), or they can be left unconnected.

Pin Information (continued)

Table 23. OR2C/2T06A, OR2C/2T08A, OR2C/2T10A, OR2C/2T12A, and OR2C/2T15A 256-Pin PBGA Pinout (continued)

Pin	2C/2T06A Pad	2C/2T08A Pad	2C/2T10A Pad	2C/2T12A Pad	2C/2T15A Pad	Function
U18	—	PR14C	PR16C	PR18C	PR20D	I/O
T17	—	PR14D	PR16D	PR18D	PR19A	I/O
V20	—	PR13A	PR15A	PR17A	PR19D	I/O
U20	PR12B	PR13B	PR15B	PR17B	PR18A	I/O
T18	PR12C	PR13C	PR15C	PR17C	PR18B	I/O
T19	PR12D	PR13D	PR15D	PR17D	PR18D	I/O
T20	PR11A	PR12A	PR14A	PR16A	PR17A	I/O
R18	PR11B	PR12B	PR14C	PR16D	PR17D	I/O
P17	PR11C	PR12C	PR14D	PR15A	PR16A	I/O
R19	PR11D	PR12D	PR13A	PR15C	PR16C	I/O
R20	PR10A	PR11A	PR13B	PR15D	PR16D	I/O-M1
P18	PR10B	PR11B	PR13C	PR14A	PR15A	I/O
P19	PR10C	PR11C	PR12A	PR14D	PR15D	I/O-$V_{DD}5$
P20	PR10D	PR11D	PR12B	PR13A	PR14A	I/O
N18	PR9A	PR10A	PR11A	PR12A	PR13A	I/O-M2
N19	PR9B	PR10B	PR11B	PR12B	PR13B	I/O
N20	PR9C	PR10C	PR11C	PR12C	PR13C	I/O
M17	PR9D	PR10D	PR11D	PR12D	PR13D	I/O
M18	PR8A	PR9A	PR10A	PR11A	PR12A	I/O-M3
M19	PR8B	PR9B	PR10B	PR11B	PR12B	I/O
M20	PR8C	PR9C	PR10C	PR11C	PR12C	I/O
L19	PR8D	PR9D	PR10D	PR11D	PR12D	I/O
L18	PR7A	PR8A	PR9A	PR10A	PR11A	I/O
L20	PR7B	PR8B	PR9B	PR10B	PR11B	I/O
K20	PR7C	PR8C	PR9C	PR10C	PR11C	I/O
K19	PR7D	PR8D	PR9D	PR10D	PR11D	I/O
K18	PR6A	PR7A	PR8A	PR9A	PR10A	I/O
K17	PR6B	PR7B	PR8B	PR9B	PR10B	I/O
J20	PR6C	PR7C	PR8C	PR9C	PR10C	I/O
J19	PR6D	PR7D	PR8D	PR9D	PR10D	I/O
J18	PR5A	PR6A	PR7A	PR8A	PR9A	I/O-$V_{DD}5$
J17	PR5B	PR6B	PR7B	PR8B	PR9B	I/O
H20	PR5C	PR6C	PR7C	PR8C	PR9C	I/O
H19	PR5D	PR6D	PR7D	PR8D	PR9D	I/O
H18	PR4A	PR5A	PR6A	PR7A	PR8A	I/O-CS1
G20	PR4B	PR5B	PR6B	PR7B	PR8B	I/O
G19	PR4C	PR5C	PR6C	PR7C	PR8C	I/O
F20	PR4D	PR5D	PR6D	PR7D	PR8D	I/O
G18	PR3A	PR4A	PR5A	PR6A	PR7A	I/O-$\overline{CS0}$

Notes:

The W3 pin on the 256-pin PBGA package is unconnected for all devices listed in this table.

The OR2C/2T08A do not have bond pads connected to the 256-pin PBGA package pins F2 and Y17.

The pins labeled "I/O-$V_{DD}5$" are user I/Os for the OR2CxxA series, but they are connected to $V_{DD}5$ for the OR2TxxA series.

The pins labeled "V_{SS}-ETC" are the 4 x 4 array of thermal balls located at the center of the package. The balls can be attached to the ground plane of the board for Enhanced Thermal Capability (see Table 28), or they can be left unconnected.

Pin Information (continued)

Table 23. OR2C/2T06A, OR2C/2T08A, OR2C/2T10A, OR2C/2T12A, and OR2C/2T15A 256-Pin PBGA Pinout (continued)

Pin	2C/2T06A Pad	2C/2T08A Pad	2C/2T10A Pad	2C/2T12A Pad	2C/2T15A Pad	Function
F19	PR3B	PR4B	PR4B	PR6B	PR7B	I/O
E20	PR3C	PR4C	PR4C	PR5B	PR6B	I/O
G17	PR3D	PR4D	PR4D	PR5D	PR6D	I/O
F18	PR2A	PR3A	PR3A	PR4A	PR5A	I/O-$\overline{RD}$
E19	PR2B	PR3B	PR3B	PR4B	PR5B	I/O
D20	PR2C	PR3C	PR3C	PR4D	PR5D	I/O
E18	PR2D	PR3D	PR3D	PR3A	PR4A	I/O-$V_{DD}5$
D19	PR1A	PR2A	PR2A	PR2A	PR3A	I/O-$\overline{WR}$
C20	PR1B	PR2B	PR2B	PR2B	PR3B	I/O
E17	PR1C	PR2C	PR2C	PR2C	PR2A	I/O
D18	PR1D	PR2D	PR2D	PR2D	PR2D	I/O
C19	—	PR1A	PR1A	PR1A	PR1A	I/O
B20	—	PR1B	PR1B	PR1B	PR1B	I/O
C18	—	PR1C	PR1C	PR1C	PR1C	I/O
B19	—	PR1D	PR1D	PR1D	PR1D	I/O
A20	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$
A19	—	PT14D	PT16D	PT18D	PT20D	I/O
B18	PT12D	PT14C	PT16C	PT18C	PT20C	I/O
B17	PT12C	PT14B	PT16B	PT18B	PT20A	I/O
C17	PT12B	PT14A	PT16A	PT18A	PT19D	I/O
D16	PT12A	PT13D	PT15D	PT17D	PT19A	I/O-RDY/RCLK
A18	—	PT13C	PT15C	PT17A	PT18A	I/O
A17	PT11D	PT13B	PT15B	PT16D	PT17D	I/O
C16	PT11C	PT13A	PT15A	PT16C	PT17C	I/O
B16	PT11B	PT12D	PT14D	PT16A	PT17A	I/O
A16	PT11A	PT12C	PT13D	PT15D	PT16D	I/O-D7
C15	—	PT12B	PT13C	PT15A	PT16A	I/O
D14	PT10D	PT12A	PT13B	PT14D	PT15D	I/O-$V_{DD}5$
B15	PT10C	PT11D	PT13A	PT14A	PT15A	I/O
A15	PT10B	PT11C	PT12D	PT13D	PT14D	I/O
C14	PT10A	PT11B	PT12B	PT13B	PT14B	I/O-D6
B14	PT9D	PT11A	PT12A	PT13A	PT14A	I/O
A14	PT9C	PT10D	PT11D	PT12D	PT13D	I/O
C13	—	PT10C	PT11C	PT12C	PT13C	I/O
B13	PT9B	PT10B	PT11B	PT12B	PT13B	I/O
A13	PT9A	PT10A	PT11A	PT12A	PT13A	I/O-D5
D12	PT8D	PT9D	PT10D	PT11D	PT12D	I/O
C12	PT8C	PT9C	PT10C	PT11C	PT12C	I/O
B12	PT8B	PT9B	PT10B	PT11B	PT12B	I/O

Notes:

The W3 pin on the 256-pin PBGA package is unconnected for all devices listed in this table.

The OR2C/2T08A do not have bond pads connected to the 256-pin PBGA package pins F2 and Y17.

The pins labeled "I/O-$V_{DD}5$" are user I/Os for the OR2CxxA series, but they are connected to $V_{DD}5$ for the OR2TxxA series.

The pins labeled "V_{SS}-ETC" are the 4 x 4 array of thermal balls located at the center of the package. The balls can be attached to the ground plane of the board for Enhanced Thermal Capability (see Table 28), or they can be left unconnected.

Pin Information (continued)

Table 23. OR2C/2T06A, OR2C/2T08A, OR2C/2T10A, OR2C/2T12A, and OR2C/2T15A 256-Pin PBGA Pinout (continued)

Pin	2C/2T06A Pad	2C/2T08A Pad	2C/2T10A Pad	2C/2T12A Pad	2C/2T15A Pad	Function
A12	PT8A	PT9A	PT10A	PT11A	PT12A	I/O-D4
B11	PT7D	PT8D	PT9D	PT10D	PT11D	I/O
C11	PT7C	PT8C	PT9C	PT10C	PT11C	I/O
A11	PT7B	PT8B	PT9B	PT10B	PT11B	I/O
A10	PT7A	PT8A	PT9A	PT10A	PT11A	I/O-D3
B10	PT6D	PT7D	PT8D	PT9D	PT10D	I/O
C10	PT6C	PT7C	PT8C	PT9C	PT10C	I/O
D10	PT6B	PT7B	PT8B	PT9B	PT10B	I/O-VDD5
A9	PT6A	PT7A	PT8A	PT9A	PT10A	I/O-D2
B9	PT5D	PT6D	PT7D	PT8D	PT9D	I/O-D1
C9	PT5C	PT6C	PT7C	PT8C	PT9C	I/O
D9	PT5B	PT6B	PT7B	PT8B	PT9B	I/O
A8	PT5A	PT6A	PT7A	PT8A	PT9A	I/O-D0/DIN
B8	PT4D	PT5D	PT6D	PT7D	PT8D	I/O
C8	PT4C	PT5C	PT6C	PT7C	PT8C	I/O
A7	PT4B	PT5B	PT6B	PT7B	PT8B	I/O
B7	PT4A	PT5A	PT6A	PT7A	PT8A	I/O-DOUT
A6	PT3D	PT4D	PT5D	PT6D	PT7D	I/O
C7	PT3C	PT4C	PT5A	PT6A	PT7A	I/O
B6	PT3B	PT4B	PT4D	PT5C	PT6C	I/O
A5	PT3A	PT4A	PT4A	PT5A	PT6A	I/O-TDI
D7	PT2D	PT3D	PT3D	PT4D	PT5D	I/O
C6	PT2C	PT3C	PT3C	PT4A	PT5A	I/O-VDD5
B5	PT2B	PT3B	PT3B	PT3D	PT4D	I/O
A4	PT2A	PT3A	PT3A	PT3A	PT4A	I/O-TMS
C5	—	PT2D	PT2D	PT2D	PT3D	I/O
B4	PT1D	PT2C	PT2C	PT2C	PT3A	I/O
A3	PT1C	PT2B	PT2B	PT2B	PT2D	I/O
D5	PT1B	PT2A	PT2A	PT2A	PT2A	I/O
C4	—	PT1D	PT1D	PT1D	PT1D	I/O
B3	—	PT1C	PT1C	PT1C	PT1C	I/O
B2	—	PT1B	PT1B	PT1B	PT1B	I/O
A2	PT1A	PT1A	PT1A	PT1A	PT1A	I/O-TCK
C3	RD_DATA/TDO	RD_DATA/TDO	RD_DATA/TDO	RD_DATA/TDO	RD_DATA/TDO	RD_DATA/TDO
A1	VSS	VSS	VSS	VSS	VSS	VSS
D4	VSS	VSS	VSS	VSS	VSS	VSS
D8	VSS	VSS	VSS	VSS	VSS	VSS
D13	VSS	VSS	VSS	VSS	VSS	VSS
D17	VSS	VSS	VSS	VSS	VSS	VSS

Notes:

The W3 pin on the 256-pin PBGA package is unconnected for all devices listed in this table.

The OR2C/2T08A do not have bond pads connected to the 256-pin PBGA package pins F2 and Y17.

The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

The pins labeled "VSS-ETC" are the 4 x 4 array of thermal balls located at the center of the package. The balls can be attached to the ground plane of the board for Enhanced Thermal Capability (see Table 28), or they can be left unconnected.

Pin Information (continued)

Table 23. OR2C/2T06A, OR2C/2T08A, OR2C/2T10A, OR2C/2T12A, and OR2C/2T15A 256-Pin PBGA Pinout (continued)

Pin	2C/2T06A Pad	2C/2T08A Pad	2C/2T10A Pad	2C/2T12A Pad	2C/2T15A Pad	Function
H4	VSS	VSS	VSS	VSS	VSS	VSS
H17	VSS	VSS	VSS	VSS	VSS	VSS
N4	VSS	VSS	VSS	VSS	VSS	VSS
N17	VSS	VSS	VSS	VSS	VSS	VSS
U4	VSS	VSS	VSS	VSS	VSS	VSS
U8	VSS	VSS	VSS	VSS	VSS	VSS
U13	VSS	VSS	VSS	VSS	VSS	VSS
U17	VSS	VSS	VSS	VSS	VSS	VSS
B1	VDD	VDD	VDD	VDD	VDD	VDD
D6	VDD	VDD	VDD	VDD	VDD	VDD
D11	VDD	VDD	VDD	VDD	VDD	VDD
D15	VDD	VDD	VDD	VDD	VDD	VDD
F4	VDD	VDD	VDD	VDD	VDD	VDD
F17	VDD	VDD	VDD	VDD	VDD	VDD
K4	VDD	VDD	VDD	VDD	VDD	VDD
L17	VDD	VDD	VDD	VDD	VDD	VDD
R4	VDD	VDD	VDD	VDD	VDD	VDD
R17	VDD	VDD	VDD	VDD	VDD	VDD
U6	VDD	VDD	VDD	VDD	VDD	VDD
U10	VDD	VDD	VDD	VDD	VDD	VDD
U15	VDD	VDD	VDD	VDD	VDD	VDD
W3	—	—	—	—	—	No Connect
J10	VSS	VSS	VSS	VSS	VSS	VSS—ETC
J11	VSS	VSS	VSS	VSS	VSS	VSS—ETC
J12	VSS	VSS	VSS	VSS	VSS	VSS—ETC
J9	VSS	VSS	VSS	VSS	VSS	VSS—ETC
K10	VSS	VSS	VSS	VSS	VSS	VSS—ETC
K11	VSS	VSS	VSS	VSS	VSS	VSS—ETC
K12	VSS	VSS	VSS	VSS	VSS	VSS—ETC
K9	VSS	VSS	VSS	VSS	VSS	VSS—ETC
L10	VSS	VSS	VSS	VSS	VSS	VSS—ETC
L11	VSS	VSS	VSS	VSS	VSS	VSS—ETC
L12	VSS	VSS	VSS	VSS	VSS	VSS—ETC
L9	VSS	VSS	VSS	VSS	VSS	VSS—ETC
M10	VSS	VSS	VSS	VSS	VSS	VSS—ETC
M11	VSS	VSS	VSS	VSS	VSS	VSS—ETC
M12	VSS	VSS	VSS	VSS	VSS	VSS—ETC
M9	VSS	VSS	VSS	VSS	VSS	VSS—ETC

Notes:

The W3 pin on the 256-pin PBGA package is unconnected for all devices listed in this table.

The OR2C/2T08A do not have bond pads connected to the 256-pin PBGA package pins F2 and Y17.

The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

The pins labeled "VSS-ETC" are the 4 x 4 array of thermal balls located at the center of the package. The balls can be attached to the ground plane of the board for Enhanced Thermal Capability (see Table 28), or they can be left unconnected.

Pin Information (continued)

Table 24. OR2C12A, OR2C15A, OR2C26A, and OR2C40A 304-Pin SQFP/SQFP2 Pinout

Pin	2C12A Pad	2C15A Pad	2C26A Pad	2C40A Pad	Function
1	VSS	VSS	VSS	VSS	VSS
2	VDD	VDD	VDD	VDD	VDD
3	VSS	VSS	VSS	VSS	VSS
4	PL1D	PL1D	PL1D	PL1D	I/O
5	PL1C	PL1C	PL1C	PL1A	I/O
6	PL1B	PL1B	PL1B	PL2D	I/O
7	PL1A	PL1A	PL1A	PL2A	I/O
8	PL2D	PL2D	PL2D	PL3D	I/O-A0
9	PL2C	PL2A	PL2A	PL3A	I/O
10	PL2B	PL3D	PL3D	PL4D	I/O
11	PL2A	PL3A	PL3A	PL4A	I/O
12	VSS	VSS	VSS	VSS	VSS
13	PL3D	PL4D	PL4D	PL5D	I/O
14	PL3A	PL4A	PL4A	PL6D	I/O
15	PL4D	PL5D	PL5D	PL7D	I/O
16	PL4A	PL5A	PL5A	PL8D	I/O-A1
17	PL5D	PL6D	PL6D	PL9D	I/O
18	PL5C	PL6C	PL6C	PL9C	I/O
19	PL5B	PL6B	PL6B	PL9B	I/O
20	PL5A	PL6A	PL6A	PL9A	I/O-A2
21	PL6D	PL7D	PL7D	PL10D	I/O
22	PL6C	PL7C	PL7C	PL10C	I/O
23	PL6B	PL7B	PL7B	PL10B	I/O
24	PL6A	PL7A	PL7A	PL10A	I/O-A3
25	VDD	VDD	VDD	VDD	VDD
26	PL7D	PL8D	PL8D	PL11D	I/O
27	PL7C	PL8C	PL8A	PL11A	I/O
28	PL7B	PL8B	PL9D	PL12D	I/O
29	PL7A	PL8A	PL9A	PL12A	I/O-A4
30	PL8D	PL9D	PL10D	PL13D	I/O-A5
31	PL8C	PL9C	PL10A	PL13A	I/O
32	PL8B	PL9B	PL11D	PL14D	I/O
33	PL8A	PL9A	PL11A	PL14A	I/O-A6
34	VSS	VSS	VSS	VSS	VSS
35	PL9D	PL10D	PL12D	PL15D	I/O
36	PL9C	PL10C	PL12C	PL15C	I/O
37	PL9B	PL10B	PL12B	PL15B	I/O
38	PL9A	PL10A	PL12A	PL15A	I/O-A7
39	VDD	VDD	VDD	VDD	VDD
40	PL10D	PL11D	PL13D	PL16D	I/O
41	PL10C	PL11C	PL13C	PL16C	I/O
42	PL10B	PL11B	PL13B	PL16B	I/O
43	PL10A	PL11A	PL13A	PL16A	I/O-A8
44	VSS	VSS	VSS	VSS	VSS
45	PL11D	PL12D	PL14D	PL17D	I/O-A9

Note: The OR2TxxA series is not offered in the 304-pin SQFP/SQFP2 packages.

Pin Information (continued)

Table 24. OR2C12A, OR2C15A, OR2C26A, and OR2C40A 304-Pin SQFP/SQFP2 Pinout (continued)

Pin	2C12A Pad	2C15A Pad	2C26A Pad	2C40A Pad	Function
46	PL11C	PL12C	PL14A	PL17A	I/O
47	PL11B	PL12B	PL15D	PL18D	I/O
48	PL11A	PL12A	PL15A	PL18A	I/O-A10
49	PL12D	PL13D	PL16D	PL19D	I/O
50	PL12C	PL13C	PL16A	PL19A	I/O
51	PL12B	PL13B	PL17D	PL20D	I/O
52	PL12A	PL13A	PL17A	PL20A	I/O-A11
53	VDD	VDD	VDD	VDD	VDD
54	PL13D	PL14D	PL18D	PL21D	I/O-A12
55	PL13B	PL14B	PL18B	PL21B	I/O
56	PL13A	PL14A	PL18A	PL21A	I/O
57	PL14D	PL15D	PL19D	PL22D	I/O
58	PL14B	PL15B	PL19B	PL22B	I/O-A13
59	PL14A	PL15A	PL19A	PL22A	I/O
60	PL15D	PL16D	PL20D	PL23D	I/O
61	PL15B	PL16B	PL20B	PL24D	I/O
62	PL15A	PL16A	PL20A	PL25D	I/O
63	PL16D	PL17D	PL21D	PL25A	I/O-A14
64	PL16A	PL17A	PL21A	PL26A	I/O
65	VSS	VSS	VSS	VSS	VSS
66	PL17D	PL18D	PL22D	PL27D	I/O
67	PL17C	PL18C	PL22C	PL27C	I/O
68	PL17B	PL18A	PL22A	PL27A	I/O
69	PL17A	PL19D	PL23D	PL28D	I/O
70	PL18D	PL19C	PL23C	PL28C	I/O
71	PL18C	PL19A	PL23A	PL28A	I/O
72	PL18B	PL20D	PL24D	PL29A	I/O
73	PL18A	PL20A	PL24A	PL30A	I/O-A15
74	VSS	VSS	VSS	VSS	VSS
75	CCLK	CCLK	CCLK	CCLK	CCLK
76	VDD	VDD	VDD	VDD	VDD
77	VSS	VSS	VSS	VSS	VSS
78	VDD	VDD	VDD	VDD	VDD
79	VSS	VSS	VSS	VSS	VSS
80	PB1A	PB1A	PB1A	PB1A	I/O-A16
81	PB1B	PB1C	PB1C	PB2A	I/O
82	PB1C	PB1D	PB1D	PB2D	I/O
83	PB1D	PB2A	PB2A	PB3A	I/O
84	PB2A	PB2D	PB2D	PB3D	I/O
85	PB2B	PB3A	PB3A	PB4A	I/O
86	PB2C	PB3C	PB3C	PB4C	I/O
87	PB2D	PB3D	PB3D	PB4D	I/O
88	VSS	VSS	VSS	VSS	VSS
89	PB3A	PB4A	PB4A	PB5A	I/O
90	PB3D	PB4D	PB4D	PB5D	I/O-A17

Note: The OR2TxxA series is not offered in the 304-pin SQFP/SQFP2 packages.

Pin Information (continued)

Table 24. OR2C12A, OR2C15A, OR2C26A, and OR2C40A 304-Pin SQFP/SQFP2 Pinout (continued)

Pin	2C12A Pad	2C15A Pad	2C26A Pad	2C40A Pad	Function
91	PB4A	PB5A	PB5A	PB6A	I/O
92	PB4D	PB5D	PB5D	PB6D	I/O
93	PB5A	PB6A	PB6A	PB7A	I/O
94	PB5B	PB6B	PB6B	PB7D	I/O
95	PB5C	PB6C	PB6C	PB8A	I/O
96	PB5D	PB6D	PB6D	PB8D	I/O
97	PB6A	PB7A	PB7A	PB9A	I/O
98	PB6B	PB7B	PB7B	PB9D	I/O
99	PB6C	PB7C	PB7C	PB10A	I/O
100	PB6D	PB7D	PB7D	PB10D	I/O
101	VDD	VDD	VDD	VDD	VDD
102	PB7A	PB8A	PB8A	PB11A	I/O
103	PB7B	PB8B	PB8D	PB11D	I/O
104	PB7C	PB8C	PB9A	PB12A	I/O
105	PB7D	PB8D	PB9D	PB12D	I/O
106	PB8A	PB9A	PB10A	PB13A	I/O
107	PB8B	PB9B	PB10D	PB13D	I/O
108	PB8C	PB9C	PB11A	PB14A	I/O
109	PB8D	PB9D	PB11D	PB14D	I/O
110	VSS	VSS	VSS	VSS	VSS
111	PB9A	PB10A	PB12A	PB15A	I/O
112	PB9B	PB10B	PB12B	PB15B	I/O
113	PB9C	PB10C	PB12C	PB15C	I/O
114	PB9D	PB10D	PB12D	PB15D	I/O
115	VSS	VSS	VSS	VSS	VSS
116	PB10A	PB11A	PB13A	PB16A	I/O
117	PB10B	PB11B	PB13B	PB16B	I/O
118	PB10C	PB11C	PB13C	PB16C	I/O
119	PB10D	PB11D	PB13D	PB16D	I/O
120	VSS	VSS	VSS	VSS	VSS
121	PB11A	PB12A	PB14A	PB17A	I/O
122	PB11B	PB12B	PB14D	PB17D	I/O
123	PB11C	PB12C	PB15A	PB18A	I/O
124	PB11D	PB12D	PB15D	PB18D	I/O
125	PB12A	PB13A	PB16A	PB19A	I/O-HDC
126	PB12B	PB13B	PB16D	PB19D	I/O
127	PB12C	PB13C	PB17A	PB20A	I/O
128	PB12D	PB13D	PB17D	PB20D	I/O
129	VDD	VDD	VDD	VDD	VDD
130	PB13A	PB14A	PB18A	PB21A	I/O-$\overline{\text{LDC}}$
131	PB13B	PB14B	PB18B	PB21D	I/O
132	PB13C	PB14C	PB18C	PB22A	I/O
133	PB13D	PB14D	PB18D	PB22D	I/O
134	PB14A	PB15A	PB19A	PB23A	I/O
135	PB14B	PB15B	PB19B	PB24A	I/O

Note: The OR2TxxA series is not offered in the 304-pin SQFP/SQFP2 packages.

Pin Information (continued)

Table 24. OR2C12A, OR2C15A, OR2C26A, and OR2C40A 304-Pin SQFP/SQFP2 Pinout (continued)

Pin	2C12A Pad	2C15A Pad	2C26A Pad	2C40A Pad	Function
136	PB14D	PB15D	PB19D	PB24D	I/O
137	PB15A	PB16A	PB20A	PB25A	I/O-$\overline{\text{INIT}}$
138	PB15D	PB16D	PB20D	PB25D	I/O
139	PB16A	PB17A	PB21A	PB26A	I/O
140	PB16D	PB17D	PB21D	PB26D	I/O
141	VSS	VSS	VSS	VSS	VSS
142	PB17A	PB18A	PB22A	PB27A	I/O
143	PB17B	PB18B	PB22B	PB27B	I/O
144	PB17C	PB18D	PB22D	PB27D	I/O
145	PB17D	PB19A	PB23A	PB28A	I/O
146	PB18A	PB19D	PB23D	PB28D	I/O
147	PB18B	PB20A	PB24A	PB29A	I/O
148	PB18C	PB20B	PB24B	PB29D	I/O
149	PB18D	PB20D	PB24D	PB30D	I/O
150	VSS	VSS	VSS	VSS	VSS
151	DONE	DONE	DONE	DONE	DONE
152	VDD	VDD	VDD	VDD	VDD
153	VSS	VSS	VSS	VSS	VSS
154	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$
155	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$
156	PR18A	PR20A	PR24A	PR30A	I/O-M0
157	PR18B	PR20C	PR24C	PR29A	I/O
158	PR18C	PR20D	PR24D	PR29D	I/O
159	PR18D	PR19A	PR23A	PR28A	I/O
160	PR17A	PR19D	PR23D	PR28D	I/O
161	PR17B	PR18A	PR22A	PR27A	I/O
162	PR17C	PR18B	PR22B	PR27B	I/O
163	PR17D	PR18D	PR22D	PR27D	I/O
164	VSS	VSS	VSS	VSS	VSS
165	PR16A	PR17A	PR21A	PR26A	I/O
166	PR16D	PR17D	PR21D	PR25A	I/O
167	PR15A	PR16A	PR20A	PR24A	I/O
168	PR15C	PR16C	PR20C	PR24D	I/O
169	PR15D	PR16D	PR20D	PR23D	I/O-M1
170	PR14A	PR15A	PR19A	PR22A	I/O
171	PR14C	PR15C	PR19C	PR22C	I/O
172	PR14D	PR15D	PR19D	PR22D	I/O
173	PR13A	PR14A	PR18A	PR21A	I/O
174	PR13C	PR14C	PR18C	PR21C	I/O
175	PR13D	PR14D	PR18D	PR21D	I/O
176	VDD	VDD	VDD	VDD	VDD
177	PR12A	PR13A	PR17A	PR20A	I/O-M2
178	PR12B	PR13B	PR17D	PR20D	I/O
179	PR12C	PR13C	PR16A	PR19A	I/O
180	PR12D	PR13D	PR16D	PR19D	I/O

Note: The OR2TxxA series is not offered in the 304-pin SQFP/SQFP2 packages.

Pin Information (continued)

Table 24. OR2C12A, OR2C15A, OR2C26A, and OR2C40A 304-Pin SQFP/SQFP2 Pinout (continued)

Pin	2C12A Pad	2C15A Pad	2C26A Pad	2C40A Pad	Function
181	PR11A	PR12A	PR15A	PR18A	I/O-M3
182	PR11B	PR12B	PR15D	PR18D	I/O
183	PR11C	PR12C	PR14A	PR17A	I/O
184	PR11D	PR12D	PR14D	PR17D	I/O
185	VSS	VSS	VSS	VSS	VSS
186	PR10A	PR11A	PR13A	PR16A	I/O
187	PR10B	PR11B	PR13B	PR16B	I/O
188	PR10C	PR11C	PR13C	PR16C	I/O
189	PR10D	PR11D	PR13D	PR16D	I/O
190	VDD	VDD	VDD	VDD	VDD
191	PR9A	PR10A	PR12A	PR15A	I/O
192	PR9B	PR10B	PR12B	PR15B	I/O
193	PR9C	PR10C	PR12C	PR15C	I/O
194	PR9D	PR10D	PR12D	PR15D	I/O
195	VSS	VSS	VSS	VSS	VSS
196	PR8A	PR9A	PR11A	PR14A	I/O
197	PR8B	PR9B	PR11D	PR14D	I/O
198	PR8C	PR9C	PR10A	PR13A	I/O
199	PR8D	PR9D	PR10D	PR13D	I/O
200	PR7A	PR8A	PR9A	PR12A	I/O-CS1
201	PR7B	PR8B	PR9D	PR12D	I/O
202	PR7C	PR8C	PR8A	PR11A	I/O
203	PR7D	PR8D	PR8D	PR11D	I/O
204	VDD	VDD	VDD	VDD	VDD
205	PR6A	PR7A	PR7A	PR10A	I/O-$\overline{CS0}$
206	PR6B	PR7B	PR7B	PR10B	I/O
207	PR6C	PR7C	PR7C	PR10C	I/O
208	PR6D	PR7D	PR7D	PR10D	I/O
209	PR5A	PR6A	PR6A	PR9A	I/O
210	PR5B	PR6B	PR6B	PR9B	I/O
211	PR5C	PR6C	PR6C	PR9C	I/O
212	PR5D	PR6D	PR6D	PR9D	I/O
213	PR4A	PR5A	PR5A	PR8A	I/O-$\overline{RD}$
214	PR4B	PR5B	PR5B	PR7A	I/O
215	PR4D	PR5D	PR5D	PR6A	I/O
216	PR3A	PR4A	PR4A	PR5A	I/O
217	VSS	VSS	VSS	VSS	VSS
218	PR2A	PR3A	PR3A	PR4A	I/O-$\overline{WR}$
219	PR2B	PR3B	PR3B	PR4B	I/O
220	PR2C	PR2A	PR2A	PR3A	I/O
221	PR2D	PR2D	PR2D	PR3D	I/O
222	PR1A	PR1A	PR1A	PR2A	I/O
223	PR1B	PR1B	PR1B	PR2D	I/O
224	PR1C	PR1C	PR1C	PR1A	I/O
225	PR1D	PR1D	PR1D	PR1D	I/O

Note: The OR2TxxA series is not offered in the 304-pin SQFP/SQFP2 packages.

Pin Information (continued)

Table 24. OR2C12A, OR2C15A, OR2C26A, and OR2C40A 304-Pin SQFP/SQFP2 Pinout (continued)

Pin	2C12A Pad	2C15A Pad	2C26A Pad	2C40A Pad	Function
226	VSS	VSS	VSS	VSS	VSS
227	RD_CFGN	RD_CFGN	RD_CFGN	RD_CFGN	RD_CFGN
228	VDD	VDD	VDD	VDD	VDD
229	VSS	VSS	VSS	VSS	VSS
230	VDD	VDD	VDD	VDD	VDD
231	VSS	VSS	VSS	VSS	VSS
232	PT18D	PT20D	PT24D	PT30D	I/O
233	PT18C	PT20C	PT24C	PT30A	I/O
234	PT18B	PT20A	PT24A	PT29A	I/O
235	PT18A	PT19D	PT23D	PT28D	I/O
236	PT17D	PT19A	PT23A	PT28A	I/O-RDY/RCLK
237	PT17C	PT18D	PT22D	PT27D	I/O
238	PT17B	PT18C	PT22C	PT27C	I/O
239	PT17A	PT18A	PT22A	PT27A	I/O
240	VSS	VSS	VSS	VSS	VSS
241	PT16D	PT17D	PT21D	PT26D	I/O
242	PT16C	PT17C	PT21C	PT26C	I/O
243	PT16A	PT17A	PT21A	PT26A	I/O
244	PT15D	PT16D	PT20D	PT25D	I/O-D7
245	PT15A	PT16A	PT20A	PT25A	I/O
246	PT14D	PT15D	PT19D	PT24D	I/O
247	PT14A	PT15A	PT19A	PT23D	I/O
248	PT13D	PT14D	PT18D	PT22D	I/O
249	PT13C	PT14C	PT18C	PT22A	I/O
250	PT13B	PT14B	PT18B	PT21D	I/O-D6
251	PT13A	PT14A	PT18A	PT21A	I/O
252	VDD	VDD	VDD	VDD	VDD
253	PT12D	PT13D	PT17D	PT20D	I/O
254	PT12C	PT13C	PT17A	PT20A	I/O
255	PT12B	PT13B	PT16D	PT19D	I/O
256	PT12A	PT13A	PT16A	PT19A	I/O-D5
257	PT11D	PT12D	PT15D	PT18D	I/O
258	PT11C	PT12C	PT15A	PT18A	I/O
259	PT11B	PT12B	PT14D	PT17D	I/O
260	PT11A	PT12A	PT14A	PT17A	I/O-D4
261	VSS	VSS	VSS	VSS	VSS
262	PT10D	PT11D	PT13D	PT16D	I/O
263	PT10C	PT11C	PT13C	PT16C	I/O
264	PT10B	PT11B	PT13B	PT16B	I/O
265	PT10A	PT11A	PT13A	PT16A	I/O-D3
266	VSS	VSS	VSS	VSS	VSS
267	PT9D	PT10D	PT12D	PT15D	I/O
268	PT9C	PT10C	PT12C	PT15C	I/O
269	PT9B	PT10B	PT12B	PT15B	I/O
270	PT9A	PT10A	PT12A	PT15A	I/O-D2

Note: The OR2TxxA series is not offered in the 304-pin SQFP/SQFP2 packages.

Pin Information (continued)

Table 24. OR2C12A, OR2C15A, OR2C26A, and OR2C40A 304-Pin SQFP/SQFP2 Pinout (continued)

Pin	2C12A Pad	2C15A Pad	2C26A Pad	2C40A Pad	Function
271	VSS	VSS	VSS	VSS	VSS
272	PT8D	PT9D	PT11D	PT14D	I/O-D1
273	PT8C	PT9C	PT11A	PT14A	I/O
274	PT8B	PT9B	PT10D	PT13D	I/O
275	PT8A	PT9A	PT10A	PT13A	I/O-D0/DIN
276	PT7D	PT8D	PT9D	PT12D	I/O
277	PT7C	PT8C	PT9A	PT12A	I/O
278	PT7B	PT8B	PT8D	PT11D	I/O
279	PT7A	PT8A	PT8A	PT11A	I/O-DOUT
280	VDD	VDD	VDD	VDD	VDD
281	PT6D	PT7D	PT7D	PT10D	I/O
282	PT6C	PT7C	PT7C	PT10A	I/O
283	PT6B	PT7B	PT7B	PT9D	I/O
284	PT6A	PT7A	PT7A	PT9A	I/O
285	PT5D	PT6D	PT6D	PT8D	I/O
286	PT5C	PT6C	PT6C	PT8A	I/O
287	PT5B	PT6B	PT6B	PT7D	I/O
288	PT5A	PT6A	PT6A	PT7A	I/O-TDI
289	PT4D	PT5D	PT5D	PT6D	I/O
290	PT4A	PT5A	PT5A	PT6A	I/O
291	PT3D	PT4D	PT4D	PT5D	I/O
292	PT3A	PT4A	PT4A	PT5A	I/O-TMS
293	VSS	VSS	VSS	VSS	VSS
294	PT2D	PT3D	PT3D	PT4D	I/O
295	PT2C	PT3A	PT3A	PT4A	I/O
296	PT2B	PT2D	PT2D	PT3D	I/O
297	PT2A	PT2A	PT2A	PT3A	I/O
298	PT1D	PT1D	PT1D	PT2D	I/O
299	PT1C	PT1C	PT1C	PT2A	I/O
300	PT1B	PT1B	PT1B	PT1D	I/O
301	PT1A	PT1A	PT1A	PT1A	I/O-TCK
302	VSS	VSS	VSS	VSS	VSS
303	RD_DATA/TDO	RD_DATA/TDO	RD_DATA/TDO	RD_DATA/TDO	RD_DATA/TDO
304	VDD	VDD	VDD	VDD	VDD

Note: The OR2TxxA series is not offered in the 304-pin SQFP/SQFP2 packages.

Pin Information (continued)

Table 25. OR2C/2T10A, OR2C/2T12A, OR2C/2T15A, and OR2C/2T26A 352-Pin PBGA Pinout

Pin	2C/2T10A Pad	2C/2T12A Pad	2C/2T15A Pad	2C/2T26A Pad	Function
B1	PL1D	PL1D	PL1D	PL1D	I/O
C2	PL1C	PL1C	PL1C	PL1C	I/O
C1	PL1B	PL1B	PL1B	PL1B	I/O
D2	PL1A	PL1A	PL1A	PL1A	I/O
D3	PL2D	PL2D	PL2D	PL2D	I/O-A0
D1	PL2C	PL2C	PL2A	PL2A	I/O
E2	PL2B	PL2B	PL3D	PL3D	I/O
E4	—	—	PL3B	PL3B	I/O
E3	PL2A	PL2A	PL3A	PL3A	I/O
E1	PL3D	PL3D	PL4D	PL4D	I/O-VDD5
F2	—	PL3C	PL4C	PL4C	I/O
G4	PL3C	PL3B	PL4B	PL4B	I/O
F3	—	PL3A	PL4A	PL4A	I/O
F1	PL3B	PL4D	PL5D	PL5D	I/O
G2	—	PL4C	PL5C	PL5C	I/O
G1	—	PL4B	PL5B	PL5B	I/O
G3	PL3A	PL4A	PL5A	PL5A	I/O-A1
H2	PL4D	PL5D	PL6D	PL6D	I/O
J4	PL4C	PL5C	PL6C	PL6C	I/O
H1	PL4B	PL5B	PL6B	PL6B	I/O
H3	PL4A	PL5A	PL6A	PL6A	I/O-A2
J2	PL5D	PL6D	PL7D	PL7D	I/O
J1	PL5C	PL6C	PL7C	PL7C	I/O
K2	PL5B	PL6B	PL7B	PL7B	I/O
J3	PL5A	PL6A	PL7A	PL7A	I/O-A3
K1	PL6D	PL7D	PL8D	PL8D	I/O
K4	PL6C	PL7C	PL8C	PL8A	I/O
L2	PL6B	PL7B	PL8B	PL9D	I/O
K3	PL6A	PL7A	PL8A	PL9A	I/O-A4
L1	PL7D	PL8D	PL9D	PL10D	I/O-A5
M2	PL7C	PL8C	PL9C	PL10A	I/O
M1	PL7B	PL8B	PL9B	PL11D	I/O
L3	PL7A	PL8A	PL9A	PL11A	I/O-A6
N2	PL8D	PL9D	PL10D	PL12D	I/O
M4	PL8C	PL9C	PL10C	PL12C	I/O
N1	PL8B	PL9B	PL10B	PL12B	I/O
M3	PL8A	PL9A	PL10A	PL12A	I/O-A7
P2	PL9D	PL10D	PL11D	PL13D	I/O
P4	PL9C	PL10C	PL11C	PL13C	I/O-VDD5
P1	PL9B	PL10B	PL11B	PL13B	I/O
N3	PL9A	PL10A	PL11A	PL13A	I/O-A8
R2	PL10D	PL11D	PL12D	PL14D	I/O-A9
P3	PL10C	PL11C	PL12C	PL14A	I/O
R1	PL10B	PL11B	PL12B	PL15D	I/O

Notes:
The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

The pins labeled "VSS-ETC" are the 6 x 6 array of thermal balls located at the center of the package. The balls can be attached to the ground plane of the board for Enhanced Thermal Capability (see Table 28), or they can be left unconnected.

Pin Information (continued)

Table 25. OR2C/2T10A, OR2C/2T12A, OR2C/2T15A, and OR2C/2T26A 352-Pin PBGA Pinout (continued)

Pin	2C/2T10A Pad	2C/2T12A Pad	2C/2T15A Pad	2C/2T26A Pad	Function
T2	PL10A	PL11A	PL12A	PL15A	I/O-A10
R3	PL11D	PL12D	PL13D	PL16D	I/O
T1	PL11C	PL12C	PL13C	PL16A	I/O
R4	PL11B	PL12B	PL13B	PL17D	I/O
U2	PL11A	PL12A	PL13A	PL17A	I/O-A11
T3	PL12D	PL13D	PL14D	PL18D	I/O-A12
U1	—	PL13C	PL14C	PL18C	I/O
U4	PL12C	PL13B	PL14B	PL18B	I/O
V2	—	PL13A	PL14A	PL18A	I/O
U3	PL12B	PL14D	PL15D	PL19D	I/O
V1	PL12A	PL14C	PL15C	PL19C	I/O
W2	PL13D	PL14B	PL15B	PL19B	I/O-A13
W1	PL13C	PL14A	PL15A	PL19A	I/O
V3	PL13B	PL15D	PL16D	PL20D	I/O
Y2	PL13A	PL15C	PL16C	PL20C	I/O
W4	PL14D	PL15B	PL16B	PL20B	I/O
Y1	—	PL15A	PL16A	PL20A	I/O
W3	PL14C	PL16D	PL17D	PL21D	I/O-A14
AA2	PL14B	PL16C	PL17C	PL21C	I/O
Y4	PL14A	PL16B	PL17B	PL21B	I/O
AA1	—	PL16A	PL17A	PL21A	I/O
Y3	PL15D	PL17D	PL18D	PL22D	I/O-$V_{DD}5$
AB2	PL15C	PL17C	PL18C	PL22C	I/O
AB1	PL15B	PL17B	PL18A	PL22A	I/O
AA3	PL15A	PL17A	PL19D	PL23D	I/O
AC2	PL16D	PL18D	PL19C	PL23C	I/O
AB4	PL16C	PL18C	PL19A	PL23A	I/O
AC1	PL16B	PL18B	PL20D	PL24D	I/O
AB3	—	—	PL20C	PL24C	I/O
AD2	—	—	PL20B	PL24B	I/O
AC3	PL16A	PL18A	PL20A	PL24A	I/O-A15
AD1	CCLK	CCLK	CCLK	CCLK	CCLK
AF2	PB1A	PB1A	PB1A	PB1A	I/O-A16
AE3	—	—	PB1B	PB1B	I/O
AF3	PB1B	PB1B	PB1C	PB1C	I/O
AE4	PB1C	PB1C	PB1D	PB1D	I/O
AD4	PB1D	PB1D	PB2A	PB2A	I/O
AF4	PB2A	PB2A	PB2D	PB2D	I/O-$V_{DD}5$
AE5	—	PB2B	PB3A	PB3A	I/O
AC5	PB2B	PB2C	PB3C	PB3C	I/O
AD5	—	PB2D	PB3D	PB3D	I/O
AF5	PB2C	PB3A	PB4A	PB4A	I/O
AE6	PB2D	PB3B	PB4B	PB4B	I/O

Notes:
The pins labeled "I/O-$V_{DD}5$" are user I/Os for the OR2CxxA series, but they are connected to $V_{DD}5$ for the OR2TxxA series.

The pins labeled "V_{SS}-ETC" are the 6 x 6 array of thermal balls located at the center of the package. The balls can be attached to the ground plane of the board for Enhanced Thermal Capability (see Table 28), or they can be left unconnected.

Pin Information (continued)

Table 25. OR2C/2T10A, OR2C/2T12A, OR2C/2T15A, and OR2C/2T26A 352-Pin PBGA Pinout (continued)

Pin	2C/2T10A Pad	2C/2T12A Pad	2C/2T15A Pad	2C/2T26A Pad	Function
AC7	PB3A	PB3C	PB4C	PB4C	I/O
AD6	PB3B	PB3D	PB4D	PB4D	I/O-A17
AF6	—	PB4A	PB5A	PB5A	I/O
AE7	PB3C	PB4B	PB5B	PB5B	I/O
AF7	—	PB4C	PB5C	PB5C	I/O
AD7	PB3D	PB4D	PB5D	PB5D	I/O
AE8	PB4A	PB5A	PB6A	PB6A	I/O
AC9	PB4B	PB5B	PB6B	PB6B	I/O
AF8	PB4C	PB5C	PB6C	PB6C	I/O
AD8	PB4D	PB5D	PB6D	PB6D	I/O
AE9	PB5A	PB6A	PB7A	PB7A	I/O
AF9	PB5B	PB6B	PB7B	PB7B	I/O
AE10	PB5C	PB6C	PB7C	PB7C	I/O
AD9	PB5D	PB6D	PB7D	PB7D	I/O
AF10	PB6A	PB7A	PB8A	PB8A	I/O
AC10	PB6B	PB7B	PB8B	PB8D	I/O
AE11	PB6C	PB7C	PB8C	PB9A	I/O
AD10	PB6D	PB7D	PB8D	PB9D	I/O
AF11	PB7A	PB8A	PB9A	PB10A	I/O
AE12	PB7B	PB8B	PB9B	PB10D	I/O
AF12	PB7C	PB8C	PB9C	PB11A	I/O
AD11	PB7D	PB8D	PB9D	PB11D	I/O
AE13	PB8A	PB9A	PB10A	PB12A	I/O
AC12	PB8B	PB9B	PB10B	PB12B	I/O
AF13	PB8C	PB9C	PB10C	PB12C	I/O
AD12	PB8D	PB9D	PB10D	PB12D	I/O
AE14	PB9A	PB10A	PB11A	PB13A	I/O
AC14	PB9B	PB10B	PB11B	PB13B	I/O
AF14	PB9C	PB10C	PB11C	PB13C	I/O
AD13	PB9D	PB10D	PB11D	PB13D	I/O
AE15	PB10A	PB11A	PB12A	PB14A	I/O-$V_{DD}5$
AD14	PB10B	PB11B	PB12B	PB14D	I/O
AF15	PB10C	PB11C	PB12C	PB15A	I/O
AE16	PB10D	PB11D	PB12D	PB15D	I/O
AD15	PB11A	PB12A	PB13A	PB16A	I/O-HDC
AF16	PB11B	PB12B	PB13B	PB16D	I/O
AC15	PB11C	PB12C	PB13C	PB17A	I/O
AE17	PB11D	PB12D	PB13D	PB17D	I/O
AD16	PB12A	PB13A	PB14A	PB18A	I/O-$\overline{LDC}$
AF17	PB12B	PB13B	PB14B	PB18B	I/O
AC17	PB12C	PB13C	PB14C	PB18C	I/O
AE18	PB12D	PB13D	PB14D	PB18D	I/O

Notes:
The pins labeled "I/O-$V_{DD}5$" are user I/Os for the OR2CxxA series, but they are connected to $V_{DD}5$ for the OR2TxxA series.

The pins labeled "V_{SS}-ETC" are the 6 x 6 array of thermal balls located at the center of the package. The balls can be attached to the ground plane of the board for Enhanced Thermal Capability (see Table 28), or they can be left unconnected.

Pin Information (continued)

Table 25. OR2C/2T10A, OR2C/2T12A, OR2C/2T15A, and OR2C/2T26A 352-Pin PBGA Pinout (continued)

Pin	2C/2T10A Pad	2C/2T12A Pad	2C/2T15A Pad	2C/2T26A Pad	Function
AD17	PB13A	PB14A	PB15A	PB19A	I/O
AF18	PB13B	PB14B	PB15B	PB19B	I/O
AE19	—	PB14C	PB15C	PB19C	I/O
AF19	PB13C	PB14D	PB15D	PB19D	I/O
AD18	PB13D	PB15A	PB16A	PB20A	I/O-$\overline{\text{INIT}}$
AE20	—	PB15B	PB16B	PB20B	I/O
AC19	PB14A	PB15C	PB16C	PB20C	I/O
AF20	—	PB15D	PB16D	PB20D	I/O
AD19	PB14B	PB16A	PB17A	PB21A	I/O-$V_{DD}5$
AE21	PB14C	PB16B	PB17B	PB21B	I/O
AC20	PB14D	PB16C	PB17C	PB21C	I/O
AF21	PB15A	PB16D	PB17D	PB21D	I/O
AD20	PB15B	PB17A	PB18A	PB22A	I/O
AE22	PB15C	PB17B	PB18B	PB22B	I/O
AF22	PB15D	PB17C	PB18D	PB22D	I/O
AD21	PB16A	PB17D	PB19A	PB23A	I/O
AE23	—	—	PB19C	PB23B	I/O
AC22	PB16B	PB18A	PB19D	PB23D	I/O
AF23	PB16C	PB18B	PB20A	PB24A	I/O
AD22	PB16D	PB18C	PB20B	PB24B	I/O
AE24	—	—	PB20C	PB24C	I/O
AD23	—	PB18D	PB20D	PB24D	I/O
AF24	DONE	DONE	DONE	DONE	DONE
AE26	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$
AD25	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$
AD26	PR16A	PR18A	PR20A	PR24A	I/O-M0
AC25	PR16B	PR18B	PR20C	PR24C	I/O
AC24	PR16C	PR18C	PR20D	PR24D	I/O
AC26	PR16D	PR18D	PR19A	PR23A	I/O
AB25	PR15A	PR17A	PR19D	PR23D	I/O
AB23	PR15B	PR17B	PR18A	PR22A	I/O
AB24	PR15C	PR17C	PR18B	PR22B	I/O
AB26	PR15D	PR17D	PR18D	PR22D	I/O
AA25	PR14A	PR16A	PR17A	PR21A	I/O
Y23	PR14B	PR16B	PR17B	PR21B	I/O
AA24	PR14C	PR16C	PR17C	PR21C	I/O
AA26	—	PR16D	PR17D	PR21D	I/O
Y25	PR14D	PR15A	PR16A	PR20A	I/O
Y26	—	PR15B	PR16B	PR20B	I/O
Y24	PR13A	PR15C	PR16C	PR20C	I/O
W25	PR13B	PR15D	PR16D	PR20D	I/O-M1
V23	PR13C	PR14A	PR15A	PR19A	I/O
W26	—	PR14B	PR15B	PR19B	I/O

Notes:
The pins labeled "I/O-$V_{DD}5$" are user I/Os for the OR2CxxA series, but they are connected to $V_{DD}5$ for the OR2TxxA series.

The pins labeled "V_{SS}-ETC" are the 6 x 6 array of thermal balls located at the center of the package. The balls can be attached to the ground plane of the board for Enhanced Thermal Capability (see Table 28), or they can be left unconnected.

Pin Information (continued)

Table 25. OR2C/2T10A, OR2C/2T12A, OR2C/2T15A, and OR2C/2T26A 352-Pin PBGA Pinout (continued)

Pin	2C/2T10A Pad	2C/2T12A Pad	2C/2T15A Pad	2C/2T26A Pad	Function
W24	PR13D	PR14C	PR15C	PR19C	I/O
V25	PR12A	PR14D	PR15D	PR19D	I/O-VDD5
V26	PR12B	PR13A	PR14A	PR18A	I/O
U25	—	PR13B	PR14B	PR18B	I/O
V24	PR12C	PR13C	PR14C	PR18C	I/O
U26	PR12D	PR13D	PR14D	PR18D	I/O
U23	PR11A	PR12A	PR13A	PR17A	I/O-M2
T25	PR11B	PR12B	PR13B	PR17D	I/O
U24	PR11C	PR12C	PR13C	PR16A	I/O
T26	PR11D	PR12D	PR13D	PR16D	I/O
R25	PR10A	PR11A	PR12A	PR15A	I/O-M3
R26	PR10B	PR11B	PR12B	PR15D	I/O
T24	PR10C	PR11C	PR12C	PR14A	I/O
P25	PR10D	PR11D	PR12D	PR14D	I/O
R23	PR9A	PR10A	PR11A	PR13A	I/O
P26	PR9B	PR10B	PR11B	PR13B	I/O
R24	PR9C	PR10C	PR11C	PR13C	I/O
N25	PR9D	PR10D	PR11D	PR13D	I/O
N23	PR8A	PR9A	PR10A	PR12A	I/O
N26	PR8B	PR9B	PR10B	PR12B	I/O
P24	PR8C	PR9C	PR10C	PR12C	I/O
M25	PR8D	PR9D	PR10D	PR12D	I/O
N24	PR7A	PR8A	PR9A	PR11A	I/O-VDD5
M26	PR7B	PR8B	PR9B	PR11D	I/O
L25	PR7C	PR8C	PR9C	PR10A	I/O
M24	PR7D	PR8D	PR9D	PR10D	I/O
L26	PR6A	PR7A	PR8A	PR9A	I/O-CS1
M23	PR6B	PR7B	PR8B	PR9D	I/O
K25	PR6C	PR7C	PR8C	PR8A	I/O
L24	PR6D	PR7D	PR8D	PR8D	I/O
K26	PR5A	PR6A	PR7A	PR7A	I/O-$\overline{CS0}$
K23	PR5B	PR6B	PR7B	PR7B	I/O
J25	PR5C	PR6C	PR7C	PR7C	I/O
K24	PR5D	PR6D	PR7D	PR7D	I/O
J26	PR4A	PR5A	PR6A	PR6A	I/O
H25	PR4B	PR5B	PR6B	PR6B	I/O
H26	PR4C	PR5C	PR6C	PR6C	I/O
J24	PR4D	PR5D	PR6D	PR6D	I/O
G25	PR3A	PR4A	PR5A	PR5A	I/O-$\overline{RD}$
H23	PR3B	PR4B	PR5B	PR5B	I/O
G26	—	PR4C	PR5C	PR5C	I/O
H24	PR3C	PR4D	PR5D	PR5D	I/O

Notes:
The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

The pins labeled "VSS-ETC" are the 6 x 6 array of thermal balls located at the center of the package. The balls can be attached to the ground plane of the board for Enhanced Thermal Capability (see Table 28), or they can be left unconnected.

Pin Information (continued)

Table 25. OR2C/2T10A, OR2C/2T12A, OR2C/2T15A, and OR2C/2T26A 352-Pin PBGA Pinout (continued)

Pin	2C/2T10A Pad	2C/2T12A Pad	2C/2T15A Pad	2C/2T26A Pad	Function
F25	PR3D	PR3A	PR4A	PR4A	I/O-VDD5
G23	—	PR3B	PR4B	PR4B	I/O
F26	—	PR3C	PR4C	PR4C	I/O
G24	—	PR3D	PR4D	PR4D	I/O
E25	PR2A	PR2A	PR3A	PR3A	I/O-$\overline{WR}$
E26	PR2B	PR2B	PR3B	PR3B	I/O
F24	—	—	PR3D	PR3D	I/O
D25	PR2C	PR2C	PR2A	PR2A	I/O
E23	PR2D	PR2D	PR2D	PR2D	I/O
D26	PR1A	PR1A	PR1A	PR1A	I/O
E24	PR1B	PR1B	PR1B	PR1B	I/O
C25	PR1C	PR1C	PR1C	PR1C	I/O
D24	PR1D	PR1D	PR1D	PR1D	I/O
C26	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$
A25	PT16D	PT18D	PT20D	PT24D	I/O
B24	PT16C	PT18C	PT20C	PT24C	I/O
A24	—	—	PT20B	PT24B	I/O
B23	PT16B	PT18B	PT20A	PT24A	I/O
C23	PT16A	PT18A	PT19D	PT23D	I/O
A23	PT15D	PT17D	PT19A	PT23A	I/O-RDY/RCLK
B22	PT15C	PT17C	PT18D	PT22D	I/O
D22	PT15B	PT17B	PT18C	PT22C	I/O
C22	PT15A	PT17A	PT18A	PT22A	I/O
A22	PT14D	PT16D	PT17D	PT21D	I/O
B21	PT14C	PT16C	PT17C	PT21C	I/O
D20	PT14B	PT16B	PT17B	PT21B	I/O
C21	PT14A	PT16A	PT17A	PT21A	I/O
A21	PT13D	PT15D	PT16D	PT20D	I/O-D7
B20	—	PT15C	PT16C	PT20C	I/O
A20	PT13C	PT15B	PT16B	PT20B	I/O
C20	—	PT15A	PT16A	PT20A	I/O
B19	PT13B	PT14D	PT15D	PT19D	I/O-VDD5
D18	—	PT14C	PT15C	PT19C	I/O
A19	PT13A	PT14B	PT15B	PT19B	I/O
C19	—	PT14A	PT15A	PT19A	I/O
B18	PT12D	PT13D	PT14D	PT18D	I/O
A18	PT12C	PT13C	PT14C	PT18C	I/O
B17	PT12B	PT13B	PT14B	PT18B	I/O-D6
C18	PT12A	PT13A	PT14A	PT18A	I/O
A17	PT11D	PT12D	PT13D	PT17D	I/O
D17	PT11C	PT12C	PT13C	PT17A	I/O
B16	PT11B	PT12B	PT13B	PT16D	I/O
C17	PT11A	PT12A	PT13A	PT16A	I/O-D5
A16	PT10D	PT11D	PT12D	PT15D	I/O

Notes:
The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

The pins labeled "VSS-ETC" are the 6 x 6 array of thermal balls located at the center of the package. The balls can be attached to the ground plane of the board for Enhanced Thermal Capability (see Table 28), or they can be left unconnected.

Pin Information (continued)

Table 25. OR2C/2T10A, OR2C/2T12A, OR2C/2T15A, and OR2C/2T26A 352-Pin PBGA Pinout (continued)

Pin	2C/2T10A Pad	2C/2T12A Pad	2C/2T15A Pad	2C/2T26A Pad	Function
B15	PT10C	PT11C	PT12C	PT15A	I/O
A15	PT10B	PT11B	PT12B	PT14D	I/O
C16	PT10A	PT11A	PT12A	PT14A	I/O-D4
B14	PT9D	PT10D	PT11D	PT13D	I/O
D15	PT9C	PT10C	PT11C	PT13C	I/O
A14	PT9B	PT10B	PT11B	PT13B	I/O
C15	PT9A	PT10A	PT11A	PT13A	I/O-D3
B13	PT8D	PT9D	PT10D	PT12D	I/O
D13	PT8C	PT9C	PT10C	PT12C	I/O
A13	PT8B	PT9B	PT10B	PT12B	I/O-VDD5
C14	PT8A	PT9A	PT10A	PT12A	I/O-D2
B12	PT7D	PT8D	PT9D	PT11D	I/O-D1
C13	PT7C	PT8C	PT9C	PT11A	I/O
A12	PT7B	PT8B	PT9B	PT10D	I/O
B11	PT7A	PT8A	PT9A	PT10A	I/O-D0/DIN
C12	PT6D	PT7D	PT8D	PT9D	I/O
A11	PT6C	PT7C	PT8C	PT9A	I/O
D12	PT6B	PT7B	PT8B	PT8D	I/O
B10	PT6A	PT7A	PT8A	PT8A/	I/O-DOUT
C11	PT5D	PT6D	PT7D	PT7D	I/O
A10	PT5C	PT6C	PT7C	PT7C	I/O
D10	PT5B	PT6B	PT7B	PT7B	I/O
B9	PT5A	PT6A	PT7A	PT7A	I/O
C10	PT4D	PT5D	PT6D	PT6D	I/O
A9	PT4C	PT5C	PT6C	PT6C	I/O
B8	PT4B	PT5B	PT6B	PT6B	I/O
A8	PT4A	PT5A	PT6A	PT6A	I/O-TDI
C9	—	PT4D	PT5D	PT5D	I/O
B7	PT3D	PT4C	PT5C	PT5C	I/O
D8	—	PT4B	PT5B	PT5B	I/O
A7	PT3C	PT4A	PT5A	PT5A	I/O-VDD5
C8	—	PT3D	PT4D	PT4D	I/O
B6	PT3B	PT3C	PT4C	PT4C	I/O
D7	—	PT3B	PT4B	PT4B	I/O
A6	PT3A	PT3A	PT4A	PT4A	I/O-TMS
C7	PT2D	PT2D	PT3D	PT3D	I/O
B5	PT2C	PT2C	PT3A	PT3A	I/O
A5	PT2B	PT2B	PT2D	PT2D	I/O
C6	—	—	PT2C	PT2C	I/O
B4	—	—	PT2B	PT2B	I/O
D5	PT2A	PT2A	PT2A	PT2A	I/O
A4	PT1D	PT1D	PT1D	PT1D	I/O
C5	PT1C	PT1C	PT1C	PT1C	I/O
B3	PT1B	PT1B	PT1B	PT1B	I/O

Notes:
The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

The pins labeled "VSS-ETC" are the 6 x 6 array of thermal balls located at the center of the package. The balls can be attached to the ground plane of the board for Enhanced Thermal Capability (see Table 28), or they can be left unconnected.

Pin Information (continued)

Table 25. OR2C/2T10A, OR2C/2T12A, OR2C/2T15A, and OR2C/2T26A 352-Pin PBGA Pinout (continued)

Pin	2C/2T10A Pad	2C/2T12A Pad	2C/2T15A Pad	2C/2T26A Pad	Function
C4	PT1A	PT1A	PT1A	PT1A	I/O-TCK
A3	RD_DATA/TDO	RD_DATA/TDO	RD_DATA/TDO	RD_DATA/TDO	RD_DATA/TDO
A1	VSS	VSS	VSS	VSS	VSS
A2	VSS	VSS	VSS	VSS	VSS
A26	VSS	VSS	VSS	VSS	VSS
AC13	VSS	VSS	VSS	VSS	VSS
AC18	VSS	VSS	VSS	VSS	VSS
AC23	VSS	VSS	VSS	VSS	VSS
AC4	VSS	VSS	VSS	VSS	VSS
AC8	VSS	VSS	VSS	VSS	VSS
AD24	VSS	VSS	VSS	VSS	VSS
AD3	VSS	VSS	VSS	VSS	VSS
AE1	VSS	VSS	VSS	VSS	VSS
AE2	VSS	VSS	VSS	VSS	VSS
AE25	VSS	VSS	VSS	VSS	VSS
AF1	VSS	VSS	VSS	VSS	VSS
AF25	VSS	VSS	VSS	VSS	VSS
AF26	VSS	VSS	VSS	VSS	VSS
B2	VSS	VSS	VSS	VSS	VSS
B25	VSS	VSS	VSS	VSS	VSS
B26	VSS	VSS	VSS	VSS	VSS
C24	VSS	VSS	VSS	VSS	VSS
C3	VSS	VSS	VSS	VSS	VSS
D14	VSS	VSS	VSS	VSS	VSS
D19	VSS	VSS	VSS	VSS	VSS
D23	VSS	VSS	VSS	VSS	VSS
D4	VSS	VSS	VSS	VSS	VSS
D9	VSS	VSS	VSS	VSS	VSS
H4	VSS	VSS	VSS	VSS	VSS
J23	VSS	VSS	VSS	VSS	VSS
N4	VSS	VSS	VSS	VSS	VSS
P23	VSS	VSS	VSS	VSS	VSS
V4	VSS	VSS	VSS	VSS	VSS
W23	VSS	VSS	VSS	VSS	VSS
AA23	VDD	VDD	VDD	VDD	VDD
AA4	VDD	VDD	VDD	VDD	VDD
AC11	VDD	VDD	VDD	VDD	VDD
AC16	VDD	VDD	VDD	VDD	VDD
AC21	VDD	VDD	VDD	VDD	VDD
AC6	VDD	VDD	VDD	VDD	VDD
D11	VDD	VDD	VDD	VDD	VDD
D16	VDD	VDD	VDD	VDD	VDD
D21	VDD	VDD	VDD	VDD	VDD
D6	VDD	VDD	VDD	VDD	VDD

Notes:
The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

The pins labeled "VSS-ETC" are the 6 x 6 array of thermal balls located at the center of the package. The balls can be attached to the ground plane of the board for Enhanced Thermal Capability (see Table 28), or they can be left unconnected.

Pin Information (continued)

Table 25. OR2C/2T10A, OR2C/2T12A, OR2C/2T15A, and OR2C/2T26A 352-Pin PBGA Pinout (continued)

Pin	2C/2T10A Pad	2C/2T12A Pad	2C/2T15A Pad	2C/2T26A Pad	Function
F23	VDD	VDD	VDD	VDD	VDD
F4	VDD	VDD	VDD	VDD	VDD
L23	VDD	VDD	VDD	VDD	VDD
L4	VDD	VDD	VDD	VDD	VDD
T23	VDD	VDD	VDD	VDD	VDD
T4	VDD	VDD	VDD	VDD	VDD
L11	VSS	VSS	VSS	VSS	VSS—ETC
L12	VSS	VSS	VSS	VSS	VSS—ETC
L13	VSS	VSS	VSS	VSS	VSS—ETC
L14	VSS	VSS	VSS	VSS	VSS—ETC
L15	VSS	VSS	VSS	VSS	VSS—ETC
L16	VSS	VSS	VSS	VSS	VSS—ETC
M11	VSS	VSS	VSS	VSS	VSS—ETC
M12	VSS	VSS	VSS	VSS	VSS—ETC
M13	VSS	VSS	VSS	VSS	VSS—ETC
M14	VSS	VSS	VSS	VSS	VSS—ETC
M15	VSS	VSS	VSS	VSS	VSS—ETC
M16	VSS	VSS	VSS	VSS	VSS—ETC
N11	VSS	VSS	VSS	VSS	VSS—ETC
N12	VSS	VSS	VSS	VSS	VSS—ETC
N13	VSS	VSS	VSS	VSS	VSS—ETC
N14	VSS	VSS	VSS	VSS	VSS—ETC
N15	VSS	VSS	VSS	VSS	VSS—ETC
N16	VSS	VSS	VSS	VSS	VSS—ETC
P11	VSS	VSS	VSS	VSS	VSS—ETC
P12	VSS	VSS	VSS	VSS	VSS—ETC
P13	VSS	VSS	VSS	VSS	VSS—ETC
P14	VSS	VSS	VSS	VSS	VSS—ETC
P15	VSS	VSS	VSS	VSS	VSS—ETC
P16	VSS	VSS	VSS	VSS	VSS—ETC
R11	VSS	VSS	VSS	VSS	VSS—ETC
R12	VSS	VSS	VSS	VSS	VSS—ETC
R13	VSS	VSS	VSS	VSS	VSS—ETC
R14	VSS	VSS	VSS	VSS	VSS—ETC
R15	VSS	VSS	VSS	VSS	VSS—ETC
R16	VSS	VSS	VSS	VSS	VSS—ETC
T11	VSS	VSS	VSS	VSS	VSS—ETC
T12	VSS	VSS	VSS	VSS	VSS—ETC
T13	VSS	VSS	VSS	VSS	VSS—ETC
T14	VSS	VSS	VSS	VSS	VSS—ETC
T15	VSS	VSS	VSS	VSS	VSS—ETC
T16	VSS	VSS	VSS	VSS	VSS—ETC

Notes:
The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

The pins labeled "VSS-ETC" are the 6 x 6 array of thermal balls located at the center of the package. The balls can be attached to the ground plane of the board for Enhanced Thermal Capability (see Table 28), or they can be left unconnected.

Pin Information (continued)

Table 26. OR2C/2T15A, OR2C/2T26A, and OR2C/2T40A 432-Pin EBGA Pinout

Pin	2C/2T15A Pad	2C/2T26A Pad	2C/2T40A Pad	Function
E28	PL1D	PL1D	PL1D	I/O
D29	PL1C	PL1C	PL1A	I/O
D30	PL1B	PL1B	PL2D	I/O
D31	PL1A	PL1A	PL2A	I/O
F28	PL2D	PL2D	PL3D	I/O-A0
E29	PL2C	PL2C	PL3C	I/O
E30	PL2B	PL2B	PL3B	I/O
E31	PL2A	PL2A	PL3A	I/O
F29	PL3D	PL3D	PL4D	I/O
F30	PL3C	PL3C	PL4C	I/O
F31	PL3B	PL3B	PL4B	I/O
H28	PL3A	PL3A	PL4A	I/O
G29	PL4D	PL4D	PL5D	I/O-VDD5
G30	PL4C	PL4C	PL5C	I/O
G31	PL4B	PL4B	PL5B	I/O
J28	PL4A	PL4A	PL6D	I/O
H29	PL5D	PL5D	PL7D	I/O
H30	PL5C	PL5C	PL7C	I/O
J29	PL5B	PL5B	PL7B	I/O
K28	PL5A	PL5A	PL8D	I/O-A1
J30	PL6D	PL6D	PL9D	I/O
J31	PL6C	PL6C	PL9C	I/O
K29	PL6B	PL6B	PL9B	I/O
K30	PL6A	PL6A	PL9A	I/O-A2
K31	PL7D	PL7D	PL10D	I/O
L29	PL7C	PL7C	PL10C	I/O
M28	PL7B	PL7B	PL10B	I/O
L30	PL7A	PL7A	PL10A	I/O-A3
L31	—	PL8D	PL11D	I/O-VDD5
M29	PL8D	PL8C	PL11C	I/O
N28	PL8C	PL8A	PL11A	I/O
M30	PL8B	PL9D	PL12D	I/O
N29	—	PL9C	PL12C	I/O
N30	PL8A	PL9A	PL12A	I/O-A4
P28	PL9D	PL10D	PL13D	I/O-A5
N31	—	PL10C	PL13C	I/O
P29	PL9C	PL10A	PL13A	I/O
P30	PL9B	PL11D	PL14D	I/O
P31	PL9A	PL11A	PL14A	I/O-A6
R29	PL10D	PL12D	PL15D	I/O
R30	PL10C	PL12C	PL15C	I/O
R31	PL10B	PL12B	PL15B	I/O
T29	PL10A	PL12A	PL15A	I/O-A7
T28	PL11D	PL13D	PL16D	I/O
T30	PL11C	PL13C	PL16C	I/O-VDD5

Notes:
The OR2T15A pin AG2 is not connected in the 432-pin EBGA package.

The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

Pin Information (continued)

Table 26. OR2C/2T15A, OR2C/2T26A, and OR2C/2T40A 432-Pin EBGA Pinout (continued)

Pin	2C/2T15A Pad	2C/2T26A Pad	2C/2T40A Pad	Function
U31	PL11B	PL13B	PL16B	I/O
U30	PL11A	PL13A	PL16A	I/O-A8
U29	PL12D	PL14D	PL17D	I/O-A9
V31	—	PL14C	PL17C	I/O
V30	PL12C	PL14A	PL17A	I/O
V29	PL12B	PL15D	PL18D	I/O
W31	—	PL15C	PL18C	I/O
V28	PL12A	PL15A	PL18A	I/O-A10
W30	PL13D	PL16D	PL19D	I/O
W29	—	PL16C	PL19C	I/O
Y30	PL13C	PL16A	PL19A	I/O
W28	PL13B	PL17D	PL20D	I/O
Y29	PL13A	PL17A	PL20A	I/O-A11
AA31	PL14D	PL18D	PL21D	I/O-A12
AA30	PL14C	PL18C	PL21C	I/O
Y28	PL14B	PL18B	PL21B	I/O
AA29	PL14A	PL18A	PL21A	I/O
AB31	PL15D	PL19D	PL22D	I/O
AB30	PL15C	PL19C	PL22C	I/O
AB29	PL15B	PL19B	PL22B	I/O-A13
AC31	PL15A	PL19A	PL22A	I/O
AC30	PL16D	PL20D	PL23D	I/O
AB28	PL16C	PL20C	PL23C	I/O
AC29	PL16B	PL20B	PL24D	I/O
AD30	PL16A	PL20A	PL25D	I/O
AD29	PL17D	PL21D	PL25A	I/O-A14
AC28	PL17C	PL21C	PL26C	I/O
AE31	PL17B	PL21B	PL26B	I/O
AE30	PL17A	PL21A	PL26A	I/O
AE29	PL18D	PL22D	PL27D	I/O-V_{DD5}
AD28	PL18C	PL22C	PL27C	I/O
AF31	PL18B	PL22B	PL27B	I/O
AF30	PL18A	PL22A	PL27A	I/O
AF29	PL19D	PL23D	PL28D	I/O
AG31	PL19C	PL23C	PL28C	I/O
AG30	PL19B	PL23B	PL28B	I/O
AG29	PL19A	PL23A	PL28A	I/O
AF28	PL20D	PL24D	PL29A	I/O
AH31	PL20C	PL24C	PL30C	I/O
AH30	PL20B	PL24B	PL30B	I/O
AH29	PL20A	PL24A	PL30A	I/O-A15
AG28	CCLK	CCLK	CCLK	CCLK
AH27	PB1A	PB1A	PB1A	I/O-A16
AJ28	PB1B	PB1B	PB1B	I/O
AK28	PB1C	PB1C	PB2A	I/O

Notes:
The OR2T15A pin AG2 is not connected in the 432-pin EBGA package.

The pins labeled "I/O-V_{DD5}" are user I/Os for the OR2CxxA series, but they are connected to V_{DD5} for the OR2TxxA series.

Pin Information (continued)

Table 26. OR2C/2T15A, OR2C/2T26A, and OR2C/2T40A 432-Pin EBGA Pinout (continued)

Pin	2C/2T15A Pad	2C/2T26A Pad	2C/2T40A Pad	Function
AL28	PB1D	PB1D	PB2D	I/O
AH26	PB2A	PB2A	PB3A	I/O
AJ27	PB2B	PB2B	PB3B	I/O
AK27	PB2C	PB2C	PB3C	I/O
AL27	PB2D	PB2D	PB3D	I/O-VDD5
AJ26	PB3A	PB3A	PB4A	I/O
AK26	PB3B	PB3B	PB4B	I/O
AL26	PB3C	PB3C	PB4C	I/O
AH24	PB3D	PB3D	PB4D	I/O
AJ25	PB4A	PB4A	PB5A	I/O
AK25	PB4B	PB4B	PB5B	I/O
AL25	PB4C	PB4C	PB5C	I/O
AH23	PB4D	PB4D	PB5D	I/O-A17
AJ24	PB5A	PB5A	PB6A	I/O
AK24	PB5B	PB5B	PB6B	I/O
AJ23	PB5C	PB5C	PB6C	I/O
AH22	PB5D	PB5D	PB6D	I/O
AK23	PB6A	PB6A	PB7A	I/O
AL23	PB6B	PB6B	PB7D	I/O
AJ22	PB6C	PB6C	PB8A	I/O
AK22	PB6D	PB6D	PB8D	I/O
AL22	PB7A	PB7A	PB9A	I/O
AJ21	PB7B	PB7B	PB9D	I/O
AH20	PB7C	PB7C	PB10A	I/O
AK21	PB7D	PB7D	PB10D	I/O
AL21	—	PB8A	PB11A	I/O-VDD5
AJ20	PB8A	PB8B	PB11B	I/O
AH19	PB8B	PB8D	PB11D	I/O
AK20	PB8C	PB9A	PB12A	I/O
AJ19	—	PB9B	PB12B	I/O
AK19	PB8D	PB9D	PB12D	I/O
AH18	PB9A	PB10A	PB13A	I/O
AL19	PB9B	PB10D	PB13D	I/O
AJ18	PB9C	PB11A	PB14A	I/O
AK18	—	PB11B	PB14B	I/O
AL18	PB9D	PB11D	PB14D	I/O
AJ17	PB10A	PB12A	PB15A	I/O
AK17	PB10B	PB12B	PB15B	I/O
AL17	PB10C	PB12C	PB15C	I/O
AJ16	PB10D	PB12D	PB15D	I/O
AH16	PB11A	PB13A	PB16A	I/O
AK16	PB11B	PB13B	PB16B	I/O
AL15	PB11C	PB13C	PB16C	I/O
AK15	PB11D	PB13D	PB16D	I/O
AJ15	PB12A	PB14A	PB17A	I/O-VDD5

Notes:
The OR2T15A pin AG2 is not connected in the 432-pin EBGA package.

The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

Pin Information (continued)

Table 26. OR2C/2T15A, OR2C/2T26A, and OR2C/2T40A 432-Pin EBGA Pinout (continued)

Pin	2C/2T15A Pad	2C/2T26A Pad	2C/2T40A Pad	Function
AL14	PB12B	PB14D	PB17D	I/O
AK14	PB12C	PB15A	PB18A	I/O
AJ14	—	PB15B	PB18B	I/O
AL13	PB12D	PB15D	PB18D	I/O
AH14	PB13A	PB16A	PB19A	I/O-HDC
AK13	—	PB16B	PB19B	I/O
AJ13	PB13B	PB16D	PB19D	I/O
AK12	PB13C	PB17A	PB20A	I/O
AH13	—	PB17B	PB20B	I/O
AJ12	PB13D	PB17D	PB20D	I/O
AL11	PB14A	PB18A	PB21A	I/O-$\overline{\text{LDC}}$
AK11	PB14B	PB18B	PB21D	I/O
AH12	PB14C	PB18C	PB22A	I/O
AJ11	PB14D	PB18D	PB22D	I/O
AL10	PB15A	PB19A	PB23A	I/O
AK10	PB15B	PB19B	PB24A	I/O
AJ10	PB15C	PB19C	PB24C	I/O
AL9	PB15D	PB19D	PB24D	I/O
AK9	PB16A	PB20A	PB25A	I/O-$\overline{\text{INIT}}$
AH10	PB16B	PB20B	PB25B	I/O
AJ9	PB16C	PB20C	PB25C	I/O
AK8	PB16D	PB20D	PB25D	I/O
AJ8	PB17A	PB21A	PB26A	I/O-$V_{DD}5$
AH9	PB17B	PB21B	PB26B	I/O
AL7	PB17C	PB21C	PB26C	I/O
AK7	PB17D	PB21D	PB26D	I/O
AJ7	PB18A	PB22A	PB27A	I/O
AH8	PB18B	PB22B	PB27B	I/O
AL6	PB18C	PB22C	PB27C	I/O
AK6	PB18D	PB22D	PB27D	I/O
AJ6	PB19A	PB23A	PB28A	I/O
AL5	PB19B	PB23B	PB28B	I/O
AK5	PB19C	PB23C	PB28C	I/O
AJ5	PB19D	PB23D	PB28D	I/O
AH6	PB20A	PB24A	PB29A	I/O
AL4	PB20B	PB24B	PB29D	I/O
AK4	PB20C	PB24C	PB30C	I/O
AJ4	PB20D	PB24D	PB30D	I/O
AH5	DONE	DONE	DONE	DONE
AG4	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$
AH3	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$
AH2	PR20A	PR24A	PR30A	I/O-M0
AH1	PR20B	PR24B	PR30B	I/O
AF4	PR20C	PR24C	PR29A	I/O
AG3	PR20D	PR24D	PR29D	I/O

Notes:
The OR2T15A pin AG2 is not connected in the 432-pin EBGA package.

The pins labeled "I/O-$V_{DD}5$" are user I/Os for the OR2CxxA series, but they are connected to $V_{DD}5$ for the OR2TxxA series.

Pin Information (continued)

Table 26. OR2C/2T15A, OR2C/2T26A, and OR2C/2T40A 432-Pin EBGA Pinout (continued)

Pin	2C/2T15A Pad	2C/2T26A Pad	2C/2T40A Pad	Function
AG2	PR19A	PR23A	PR28A	I/O-VDD5
AG1	PR19B	PR23B	PR28B	I/O
AF3	PR19C	PR23C	PR28C	I/O
AF2	PR19D	PR23D	PR28D	I/O
AF1	PR18A	PR22A	PR27A	I/O
AD4	PR18B	PR22B	PR27B	I/O
AE3	PR18C	PR22C	PR27C	I/O
AE2	PR18D	PR22D	PR27D	I/O
AE1	PR17A	PR21A	PR26A	I/O
AC4	PR17B	PR21B	PR26B	I/O
AD3	PR17C	PR21C	PR26C	I/O
AD2	PR17D	PR21D	PR25A	I/O
AC3	PR16A	PR20A	PR24A	I/O
AB4	PR16B	PR20B	PR24B	I/O
AC2	PR16C	PR20C	PR24D	I/O
AC1	PR16D	PR20D	PR23D	I/O-M1
AB3	PR15A	PR19A	PR22A	I/O
AB2	PR15B	PR19B	PR22B	I/O
AB1	PR15C	PR19C	PR22C	I/O
AA3	PR15D	PR19D	PR22D	I/O-VDD5
Y4	PR14A	PR18A	PR21A	I/O
AA2	PR14B	PR18B	PR21B	I/O
AA1	PR14C	PR18C	PR21C	I/O
Y3	PR14D	PR18D	PR21D	I/O
W4	PR13A	PR17A	PR20A	I/O-M2
Y2	PR13B	PR17D	PR20D	I/O
W3	PR13C	PR16A	PR19A	I/O
W2	PR13D	PR16B	PR19B	I/O
V4	—	PR16D	PR19D	I/O
W1	PR12A	PR15A	PR18A	I/O-M3
V3	—	PR15D	PR18D	I/O
V2	PR12B	PR14A	PR17A	I/O
V1	PR12C	PR14B	PR17B	I/O
U3	PR12D	PR14D	PR17D	I/O
U2	PR11A	PR13A	PR16A	I/O
U1	PR11B	PR13B	PR16B	I/O
T3	PR11C	PR13C	PR16C	I/O
T4	PR11D	PR13D	PR16D	I/O
T2	PR10A	PR12A	PR15A	I/O
R1	PR10B	PR12B	PR15B	I/O
R2	PR10C	PR12C	PR15C	I/O
R3	PR10D	PR12D	PR15D	I/O
P1	PR9A	PR11A	PR14A	I/O-VDD5
P2	PR9B	PR11C	PR14C	I/O
P3	PR9C	PR11D	PR14D	I/O

Notes:
The OR2T15A pin AG2 is not connected in the 432-pin EBGA package.

The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

Pin Information (continued)

Table 26. OR2C/2T15A, OR2C/2T26A, and OR2C/2T40A 432-Pin EBGA Pinout (continued)

Pin	2C/2T15A Pad	2C/2T26A Pad	2C/2T40A Pad	Function
N1	—	PR10A	PR13A	I/O
P4	PR9D	PR10C	PR13C	I/O
N2	—	PR10D	PR13D	I/O
N3	PR8A	PR9A	PR12A	I/O-CS1
M2	PR8B	PR9D	PR12D	I/O
N4	PR8C	PR8A	PR11A	I/O
M3	PR8D	PR8D	PR11D	I/O
L1	PR7A	PR7A	PR10A	I/O-$\overline{CS0}$
L2	PR7B	PR7B	PR10B	I/O
M4	PR7C	PR7C	PR10C	I/O
L3	PR7D	PR7D	PR10D	I/O
K1	PR6A	PR6A	PR9A	I/O
K2	PR6B	PR6B	PR9B	I/O
K3	PR6C	PR6C	PR9C	I/O
J1	PR6D	PR6D	PR9D	I/O
J2	PR5A	PR5A	PR8A	I/O-$\overline{RD}$
K4	PR5B	PR5B	PR7A	I/O
J3	PR5C	PR5C	PR7C	I/O
H2	PR5D	PR5D	PR6A	I/O
H3	PR4A	PR4A	PR5A	I/O-$V_{DD}5$
J4	PR4B	PR4B	PR5B	I/O
G1	PR4C	PR4C	PR5C	I/O
G2	PR4D	PR4D	PR5D	I/O
G3	PR3A	PR3A	PR4A	I/O-$\overline{WR}$
H4	PR3B	PR3B	PR4B	I/O
F1	PR3C	PR3C	PR4C	I/O
F2	PR3D	PR3D	PR4D	I/O
F3	PR2A	PR2A	PR3A	I/O
E1	PR2B	PR2B	PR3B	I/O
E2	PR2C	PR2C	PR3C	I/O
E3	PR2D	PR2D	PR3D	I/O
F4	PR1A	PR1A	PR2A	I/O
D1	PR1B	PR1B	PR2D	I/O
D2	PR1C	PR1C	PR1A	I/O
D3	PR1D	PR1D	PR1D	I/O
E4	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$
D5	PT20D	PT24D	PT30D	I/O
C4	PT20C	PT24C	PT30A	I/O
B4	PT20B	PT24B	PT29B	I/O
A4	PT20A	PT24A	PT29A	I/O
D6	PT19D	PT23D	PT28D	I/O
C5	PT19C	PT23C	PT28C	I/O
B5	PT19B	PT23B	PT28B	I/O
A5	PT19A	PT23A	PT28A	I/O-RDY/RCLK
C6	PT18D	PT22D	PT27D	I/O

Notes:
The OR2T15A pin AG2 is not connected in the 432-pin EBGA package.

The pins labeled "I/O-$V_{DD}5$" are user I/Os for the OR2CxxA series, but they are connected to $V_{DD}5$ for the OR2TxxA series.

Pin Information (continued)

Table 26. OR2C/2T15A, OR2C/2T26A, and OR2C/2T40A 432-Pin EBGA Pinout (continued)

Pin	2C/2T15A Pad	2C/2T26A Pad	2C/2T40A Pad	Function
B6	PT18C	PT22C	PT27C	I/O
A6	PT18B	PT22B	PT27B	I/O
D8	PT18A	PT22A	PT27A	I/O
C7	PT17D	PT21D	PT26D	I/O
B7	PT17C	PT21C	PT26C	I/O
A7	PT17B	PT21B	PT26B	I/O
D9	PT17A	PT21A	PT26A	I/O
C8	PT16D	PT20D	PT25D	I/O-D7
B8	PT16C	PT20C	PT25C	I/O
C9	PT16B	PT20B	PT25B	I/O
D10	PT16A	PT20A	PT25A	I/O
B9	PT15D	PT19D	PT24D	I/O-VDD5
A9	PT15C	PT19C	PT24C	I/O
C10	PT15B	PT19B	PT24B	I/O
B10	PT15A	PT19A	PT23D	I/O
A10	PT14D	PT18D	PT22D	I/O
C11	PT14C	PT18C	PT22A	I/O
D12	PT14B	PT18B	PT21D	I/O-D6
B11	PT14A	PT18A	PT21A	I/O
A11	PT13D	PT17D	PT20D	I/O
C12	PT13C	PT17A	PT20A	I/O
D13	—	PT16D	PT19D	I/O-VDD5
B12	PT13B	PT16B	PT19B	I/O
C13	PT13A	PT16A	PT19A	I/O-D5
B13	PT12D	PT15D	PT18D	I/O
D14	—	PT15B	PT18B	I/O
A13	PT12C	PT15A	PT18A	I/O
C14	PT12B	PT14D	PT17D	I/O
B14	—	PT14B	PT17B	I/O
A14	PT12A	PT14A	PT17A	I/O-D4
C15	PT11D	PT13D	PT16D	I/O
B15	PT11C	PT13C	PT16C	I/O
A15	PT11B	PT13B	PT16B	I/O
C16	PT11A	PT13A	PT16A	I/O-D3
D16	PT10D	PT12D	PT15D	I/O
B16	PT10C	PT12C	PT15C	I/O
A17	PT10B	PT12B	PT15B	I/O-VDD5
B17	PT10A	PT12A	PT15A	I/O-D2
C17	PT9D	PT11D	PT14D	D1
A18	—	PT11C	PT14C	I/O
B18	PT9C	PT11A	PT14A	I/O
C18	PT9B	PT10D	PT13D	I/O
A19	—	PT10C	PT13C	I/O
D18	PT9A	PT10A	PT13A	I/O-D0/DIN
B19	PT8D	PT9D	PT12D	I/O

Notes:
The OR2T15A pin AG2 is not connected in the 432-pin EBGA package.

The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

Pin Information (continued)

Table 26. OR2C/2T15A, OR2C/2T26A, and OR2C/2T40A 432-Pin EBGA Pinout (continued)

Pin	2C/2T15A Pad	2C/2T26A Pad	2C/2T40A Pad	Function
C19	—	PT9C	PT12C	I/O
B20	PT8C	PT9A	PT12A	I/O
D19	PT8B	PT8D	PT11D	I/O
C20	PT8A	PT8A	PT11A	I/O-DOUT
A21	PT7D	PT7D	PT10D	I/O
B21	PT7C	PT7C	PT10A	I/O
D20	PT7B	PT7B	PT9D	I/O
C21	PT7A	PT7A	PT9A	I/O
A22	PT6D	PT6D	PT8D	I/O
B22	PT6C	PT6C	PT8A	I/O
C22	PT6B	PT6B	PT7D	I/O
A23	PT6A	PT6A	PT7A	I/O-TDI
B23	PT5D	PT5D	PT6D	I/O
D22	PT5C	PT5C	PT6C	I/O
C23	PT5B	PT5B	PT6B	I/O
B24	PT5A	PT5A	PT6A	I/O-$V_{DD}5$
C24	PT4D	PT4D	PT5D	I/O
D23	PT4C	PT4C	PT5C	I/O
A25	PT4B	PT4B	PT5B	I/O
B25	PT4A	PT4A	PT5A	I/O-TMS
C25	PT3D	PT3D	PT4D	I/O
D24	PT3C	PT3C	PT4C	I/O
A26	PT3B	PT3B	PT4B	I/O
B26	PT3A	PT3A	PT4A	I/O
C26	PT2D	PT2D	PT3D	I/O
A27	PT2C	PT2C	PT3C	I/O
B27	PT2B	PT2B	PT3B	I/O
C27	PT2A	PT2A	PT3A	I/O
D26	PT1D	PT1D	PT2D	I/O
A28	PT1C	PT1C	PT2A	I/O
B28	PT1B	PT1B	PT1D	I/O
C28	PT1A	PT1A	PT1A	I/O-TCK
D27	RD_DATA/TDO	RD_DATA/TDO	RD_DATA/TDO	RD_DATA/TDO
A12	V_{SS}	V_{SS}	V_{SS}	V_{SS}
A16	V_{SS}	V_{SS}	V_{SS}	V_{SS}
A2	V_{SS}	V_{SS}	V_{SS}	V_{SS}
A20	V_{SS}	V_{SS}	V_{SS}	V_{SS}
A24	V_{SS}	V_{SS}	V_{SS}	V_{SS}
A29	V_{SS}	V_{SS}	V_{SS}	V_{SS}
A3	V_{SS}	V_{SS}	V_{SS}	V_{SS}
A30	V_{SS}	V_{SS}	V_{SS}	V_{SS}
A8	V_{SS}	V_{SS}	V_{SS}	V_{SS}
AD1	V_{SS}	V_{SS}	V_{SS}	V_{SS}
AD31	V_{SS}	V_{SS}	V_{SS}	V_{SS}
AJ1	V_{SS}	V_{SS}	V_{SS}	V_{SS}

Notes:
The OR2T15A pin AG2 is not connected in the 432-pin EBGA package.

The pins labeled "I/O-$V_{DD}5$" are user I/Os for the OR2CxxA series, but they are connected to $V_{DD}5$ for the OR2TxxA series.

Pin Information (continued)

Table 26. OR2C/2T15A, OR2C/2T26A, and OR2C/2T40A 432-Pin EBGA Pinout (continued)

Pin	2C/2T15A Pad	2C/2T26A Pad	2C/2T40A Pad	Function
AJ2	VSS	VSS	VSS	VSS
AJ30	VSS	VSS	VSS	VSS
AJ31	VSS	VSS	VSS	VSS
AK1	VSS	VSS	VSS	VSS
AK29	VSS	VSS	VSS	VSS
AK3	VSS	VSS	VSS	VSS
AK31	VSS	VSS	VSS	VSS
AL12	VSS	VSS	VSS	VSS
AL16	VSS	VSS	VSS	VSS
AL2	VSS	VSS	VSS	VSS
AL20	VSS	VSS	VSS	VSS
AL24	VSS	VSS	VSS	VSS
AL29	VSS	VSS	VSS	VSS
AL3	VSS	VSS	VSS	VSS
AL30	VSS	VSS	VSS	VSS
AL8	VSS	VSS	VSS	VSS
B1	VSS	VSS	VSS	VSS
B29	VSS	VSS	VSS	VSS
B3	VSS	VSS	VSS	VSS
B31	VSS	VSS	VSS	VSS
C1	VSS	VSS	VSS	VSS
C2	VSS	VSS	VSS	VSS
C30	VSS	VSS	VSS	VSS
C31	VSS	VSS	VSS	VSS
H1	VSS	VSS	VSS	VSS
H31	VSS	VSS	VSS	VSS
M1	VSS	VSS	VSS	VSS
M31	VSS	VSS	VSS	VSS
T1	VSS	VSS	VSS	VSS
T31	VSS	VSS	VSS	VSS
Y1	VSS	VSS	VSS	VSS
Y31	VSS	VSS	VSS	VSS
A1	VDD	VDD	VDD	VDD
A31	VDD	VDD	VDD	VDD
AA28	VDD	VDD	VDD	VDD
AA4	VDD	VDD	VDD	VDD
AE28	VDD	VDD	VDD	VDD
AE4	VDD	VDD	VDD	VDD
AH11	VDD	VDD	VDD	VDD
AH15	VDD	VDD	VDD	VDD
AH17	VDD	VDD	VDD	VDD
AH21	VDD	VDD	VDD	VDD
AH25	VDD	VDD	VDD	VDD
AH28	VDD	VDD	VDD	VDD
AH4	VDD	VDD	VDD	VDD

Notes:
The OR2T15A pin AG2 is not connected in the 432-pin EBGA package.

The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

Pin Information (continued)

Table 26. OR2C/2T15A, OR2C/2T26A, and OR2C/2T40A 432-Pin EBGA Pinout (continued)

Pin	2C/2T15A Pad	2C/2T26A Pad	2C/2T40A Pad	Function
AH7	VDD	VDD	VDD	VDD
AJ29	VDD	VDD	VDD	VDD
AJ3	VDD	VDD	VDD	VDD
AK2	VDD	VDD	VDD	VDD
AK30	VDD	VDD	VDD	VDD
AL1	VDD	VDD	VDD	VDD
AL31	VDD	VDD	VDD	VDD
B2	VDD	VDD	VDD	VDD
B30	VDD	VDD	VDD	VDD
C29	VDD	VDD	VDD	VDD
C3	VDD	VDD	VDD	VDD
D11	VDD	VDD	VDD	VDD
D15	VDD	VDD	VDD	VDD
D17	VDD	VDD	VDD	VDD
D21	VDD	VDD	VDD	VDD
D25	VDD	VDD	VDD	VDD
D28	VDD	VDD	VDD	VDD
D4	VDD	VDD	VDD	VDD
D7	VDD	VDD	VDD	VDD
G28	VDD	VDD	VDD	VDD
G4	VDD	VDD	VDD	VDD
L28	VDD	VDD	VDD	VDD
L4	VDD	VDD	VDD	VDD
R28	VDD	VDD	VDD	VDD
R4	VDD	VDD	VDD	VDD
U28	VDD	VDD	VDD	VDD
U4	VDD	VDD	VDD	VDD

Notes:
The OR2T15A pin AG2 is not connected in the 432-pin EBGA package.

The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

Pin Information (continued)

Table 27. OR2C/2T26A and OR2C/2T40A 600-Pin EBGA Pinout

Pin	OR2C/2T26A Pad	OR2C/2T40A Pad	Function
E32	PL1D	PL1D	I/O
E33	—	PL1C	I/O
E34	PL1C	PL1A	I/O
F31	PL1B	PL2D	I/O
F32	PL1A	PL2A	I/O
F33	PL2D	PL3D	I/O-A0
F34	PL2C	PL3C	I/O
G31	PL2B	PL3B	I/O
G32	PL2A	PL3A	I/O
G33	PL3D	PL4D	I/O
G34	PL3C	PL4C	I/O
H31	PL3B	PL4B	I/O
H32	PL3A	PL4A	I/O
H33	PL4D	PL5D	I/O-VDD5
H34	PL4C	PL5C	I/O
J31	PL4B	PL5B	I/O
J32	—	PL5A	I/O
J33	PL4A	PL6D	I/O
J34	—	PL6C	I/O
J35	—	PL6B	I/O
K31	—	PL6A	I/O
K32	PL5D	PL7D	I/O
K33	PL5C	PL7C	I/O
K34	PL5B	PL7B	I/O
K35	—	PL7A	I/O
L32	PL5A	PL8D	I/O-A1
L33	—	PL8C	I/O
L34	—	PL8B	I/O
L35	—	PL8A	I/O
M31	PL6D	PL9D	I/O
M32	PL6C	PL9C	I/O
M33	PL6B	PL9B	I/O
M34	PL6A	PL9A	I/O-A2
M35	PL7D	PL10D	I/O
N31	PL7C	PL10C	I/O
N32	PL7B	PL10B	I/O
N33	PL7A	PL10A	I/O-A3
N34	PL8D	PL11D	I/O-VDD5
P31	PL8C	PL11C	I/O
P32	PL8B	PL11B	I/O
P33	PL8A	PL11A	I/O
P34	PL9D	PL12D	I/O
P35	PL9C	PL12C	I/O
R31	PL9B	PL12B	I/O
R32	PL9A	PL12A	I/O-A4
R33	PL10D	PL13D	I/O-A5
R34	PL10C	PL13C	I/O
R35	PL10B	PL13B	I/O
T32	PL10A	PL13A	I/O
T33	PL11D	PL14D	I/O
T34	PL11C	PL14C	I/O
U31	PL11B	PL14B	I/O
U32	PL11A	PL14A	I/O-A6
U33	PL12D	PL15D	I/O
U34	PL12C	PL15C	I/O
V33	PL12B	PL15B	I/O
V31	PL12A	PL15A	I/O-A7
V32	PL13D	PL16D	I/O
V34	PL13C	PL16C	I/O-VDD5
W35	PL13B	PL16B	I/O
W33	PL13A	PL16A	I/O-A8
W31	PL14D	PL17D	I/O-A9
W32	PL14C	PL17C	I/O
W34	PL14B	PL17B	I/O
Y34	PL14A	PL17A	I/O
Y33	PL15D	PL18D	I/O
Y32	PL15C	PL18C	I/O
AA35	PL15B	PL18B	I/O
AA34	PL15A	PL18A	I/O-A10
AA33	PL16D	PL19D	I/O
AA31	PL16C	PL19C	I/O
AA32	PL16B	PL19B	I/O
AB35	PL16A	PL19A	I/O
AB34	PL17D	PL20D	I/O
AB33	PL17C	PL20C	I/O
AB31	PL17B	PL20B	I/O
AB32	PL17A	PL20A	I/O-A11
AC34	PL18D	PL21D	I/O-A12
AC33	PL18C	PL21C	I/O
AC32	PL18B	PL21B	I/O
AC31	PL18A	PL21A	I/O
AD35	PL19D	PL22D	I/O
AD34	PL19C	PL22C	I/O
AD33	PL19B	PL22B	I/O-A13
AD32	PL19A	PL22A	I/O
AD31	PL20D	PL23D	I/O
AE35	PL20C	PL23C	I/O
AE34	—	PL23B	I/O
AE33	—	PL23A	I/O
AE32	PL20B	PL24D	I/O
AF35	—	PL24C	I/O
AF34	—	PL24B	I/O
AF33	—	PL24A	I/O
AF32	PL20A	PL25D	I/O

Note: The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

Pin Information (continued)

Table 27. OR2C/2T26A and OR2C/2T40A 600-Pin EBGA Pinout (continued)

Pin	OR2C/2T26A Pad	OR2C/2T40A Pad	Function
AF31	—	PL25C	I/O
AG35	—	PL25B	I/O
AG34	PL21D	PL25A	I/O-A14
AG33	—	PL26D	I/O
AG32	PL21C	PL26C	I/O
AG31	PL21B	PL26B	I/O
AH34	PL21A	PL26A	I/O
AH33	PL22D	PL27D	I/O-VDD5
AH32	PL22C	PL27C	I/O
AH31	PL22B	PL27B	I/O
AJ34	PL22A	PL27A	I/O
AJ33	PL23D	PL28D	I/O
AJ32	PL23C	PL28C	I/O
AJ31	PL23B	PL28B	I/O
AK34	PL23A	PL28A	I/O
AK33	—	PL29D	I/O
AK32	PL24D	PL29A	I/O
AK31	PL24C	PL30C	I/O
AL34	PL24B	PL30B	I/O
AL33	PL24A	PL30A	I/O-A15
AL32	CCLK	CCLK	CCLK
AM31	PB1A	PB1A	I/O-A16
AN31	PB1B	PB1B	I/O
AP31	—	PB1D	I/O
AL30	PB1C	PB2A	I/O
AM30	PB1D	PB2D	I/O
AN30	PB2A	PB3A	I/O
AP30	PB2B	PB3B	I/O
AL29	PB2C	PB3C	I/O
AM29	PB2D	PB3D	I/O-VDD5
AN29	PB3A	PB4A	I/O
AP29	PB3B	PB4B	I/O
AL28	PB3C	PB4C	I/O
AM28	PB3D	PB4D	I/O
AN28	PB4A	PB5A	I/O
AP28	PB4B	PB5B	I/O
AL27	PB4C	PB5C	I/O
AM27	PB4D	PB5D	I/O-A17
AN27	PB5A	PB6A	I/O
AP27	PB5B	PB6B	I/O
AR27	PB5C	PB6C	I/O
AL26	PB5D	PB6D	I/O
AM26	PB6A	PB7A	I/O
AN26	—	PB7B	I/O
AP26	—	PB7C	I/O
AR26	PB6B	PB7D	I/O
AM25	PB6C	PB8A	I/O
AN25	—	PB8B	I/O
AP25	—	PB8C	I/O
AR25	PB6D	PB8D	I/O
AL24	PB7A	PB9A	I/O
AM24	—	PB9B	I/O
AN24	—	PB9C	I/O
AP24	PB7B	PB9D	I/O
AR24	PB7C	PB10A	I/O
AL23	—	PB10B	I/O
AM23	—	PB10C	I/O
AN23	PB7D	PB10D	I/O
AP23	PB8A	PB11A	I/O-VDD5
AL22	PB8B	PB11B	I/O
AM22	PB8C	PB11C	I/O
AN22	PB8D	PB11D	I/O
AP22	PB9A	PB12A	I/O
AR22	PB9B	PB12B	I/O
AL21	PB9C	PB12C	I/O
AM21	PB9D	PB12D	I/O
AN21	PB10A	PB13A	I/O
AP21	PB10B	PB13B	I/O
AR21	PB10C	PB13C	I/O
AM20	PB10D	PB13D	I/O
AN20	PB11A	PB14A	I/O
AP20	PB11B	PB14B	I/O
AL19	PB11C	PB14C	I/O
AM19	PB11D	PB14D	I/O
AN19	PB12A	PB15A	I/O
AP19	PB12B	PB15B	I/O
AN18	PB12C	PB15C	I/O
AL18	PB12D	PB15D	I/O
AM18	PB13A	PB16A	I/O
AP18	PB13B	PB16B	I/O
AR17	PB13C	PB16C	I/O
AN17	PB13D	PB16D	I/O
AL17	PB14A	PB17A	I/O-VDD5
AM17	PB14B	PB17B	I/O
AP17	PB14C	PB17C	I/O
AP16	PB14D	PB17D	I/O
AN16	PB15A	PB18A	I/O
AM16	PB15B	PB18B	I/O
AR15	PB15C	PB18C	I/O
AP15	PB15D	PB18D	I/O
AN15	PB16A	PB19A	I/O-HDC
AL15	PB16B	PB19B	I/O
AM15	PB16C	PB19C	I/O
AR14	PB16D	PB19D	I/O

Note: The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

Pin Information (continued)

Table 27. OR2C/2T26A and OR2C/2T40A 600-Pin EBGA Pinout (continued)

Pin	OR2C/2T26A Pad	OR2C/2T40A Pad	Function
AP14	PB17A	PB20A	I/O
AN14	PB17B	PB20B	I/O
AL14	PB17C	PB20C	I/O
AM14	PB17D	PB20D	I/O
AP13	PB18A	PB21A	I/O-$\overline{LDC}$
AN13	—	PB21B	I/O
AM13	—	PB21C	I/O
AL13	PB18B	PB21D	I/O
AR12	PB18C	PB22A	I/O
AP12	—	PB22B	I/O
AN12	—	PB22C	I/O
AM12	PB18D	PB22D	I/O
AL12	PB19A	PB23A	I/O
AR11	—	PB23B	I/O
AP11	—	PB23C	I/O
AN11	—	PB23D	I/O
AM11	PB19B	PB24A	I/O
AR10	—	PB24B	I/O
AP10	PB19C	PB24C	I/O
AN10	PB19D	PB24D	I/O
AM10	PB20A	PB25A	I/O-$\overline{INIT}$
AL10	PB20B	PB25B	I/O
AR9	PB20C	PB25C	I/O
AP9	PB20D	PB25D	I/O
AN9	PB21A	PB26A	I/O-V_{DD5}
AM9	PB21B	PB26B	I/O
AL9	PB21C	PB26C	I/O
AP8	PB21D	PB26D	I/O
AN8	PB22A	PB27A	I/O
AM8	PB22B	PB27B	I/O
AL8	PB22C	PB27C	I/O
AP7	PB22D	PB27D	I/O
AN7	PB23A	PB28A	I/O
AM7	PB23B	PB28B	I/O
AL7	PB23C	PB28C	I/O
AP6	PB23D	PB28D	I/O
AN6	PB24A	PB29A	I/O
AM6	PB24B	PB29D	I/O
AL6	—	PB30A	I/O
AP5	PB24C	PB30C	I/O
AN5	PB24D	PB30D	I/O
AM5	DONE	DONE	DONE
AL4	$\overline{RESET}$	$\overline{RESET}$	$\overline{RESET}$
AL3	$\overline{PRGM}$	$\overline{PRGM}$	$\overline{PRGM}$
AL2	PR24A	PR30A	I/O-M0
AK5	PR24B	PR30B	I/O
AK4	PR24C	PR29A	I/O
AK3	PR24D	PR29D	I/O
AK2	PR23A	PR28A	I/O-V_{DD5}
AJ5	PR23B	PR28B	I/O
AJ4	PR23C	PR28C	I/O
AJ3	PR23D	PR28D	I/O
AJ2	PR22A	PR27A	I/O
AH5	PR22B	PR27B	I/O
AH4	PR22C	PR27C	I/O
AH3	PR22D	PR27D	I/O
AH2	PR21A	PR26A	I/O
AG5	PR21B	PR26B	I/O
AG4	PR21C	PR26C	I/O
AG3	—	PR26D	I/O
AG2	PR21D	PR25A	I/O
AG1	—	PR25B	I/O
AF5	—	PR25C	I/O
AF4	—	PR25D	I/O
AF3	PR20A	PR24A	I/O
AF2	PR20B	PR24B	I/O
AF1	—	PR24C	I/O
AE4	PR20C	PR24D	I/O
AE3	—	PR23A	I/O
AE2	—	PR23B	I/O
AE1	—	PR23C	I/O
AD5	PR20D	PR23D	I/O-M1
AD4	PR19A	PR22A	I/O
AD3	PR19B	PR22B	I/O
AD2	PR19C	PR22C	I/O
AD1	PR19D	PR22D	I/O-V_{DD5}
AC5	PR18A	PR21A	I/O
AC4	PR18B	PR21B	I/O
AC3	PR18C	PR21C	I/O
AC2	PR18D	PR21D	I/O
AB5	PR17A	PR20A	I/O-M2
AB4	PR17B	PR20B	I/O
AB3	PR17C	PR20C	I/O
AB2	PR17D	PR20D	I/O
AB1	PR16A	PR19A	I/O
AA5	PR16B	PR19B	I/O
AA4	PR16C	PR19C	I/O
AA3	PR16D	PR19D	I/O
AA2	PR15A	PR18A	I/O-M3
AA1	PR15B	PR18B	I/O
Y4	PR15C	PR18C	I/O
Y3	PR15D	PR18D	I/O
Y2	PR14A	PR17A	I/O
W5	PR14B	PR17B	I/O

Note: The pins labeled "I/O-V_{DD5}" are user I/Os for the OR2CxxA series, but they are connected to V_{DD5} for the OR2TxxA series.

Pin Information (continued)

Table 27. OR2C/2T26A and OR2C/2T40A 600-Pin EBGA Pinout (continued)

Pin	OR2C/2T26A Pad	OR2C/2T40A Pad	Function
W4	PR14C	PR17C	I/O
W3	PR14D	PR17D	I/O
W2	PR13A	PR16A	I/O
V3	PR13B	PR16B	I/O
V5	PR13C	PR16C	I/O
V4	PR13D	PR16D	I/O
V2	PR12A	PR15A	I/O
U1	PR12B	PR15B	I/O
U3	PR12C	PR15C	I/O
U5	PR12D	PR15D	I/O
U4	PR11A	PR14A	I/O-$V_{DD}5$
U2	PR11B	PR14B	I/O
T2	PR11C	PR14C	I/O
T3	PR11D	PR14D	I/O
T4	PR10A	PR13A	I/O
R1	PR10B	PR13B	I/O
R2	PR10C	PR13C	I/O
R3	PR10D	PR13D	I/O
R5	PR9A	PR12A	I/O-CS1
R4	PR9B	PR12B	I/O
P1	PR9C	PR12C	I/O
P2	PR9D	PR12D	I/O
P3	PR8A	PR11A	I/O
P5	PR8B	PR11B	I/O
P4	PR8C	PR11C	I/O
N2	PR8D	PR11D	I/O
N3	PR7A	PR10A	I/O-$\overline{CS0}$
N4	PR7B	PR10B	I/O
N5	PR7C	PR10C	I/O
M1	PR7D	PR10D	I/O
M2	PR6A	PR9A	I/O
M3	PR6B	PR9B	I/O
M4	PR6C	PR9C	I/O
M5	PR6D	PR9D	I/O
L1	PR5A	PR8A	I/O-$\overline{RD}$
L2	—	PR8B	I/O
L3	—	PR8C	I/O
L4	—	PR8D	I/O
K1	PR5B	PR7A	I/O
K2	—	PR7B	I/O
K3	PR5C	PR7C	I/O
K4	—	PR7D	I/O
K5	PR5D	PR6A	I/O
J1	—	PR6B	I/O
J2	—	PR6C	I/O
J3	—	PR6D	I/O
J4	PR4A	PR5A	I/O-$V_{DD}5$
J5	PR4B	PR5B	I/O
H2	PR4C	PR5C	I/O
H3	PR4D	PR5D	I/O
H4	PR3A	PR4A	I/O-$\overline{WR}$
H5	PR3B	PR4B	I/O
G2	PR3C	PR4C	I/O
G3	PR3D	PR4D	I/O
G4	PR2A	PR3A	I/O
G5	PR2B	PR3B	I/O
F2	PR2C	PR3C	I/O
F3	PR2D	PR3D	I/O
F4	PR1A	PR2A	I/O
F5	PR1B	PR2D	I/O
E2	PR1C	PR1A	I/O
E3	PR1D	PR1D	I/O
E4	RD_CFGN	RD_CFGN	RD_CFGN
D5	PT24D	PT30D	I/O
C5	—	PT30C	I/O
B5	PT24C	PT30A	I/O
E6	PT24B	PT29B	I/O
D6	PT24A	PT29A	I/O
C6	PT23D	PT28D	I/O
B6	PT23C	PT28C	I/O
E7	PT23B	PT28B	I/O
D7	PT23A	PT28A	I/O-RDY/RCLK
C7	PT22D	PT27D	I/O
B7	PT22C	PT27C	I/O
E8	PT22B	PT27B	I/O
D8	PT22A	PT27A	I/O
C8	PT21D	PT26D	I/O
B8	PT21C	PT26C	I/O
E9	PT21B	PT26B	I/O
D9	PT21A	PT26A	I/O
C9	PT20D	PT25D	I/O-D7
B9	PT20C	PT25C	I/O
A9	PT20B	PT25B	I/O
E10	PT20A	PT25A	I/O
D10	PT19D	PT24D	I/O-$V_{DD}5$
C10	PT19C	PT24C	I/O
B10	PT19B	PT24B	I/O
A10	—	PT24A	I/O
D11	PT19A	PT23D	I/O
C11	—	PT23C	I/O
B11	—	PT23B	I/O
A11	—	PT23A	I/O
E12	PT18D	PT22D	I/O
D12	—	PT22C	I/O

Note: The pins labeled "I/O-$V_{DD}5$" are user I/Os for the OR2CxxA series, but they are connected to $V_{DD}5$ for the OR2TxxA series.

Pin Information (continued)

Table 27. OR2C/2T26A and OR2C/2T40A 600-Pin EBGA Pinout (continued)

Pin	OR2C/2T26A Pad	OR2C/2T40A Pad	Function
C12	—	PT22B	I/O
B12	PT18C	PT22A	I/O
A12	PT18B	PT21D	I/O-D6
E13	—	PT21C	I/O
D13	—	PT21B	I/O
C13	PT18A	PT21A	I/O
B13	PT17D	PT20D	I/O
E14	PT17C	PT20C	I/O
D14	PT17B	PT20B	I/O
C14	PT17A	PT20A	I/O
B14	PT16D	PT19D	I/O-VDD5
A14	PT16C	PT19C	I/O
E15	PT16B	PT19B	I/O
D15	PT16A	PT19A	I/O-D5
C15	PT15D	PT18D	I/O
B15	PT15C	PT18C	I/O
A15	PT15B	PT18B	I/O
D16	PT15A	PT18A	I/O
C16	PT14D	PT17D	I/O
B16	PT14C	PT17C	I/O
E17	PT14B	PT17B	I/O
D17	PT14A	PT17A	I/O-D4
C17	PT13D	PT16D	I/O
B17	PT13C	PT16C	I/O
C18	PT13B	PT16B	I/O
E18	PT13A	PT16A	I/O-D3
D18	PT12D	PT15D	I/O
B18	PT12C	PT15C	I/O
A19	PT12B	PT15B	I/O-VDD5
C19	PT12A	PT15A	I/O-D2
E19	PT11D	PT14D	I/O-D1
D19	PT11C	PT14C	I/O
B19	PT11B	PT14B	I/O
B20	PT11A	PT14A	I/O
C20	PT10D	PT13D	I/O
D20	PT10C	PT13C	I/O
A21	PT10B	PT13B	I/O
B21	PT10A	PT13A	I/O-D0/DIN
C21	PT9D	PT12D	I/O
E21	PT9C	PT12C	I/O
D21	PT9B	PT12B	I/O
A22	PT9A	PT12A	I/O
B22	PT8D	PT11D	I/O
C22	PT8C	PT11C	I/O
E22	PT8B	PT11B	I/O
D22	PT8A	PT11A	I/O-DOUT
B23	PT7D	PT10D	I/O
C23	—	PT10C	I/O
D23	—	PT10B	I/O
E23	PT7C	PT10A	I/O
A24	PT7B	PT9D	I/O
B24	—	PT9C	I/O
C24	—	PT9B	I/O
D24	PT7A	PT9A	I/O
E24	PT6D	PT8D	I/O
A25	—	PT8C	I/O
B25	—	PT8B	I/O
C25	PT6C	PT8A	I/O
D25	PT6B	PT7D	I/O
A26	—	PT7C	I/O
B26	—	PT7B	I/O
C26	PT6A	PT7A	I/O-TDI
D26	PT5D	PT6D	I/O
E26	PT5C	PT6C	I/O
A27	PT5B	PT6B	I/O
B27	PT5A	PT6A	I/O-VDD5
C27	PT4D	PT5D	I/O
D27	PT4C	PT5C	I/O
E27	PT4B	PT5B	I/O
B28	PT4A	PT5A	I/O-TMS
C28	PT3D	PT4D	I/O
D28	PT3C	PT4C	I/O
E28	PT3B	PT4B	I/O
B29	PT3A	PT4A	I/O
C29	PT2D	PT3D	I/O
D29	PT2C	PT3C	I/O
E29	PT2B	PT3B	I/O
B30	PT2A	PT3A	I/O
C30	PT1D	PT2D	I/O
D30	PT1C	PT2A	I/O
E30	PT1B	PT1D	I/O
B31	—	PT1C	I/O
C31	PT1A	PT1A	I/O-TCK
D31	RD_DATA/ PTDO	RD_DATA/ PTDO	RD_DATA/ TDO
A13	VSS	VSS	VSS
A16	VSS	VSS	VSS
A20	VSS	VSS	VSS
A23	VSS	VSS	VSS
A28	VSS	VSS	VSS
A29	VSS	VSS	VSS
A3	VSS	VSS	VSS
A32	VSS	VSS	VSS
A33	VSS	VSS	VSS

Note: The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

Pin Information (continued)

Table 27. OR2C/2T26A and OR2C/2T40A 600-Pin EBGA Pinout (continued)

Pin	OR2C/2T26A Pad	OR2C/2T40A Pad	Function
A4	VSS	VSS	VSS
A7	VSS	VSS	VSS
A8	VSS	VSS	VSS
AC1	VSS	VSS	VSS
AC35	VSS	VSS	VSS
AH1	VSS	VSS	VSS
AH35	VSS	VSS	VSS
AJ1	VSS	VSS	VSS
AJ35	VSS	VSS	VSS
AM1	VSS	VSS	VSS
AM2	VSS	VSS	VSS
AM3	VSS	VSS	VSS
AM33	VSS	VSS	VSS
AM34	VSS	VSS	VSS
AM35	VSS	VSS	VSS
AN1	VSS	VSS	VSS
AN2	VSS	VSS	VSS
AN32	VSS	VSS	VSS
AN34	VSS	VSS	VSS
AN35	VSS	VSS	VSS
AN4	VSS	VSS	VSS
AP3	VSS	VSS	VSS
AP32	VSS	VSS	VSS
AP33	VSS	VSS	VSS
AP4	VSS	VSS	VSS
AR13	VSS	VSS	VSS
AR16	VSS	VSS	VSS
AR20	VSS	VSS	VSS
AR23	VSS	VSS	VSS
AR28	VSS	VSS	VSS
AR29	VSS	VSS	VSS
AR3	VSS	VSS	VSS
AR32	VSS	VSS	VSS
AR33	VSS	VSS	VSS
AR4	VSS	VSS	VSS
AR7	VSS	VSS	VSS
AR8	VSS	VSS	VSS
B3	VSS	VSS	VSS
B32	VSS	VSS	VSS
B33	VSS	VSS	VSS
B4	VSS	VSS	VSS
C1	VSS	VSS	VSS
C2	VSS	VSS	VSS
C32	VSS	VSS	VSS
C34	VSS	VSS	VSS
C35	VSS	VSS	VSS
C4	VSS	VSS	VSS
D1	VSS	VSS	VSS
D2	VSS	VSS	VSS
D3	VSS	VSS	VSS
D33	VSS	VSS	VSS
D34	VSS	VSS	VSS
D35	VSS	VSS	VSS
G1	VSS	VSS	VSS
G35	VSS	VSS	VSS
H1	VSS	VSS	VSS
H35	VSS	VSS	VSS
N1	VSS	VSS	VSS
N35	VSS	VSS	VSS
T1	VSS	VSS	VSS
T35	VSS	VSS	VSS
Y1	VSS	VSS	VSS
Y35	VSS	VSS	VSS
A1	VDD	VDD	VDD
A17	VDD	VDD	VDD
A18	VDD	VDD	VDD
A2	VDD	VDD	VDD
A30	VDD	VDD	VDD
A31	VDD	VDD	VDD
A34	VDD	VDD	VDD
A35	VDD	VDD	VDD
A5	VDD	VDD	VDD
A6	VDD	VDD	VDD
AE31	VDD	VDD	VDD
AE5	VDD	VDD	VDD
AK1	VDD	VDD	VDD
AK35	VDD	VDD	VDD
AL1	VDD	VDD	VDD
AL11	VDD	VDD	VDD
AL16	VDD	VDD	VDD
AL20	VDD	VDD	VDD
AL25	VDD	VDD	VDD
AL31	VDD	VDD	VDD
AL35	VDD	VDD	VDD
AL5	VDD	VDD	VDD
AM32	VDD	VDD	VDD
AM4	VDD	VDD	VDD
AN3	VDD	VDD	VDD
AN33	VDD	VDD	VDD
AP1	VDD	VDD	VDD
AP2	VDD	VDD	VDD
AP34	VDD	VDD	VDD
AP35	VDD	VDD	VDD
AR1	VDD	VDD	VDD

Note: The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

Pin Information (continued)

Table 27. OR2C/2T26A and OR2C/2T40A 600-Pin EBGA Pinout (continued)

Pin	OR2C/2T26A Pad	OR2C/2T40A Pad	Function
AR18	VDD	VDD	VDD
AR19	VDD	VDD	VDD
AR2	VDD	VDD	VDD
AR30	VDD	VDD	VDD
AR31	VDD	VDD	VDD
AR34	VDD	VDD	VDD
AR35	VDD	VDD	VDD
AR5	VDD	VDD	VDD
AR6	VDD	VDD	VDD
B1	VDD	VDD	VDD
B2	VDD	VDD	VDD
B34	VDD	VDD	VDD
B35	VDD	VDD	VDD
C3	VDD	VDD	VDD
C33	VDD	VDD	VDD
D32	VDD	VDD	VDD
D4	VDD	VDD	VDD
E1	VDD	VDD	VDD
E11	VDD	VDD	VDD
E16	VDD	VDD	VDD
E20	VDD	VDD	VDD
E25	VDD	VDD	VDD
E31	VDD	VDD	VDD
E35	VDD	VDD	VDD
E5	VDD	VDD	VDD
F1	VDD	VDD	VDD
F35	VDD	VDD	VDD
L31	VDD	VDD	VDD
L5	VDD	VDD	VDD
T31	VDD	VDD	VDD
T5	VDD	VDD	VDD
U35	VDD	VDD	VDD
V1	VDD	VDD	VDD
V35	VDD	VDD	VDD
W1	VDD	VDD	VDD
Y31	VDD	VDD	VDD
Y5	VDD	VDD	VDD

Note: The pins labeled "I/O-VDD5" are user I/Os for the OR2CxxA series, but they are connected to VDD5 for the OR2TxxA series.

Package Thermal Characteristics

When silicon die junction temperature is below the recommended junction temperature of 125 °C, the temperature-activated failure mechanisms are minimized. There are four major factors that affect the thermal resistance value: silicon device size/paddle size, board mounting configuration (board density, multilayer nature of board), package type and size, and system airflow over the package. The values in the table below reflect the capability of the various package types to dissipate heat at given airflow rates. The numbers represent the delta °C/W between the ambient temperature and the device junction temperature.

To test package thermal characteristics, a single package containing a 0.269 in. sq. test IC of each configuration is mounted at the center of a printed-circuit board (PCB) measuring 8 in. x 13 in. x 0.062 in. The assembled PCB is mounted vertically in the center of the rectangular test section of a wind tunnel. The walls of the wind tunnel simulate adjacent boards in the electronic rack and can be adjusted to study the effects of PCB spacing. Forced air at room temperature is supplied by a pair of push-pull blowers which can be regulated to supply the desired air velocities. The air velocity is measured with a hot-wire anemometer at the center of the channel, 3 in. upstream from the package.

A typical test consists of regulating the wind tunnel blowers to obtain the desired air velocity and applying power to the test IC. The power to the IC is adjusted until the maximum junction temperature (as measured by its diodes) reaches 115 °C to 120 °C. The thermal resistance Θ_{JA} (°C/W) is computed by using the power supplied to the IC, junction temperature, ambient temperature, and air velocity:

$$\Theta_{JA} = \frac{T_J - T_A}{Q_C}$$

where:

T_J = peak temperature on the active surface of the IC

T_A = ambient air temperature

Q_C = IC power

The tests are repeated at several velocities from 0 fpm (feet per minute) to 1000 fpm.

The definition of the junction-to-case thermal resistance (Θ_{JC}) is:

$$\Theta_{JC} = \frac{T_J - T_C}{Q_C}$$

where:

T_C = temperature measured to the thermocouple at the top dead center of the package

The actual Θ_{JC} measurement performed at Lucent Technologies, Θ_{J-TDC}, uses a different package mounting arrangement than the one defined for Θ_{JC} in MIL-STD-883D and SEMI standards. Please contact Lucent for a diagram.

The maximum power dissipation for a package is calculated from the maximum allowed junction temperature (T_{Jmax}, 125 °C), the maximum ambient temperature (T_{Amax}), and the junction-to-ambient thermal characteristic for the given package (Θ_{JA}). The maximum power for the package is calculated as follows:

$$\text{Max. Power (Watts)} = (125\ °C - T_{Amax}) \times (1/\Theta_{JA})$$

In Table 28, a maximum power dissipation for each package is shown with T_{Amax} = 70 °C for the commercial temperature range and the Θ_{JA} used is for 0 feet per minute of air flowing over the package. If your application does not correspond to these parameters, the maximum power dissipation should be recalculated using the formula above.

Once the power dissipated by the FPGA has been determined (see the Estimating Power Dissipation section), the maximum junction temperature of the FPGA can be found. This is needed to determine if speed derating of the device from the 85 °C junction temperature used in all of the delay tables is needed. Using the maximum ambient temperature, T_{Amax}, and the power dissipated by the device, P, the maximum junction temperature is given by:

$$T_{Jmax} = T_{Amax} + (P \bullet \Theta_{JA})\ °C$$

Table 28 lists the thermal characteristics for all packages used with the *ORCA* OR2CxxA/OR2TxxA Series of FPGAs.

Package Thermal Characteristics (continued)

Table 28. *ORCA* OR2CxxA/OR2TxxA Plastic Package Thermal Guidelines

Package	Θ_{JA} (°C/W)			Θ_{JC} (°C/W)	Max Power (W) (70 °C—0 fpm)
	0 fpm	200 fpm	400 fpm		
84-Pin PLCC	40	35	32	9	1.38
100-Pin TQFP	61	49	46	6	0.90
144-Pin TQFP	52	39	36	4	1.05
160-Pin QFP	40	36	32	8	1.38
208-Pin SQFP	37	33	29	8	1.49
208-Pin SQFP2	16	14	12	2	3.43
240-Pin SQFP	35	31	28	7	1.57
240-Pin SQFP2	15	12	10	2	3.66
256-Pin PBGA[1,3]	22	14	17	3 (est.)	2.62
256-Pin PBGA[2,3]	26	23	21	TBD	1.97
304-Pin SQFP	33	30	27	6	1.67
304-Pin SQFP2	12	10	8	2	4.58
352-Pin PBGA[1,3]	18	15	13	2 (est.)	3.06
352-Pin PBGA[2,3]	25	21	19	TBD	2.20
432-Pin EBGA	13	10	9.3	<1	5.50
600-Pin EBGA	12	9	8.3	<1	6.88

1. With thermal balls connected to board ground plane.
2. Without thermal balls connected to board ground plane.
3. Mounted on 4-layer board with two power/ground planes.

Package Coplanarity

The coplanarity limits of the *ORCA* OR2CxxA/ OR2TxxA series packages are as follows:

- TQFP: 3.15 mils
- PLCC and QFP: 4.0 mils
- PBGA: 7.5 mils
- SQFP: 4.0 mils (240 and 304 only)
  3.15 mils (all other sizes)
- EBGA: 8.0 mils

Package Parasitics

The electrical performance of an IC package, such as signal quality and noise sensitivity, is directly affected by the package parasitics. Table 29 lists eight parasitics associated with the *ORCA* packages. These parasitics represent the contributions of all components of a package, which include the bond wires, all internal package routing, and the external leads.

Four inductances in nH are listed: L_{SW} and L_{SL}, the self-inductance of the lead; and L_{MW} and L_{ML}, the mutual inductance to the nearest neighbor lead. These parameters are important in determining ground bounce noise and inductive crosstalk noise. Three capacitances in pF are listed: C_M, the mutual capacitance of the lead to the nearest neighbor lead; and C_1 and C_2, the total capacitance of the lead to all other leads (all other leads are assumed to be grounded). These parameters are important in determining capacitive crosstalk and the capacitive loading effect of the lead.

The parasitic values in Table 29 are for the circuit model of bond wire and package lead parasitics. If the mutual capacitance value is not used in the designer's model, then the value listed as mutual capacitance should be added to each of the C_1 and C_2 capacitors. The PGAs contain power and ground planes that will make the inductance value for power and ground leads the minimum value listed. The PGAs also have a significant range of parasitic values. This is due to the large variation in internal trace lengths and is also due to two signal metal layers that are separated from the ground plane by different distances. The upper signal layer is more inductive but less capacitive than the closer, lower signal layer.

Package Parasitics (continued)

Table 29. *ORCA* OR2CxxA/OR2TxxA Package Parasitics

Package Type	L_{SW}	L_{MW}	R_W	C_1	C_2	C_M	L_{SL}	L_{ML}
84-Pin PLCC	3	1	160	1	1	0.5	7—11	3—6
100-Pin TQFP	3	1	160	0.7	0.7	0.94	3—4	1.5—2
144-Pin TQFP	3.5	1.5	175	1	1	0.6	4—6	2—2.5
160-Pin QFP	4	2	200	1	1	1	13—17	8—11
208-Pin SQFP	4	2	200	1	1	1	7—10	4—6
208-Pin SQFP2	4	2	200	1	1	1	6—9	4—6
240-Pin SQFP	4	2	200	1	1	1	8—12	5—8
240-Pin SQFP2	4	2	200	1	1	1	7—11	4—7
256-Pin PBGA	5	2	220	1	1	1	5—13	2—6
304-Pin SQFP	5	2	220	1	1	1	12—18	7—12
304-Pin SQFP2	5	2	220	1	1	1	11—17	7—12
352-Pin PBGA	5	2	220	1.5	1.5	1.5	7—17	3—8
432-Pin EBGA	4	1.5	500	1	1	0.3	3—5.5	0.5—1
600-Pin EBGA	4	1.5	500	1	1	0.4	3—6	0.5—1

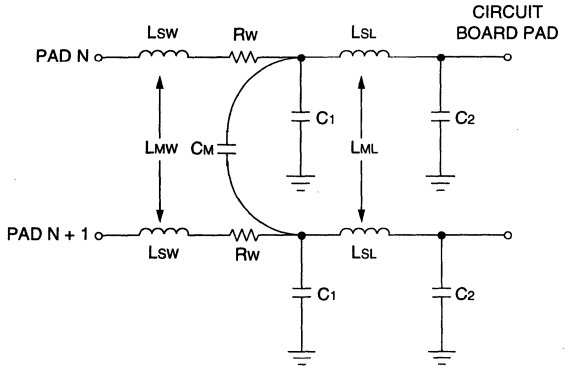


Figure 54. Package Parasitics

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 operations sections of this data sheet. Exposure to absolute maximum ratings for extended periods can adversely affect device reliability.

The *ORCA* Series FPGAs include circuitry designed to protect the chips from damaging substrate injection currents and prevent accumulations of static charge. Nevertheless, conventional precautions should be observed during storage, handling, and use to avoid exposure to excessive electrical stress.

Parameter	Symbol	Min	Max	Unit
Storage Temperature	T_{stg}	–65	150	°C
Supply Voltage with Respect to Ground	V_{DD}	–0.5	7.0	V
V_{DD5} Supply Voltage with Respect to Ground (OR2TxxA)	V_{DD5}	V_{DD}	7.0	V
Input Signal with Respect to Ground	—	–0.5	V_{DD} + 0.3	V
Signal Applied to High-impedance Output	—	–0.5	V_{DD} + 0.3	V
Maximum Soldering Temperature	—	—	260	°C

Note: During powerup and powerdown sequencing, V_{DD} is allowed to be at a higher voltage level than V_{DD5} for up to 100 ms.

Recommended Operating Conditions

Mode	OR2CxxA		OR2TxxA		
	Temperature Range (Ambient)	Supply Voltage (V_{DD})	Temperature Range (Ambient)	Supply Voltage (V_{DD})	Supply Voltage (V_{DD5})
Commercial	0 °C to 70 °C	5 V ± 5%	0 °C to 70 °C	3.0 V to 3.6 V	V_{DD} to 5.25 V
Industrial	–40 °C to +85 °C	5 V ± 10%	–40 °C to +85 °C	3.0 V to 3.6 V	V_{DD} to 5.25 V

Note: The maximum recommended junction temperature (T_J) during operation is 125 °C.

Electrical Characteristics

Table 30. Electrical Characteristics

OR2CxxA Commercial: VDD = 5.0 V ± 5%, 0 °C ≤ TA ≤ 70 °C; OR2CxxA Industrial: VDD = 5.0 V ± 10%, –40 °C ≤ TA ≤ +85 °C.
OR2TxxA Commercial: VDD = 3.0 V to 3.6 V, 0 °C ≤ TA ≤ 70 °C; OR2TxxA Industrial: VDD = 3.0 V to 3.6 V, –40 °C ≤ TA ≤ +85 °C.

Parameter	Symbol	Test Conditions	OR2CxxA		OR2TxxA		Unit
			Min	Max	Min	Max	
Input Voltage: High Low	 VIH VIL	Input configured as CMOS (Includes OR2TxxA)	 70% VDD GND – 0.5	 VDD + 0.3 20% VDD	 80% VDD GND – 0.5	 VDD5 + 0.3 15% VDD	 V V
Input Voltage: High Low	 VIH VIL	Input configured as TTL (Not valid for OR2TxxA)	 2.0 –0.5	 VDD + 0.3 0.8	 — —	 — —	 V V
Output Voltage: High Low	 VOH VOL	 VDD = Min, IOH = 6 mA or 3 mA VDD = Min, IOL = 12 mA or 6 mA	 2.4 —	 — 0.4	 2.4 —	 — 0.4	 V V
Input Leakage Current	IL	VDD = Max, VIN = VSS or VDD	–10	10	–10	10	µA
Standby Current: OR2C04A/OR2T04A OR2C06A/OR2T06A OR2C08A/OR2T08A OR2C10A/OR2T10A OR2C12A/OR2T12A OR2C15A/OR2T15A OR2C26A/OR2T26A OR2C40A/OR2T40A	IDDSB	OR2CxxA (TA = 25 °C, VDD = 5.0 V) OR2TxxA (TA = 25 °C, VDD = 3.3 V) internal oscillator running, no output loads, inputs at VDD or GND	 — — — — — — — —	 6.5 7.0 7.7 8.4 9.2 10.0 12.2 16.3	 — — — — — — — —	 4.0 4.3 4.8 5.3 5.8 6.3 7.8 10.6	 mA mA mA mA mA mA mA mA
Standby Current: OR2C04A/OR2T04A OR2C06A/OR2T06A OR2C08A/OR2T08A OR2C10A/OR2T10A OR2C12A/OR2T12A OR2C15A/OR2T15A OR2C26A/OR2T26A OR2C40A/OR2T40A	IDDSB	OR2CxxA (TA = 25 °C, VDD = 5.0 V) OR2TxxA (TA = 25 °C, VDD = 3.3 V) internal oscillator stopped, no output loads, inputs at VDD or GND	 — — — — — — — —	 1.5 2.0 2.7 3.4 4.2 5.0 7.2 11.3	 — — — — — — — —	 1.0 1.3 1.8 2.3 2.8 3.3 4.8 7.6	 mA mA mA mA mA mA mA mA
Data Retention Voltage	VDR	TA = 25 °C	2.3	—	2.3	—	V
Input Capacitance	CIN	OR2CxxA (TA = 25 °C, VDD = 5.0 V) OR2TxxA (TA = 25 °C, VDD = 3.3 V) Test frequency = 1 MHz	—	9	—	9	pF
Output Capacitance	COUT	OR2CxxA (TA = 25 °C, VDD = 5.0 V) OR2TxxA (TA = 25 °C, VDD = 3.3 V) Test frequency = 1 MHz	—	9	—	9	pF
DONE Pull-up Resistor	RDONE	—	100K	—	100K	—	Ω
M3, M2, M1, and M0 Pull-up Resistors	RM	—	100K	—	100K	—	Ω
I/O Pad Static Pull-up Current	IPU	VDD = 5.25 V, VIN = VSS, TA = 0 °C	14.4	50.9	14.4	50.9	µA
I/O Pad Static Pull-down Current	IPD	VDD = 5.25 V, VIN = VDD, TA = 0 °C	26	103	26	103	µA
I/O Pad Pull-up Resistor	RPU	VDD = 5.25 V, VIN = VSS, TA = 0 °C	100K	—	100K	—	Ω
I/O Pad Pull-down Resistor	RPD	VDD = 5.25 V, VIN = VDD, TA = 0 °C	50K	—	50K	—	Ω

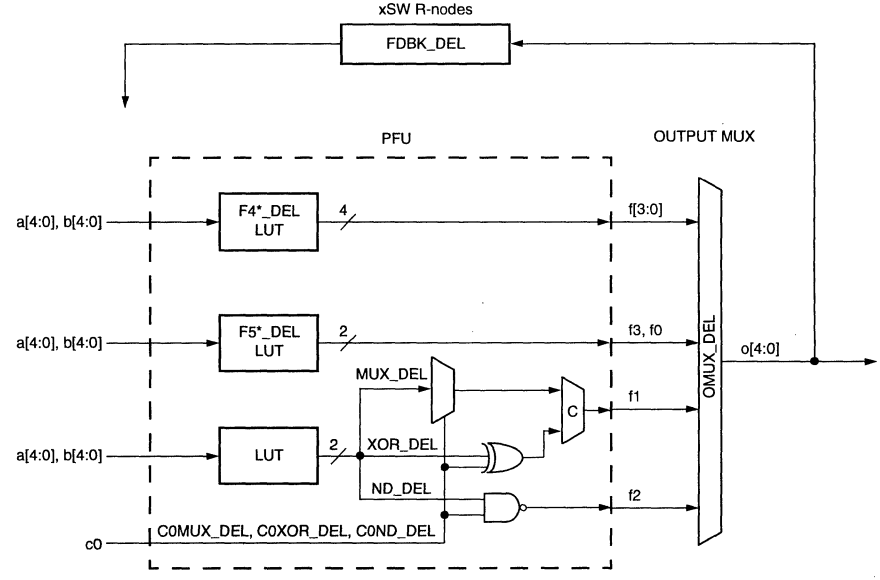
Timing Characteristics

Table 31. Combinatorial PFU Timing Characteristics

OR2CxxA Commercial: V_{DD} = 5.0 V ± 5%, 0 °C ≤ TA ≤ 70 °C; OR2CxxA Industrial: V_{DD} = 5.0 V ± 10%, –40 °C ≤ TA ≤ +85 °C.
OR2TxxA Commercial: V_{DD} = 3.0 V to 3.6 V, 0 °C ≤ TA ≤ 70 °C; OR2TxxA Industrial: V_{DD} = 3.0 V to 3.6 V, –40 °C ≤ TA ≤ +85 °C.

Parameter	Symbol	Speed								Unit
		-2		-3		-4		-5		
		Min	Max	Min	Max	Min	Max	Min	Max	
Combinatorial Delays (T_J = +85 °C, V_{DD} = Min):										
Four Input Variables (a[4:0], b[4:0] to f[3:0])	F4*_DEL	—	4.0	—	2.8	—	2.1	—	1.7	ns
Five Input Variables (a[4:0], b[4:0] to f3, f0)	F5*_DEL	—	4.1	—	2.9	—	2.2	—	1.8	ns
PFUMUX (a[4:0], b[4:0] to f1)	MUX_DEL	—	4.7	—	3.8	—	3.2	—	2.6	ns
PFUMUX (c0 to f1)	C0MUX_DEL	—	3.0	—	2.2	—	1.9	—	1.5	ns
PFUNAND (a[4:0], b[4:0] to f2)	ND_DEL	—	4.7	—	4.0	—	3.3	—	2.7	ns
PFUNAND (c0 to f2)	C0ND_DEL	—	2.7	—	2.2	—	1.8	—	1.5	ns
PFUXOR (a[4:0], b[4:0] to f1)	XOR_DEL	—	5.6	—	4.5	—	3.8	—	3.1	ns
PFUXOR (c0 to f1)	C0XOR_DEL	—	3.1	—	2.2	—	2.0	—	1.6	ns

Note: Shaded values are preliminary.



C = controlled by configuration RAM.

Notes:
The parameters MUX_DEL, XOR_DEL, and ND_DEL include the delay through the LUT in F5A/F5B modes.

See Table 40 for an explanation of FDBK_DEL and OMUX_DEL.

Figure 55. Combinatorial PFU Timing

Timing Characteristics (continued)

Table 32. Sequential PFU Timing Characteristics

OR2CxxA Commercial: VDD = 5.0 V ± 5%, 0 °C ≤ TA ≤ 70 °C; OR2CxxA Industrial: VDD = 5.0 V ± 10%, –40 °C ≤ TA ≤ +85 °C.
OR2TxxA Commercial: VDD = 3.0 V to 3.6 V, 0 °C ≤ TA ≤ 70 °C; OR2TxxA Industrial: VDD = 3.0 V to 3.6 V, –40 °C ≤ TA ≤ +85 °C.

Parameter	Symbol	Speed								Unit
		-2		-3		-4		-5		
		Min	Max	Min	Max	Min	Max	Min	Max	
Input Requirements										
Clock Low Time	TCL	3.2	—	2.5	—	2.0	—	1.8	—	ns
Clock High Time	TCH	3.2	—	2.5	—	2.0	—	1.8	—	ns
Global S/R Pulse Width (gsrn)	TRW	2.8	—	2.5	—	2.0	—	1.8	—	ns
Local S/R Pulse Width	TPW	3.0	—	2.5	—	2.0	—	1.8	—	ns
Combinatorial Setup Times (TJ = +85 °C, VDD = Min):										
Four Input Variables to Clock (a[4:0], b[4:0] to ck)	F4*_SET	2.4	—	1.7	—	1.3	—	1.1	—	ns
Five Input Variables to Clock (a[4:0], b[4:0] to ck)	F5*_SET	2.5	—	1.9	—	1.3	—	1.2	—	ns
PFUMUX to Clock (a[4:0], b[4:0] to ck)	MUX_SET	3.9	—	2.9	—	2.3	—	2.1	—	ns
PFUMUX to Clock (c0 to ck)	C0MUX_SET	1.5	—	1.2	—	0.9	—	0.8	—	ns
PFUNAND to Clock (a[4:0], b[4:0] to ck)	ND_SET	3.9	—	2.9	—	2.2	—	2.0	—	ns
PFUNAND to Clock (c0 to ck)	C0ND_SET	1.7	—	1.2	—	0.6	—	0.5	—	ns
PFUXOR to Clock (a[4:0], b[4:0] to ck)	XOR_SET	4.8	—	3.6	—	3.0	—	2.7	—	ns
PFUXOR to Clock (c0 to ck)	C0XOR_SET	1.6	—	1.2	—	0.9	—	0.8	—	ns
Data In to Clock (wd[3:0] to ck)	D*_SET	0.5	—	0.1	—	0.1	—	0.0	—	ns
Clock Enable to Clock (ce to ck)	CKEN_SET	1.6	—	1.2	—	1.0	—	0.9	—	ns
Local Set/Reset (synchronous) (lsr to ck)	LSR_SET	1.7	—	1.4	—	1.3	—	1.2	—	ns
Data Select to Clock (sel to ck)	SELECT_SET	1.9	—	1.5	—	1.4	—	1.3	—	ns
Pad Direct In	PDIN_SET	0.0	—	0.0	—	0.0	—	0.0	—	ns
Combinatorial Hold Times (TJ = All, VDD = All):										
Data In (wd[3:0] from ck)	D*_HLD	0.6	—	0.4	—	0.4	—	0.4	—	ns
Clock Enable (ce from ck)	CKEN_HLD	0.6	—	0.4	—	0.0	—	0.0	—	ns
Local Set/Reset (synchronous) (lsr from ck)	LSR_HLD	0.0	—	0.0	—	0.0	—	0.0	—	ns
Data Select (sel from ck)	SELECT_HLD	0.0	—	0.0	—	0.0	—	0.0	—	ns
Pad Direct In Hold (dia[3:0], dib[3:0] to ck)*	PDIN_HLD	1.5	—	1.4	—	1.0	—	0.9	—	ns
All Others	—	0.0	—	0.0	—	0.0	—	0.0	—	ns
Output Characteristics										
Sequential Delays (TJ = +85 °C, VDD = Min):										
Local S/R (async) to PFU Out (lsr to q[3:0])	LSR_DEL	—	4.5	—	3.4	—	3.1	—	2.5	ns
Global S/R to PFU Out (gsrn to q[3:0])	GSR_DEL	—	2.9	—	2.3	—	2.0	—	1.6	ns
Clock to PFU Out (ck to q[3:0]) — Register	REG_DEL	—	2.8	—	2.0	—	1.9	—	1.5	ns
Clock to PFU Out (ck to q[3:0]) — Latch	LTCH_DEL	—	2.9	—	2.0	—	1.9	—	1.5	ns
Transparent Latch (wd[3:0] to q[3:0])	LTCH_DDEL	—	3.9	—	2.7	—	2.5	—	2.0	ns

* The input buffers contain a programmable delay to allow the hold time vs. the external clock pin to be equal to 0.

Note: Shaded values are preliminary.

Timing Characteristics (continued)

Table 33. Ripple Mode PFU Timing Characteristics

OR2CxxA Commercial: VDD = 5.0 V ± 5%, 0 °C ≤ TA ≤ 70 °C; OR2CxxA Industrial: VDD = 5.0 V ± 10%, –40 °C ≤ TA ≤ +85 °C.
OR2TxxA Commercial: VDD = 3.0 V to 3.6 V, 0 °C ≤ TA ≤ 70 °C; OR2TxxA Industrial: VDD = 3.0 V to 3.6 V, –40 °C ≤ TA ≤ +85 °C.

Parameter	Symbol	Speed								Unit
		-2		-3		-4		-5		
		Min	Max	Min	Max	Min	Max	Min	Max	
Ripple Setup Times (TJ = +85 °C, VDD = Min):										
Operands to Clock (a[3:0], b[3:0] to ck)	RIP_SET	6.7	—	5.0	—	3.7	—	3.3	—	ns
Bit-Wise Operands to Clock (a[i], b[i] to ck at f[i])	FRIP_SET	2.4	—	1.7	—	1.3	—	1.2	—	ns
Carry-In from Fast Carry to Clock (cin to ck)	CIN_SET	4.0	—	3.2	—	1.9	—	1.7	—	ns
Carry-In from General Routing to Clock (b4 to ck)	B4_SET	4.0	—	3.2	—	1.9	—	1.7	—	ns
Add/Subtract to Clock (a4 to ck)	AS_SET	8.2	—	5.6	—	4.3	—	3.9	—	ns
Ripple Hold Times (TJ = All, VDD = All): All	TH	0	—	0	—	0	—	0	—	ns
Ripple Delays (TJ = +85 °C, VDD = Min):										
Operands to Carry-Out (a[3:0], b[3:0] to cout)	RIP_CODEL	—	5.4	—	3.8	—	3.3	—	2.6	ns
Operands to Carry-Out (a[3:0], b[3:0] to o4)	RIP_O4DEL	—	6.9	—	4.8	—	4.2	—	3.4	ns
Operands to PFU Out (a[3:0], b[3:0] to f[3:0])	RIP_DEL	—	8.2	—	6.0	—	4.7	—	3.8	ns
Bit-Wise Operands to PFU Out (a[i], b[i] to f[i])	FRIP_DEL	—	4.0	—	2.8	—	2.1	—	1.7	ns
Carry-In from Fast Carry to Carry-Out (cin to cout)	CIN_CODEL	—	1.9	—	1.6	—	1.1	—	0.9	ns
Carry-In from Fast Carry to Carry-Out (cin to o4)	CIN_O4DEL	—	3.5	—	2.6	—	2.1	—	1.7	ns
Carry-In from Fast Carry to PFU Out (cin to f[3:0])	CIN_DEL	—	5.6	—	4.2	—	2.9	—	2.3	ns
Carry-In from General Routing to Carry-Out (b4 to cout)	B4_CODEL	—	1.9	—	1.6	—	1.1	—	0.9	ns
Carry-In from General Routing to Carry-Out (b4 to o4)	B4_O4DEL	—	3.5	—	2.6	—	2.1	—	1.7	ns
Carry-In from General Routing to PFU Out (b4 to f[3:0])	B4_DEL	—	5.6	—	4.2	—	2.9	—	2.3	ns
Add/Subtract to Carry-Out (a4 to cout)	AS_CODEL	—	6.1	—	4.5	—	3.9	—	3.1	ns
Add/Subtract to Carry-Out (a4 to o4)	AS_O4DEL	—	7.6	—	5.6	—	4.9	—	3.9	ns
Add/Subtract to PFU Out (a4 to f[3:0])	AS_DEL	—	9.7	—	6.8	—	5.3	—	4.3	ns

Notes:
The new 4 x 1 multiplier and 4-bit comparator submodes use the appropriate ripple mode timing shown above.

Shaded values are preliminary.

Timing Characteristics (continued)

Table 34. Asynchronous Memory Read Characteristics (MA/MB Modes)

OR2CxxA Commercial: V_{DD} = 5.0 V ± 5%, 0 °C ≤ T_A ≤ 70 °C; OR2CxxA Industrial: V_{DD} = 5.0 V ± 10%, –40 °C ≤ T_A ≤ +85 °C.
OR2TxxA Commercial: V_{DD} = 3.0 V to 3.6 V, 0 °C ≤ T_A ≤ 70 °C; OR2TxxA Industrial: V_{DD} = 3.0 V to 3.6 V, –40 °C ≤ T_A ≤ +85 °C.

Parameter	Symbol	Speed								Unit
		-2		-3		-4		-5		
		Min	Max	Min	Max	Min	Max	Min	Max	
Read Operation (T_J = 85 °C, V_{DD} = Min):										
Read Cycle Time	T_{RC}	5.1	—	3.6	—	2.7	—	2.4	—	ns
Data Valid after Address (a[3:0], b[3:0] to f[3:0])	MEM*_ADEL	—	4.0	—	2.8	—	2.1	—	1.7	ns
Read Operation, Clocking Data into Latch/Flip-Flop (T_J = 85 °C, V_{DD} = Min):										
Address to Clock Setup Time (a[3:0], b[3:0] to ck)	MEM*_ASET	2.4	—	1.8	—	1.2	—	1.1	—	ns
Clock to PFU Out (ck to q[3:0]) — Register	REG_DEL	—	2.8	—	2.0	—	1.9	—	1.5	ns

Note: Shaded values are preliminary.

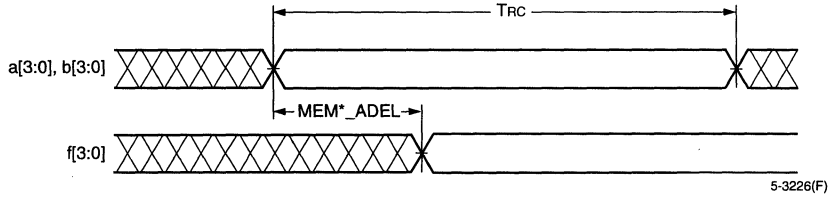


Figure 56. Read Operation—Flip-Flop Bypass

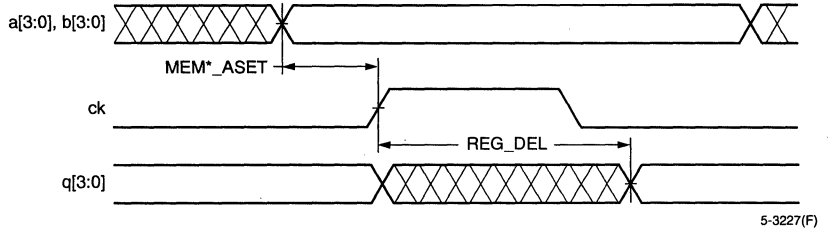


Figure 57. Read Operation—LUT Memory Loading Flip-Flops

Timing Characteristics (continued)

Table 35. Asynchronous Memory Write Characteristics (MA/MB Modes)

OR2CxxA Commercial: V_{DD} = 5.0 V ± 5%, 0 °C ≤ TA ≤ 70 °C; OR2CxxA Industrial: V_{DD} = 5.0 V ± 10%, –40 °C ≤ TA ≤ +85 °C.
OR2TxxA Commercial: V_{DD} = 3.0 V to 3.6 V, 0 °C ≤ TA ≤ 70 °C; OR2TxxA Industrial: V_{DD} = 3.0 V to 3.6 V, –40 °C ≤ TA ≤ +85 °C.

Parameter	Symbol	Speed								Unit
		-2		-3		-4		-5		
		Min	Max	Min	Max	Min	Max	Min	Max	
Write Operation (T_J = +85 °C, V_{DD} = Min):										
Write Cycle Time	T_{WC}	5.5	—	4.5	—	3.8	—	3.4	—	ns
Write Enable (wren) Pulse Width (a4/b4)	T_{PW}	3.0	—	2.5	—	2.0	—	1.8	—	ns
Setup Time (T_J = +85 °C, V_{DD} = Min):										
Address to wren (a[3:0]/b[3:0] to a4/b4)	MEM*_AWRSET	0.1	—	0.1	—	0.0	—	0.0	—	ns
Data to wren (wd[3:0] to a4/b4)	MEM*_DWRSET	0.0	—	0.0	—	0.0	—	0.0	—	ns
Address to wpe (a[3:0]/b[3:0] to c0)	MEM*_APWRSET	0.0	—	0.0	—	0.0	—	0.0	—	ns
Data to wpe (wd[3:0] to c0)	MEM*_DPWRSET	0.0	—	0.0	—	0.0	—	0.0	—	ns
wpe to wren (c0 to a4/b4)	MEM*_WPESET	2.5	—	2.0	—	1.5	—	1.4	—	ns
Hold Time (T_J = All, V_{DD} = All):										
Address from wren (a[3:0]/b[3:0] from a4/b4)	MEM*_WRAHLD	2.4	—	1.7	—	1.8	—	1.6	—	ns
Data from wren (wd[3:0] from a4/b4)	MEM*_WRDHLD	2.4	—	2.0	—	1.9	—	1.5	—	ns
Address from wpe (a[3:0/b[3:0] to c0)	MEM*_PWRAHLD	3.8	—	3.3	—	2.8	—	2.5	—	ns
Data from wpe (wd[3:0] to c0)	MEM*_PWRDHLD	3.9	—	3.4	—	2.9	—	2.6	—	ns
wpe from wren (c0 from a4/b4)	MEM*_WPEHLD	0.0	—	0.0	—	0.0	—	0.0	—	ns

Note: Shaded values are preliminary.

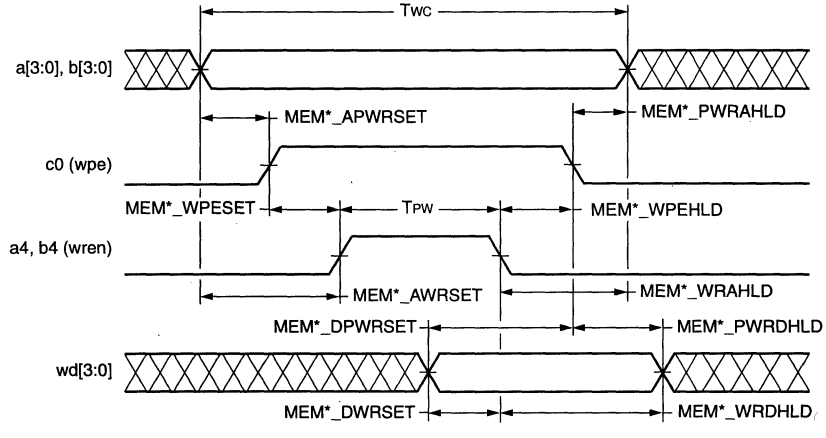


Figure 58. Write Operation

Timing Characteristics (continued)

Table 36. Asynchronous Memory Read During Write Operation (MA/MB Modes)

OR2CxxA Commercial: VDD = 5.0 V ± 5%, 0 °C ≤ TA ≤ 70 °C; OR2CxxA Industrial: VDD = 5.0 V ± 10%, –40 °C ≤ TA ≤ +85 °C.
OR2TxxA Commercial: VDD = 3.0 V to 3.6 V, 0 °C ≤ TA ≤ 70 °C; OR2TxxA Industrial: VDD = 3.0 V to 3.6 V, –40 °C ≤ TA ≤ +85 °C.

Parameter	Symbol	Speed								Unit
		-2		-3		-4		-5		
		Min	Max	Min	Max	Min	Max	Min	Max	
Read During Write Operation (TJ = +85 °C, VDD = Min):										
Write Enable (wren) to PFU Output Delay (a4/b4 to f[3:0])	MEM*_WRDEL	—	7.0	—	4.9	—	4.8	—	3.9	ns
Write-Port Enable (wpe) to PFU Output Delay (c0 to f[3:0])	MEM*_PWRDEL	—	9.0	—	6.4	—	5.8	—	4.7	ns
Data to PFU Output Delay (wd[3:0] to f[3:0])	MEM*_DDEL	—	5.0	—	3.6	—	3.1	—	2.5	ns

Note: Shaded values are preliminary.

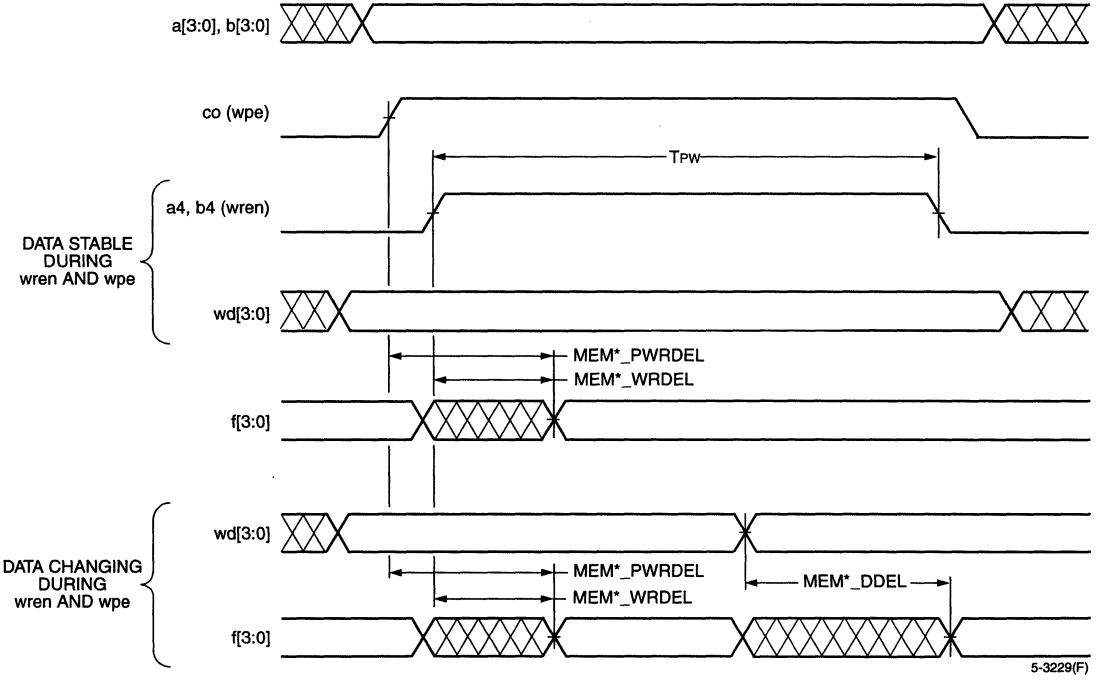


Figure 59. Read During Write

Timing Characteristics (continued)

Table 37. Asynchronous Memory Read During Write, Clocking Data Into Latch/Flip-Flop (MA/MB Modes)

OR2CxxA Commercial: V_{DD} = 5.0 V ± 5%, 0 °C ≤ T_A ≤ 70 °C; OR2CxxA Industrial: V_{DD} = 5.0 V ± 10%, –40 °C ≤ T_A ≤ +85 °C.
OR2TxxA Commercial: V_{DD} = 3.0 V to 3.6 V, 0 °C ≤ T_A ≤ 70 °C; OR2TxxA Industrial: V_{DD} = 3.0 V to 3.6 V, –40 °C ≤ T_A ≤ +85 °C.

Parameter	Symbol	Speed								Unit
		-2		-3		-4		-5		
		Min	Max	Min	Max	Min	Max	Min	Max	
Setup Time (T_J = +85 °C, V_{DD} = Min):										
Address to Clock (a[3:0], b[3:0] to ck)	MEM*_ASET	2.4	—	1.8	—	1.2	—	1.1	—	ns
Write Enable (wren) to Clock (a4/b4 to ck)	MEM*_WRSET	5.4	—	4.4	—	3.8	—	3.4	—	ns
Write-Port Enable (wpe) to Clock (c0 to ck)	MEM*_PWRSET	7.4	—	5.9	—	4.8	—	4.3	—	ns
Data (wd[3:0] to ck)	MEM*_DSET	3.5	—	2.6	—	2.6	—	2.3	—	ns
Hold Time (T_J = All, V_{DD} = All): All	T_H	0	—	0	—	0	—	0	—	ns
Clock to PFU Out (ck to q[3:0]) — Register	REG_DEL	—	2.8	—	2.0	—	1.9	—	1.5	ns

Note: Shaded values are preliminary.

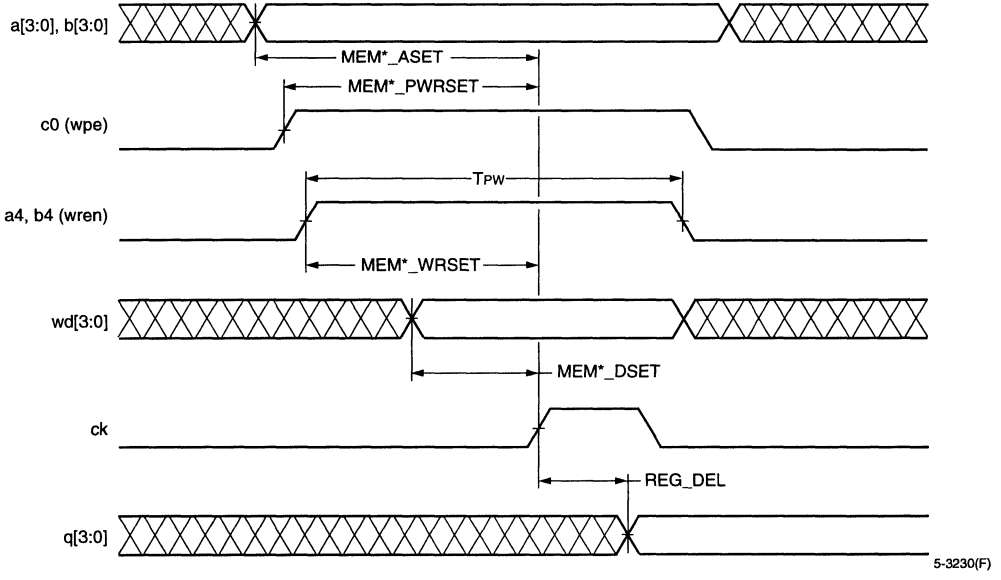


Figure 60. Read During Write—Clocking Data into Flip-Flop

Timing Characteristics (continued)

Table 38. Synchronous Memory Write Characteristics (SSPM and SDPM Modes)

OR2CxxA Commercial: VDD = 5.0 V ± 5%, 0 °C ≤ TA ≤ 70 °C; OR2CxxA Industrial: VDD = 5.0 V ± 10%, –40 °C ≤ TA ≤ +85 °C.
OR2TxxA Commercial: VDD = 3.0 V to 3.6 V, 0 °C ≤ TA ≤ 70 °C; OR2TxxA Industrial: VDD = 3.0 V to 3.6 V, –40 °C ≤ TA ≤ +85 °C.

Parameter	Symbol	Speed								Unit
		-2		-3		-4		-5		
		Min	Max	Min	Max	Min	Max	Min	Max	
Write Operation for Fast-RAM Mode[1]:										
Maximum Frequency	FFSCK	38.2	—	52.6	—	83.3	—	95.0	—	MHz
Clock Low Time	TFSCL	13.1	—	9.5	—	6.0	—	5.2	—	ns
Clock High Time	TFSCH	13.1	—	9.5	—	6.0	—	5.2	—	ns
Clock to Data Valid (ck to f[3:0])[2]	FMEMS_DEL	—	9.0	—	7.4	—	6.2	—	5.0	ns
Write Operation for Normal RAM Mode:										
Maximum Frequency	FSCK	24.3	—	33.3	—	52.6	—	58.0	—	MHz
Clock Low Time	TSCL	20.6	—	15.0	—	9.5	—	8.5	—	ns
Clock High Time	TSCH	20.6	—	15.0	—	9.5	—	8.5	—	ns
Clock to Data Valid (ck to f[3:0])	MEMS_DEL	—	10.9	—	8.6	—	7.5	—	6.0	ns
Write Operation Setup Time:										
Address to Clock (a[3:0]/b[3:0] to ck)	MEMS_ASET	0.0	—	0.0	—	0.0	—	0.0	—	ns
Data to Clock (wd[3:0] to ck)	MEMS_DSET	0.0	—	0.0	—	0.0	—	0.0	—	ns
Write Enable (wren) to Clock (a4 to ck)	MEMS_WRSET	0.0	—	0.0	—	0.0	—	0.0	—	ns
Write-Port Enable (wpe) to Clock (c0 to ck)	MEMS_PWRSET	0.0	—	0.0	—	0.0	—	0.0	—	ns
Write Operation Hold Time:										
Address to Clock (a[3:0]/b[3:0] to ck)	MEMS_AHLD	3.8	—	3.0	—	2.2	—	2.0	—	ns
Data to Clock (wd[3:0] to ck)	MEMS_DHLD	3.8	—	3.0	—	2.2	—	2.0	—	ns
Write Enable (wren) to Clock (a4 to ck)	MEMS_WRHLD	3.8	—	3.0	—	2.2	—	2.0	—	ns
Write-Port Enable (wpe) to Clock (c0 to ck)	MEMS_PWRHLD	3.3	—	2.3	—	1.5	—	1.4	—	ns

1. Readback of the configuration bit stream when simultaneously writing to a PFU in either SSPM fast mode or SDPM fast mode is not allowed.
2. Because the setup time of data into the latches/FFs is less than 0 ns, data written into the RAM can be loaded into a latch/FF in the same PFU on the next opposite clock edge (one-half clock period).

Note: Shaded values are preliminary.

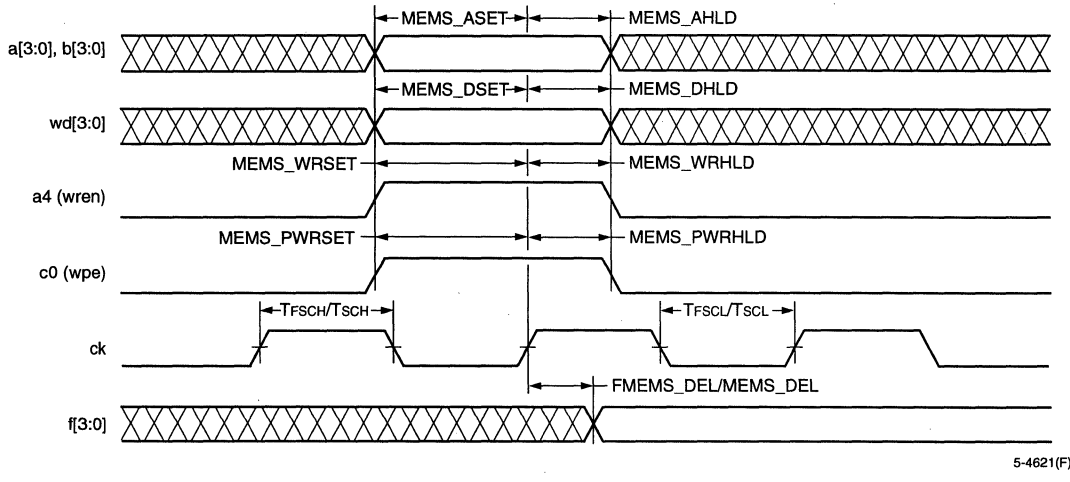


Figure 61. Synchronous Memory Write Characteristics

Timing Characteristics (continued)

Table 39. Synchronous Memory Read Characteristics (SSPM and SDPM Modes)

OR2CxxA Commercial: V_{DD} = 5.0 V ± 5%, 0 °C ≤ T_A ≤ 70 °C; OR2CxxA Industrial: V_{DD} = 5.0 V ± 10%, –40 °C ≤ T_A ≤ +85 °C.
OR2TxxA Commercial: V_{DD} = 3.0 V to 3.6 V, 0 °C ≤ T_A ≤ 70 °C; OR2TxxA Industrial: V_{DD} = 3.0 V to 3.6 V, –40 °C ≤ T_A ≤ +85 °C.

Parameter	Symbol	Speed								Unit
		-2		-3		-4		-5		
		Min	Max	Min	Max	Min	Max	Min	Max	
Read Operation (T_J = 85 °C, V_{DD} = Min):										
Read Cycle Time	T_{RC}	5.1	—	3.6	—	2.7	—	2.4	—	ns
Data Valid After Address (a[3:0], b[3:0] to f[3:0])	MEM*_ADEL	—	4.0	—	2.8	—	2.1	—	1.7	ns
Read Operation, Clocking Data Into Latch/FF (T_J = 85 °C, V_{DD} = Min):										
Address to Clock Setup Time (a[3:0], b[3:0] to ck)	MEM*_ASET	2.4	—	1.8	—	1.2	—	1.1	—	ns
Clock to PFU Output—Register (ck to q[3:0])	REG_DEL	—	2.8	—	2.0	—	1.9	—	1.5	ns

Note: Shaded values are preliminary.

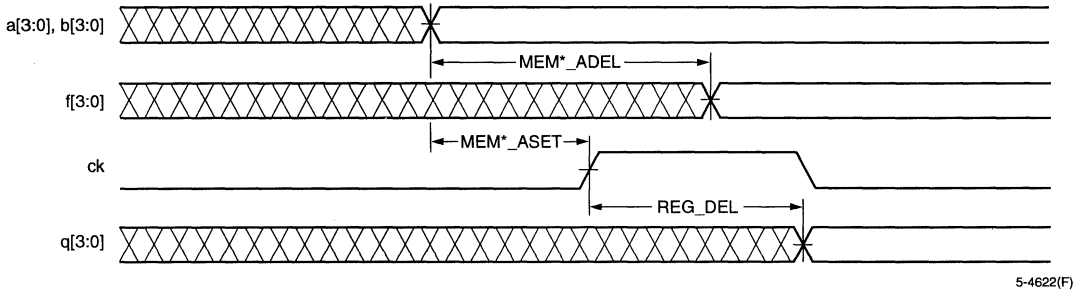


Figure 62. Synchronous Memory Read Cycle

Timing Characteristics (continued)

Table 40. PFU Output MUX, PLC BIDI, and Direct Routing Timing Characteristics

OR2CxxA Commercial: V_{DD} = 5.0 V ± 5%, 0 °C ≤ T_A ≤ 70 °C; OR2CxxA Industrial: V_{DD} = 5.0 V ± 10%, –40 °C ≤ T_A ≤ +85 °C.
OR2TxxA Commercial: V_{DD} = 3.0 V to 3.6 V, 0 °C ≤ T_A ≤ 70 °C; OR2TxxA Industrial: V_{DD} = 3.0 V to 3.6 V, –40 °C ≤ T_A ≤ +85 °C.

Parameter	Symbol	Speed								Unit
		-2		-3		-4		-5		
		Min	Max	Min	Max	Min	Max	Min	Max	
PFU Output MUX (T_J = 85 °C, V_{DD} = Min)										
Output MUX Delay (f[3:0]/q[3:0] to o[4:0])	OMUX_DEL	—	1.1	—	0.8	—	0.6	—	0.5	ns
PLC 3-Statable BIDIs (T_J = 85 °C, V_{DD} = Min)										
BIDI Propagation Delay	TRI_DEL	—	1.2	—	1.0	—	0.8	—	0.7	ns
BIDI 3-State Enable/Disable Delay	TRIEN_DEL	—	1.7	—	1.3	—	1.0	—	0.8	ns
Direct Routing (T_J = 85 °C, V_{DD} = Min)										
PFU to PFU Delay (xSW)	DIR_DEL	—	1.4	—	1.1	—	0.9	—	0.7	ns
PFU Feedback (xSW)	FDBK_DEL	—	1.0	—	0.8	—	0.7	—	0.6	ns

Note: Shaded values are preliminary.

Timing Characteristics (continued)

Table 41. Clock Delay

OR2CxxA Commercial: V_{DD} = 5.0 V ± 5%, 0 °C ≤ TA ≤ 70 °C; OR2CxxA Industrial: V_{DD} = 5.0 V ± 10%, –40 °C ≤ TA ≤ +85 °C.
OR2TxxA Commercial: V_{DD} = 3.0 V to 3.6 V, 0 °C ≤ TA ≤ 70 °C; OR2TxxA Industrial: V_{DD} = 3.0 V to 3.6 V, –40 °C ≤ TA ≤ +85 °C.

Device (T_J = 85 °C, V_{DD} = Min)	Symbol	Speed								Unit
		-2		-3		-4		-5		
		Min	Max	Min	Max	Min	Max	Min	Max	
OR2C04A/OR2T04A	CLK_DEL	—	5.5	—	4.4	—	4.3	—	3.4	ns
OR2C06A/OR2T06A	CLK_DEL	—	5.6	—	4.5	—	4.4	—	3.5	ns
OR2C08A/OR2T08A	CLK_DEL	—	5.8	—	4.6	—	4.5	—	3.6	ns
OR2C10A/OR2T10A	CLK_DEL	—	5.9	—	4.7	—	4.6	—	3.7	ns
OR2C12A/OR2T12A	CLK_DEL	—	6.1	—	4.9	—	4.7	—	3.8	ns
OR2C15A/OR2T15A	CLK_DEL	—	6.2	—	5.0	—	4.8	—	3.9	ns
OR2C26A/OR2T26A	CLK_DEL	—	6.4	—	5.2	—	5.0	—	4.1	ns
OR2C40A/OR2T40A	CLK_DEL	—	6.9	—	5.8	—	5.3	—	4.3	ns

Notes:
This clock delay is for a fully routed clock tree that uses the primary clock network. It includes both the input buffer delay and the clock routing to the PFU CLK input. The delay will be reduced if any of the clock branches are not used.

Shaded values are preliminary.

Timing Characteristics (continued)

Table 42A. OR2CxxA/OR2TxxA Global Clock to Output Delay (Pin-to-Pin)—Output on Same Side of the Device as the Clock Pin

OR2CxxA Commercial: V_{DD} = 5.0 V ± 5%, 0 °C ≤ TA ≤ 70 °C; Industrial: V_{DD} = 5.0 V ± 10%, –40 °C ≤ TA ≤ +85 °C; CL = 50 pF.
OR2TxxA Commercial: V_{DD} = 3.0 V to 3.6 V, 0 °C ≤ TA ≤ 70 °C; Industrial: V_{DD} = 3.0 V to 3.6 V, –40 °C ≤ TA ≤ +85 °C; CL = 50 pF.

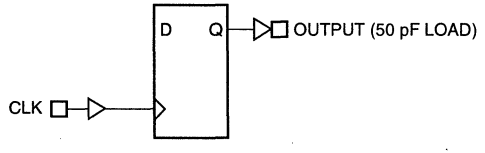
Description (T_J = 85 °C, V_{DD} = Min)	Device	Speed								Unit
		-2		-3		-4		-5		
		Min	Max	Min	Max	Min	Max	Min	Max	
CLK Input Pin → OUTPUT Pin (Fast)	OR2C/2T04A	—	11.7	—	10.1	—	9.8	—	8.4	ns
	OR2C/2T06A	—	11.8	—	10.2	—	9.9	—	8.5	ns
	OR2C/2T08A	—	11.9	—	10.3	—	10.0	—	8.6	ns
	OR2C/2T10A	—	12.0	—	10.4	—	10.1	—	8.7	ns
	OR2C/2T12A	—	12.2	—	10.6	—	10.2	—	8.8	ns
	OR2C/2T15A	—	12.3	—	10.7	—	10.3	—	8.9	ns
	OR2C/2T26A	—	12.5	—	10.9	—	10.5	—	9.1	ns
	OR2C/2T40A	—	13.3	—	11.5	—	10.8	—	9.3	ns
CLK Input Pin → OUTPUT Pin (Slewlim)	OR2C/2T04A	—	14.1	—	12.2	—	11.8	—	10.1	ns
	OR2C/2T06A	—	14.2	—	12.3	—	11.9	—	10.2	ns
	OR2C/2T08A	—	14.3	—	12.4	—	12.0	—	10.3	ns
	OR2C/2T10A	—	14.4	—	12.5	—	12.1	—	10.4	ns
	OR2C/2T12A	—	14.5	—	12.6	—	12.2	—	10.5	ns
	OR2C/2T15A	—	14.6	—	12.7	—	12.3	—	10.6	ns
	OR2C/2T26A	—	14.8	—	12.9	—	12.5	—	10.8	ns
	OR2C/2T40A	—	15.5	—	13.5	—	12.8	—	11.0	ns
CLK Input Pin → OUTPUT Pin (Sinklim)	OR2C/2T04A	—	15.3	—	13.3	—	13.1	—	11.2	ns
	OR2C/2T06A	—	15.4	—	13.4	—	13.2	—	11.3	ns
	OR2C/2T08A	—	15.5	—	13.5	—	13.3	—	11.4	ns
	OR2C/2T10A	—	15.6	—	13.6	—	13.4	—	11.5	ns
	OR2C/2T12A	—	15.9	—	13.8	—	13.5	—	11.6	ns
	OR2C/2T15A	—	16.0	—	13.9	—	13.6	—	11.7	ns
	OR2C/2T26A	—	16.2	—	14.1	—	13.8	—	11.9	ns
	OR2C/2T40A	—	16.9	—	14.7	—	14.1	—	12.1	ns

Notes:
The pin-to-pin timing parameters in this table should be used instead of results reported by *ORCA* Foundry.

This clock delay is for a fully routed clock tree that uses the primary clock network. It includes both the input buffer delay, the clock routing to the PFU CLK input, the clock→Q of the FF, and the delay through the output buffer. The delay will be reduced if any of the clock branches are not used. The given timing requires that the input clock pin be located at one of the four center PICs on any side of the device and that the direct FF→I/O routing be used.

If the clock pin is not located at one of the four center PICs, this delay must be increased by up to the following amounts: OR2C/2T04A = 1.5%, OR2C/2T06A = 2.0%, OR2C/2T08A = 3.1%, OR2C/2T10A = 3.9%, OR2C/2T12A = 4.9%, OR2C/2T15A = 5.7%, OR2C/2T26A = 8.1%, OR2C/2T40A = 12.5%.

Shaded values are preliminary.



5-4846(F)

Figure 63. Global Clock to Output Delay

Timing Characteristics (continued)

Table 42B. OR2CxxA/OR2TxxA Global Clock to Output Delay (Pin-to-Pin)—Output Not On Same Side of the Device as the Clock Pin

OR2CxxA Commercial: V_{DD} = 5.0 V ± 5%, 0 °C ≤ T_A ≤ 70 °C; Industrial: V_{DD} = 5.0 V ± 10%, –40 °C ≤ T_A ≤ +85 °C; C_L = 50 pF.
OR2TxxA Commercial: V_{DD} = 3.0 V to 3.6 V, 0 °C ≤ T_A ≤ 70 °C; Industrial: V_{DD} = 3.0 V to 3.6 V, –40 °C ≤ T_A ≤ +85 °C; C_L = 50 pF.

Description (T_J = 85 °C, V_{DD} = Min)	Device	Speed								Unit
		-2		-3		-4		-5		
		Min	Max	Min	Max	Min	Max	Min	Max	
CLK Input Pin → OUTPUT Pin (Fast)	OR2C/2T04A	—	12.0	—	10.3	—	10.0	—	8.6	ns
	OR2C/2T06A	—	12.2	—	10.4	—	10.1	—	8.7	ns
	OR2C/2T08A	—	12.4	—	10.6	—	10.3	—	8.9	ns
	OR2C/2T10A	—	12.6	—	10.8	—	10.5	—	9.1	ns
	OR2C/2T12A	—	13.0	—	11.1	—	10.7	—	9.3	ns
	OR2C/2T15A	—	13.3	—	11.3	—	10.9	—	9.5	ns
	OR2C/2T26A	—	13.9	—	11.8	—	11.4	—	10.0	ns
	OR2C/2T40A	—	15.3	—	12.9	—	12.2	—	10.7	ns
CLK Input Pin → OUTPUT Pin (Slewlim)	OR2C/2T04A	—	14.4	—	12.4	—	12.0	—	10.3	ns
	OR2C/2T06A	—	14.6	—	12.5	—	12.1	—	10.4	ns
	OR2C/2T08A	—	14.8	—	12.7	—	12.3	—	10.6	ns
	OR2C/2T10A	—	15.0	—	12.9	—	12.5	—	10.8	ns
	OR2C/2T12A	—	15.3	—	13.1	—	12.7	—	11.0	ns
	OR2C/2T15A	—	15.6	—	13.3	—	12.9	—	11.2	ns
	OR2C/2T26A	—	16.2	—	13.8	—	13.4	—	11.7	ns
	OR2C/2T40A	—	17.5	—	14.9	—	14.2	—	12.4	ns
CLK Input Pin → OUTPUT Pin (Sinklim)	OR2C/2T04A	—	15.6	—	13.5	—	13.3	—	11.4	ns
	OR2C/2T06A	—	15.8	—	13.6	—	13.4	—	11.5	ns
	OR2C/2T08A	—	16.0	—	13.8	—	13.6	—	11.7	ns
	OR2C/2T10A	—	16.2	—	14.0	—	13.8	—	11.9	ns
	OR2C/2T12A	—	16.7	—	14.3	—	14.0	—	12.1	ns
	OR2C/2T15A	—	17.0	—	14.5	—	14.2	—	12.3	ns
	OR2C/2T26A	—	17.6	—	15.0	—	14.7	—	12.8	ns
	OR2C/2T40A	—	18.9	—	16.1	—	15.5	—	13.5	ns

Notes:
The pin-to-pin timing parameters in this table should be used instead of results reported by *ORCA* Foundry.

This clock delay is for a fully routed clock tree that uses the primary clock network. It includes both the input buffer delay, the clock routing to the PFU CLK input, the clock→Q of the FF, and the delay through the output buffer. The delay will be reduced if any of the clock branches are not used. The given timing requires that the input clock pin be located at one of the four center PICs on any side of the device and that the direct FF→I/O routing be used.

If the clock pin is not located at one of the four center PICs, this delay must be increased by up to the following amounts: OR2C/2T04A = 1.5%, OR2C/2T06A = 2.0%, OR2C/2T08A = 3.1%, OR2C/2T10A = 3.9%, OR2C/2T12A = 4.9%, OR2C/2T15A = 5.7%, OR2C/2T26A = 8.1%, OR2C/2T40A = 12.5%.

Shaded values are preliminary.

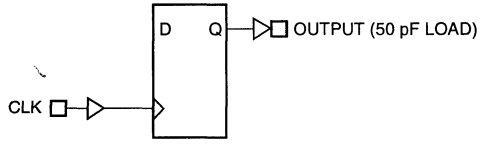


Figure 64. Global Clock to Output Delay

Timing Characteristics (continued)

Table 43. OR2CxxA/OR2TxxA Global Input to Clock Setup/Hold Time (Pin-to-Pin)

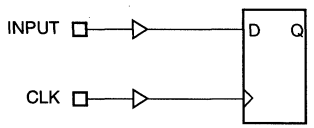
OR2CxxA Commercial: V_{DD} = 5.0 V ± 5%, 0 °C ≤ T_A ≤ 70 °C; Industrial: V_{DD} = 5.0 V ± 10%, –40 °C ≤ T_A ≤ +85 °C.
OR2TxxA Commercial: V_{DD} = 3.0 V to 3.6 V, 0 °C ≤ T_A ≤ 70 °C; Industrial: V_{DD} = 3.0 V to 3.6 V, –40 °C ≤ T_A ≤ +85 °C.

Description (T_J = All, V_{DD} = All)	Device	Speed								Unit
		-2		-3		-4		-5		
		Min	Max	Min	Max	Min	Max	Min	Max	
Input to CLK (TTL/CMOS) Setup Time (No delay)	OR2C/2T04A	0.0	—	0.0	—	0.0	—	0.0	—	ns
	OR2C/2T06A	0.0	—	0.0	—	0.0	—	0.0	—	ns
	OR2C/2T08A	0.0	—	0.0	—	0.0	—	0.0	—	ns
	OR2C/2T10A	0.0	—	0.0	—	0.0	—	0.0	—	ns
	OR2C/2T12A	0.0	—	0.0	—	0.0	—	0.0	—	ns
	OR2C/2T15A	0.0	—	0.0	—	0.0	—	0.0	—	ns
	OR2C/2T26A	0.0	—	0.0	—	0.0	—	0.0	—	ns
	OR2C/2T40A	0.0	—	0.0	—	0.0	—	0.0	—	ns
Input to CLK (TTL/CMOS) Setup Time (Delayed)	OR2C/2T04A	5.8	—	5.5	—	4.2	—	4.0	—	ns
	OR2C/2T06A	5.7	—	5.4	—	4.1	—	3.9	—	ns
	OR2C/2T08A	5.6	—	5.3	—	4.0	—	3.8	—	ns
	OR2C/2T10A	5.3	—	5.0	—	3.9	—	3.7	—	ns
	OR2C/2T12A	5.2	—	4.9	—	3.8	—	3.6	—	ns
	OR2C/2T15A	4.9	—	4.7	—	3.6	—	3.4	—	ns
	OR2C/2T26A	7.3	—	6.9	—	6.0	—	5.7	—	ns
	OR2C/2T40A	6.8	—	6.4	—	5.5	—	5.2	—	ns
Input to CLK (TTL/CMOS) Hold Time (No delay)	OR2C/2T04A	4.2	—	4.0	—	3.8	—	3.6	—	ns
	OR2C/2T06A	4.3	—	4.1	—	3.9	—	3.7	—	ns
	OR2C/2T08A	4.5	—	4.3	—	4.1	—	3.9	—	ns
	OR2C/2T10A	4.8	—	4.6	—	4.4	—	4.2	—	ns
	OR2C/2T12A	5.0	—	4.8	—	4.6	—	4.4	—	ns
	OR2C/2T15A	5.4	—	5.1	—	4.9	—	4.7	—	ns
	OR2C/2T26A	6.2	—	5.8	—	5.6	—	5.3	—	ns
	OR2C/2T40A	7.9	—	6.8	—	6.6	—	6.3	—	ns
Input to CLK (TTL/CMOS) Hold Time (Delayed)	OR2C/2T04A	0.0	—	0.0	—	0.0	—	0.0	—	ns
	OR2C/2T06A	0.0	—	0.0	—	0.0	—	0.0	—	ns
	OR2C/2T08A	0.0	—	0.0	—	0.0	—	0.0	—	ns
	OR2C/2T10A	0.0	—	0.0	—	0.0	—	0.0	—	ns
	OR2C/2T12A	0.0	—	0.0	—	0.0	—	0.0	—	ns
	OR2C/2T15A	0.0	—	0.0	—	0.0	—	0.0	—	ns
	OR2C/2T26A	0.0	—	0.0	—	0.0	—	0.0	—	ns
	OR2C/2T40A	0.0	—	0.0	—	0.0	—	0.0	—	ns

Notes:
The pin-to-pin timing parameters in this table should be used instead of results reported by *ORCA* Foundry.

This clock delay is for a fully routed clock tree that uses the primary clock network. It includes both the input buffer delay and the clock routing to the PFU CLK input. The delay will be reduced if any of the clock branches are not used. The given Setup (Delayed and No delay) and Hold (Delayed) timing allows the input clock pin to be located in any PIC on any side of the device, but direct I/O→FF routing must be used. The Hold (No delay) timing assumes the clock pin is located at one of the four center PICs and direct I/O→FF routing is used. If it is not located at one of the four center PICs, this delay must be increased by up to the following amounts: OR2C/2T04A = 5.3%, OR2C/2T06A = 6.4%, OR2C/2T08A = 7.3%, OR2C/2T10A = 9.1%, OR2C/2T12A = 10.8%, OR2C/2T15A = 12.2%, OR2C/2T26A = 16.1%, OR2C/2T40A = 21.2%.

Shaded values are preliminary.



5-4847(F)

Figure 65. Global Input to Clock Setup/Hold Time

Timing Characteristics (continued)

Table 44. Programmable I/O Cell Timing Characteristics

OR2CxxA Commercial: V_{DD} = 5.0 V ± 5%, 0 °C ≤ T_A ≤ 70 °C; OR2CxxA Industrial: V_{DD} = 5.0 V ± 10%, –40 °C ≤ T_A ≤ +85 °C.
OR2TxxA Commercial: V_{DD} = 3.0 V to 3.6 V, 0 °C ≤ T_A ≤ 70 °C; OR2TxxA Industrial: V_{DD} = 3.0 V to 3.6 V, –40 °C ≤ T_A ≤ +85 °C.

Parameter	Symbol	Speed								Unit
		-2		-3		-4		-5		
		Min	Max	Min	Max	Min	Max	Min	Max	
Inputs (T_J = 85 °C, V_{DD} = Min)										
Input Rise Time	T_R	—	500	—	500	—	500	—	500	ns
Input Fall Time	T_F	—	500	—	500	—	500	—	500	ns
Pad to In Delay	PAD_IN_DEL	—	1.9	—	1.6	—	1.3	—	1.1	ns
Pad to Nearest PFU Latch Output	CHIP_LATCH	—	6.2	—	4.7	—	4.1	—	3.3	ns
Delay Added to General Routing (input buffer in delay mode for OR2C/2T15A and smaller devices)	—	—	8.1	—	7.0	—	6.3	—	5.1	ns
Delay Added to General Routing (input buffer in delay mode for OR2C/2T26A and OR2C/2T40A)	—	—	11.0	—	9.7	—	8.9	—	7.2	ns
Delay Added to Direct-FF Routing (input buffer in delay mode for OR2C/2T15A and smaller devices)	—	—	8.0	—	6.8	—	5.9	—	4.7	ns
Delay Added to Direct-FF Routing (input buffer in delay mode for OR2C/2T26A and OR2C/2T40A)	—	—	10.9	—	9.6	—	8.5	—	6.8	ns
Outputs (T_J = 85 °C, V_{DD} = Min, C_L = 50 pF)										
PFU ck to Pad Delay (dout[3:0] to pad):										
Fast	DOUT_DEL(F)	—	7.6	—	5.7	—	5.5	—	4.7	ns
Slewlim	DOUT_DEL(SL)	—	9.3	—	8.0	—	7.5	—	6.4	ns
Sinklim	DOUT_DEL(SI)	—	12.4	—	8.9	—	8.8	—	7.5	ns
Output to Pad Delay (out[3:0] to pad):										
Fast	OUT_DEL(F)	—	5.0	—	4.0	—	3.6	—	3.1	ns
Slewlim	OUT_DEL(SL)	—	6.7	—	6.3	—	5.6	—	4.8	ns
Sinklim	OUT_DEL(SI)	—	9.8	—	7.2	—	6.9	—	5.9	ns
3-state Enable Delay (ts[3:0] to pad):										
Fast	TS_DEL(F)	—	5.8	—	4.7	—	4.0	—	3.4	ns
Slewlim	TS_DEL(SL)	—	7.5	—	7.0	—	6.0	—	5.1	ns
Sinklim	TS_DEL(SI)	—	10.6	—	7.9	—	7.3	—	6.2	ns

Notes:
If the input buffer is placed in delay mode, the chip hold time to the nearest PFU latch is guaranteed to be 0 if the clock is routed using the primary clock network; (T_J = All, V_{DD} = All). It should also be noted that any signals routed on the clock R-nodes or using the TRIDI buffers directly from the input buffer do not get delayed at any time.

The delays for all input buffers assume an input rise/fall time of ≤1 V/ns.

Shaded values are preliminary.

Timing Characteristics (continued)

Table 45. General Configuration Mode Timing Characteristics

OR2CxxA Commercial: VDD = 5.0 V ± 5%, 0 °C ≤ TA ≤ 70 °C; OR2CxxA Industrial: VDD = 5.0 V ± 10%, –40 °C ≤ TA ≤ +85 °C.
OR2TxxA Commercial: VDD = 3.0 V to 3.6 V, 0 °C ≤ TA ≤ 70 °C; OR2TxxA Industrial: VDD = 3.0 V to 3.6 V, –40 °C ≤ TA ≤ +85 °C.

Parameter		Symbol	Min	Max	Unit
All Configuration Modes					
M[3:0] Setup Time to $\overline{\text{INIT}}$ High		TSMODE	50.0	—	ns
M[3:0] Hold Time from $\overline{\text{INIT}}$ High		THMODE	600.0	—	ns
$\overline{\text{RESET}}$ Pulse Width Low to Start Reconfiguration		TRW	50.0	—	ns
$\overline{\text{PRGM}}$ Pulse Width Low to Start Reconfiguration		TPGW	50.0	—	ns
Master and Asynchronous Peripheral Modes					
Power-on Reset Delay		TPO	17.30	69.47	ms
CCLK Period (M3 = 0)		TCCLK	66.00	265.00	ns
(M3 = 1)			528.00	2120.00	ns
Configuration Latency (noncompressed)		TCL			
OR2C/2T04A	(M3 = 0)		4.31	17.30*	ms
	(M3 = 1)		34.48	138.40*	ms
OR2C/2T06A	(M3 = 0)		6.00	24.08*	ms
	(M3 = 1)		48.00	192.64*	ms
OR2C/2T08A	(M3 = 0)		7.62	30.60*	ms
	(M3 = 1)		60.96	244.80*	ms
OR2C/2T10A	(M3 = 0)		9.82	39.43*	ms
	(M3 = 1)		78.56	315.44*	ms
OR2C/2T12A	(M3 = 0)		11.86	47.62*	ms
	(M3 = 1)		94.88	380.96*	ms
OR2C/2T15A	(M3 = 0)		14.57	58.51*	ms
	(M3 = 1)		116.56	468.08*	ms
OR2C/2T26A	(M3 = 0)		20.25	81.32*	ms
	(M3 = 1)		162.00	650.56*	ms
OR2C/2T40A	(M3 = 0)		31.29	125.62*	ms
	(M3 = 1)		250.32	1004.96*	ms
Slave Serial and Synchronous Peripheral Modes					
Power-on Reset Delay		TPO	4.33	17.37	ms
CCLK Period		TCCLK	100.00	—	ns
Configuration Latency (noncompressed):		TCL			
OR2C/2T04A			6.53	—	ms
OR2C/2T06A			9.09	—	ms
OR2C/2T08A			11.55	—	ms
OR2C/2T10A			14.88	—	ms
OR2C/2T12A			17.97	—	ms
OR2C/2T15A			22.08	—	ms
OR2C/2T26A			30.69	—	ms
OR2C/2T40A			47.40	—	ms

* Not applicable to asynchronous peripheral mode.

Timing Characteristics (continued)

Table 45. General Configuration Mode Timing Characteristics (continued)

OR2CxxA Commercial: VDD = 5.0 V ± 5%, 0 °C ≤ TA ≤ 70 °C; OR2CxxA Industrial: VDD = 5.0 V ± 10%, –40 °C ≤ TA ≤ +85 °C.
OR2TxxA Commercial: VDD = 3.0 V to 3.6 V, 0 °C ≤ TA ≤ 70 °C; OR2TxxA Industrial: VDD = 3.0 V to 3.6 V, –40 °C ≤ TA ≤ +85 °C.

Parameter	Symbol	Min	Max	Unit
Slave Parallel Mode				
Power-on Reset Delay	TPO	4.33	17.37	ms
CCLK Period	TCCLK	100.00	—	ns
Configuration Latency (Noncompressed):	TCL			
OR2C/2T04A		0.82	—	ms
OR2C/2T06A		1.14	—	ms
OR2C/2T08A		1.44	—	ms
OR2C/2T10A		1.86	—	ms
OR2C/2T12A		2.25	—	ms
OR2C/2T15A		2.76	—	ms
OR2C/2T26A		3.84	—	ms
OR2C/2T40A		5.93	—	ms
Partial Reconfiguration (Noncompressed)	TPR			
OR2C/2T04A		1.70	—	µs/frame
OR2C/2T06A		2.00	—	µs/frame
OR2C/2T08A		2.20	—	µs/frame
OR2C/2T10A		2.50	—	µs/frame
OR2C/2T12A		2.70	—	µs/frame
OR2C/2T15A		3.00	—	µs/frame
OR2C/2T26A		3.50	—	µs/frame
OR2C/2T40A		4.30	—	µs/frame
$\overline{\text{INIT}}$ Timing				
$\overline{\text{INIT}}$ High to CCLK Delay	TINIT_CCLK			
Slave Parallel		1.00	—	µs
Slave Serial		1.00	—	µs
Synchronous Peripheral		1.00	—	µs
Master Serial				
(M3 = 1)		1.06	4.51	µs
(M3 = 0)		0.59	2.65	µs
Master Parallel				
(M3 = 1)		5.28	21.47	µs
(M3 = 0)		1.12	4.77	µs
Initialization Latency ($\overline{\text{PRGM}}$ high to $\overline{\text{INIT}}$ high)	TIL			
OR2C/2T04A		63.36	254.40	µs
OR2C/2T06A		74.98	301.04	µs
OR2C/2T08A		86.59	347.68	µs
OR2C/2T10A		98.21	394.32	µs
OR2C/2T12A		109.82	440.96	µs
OR2C/2T15A		121.44	487.60	µs
OR2C/2T26A		144.67	580.88	µs
OR2C/2T40A		181.90	730.34	µs
$\overline{\text{INIT}}$ High to $\overline{\text{WR}}$, Asynchronous Peripheral	TINIT_WR	1.50	—	µs

Note: TPO is triggered when VDD reaches between 3.0 V to 4.0 V for the OR2CxxA and between 2.7 V and 3.0 V for the OR2TxxA.

Timing Characteristics (continued)

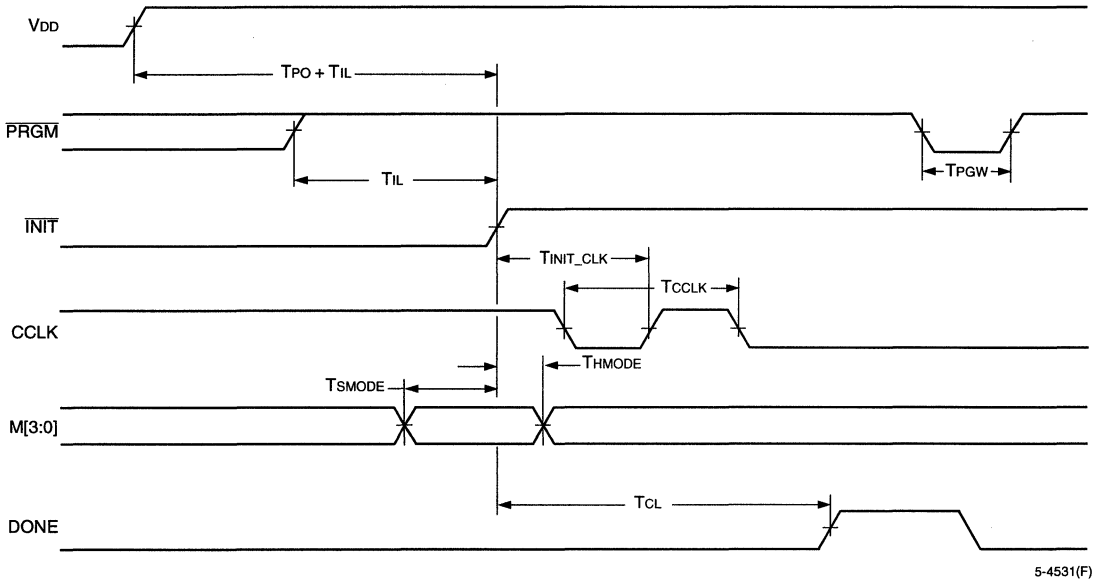


Figure 66. General Configuration Mode Timing Diagram

Timing Characteristics (continued)

Table 46. Master Serial Configuration Mode Timing Characteristics

OR2CxxA Commercial: V_{DD} = 5.0 V ± 5%, 0 °C ≤ T_A ≤ 70 °C; OR2CxxA Industrial: V_{DD} = 5.0 V ± 10%, –40 °C ≤ T_A ≤ +85 °C.
OR2TxxA Commercial: V_{DD} = 3.0 V to 3.6 V, 0 °C ≤ T_A ≤ 70 °C; OR2TxxA Industrial: V_{DD} = 3.0 V to 3.6 V, –40 °C ≤ T_A ≤ +85 °C.

Parameter	Symbol	Min	Nom	Max	Unit
DIN Setup Time	T_S	60.0	—	—	ns
DIN Hold Time	T_H	0	—	—	ns
CCLK Frequency (M3 = 0)	F_C	3.8	10.0	15.2	MHz
CCLK Frequency (M3 = 1)	F_C	0.48	1.25	1.9	MHz
CCLK to DOUT Delay	T_D	—	—	30	ns

Note: Serial configuration data is transmitted out on DOUT on the falling edge of CCLK after it is input DIN.

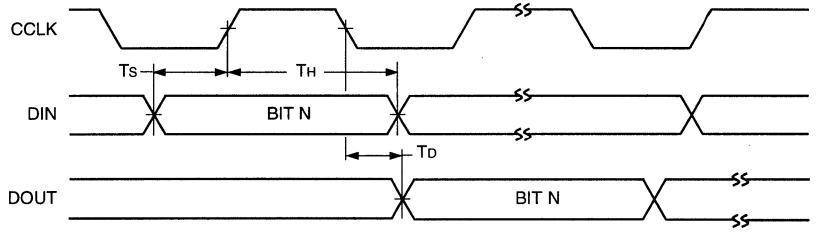


Figure 67. Master Serial Configuration Mode Timing Diagram

Timing Characteristics (continued)

Table 47. Master Parallel Configuration Mode Timing Characteristics

OR2CxxA Commercial: VDD = 5.0 V ± 5%, 0 °C ≤ TA ≤ 70 °C; OR2CxxA Industrial: VDD = 5.0 V ± 10%, –40 °C ≤ TA ≤ +85 °C.
OR2TxxA Commercial: VDD = 3.0 V to 3.6 V, 0 °C ≤ TA ≤ 70 °C; OR2TxxA Industrial: VDD = 3.0 V to 3.6 V, –40 °C ≤ TA ≤ +85 °C.

Parameter	Symbol	Min	Max	Unit
RCLK to Address Valid	TAV	0	200	ns
D[7:0] Setup Time to RCLK High	TS	60	—	ns
D[7:0] Hold Time to RCLK High	TH	0	—	ns
RCLK Low Time (M3 = 0)	TCL	462	1855	ns
RCLK High Time (M3 = 0)	TCH	66	265	ns
RCLK Low Time (M3 = 1)	TCL	3696	14840	ns
RCLK High Time (M3 = 1)	TCH	528	2120	ns
CCLK to DOUT	TD	—	30	ns

Notes:
The RCLK period consists of seven CCLKs for RCLK low and one CCLK for RCLK high.
Serial data is transmitted out on DOUT 1.5 CCLK cycles after the byte is input D[7:0].

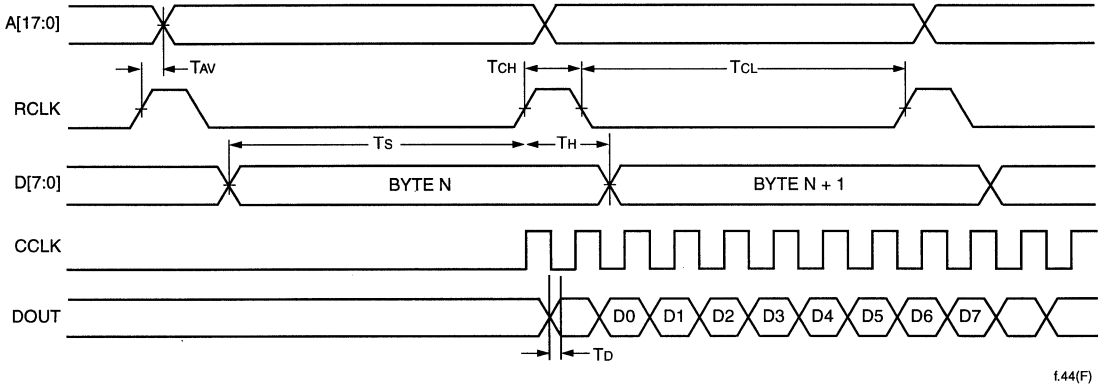


Figure 68. Master Parallel Configuration Mode Timing Diagram

Timing Characteristics (continued)

Table 48. Asynchronous Peripheral Configuration Mode Timing Characteristics

OR2CxxA Commercial: V_{DD} = 5.0 V ± 5%, 0 °C ≤ T_A ≤ 70 °C; OR2CxxA Industrial: V_{DD} = 5.0 V ± 10%, –40 °C ≤ T_A ≤ +85 °C.
OR2TxxA Commercial: V_{DD} = 3.0 V to 3.6 V, 0 °C ≤ T_A ≤ 70 °C; OR2TxxA Industrial: V_{DD} = 3.0 V to 3.6 V, –40 °C ≤ T_A ≤ +85 °C.

Parameter	Symbol	Min	Max	Unit
$\overline{WR}$, $\overline{CS0}$, and CS1 Pulse Width	T_{WR}	100	—	ns
D[7:0] Setup Time	T_S	20	—	ns
D[7:0] Hold Time	T_H	0	—	ns
RDY/$\overline{BUSY}$ Delay	T_{RDY}	—	60	ns
RDY/$\overline{BUSY}$ Low	T_B	1	8	CCLK Periods
Earliest $\overline{WR}$ After End of $\overline{BUSY}$	T_{WR2}	0	—	ns
CCLK to DOUT	T_D	—	30	ns

Notes:
Serial data is transmitted out on DOUT on the falling edge of CCLK after the byte is input D[7:0].

When the RDY/$\overline{BUSY}$ status is read on D7, the timing from $\overline{RD}$ to D7 is the same as the $\overline{WR}$ to RDY/$\overline{BUSY}$ delay (T_{RDY}).

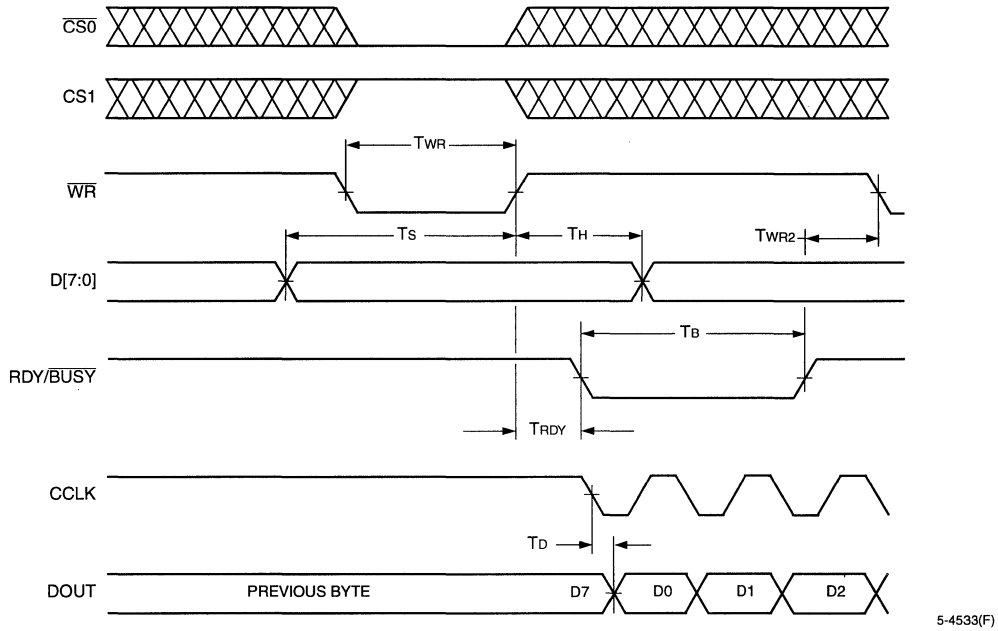


Figure 69. Asynchronous Peripheral Configuration Mode Timing Diagram

Timing Characteristics (continued)

Table 49. Synchronous Peripheral Configuration Mode Timing Characteristics

OR2CxxA Commercial: V_{DD} = 5.0 V ± 5%, 0 °C ≤ T_A ≤ 70 °C; OR2CxxA Industrial: V_{DD} = 5.0 V ± 10%, –40 °C ≤ T_A ≤ +85 °C.
OR2TxxA Commercial: V_{DD} = 3.0 V to 3.6 V, 0 °C ≤ T_A ≤ 70 °C; OR2TxxA Industrial: V_{DD} = 3.0 V to 3.6 V, –40 °C ≤ T_A ≤ +85 °C.

Parameter	Symbol	Min	Max	Unit
D[7:0] Setup Time	T_S	20	—	ns
D[7:0] Hold Time	T_H	0	—	ns
CCLK High Time	T_{CH}	50	—	ns
CCLK Low Time	T_{CL}	50	—	ns
CCLK Frequency	F_C	—	10	MHz
CCLK to DOUT	T_D	—	30	ns

Note: Serial data is transmitted out on DOUT 1.5 clock cycles after the byte is input D[7:0].

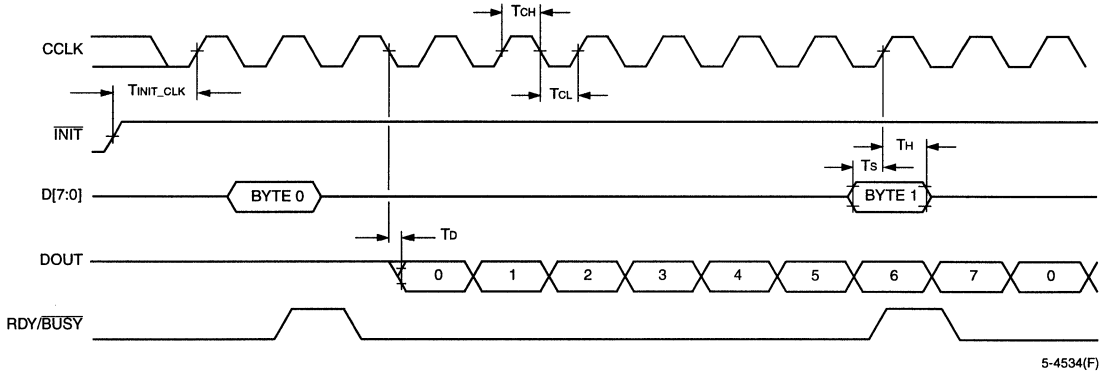


Figure 70. Synchronous Peripheral Configuration Mode Timing Diagram

Timing Characteristics (continued)

Table 50. Slave Serial Configuration Mode Timing Characteristics

OR2CxxA Commercial: V_{DD} = 5.0 V ± 5%, 0 °C ≤ T_A ≤ 70 °C; OR2CxxA Industrial: V_{DD} = 5.0 V ± 10%, –40 °C ≤ T_A ≤ +85 °C.
OR2TxxA Commercial: V_{DD} = 3.0 V to 3.6 V, 0 °C ≤ T_A ≤ 70 °C; OR2TxxA Industrial: V_{DD} = 3.0 V to 3.6 V, –40 °C ≤ T_A ≤ +85 °C.

Parameter	Symbol	Min	Max	Unit
DIN Setup Time	T_S	20	—	ns
DIN Hold Time	T_H	0	—	ns
CCLK High Time	T_{CH}	50	—	ns
CCLK Low Time	T_{CL}	50	—	ns
CCLK Frequency	F_C	—	10	MHz
CCLK to DOUT	T_D	—	30	ns

Note: Serial configuration data is transmitted out on DOUT on the falling edge of CCLK after it is input on DIN.

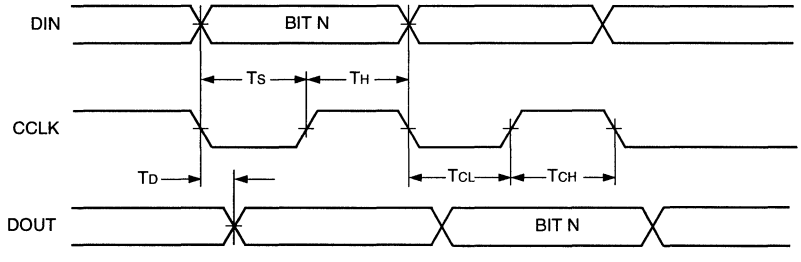


Figure 71. Slave Serial Configuration Mode Timing Diagram

Timing Characteristics (continued)

Table 51. Slave Parallel Configuration Mode Timing Characteristics

OR2CxxA Commercial: V_{DD} = 5.0 V ± 5%, 0 °C ≤ T_A ≤ 70 °C; OR2CxxA Industrial: V_{DD} = 5.0 V ± 10%, –40 °C ≤ T_A ≤ +85 °C.
OR2TxxA Commercial: V_{DD} = 3.0 V to 3.6 V, 0 °C ≤ T_A ≤ 70 °C; OR2TxxA Industrial: V_{DD} = 3.0 V to 3.6 V, –40 °C ≤ T_A ≤ +85 °C.

Parameter	Symbol	Min	Max	Unit
$\overline{CS0}$, CS1, $\overline{WR}$ Setup Time	T_{S1}	60	—	ns
$\overline{CS0}$, CS1, $\overline{WR}$ Hold Time	T_{H1}	20	—	ns
D[7:0] Setup Time	T_{S2}	20	—	ns
D[7:0] Hold Time	T_{H2}	0	—	ns
CCLK High Time	T_{CH}	50	—	ns
CCLK Low Time	T_{CL}	50	—	ns
CCLK Frequency	F_C	—	10	MHz

Note: Daisy chaining of FPGAs is not supported in this mode.

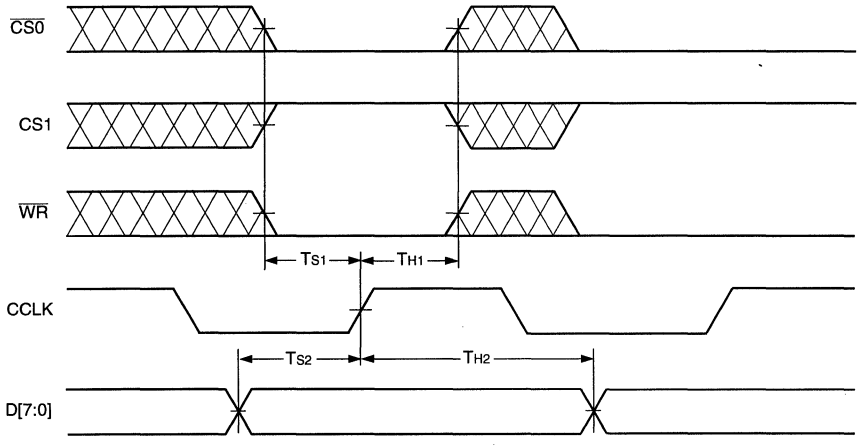


Figure 72. Slave Parallel Configuration Mode Timing Diagram

Timing Characteristics (continued)

Table 52. Readback Timing Characteristics

OR2CxxA Commercial: V_{DD} = 5.0 V ± 5%, 0 °C ≤ T_A ≤ 70 °C; OR2CxxA Industrial: V_{DD} = 5.0 V ± 10%, –40 °C ≤ T_A ≤ +85 °C.
OR2TxxA Commercial: V_{DD} = 3.0 V to 3.6 V, 0 °C ≤ T_A ≤ 70 °C; OR2TxxA Industrial: V_{DD} = 3.0 V to 3.6 V, –40 °C ≤ T_A ≤ +85 °C.

Parameter	Symbol	Min	Max	Unit
$\overline{RD\_CFGN}$ to CCLK Setup Time	T_S	50	—	ns
$\overline{RD\_CFGN}$ High Width to Abort Readback	T_{RBA}	2	—	CCLK
CCLK Low Time	T_{CL}	50	—	ns
CCLK High Time	T_{CH}	50	—	ns
CCLK Frequency	F_C	—	10	MHz
CCLK to RD_DATA Delay	T_D	—	50	ns

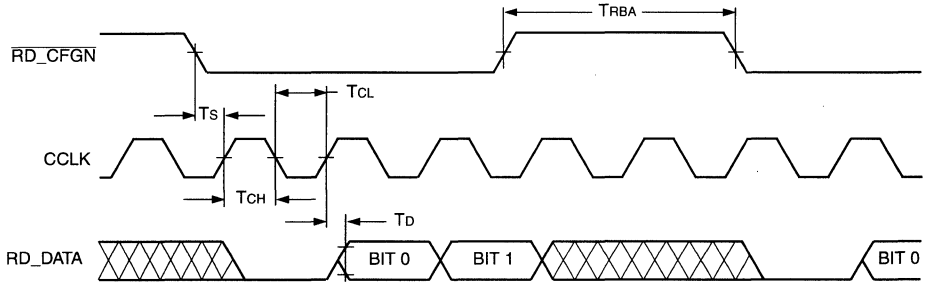


Figure 73. Readback Timing Diagram

Timing Characteristics (continued)

Table 53. Boundary-Scan Timing Characteristics

OR2CxxA Commercial: V_{DD} = 5.0 V ± 5%, 0 °C ≤ T_A ≤ 70 °C; OR2CxxA Industrial: V_{DD} = 5.0 V ± 10%, –40 °C ≤ T_A ≤ +85 °C.
OR2TxxA Commercial: V_{DD} = 3.0 V to 3.6 V, 0 °C ≤ T_A ≤ 70 °C; OR2TxxA Industrial: V_{DD} = 3.0 V to 3.6 V, –40 °C ≤ T_A ≤ +85 °C.

Parameter	Symbol	Min	Max	Unit
TDI/TMS to TCK Setup Time	T_S	25	—	ns
TDI/TMS Hold Time from TCK	T_H	0	—	ns
TCK Low Time	T_{CL}	50	—	ns
TCK High Time	T_{CH}	50	—	ns
TCK to TDO Delay	T_D	—	20	ns
TCK Frequency	T_{TCK}	—	10	MHz

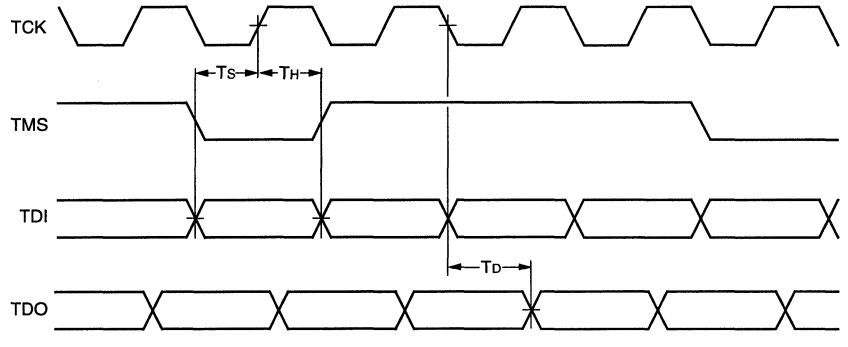
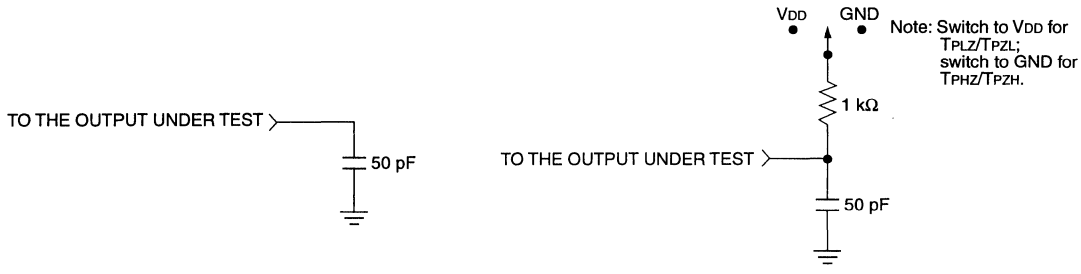


Figure 74. Boundary-Scan Timing Diagram

Measurement Conditions

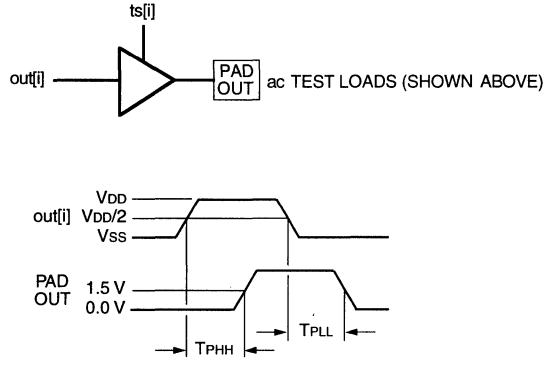


A. Load Used to Measure Propagation Delay

B. Load Used to Measure Rising/Falling Edges

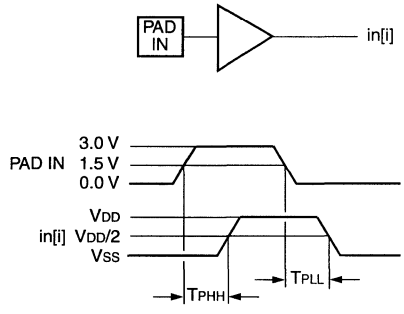
5-3234(F)

Figure 75. ac Test Loads



5-3233.a(F)

Figure 76. Output Buffer Delays



5-3235(F)

Figure 77. Input Buffer Delays

Output Buffer Characteristics
OR2CxxA

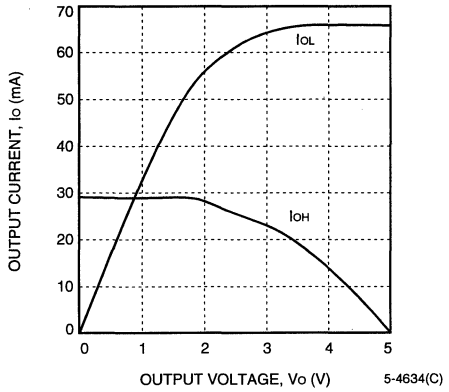


Figure 78. Sinklim (T_J = 25 °C, V_{DD} = 5.0 V)

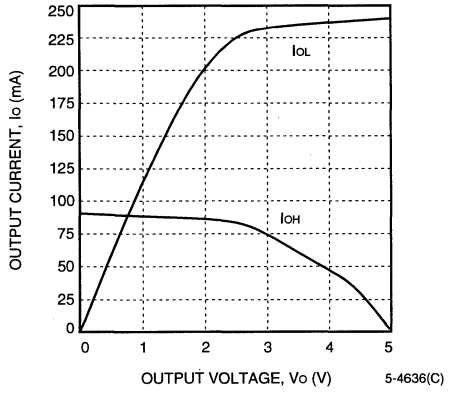


Figure 79. Slewlim (T_J = 25 °C, V_{DD} = 5.0 V)

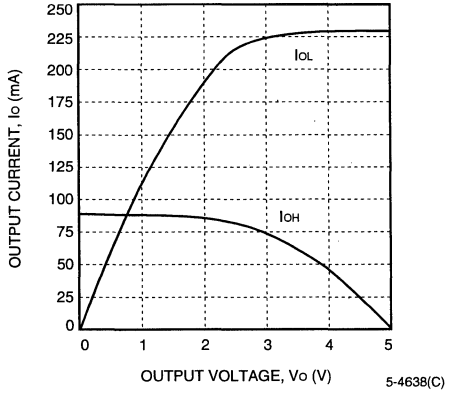


Figure 80. Fast (T_J = 25 °C, V_{DD} = 5.0 V)

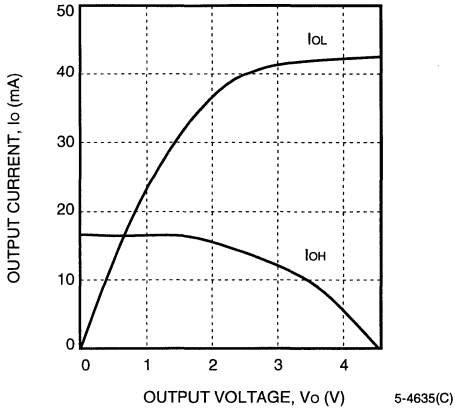


Figure 81. Sinklim (T_J = 125 °C, V_{DD} = 4.5 V)

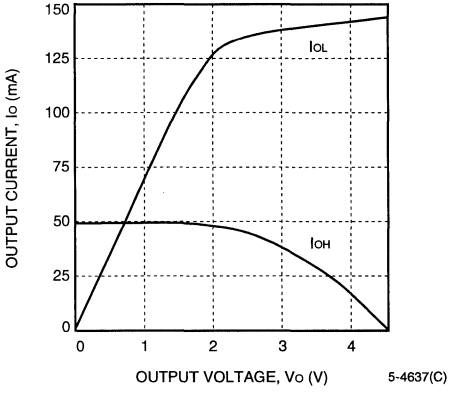


Figure 82. Slewlim (T_J = 125 °C, V_{DD} = 4.5 V)

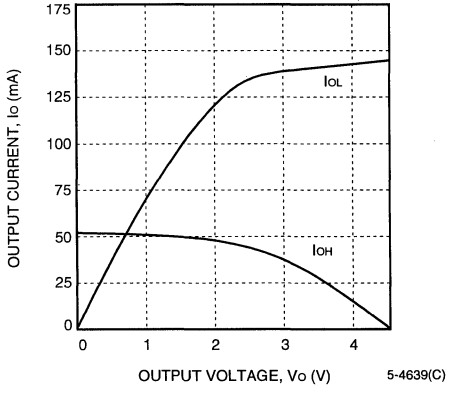


Figure 83. Fast (T_J = 125 °C, V_{DD} = 4.5 V)

Output Buffer Characteristics (continued)
OR2TxxA

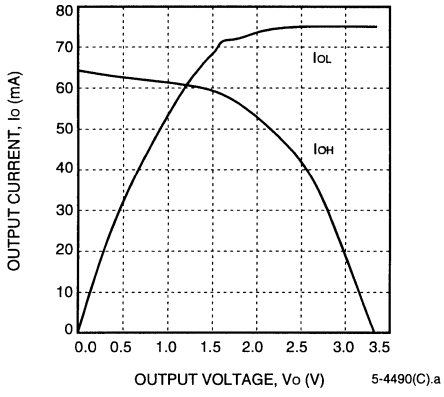


Figure 84. Sinklim (T_J = 25 °C, V_{DD} = 3.3 V)

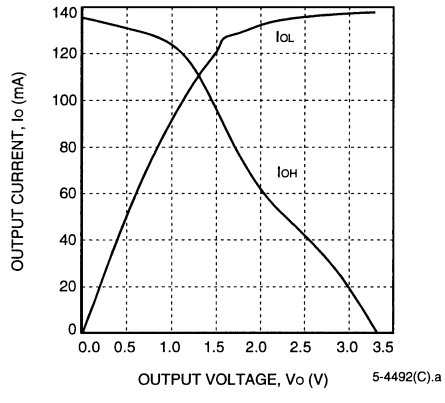


Figure 85. Slewlim (T_J = 25 °C, V_{DD} = 3.3 V)

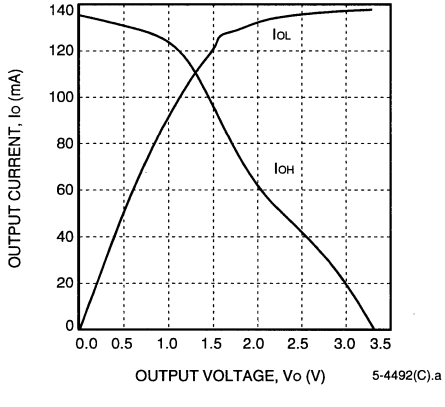


Figure 86. Fast (T_J = 25 °C, V_{DD} = 3.3 V)

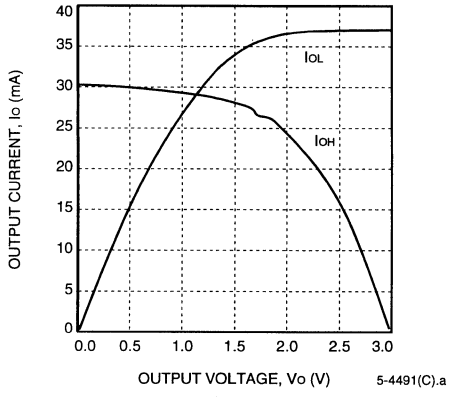


Figure 87. Sinklim (T_J = 125 °C, V_{DD} = 3.0 V)

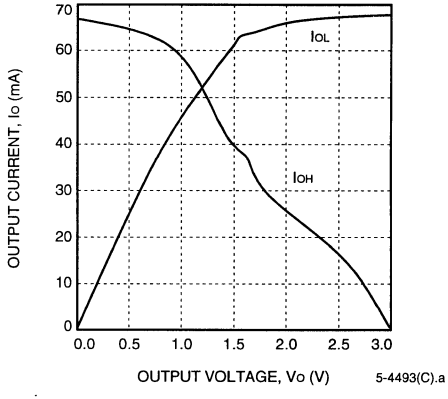


Figure 88. Slewlim (T_J = 125 °C, V_{DD} = 3.0 V)

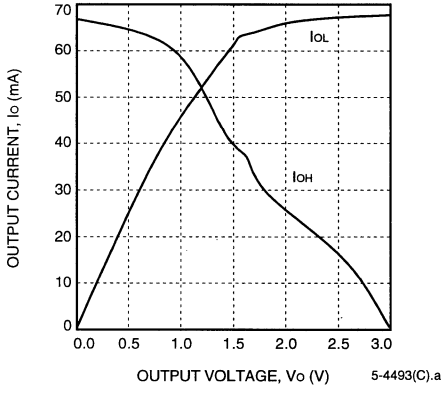
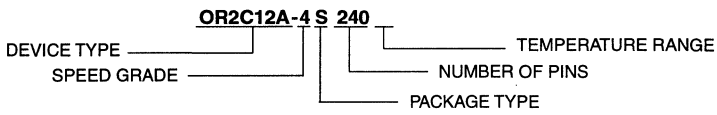


Figure 89. Fast (T_J = 125 °C, V_{DD} = 3.0 V)

Ordering Information

Example:



OR2C12A, -4 Speed Grade, 240-pin Shrink Quad Flat Pack, Commercial Temperature.

Table 54. FPGA Voltage Options

Device	Voltage
OR2CxxA	5.0
OR2TxxA	3.3

Table 55. FPGA Temperature Options

Symbol	Description	Temperature
(Blank)	Commercial	0 °C to 70 °C
I	Industrial	–40 °C to +85 °C

Table 56. FPGA Package Options

Symbol	Description
BA	Plastic Ball Grid Array (PBGA)
BC	Enhanced Ball Grid Array (EBGA)
J	Quad Flat Package (QFP)
M	Plastic Leaded Chip Carrier (PLCC)
PS	Power Quad Shrink Flat Package (SQFP2)
S	Shrink Quad Flat Package (SQFP)
T	Thin Quad Flat Package (TQFP)

Table 57. *ORCA* OR2CxxA/OR2TxxA Series Package Matrix

Packages	84-Pin PLCC	100-Pin TQFP	144-Pin TQFP	160-Pin QFP	208-Pin EIAJ SQFP/ SQFP2	240-Pin EIAJ SQFP/ SQFP2	256-Pin PBGA	304-Pin EIAJ SQFP/ SQFP2	352-Pin PBGA	432-Pin EBGA	600-Pin EBGA
	M84	T100	T144	J160	S208/ PS208	S240/ PS240	BA256	S304/ PS304	BA352	BC432	BC600
OR2C/2T04A	CI	CI	CI	CI	CI	—	—	—	—	—	—
OR2C/2T06A	CI	CI	CI	CI	CI	CI	CI	—	—	—	—
OR2C/2T08A	CI	—	—	CI	CI	CI	CI	—	—	—	—
OR2C/2T10A	CI	—	—	CI	CI	CI	CI	—	CI	—	—
OR2C/2T12A	CI	—	—	—	CI	CI	CI	CI	CI	—	—
OR2C/2T15A	CI	—	—	—	CI	CI	CI	CI	CI	CI	—
OR2C/2T26A	—	—	—	—	CI	CI	—	CI	CI	CI	CI
OR2C/2T40A	—	—	—	—	CI	CI	—	CI	—	CI	CI

Key: C = commercial, I = industrial.

Notes:
The package options with the SQFP/SQFP2 designation in the table above use the SQFP package for all densities up to and including the OR2C/2T15A, while the OR2C/2T26A and the OR2C/2T40A use the SQFP2. Availability of the OR2C15A in the 208-pin and 240-pin SQFP2 is to be announced.

The OR2TxxA series is not offered in the 304-pin SQFP/SQFP2 packages.

Lucent Technologies
Bell Labs Innovations

Optimized Reconfigurable Cell Array (*ORCA*) ATT2Cxx Series Field-Programmable Gate Arrays

Features

- High-performance, cost-effective 0.5 µm technology (four-input look-up table delay less than 3.6 ns)
- High density (up to 43,200 usable, logic-only gates, or 99,400 gates including RAM)
- Up to 480 user I/Os
- Fast on-chip user SRAM: 64 bits/logic block
- Nibble-oriented architecture for implementing 4-, 8-, 16-, 32-bit (or wider) bus structures
- Innovative, abundant, and hierarchical nibble-oriented routing resources that allow automatic use of internal gates for all device densities without sacrificing performance
- Four 16-bit look-up tables and four latches/ flip-flops per logic block
- Internal fast carry for arithmetic functions
- TTL or CMOS input thresholds programmable per pin
- Individually programmable drive capability: 12 mA sink/6 mA source or 6 mA sink/3 mA source
- Built-in boundary scan (*IEEE* 1149.1)
- Low power consumption from submicron CMOS process
- Full PCI-bus compliance
- Supported by industry-standard CAE tools for design entry, synthesis, and simulation
- *ORCA* Foundry Development System support

Description

The Lucent Technologies Optimized Reconfigurable Cell Array (*ORCA*) series is the second generation of SRAM-based field-programmable gate arrays (FPGAs) from Lucent. The *ORCA* 2C FPGA series provides seven CMOS FPGAs ranging in complexity from 4,800 to 43,200 usable gates in a variety of packages, speed grades, and temperature ranges. Table 1 lists the usable gates for the 0.5 µm *ORCA* 2C series FPGAs.

The *ORCA* series FPGA consists of two basic elements: programmable logic cells (PLCs) and programmable input/output cells (PICs). An array of PLCs is surrounded by PICs. Each PLC contains a programmable function unit (PFU). The PLCs and PICs also contain routing resources and configuration RAM. All logic is done in the PFU. Each PFU contains four 16-bit look-up tables (LUTs) and four latches/flip-flops (FFs).

The PLC architecture provides a balanced mix of logic and routing which allows a higher utilized gate/PFU than alternative architectures. The routing resources carry logic signals between PFUs and I/O pads. The routing in the PLC is symmetrical about the horizontal and vertical axes. This improves routability by allowing a signal to be routed into the PLC from any direction.

Table 1. *ORCA* 2C Series FPGAs

Device	Usable Gates*	Latches/ Flip-Flops	Max User RAM Bits	User I/Os	Array Size
2C04	4,800—11,000	400	6,400	160	10 x 10
2C06	6,900—15,900	576	9,216	192	12 x 12
2C08	9,400—21,600	784	12,544	224	14 x 14
2C10	12,300—28,300	1024	16,384	256	16 x 16
2C12	15,600—35,800	1296	20,736	288	18 x 18
2C15	19,200—44,200	1600	25,600	320	20 x 20
2C26	27,600—63,600	2304	36,864	384	24 x 24
2C40	43,200—99,400	3600	57,600	480	30 x 30

* The first number in the usable gates column assumes 48 gates per PFU for logic-only designs. The second number assumes 30% of a design is RAM. PFUs used as RAM are counted at 4 gates per bit, with each PFU capable of implementing a 16 x 4 RAM (or 256 gates) per PFU.

Table of Contents

Description (continued)

The *ORCA* Foundry Development System is used to process a design from a netlist to a configured FPGA. Lucent provides interfaces and libraries to popular CAE tools for design entry and simulation.

The FPGA's functionality is determined by internal configuration RAM. The FPGA's internal initialization/ configuration circuitry loads the configuration data at powerup or under system control. The RAM is loaded by using one of several configuration modes. The configuration data resides externally in an EEPROM, EPROM, or ROM on the circuit board, or any other storage media. Serial ROMs provide a simple, low pin count method for configuring FPGAs.

ORCA Foundry Development System Overview

The *ORCA* Foundry Development System interfaces to front-end design entry tools and provides the tools to produce a configured FPGA. In the design flow, the user defines the functionality of the FPGA at two points in the design flow: at design entry and at the bit stream generation stage.

Following design entry, the development system's map, place, and route tools translate the netlist into a routed FPGA. Its bit stream generator is then used to generate the configuration data which is loaded into the FPGA's internal configuration RAM. When using the bit stream generator, the user selects options that affect the functionality of the FPGA. Combined with the front-end tools, *ORCA* Foundry produces configuration data that implements the various logic and routing options discussed in this data sheet.

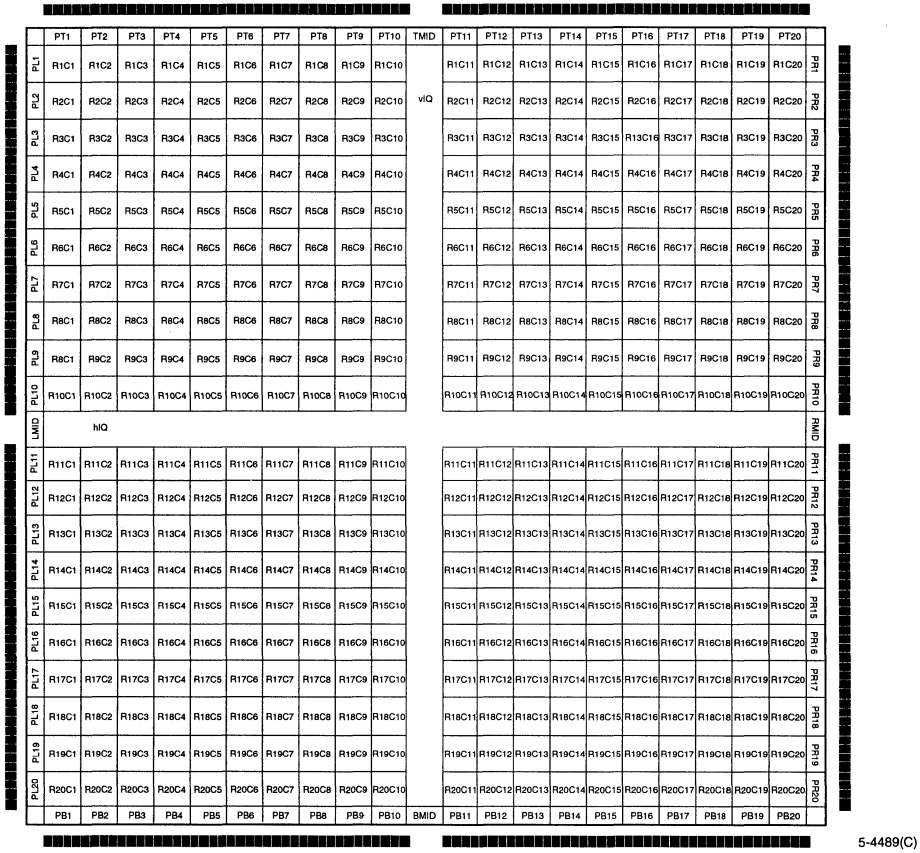


Figure 1. ATT2C15 Array

Architecture

The *ORCA* Series FPGA is comprised of two basic elements: PLCs and PICs. Figure 1 shows an array of programmable logic cells (PLCs) surrounded by programmable input/output cells (PICs). The ATT2C15 has PLCs arranged in an array of 20 rows and 20 columns. PICs are located on all four sides of the FPGA between the PLCs and the IC edge. The location of a PLC is indicated by its row and column so that a PLC in the second row and third column is R2C3. PICs are indicated similarly, with PT (top) and PB (bottom) designating rows and PL (left) and PR (right) designating columns, followed by a number. The routing resources and configuration RAM are not shown, but the interquad routing blocks (hIQ, vIQ) present in the 2C series are shown.

Each PIC contains the necessary I/O buffers to interface to bond pads. The PICs also contain the routing resources needed to connect signals from the bond pads to/from PLCs. The PICs do not contain any user-accessible logic elements, such as flip-flops.

Combinatorial logic is done in look-up tables (LUTs) located in the PFU. The PFU can be used in different modes to meet different logic requirements. The LUT's configurable medium-/large-grain architecture can be used to implement from one to four combinatorial logic functions. The flexibility of the LUT to handle wide input functions as well as multiple smaller input functions maximizes the gate count/PFU.

Programmable Logic Cells

The programmable logic cell (PLC) consists of a programmable function unit (PFU) and routing resources. All PLCs in the array are identical. The PFU, which contains four LUTs and four latches/FFs for logic implementation, is discussed in the next section.

Programmable Function Unit

The programmable function units (PFUs) are used for logic. The PFU has 19 external inputs and six outputs and can operate in several modes. The functionality of the inputs and outputs depends on the operating mode.

The PFU uses three input data buses (a[4:0], b[4:0], wd[3:0]), four control inputs (c0, ck, ce, lsr), and a carry input (cin); the last is used for fast arithmetic functions. There is a 5-bit output bus (o[4:0]) and a carry-out (cout).

Figure 2 and Figure 3 show high-level and detailed views of the ports in the PFU, respectively. The ports are referenced with a two- to four-character suffix to a PFU's location. As mentioned, there are two 5-bit input data buses (a[4:0] and b[4:0]) to the LUT, one 4-bit input data bus (wd[3:0]) to the latches/FFs, and an output data bus (o[4:0]).

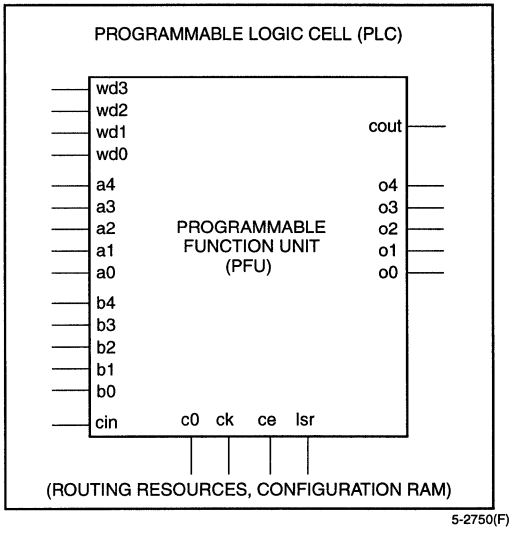


Figure 2. PFU Ports

The PFU is used in a variety of modes, as illustrated in Figures 4 through 11, and it is these specific modes that are most relevant to PFU functionality.

The PFU does combinatorial logic in the LUT and sequential logic in the latches/FFs. The LUT is static random access memory (SRAM) and can be used for read/write or read-only memory. Table 2 lists the basic operating modes of the LUT. The operating mode affects the functionality of the PFU input and output ports and internal PFU routing.

Programmable Logic Cells (continued)

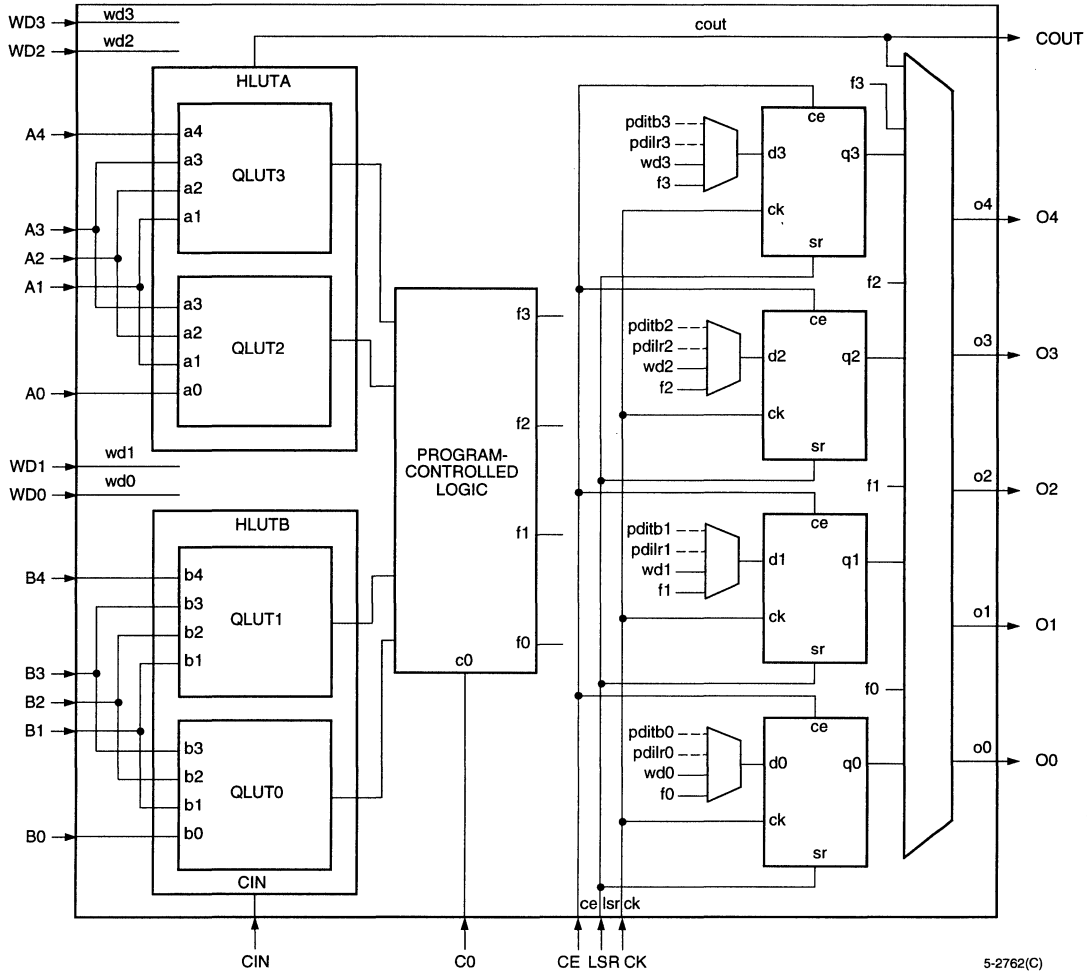


Figure 3. Simplified PFU Diagram

For example, in some operating modes, the wd[3:0] inputs are direct data inputs to the PFU latches/FFs. In the dual 16 x 2 memory mode, the same wd[3:0] inputs are used as a 4-bit data input bus into LUT memory.

Figure 3 shows the four latches/FFs and the 64-bit look-up table (LUT) in the PFU. Each latch/FF can accept data from the LUT. Alternately, the latches/FFs can accept direct data from wd[3:0], eliminating the LUT delay if no combinatorial function is needed.

The LUT outputs can bypass the latches/FFs, which reduces the delay out of the PFU. The pdilr[3:0] and pditb[3:0] inputs allow fast input from an I/O pad to the latches/FFs in the two closest PLCs perpendicular to the PIC containing the I/O pad. It is possible to use the LUT and latches/FFs more or less independently. For example, the latches/FFs can be used as a 4-bit shift register, and the LUT can be used to detect when a register has a particular pattern in it.

Programmable Logic Cells (continued)

PFU Control Inputs

The four control inputs to the PFU are clock (ck), local set/reset (lsr), clock enable (ce), and c0. The ck, ce, and lsr inputs control the operation of all four latches in the PFU. An active-low global set/reset (gsrn) signal is also available to the latches/FFs in every PFU. Their operation is discussed briefly here, and in more detail in the Latches/Flip-Flops section. The polarity of the control inputs can be inverted.

The ck input is distributed to each PFU from a vertical or horizontal net. The ce input inhibits the latches/FFs from responding to data inputs. The ce input can be disabled, always enabling the clock. Each latch/FF can be independently programmed to be a set or reset by the lsr and the global set/reset (gsrn) signals. Each PFU's lsr input can be configured as synchronous or asynchronous. The gsrn signal is always asynchronous. The lsr signal applies to all four latches/FFs in a PFU. The lsr input can be disabled (the default). The asynchronous set/reset is dominant over clocked inputs.

The c0 input is used as an input in combinatorial logic functions and as a carry input. It is used as an input into special PFU logic gates in wide input functions. The c0 input can be disabled (the default).

Look-Up Table Operating Modes

The LUT can be configured to operate in one of three general modes:

- Combinatorial logic mode
- Ripple mode
- Memory mode

The combinatorial logic mode uses a 64-bit look-up table (LUT) to implement Boolean functions. The two 5-bit logic inputs, a[4:0] and b[4:0], and the c0 input are used as LUT inputs. The use of these ports changes based on the PFU operating mode.

Table 2. Look-Up Table Operating Modes

Mode	Function
F4A	Two functions of four inputs, some inputs shared (QLUT2/QLUT3)
F4B	Two functions of four inputs, some inputs shared (QLUT0/QLUT1)
F5A	One function of five inputs (HLUTA)
F5B	One function of five inputs (HLUTB)
MA	16 x 2 memory (HLUTA)
MB	16 x 2 memory (HLUTB)
R	Ripple—LUT

For combinatorial logic, the LUT can be used to do any single function of six inputs, any two functions of five inputs, or four functions of four inputs (with some inputs shared), and three special functions based on the two five-input functions and c0.

The functionality of the LUT is determined by its operating mode. The entries in Table 2 show the basic modes of operation for combinatorial logic, ripple, and memory functions in the LUT. Depending on the operating mode, the LUT can be divided into sub-LUTs. The LUT is comprised of two 32-bit half look-up tables, HLUTA and HLUTB. Each half look-up table (HLUT) is comprised of two quarter look-up tables (QLUTs). HLUTA consists of QLUT2 and QLUT3, while HLUTB consists of QLUT0 and QLUT1. The outputs of QLUT0, QLUT1, QLUT2, and QLUT3 are f0, f1, f2, and f3, respectively.

If the LUT is configured to operate in the ripple mode, it cannot be used for basic combinatorial logic or memory functions. In modes other than the ripple mode, combinations of operating modes are possible. For example, the LUT can be configured as a 16 x 2 RAM in one HLUT and a five-input combinatorial logic function in the second HLUT. This can be done by configuring HLUTA in the MA mode and HLUTB in the F5B mode (or vice versa).

F4A/F4B Mode — Two Four-Input Functions

Each HLUT can be used to implement two four-input combinatorial functions, but the total number of inputs into each HLUT cannot exceed five. The two QLUTs within each HLUT share three inputs. In HLUTA, the a1, a2, and a3 inputs are shared by QLUT2 and QLUT3. Similarly, in HLUTB, the b1, b2, and b3 inputs are shared by QLUT0 and QLUT1. The four outputs are f0, f1, f2, and f3. The use of the LUT for four functions of up to four inputs each is given in Figure 4.

Programmable Logic Cells (continued)

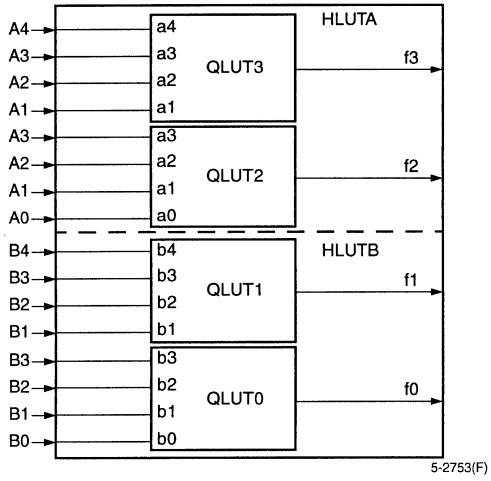


Figure 4. F4 Mode—Four Functions of Four Input Variables

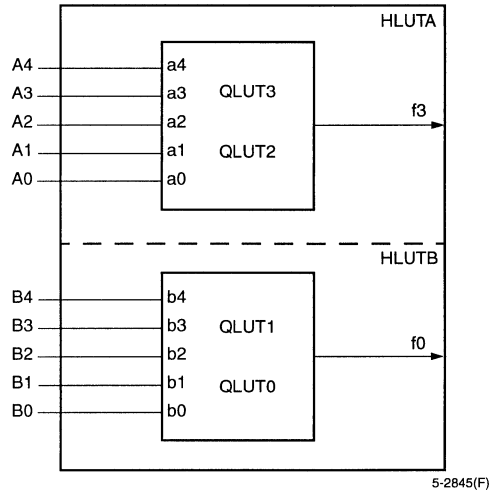


Figure 5. F5 Mode—Two Functions of Five Input Variables

F5A/F5B Mode—One Five-Input Variable Function

Each HLUT can be used to implement any five-input combinatorial function. The input ports are a[4:0] and b[4:0], and the output ports are f0 and f3. One five or less input function is input into a[4:0], and the second five or less input function is input into b[4:0]. The results are routed to the latch/FF d0 and latch/FF d3 inputs, or directly to the outputs o0 and o3. The use of the LUT for two independent functions of up to five inputs is given in Figure 5. In this case, the LUT is configured in the F5A and F5B modes. As a variation, the LUT can do one function of up to five input variables and two four-input functions using F5A and F4B modes or F4A and F5B modes.

F5M and F5X Modes — Special Function Modes

The PFU contains logic to implement two special function modes which are variations on the F5 mode. As with the F5 mode, the LUT implements two independent five-input functions. Figure 6 and Figure 7 show the schematics for F5M and F5X modes. The F5X and F5M functions differ from the basic F5A/F5B functions in that there are three logic gates which have inputs from the LUT. In some cases, this can be used for faster and/or wider logic functions. The HLUTs operate as in the F5 mode, providing outputs on f0 and f3. The resulting output is then input into a NAND and either a multiplexer in F5M mode or an exclusive OR in F5X mode.

As shown, two of the three inputs into the NAND, XOR, and MUX gates, f0 and f3, are from the LUT. The third input is from the c0 input into PFU. The output of the special function (either XOR or MUX) is f1. Since the XOR and multiplexer share the f1 output, the F5X and F5M modes are mutually exclusive. The output of the NAND is f2.

To use either the F5M or F5X functions, the LUT must be in the F5A/F5B mode. In both the F5X and F5M functions, the outputs of the five-input combinatorial functions, f0 and f3, are also usable simultaneously with the logic gate outputs.

The output of the multiplexer is:

$f1 = (HLUTA \times c0) + (HLUTB \times \overline{c0})$

$f1 = (f3 \times c0) + (f0 \times \overline{c0})$

The output of the exclusive OR is:

$f1 = HLUTA \oplus HLUTB \oplus c0$

$f1 = f3 \oplus f0 \oplus c0$

The output of the NAND is:

$f2 = \overline{HLUTA \times HLUTB \times c0}$

$f2 = \overline{f3 \times f0 \times c0}$

Programmable Logic Cells (continued)

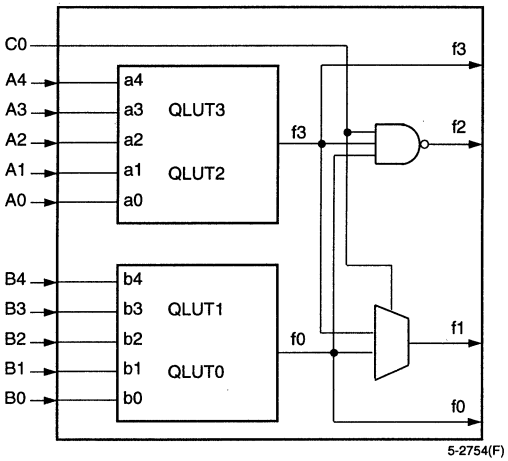


Figure 6. F5M Mode—Multiplexed Function of Two Independent Five-Input Variable Functions

F5M Mode — One Six-Input Variable Function

The LUT can be used to implement any function of six input variables. As shown in Figure 8, five input signals are routed into both the a[4:0] and b[4:0] ports, and the c0 port is used for the sixth input. The output port is f1.

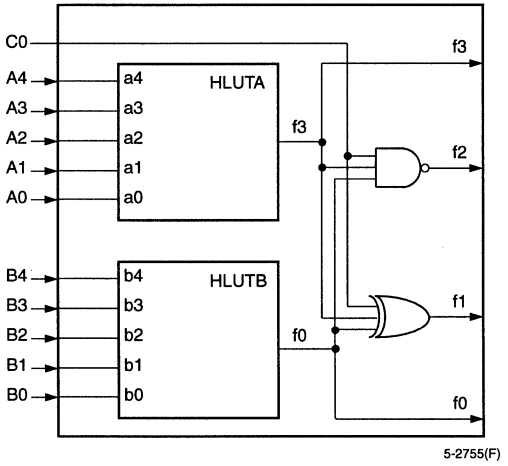


Figure 7. F5X Mode—Exclusive OR Function of Two Independent Five-Input Variable Functions

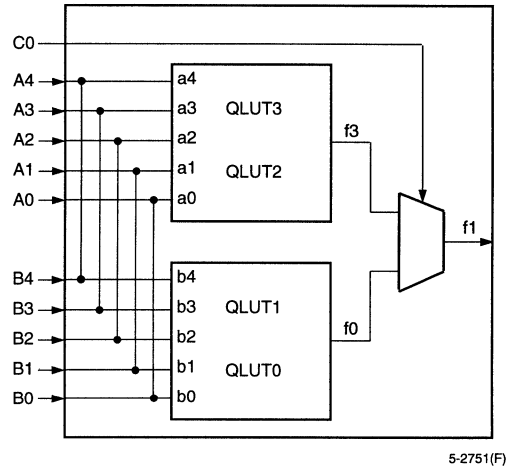


Figure 8. F5M Mode—One Six-Input Variable Function

Ripple Mode

The LUT can do nibble-wide ripple functions with high-speed carry logic. The QLUTs each have a dedicated carry-out net to route the carry to/from the adjacent QLUT. Using the internal carry circuits, fast arithmetic and counter functions can be implemented in one PFU. Similarly, each PFU has carry-in and carry-out ports for fast carry routing between adjacent PFUs.

The ripple mode is generally used in operations on two 4-bit buses. Each QLUT has two operands and a ripple input, and provides a result and ripple (generally carry) output. A single bit is rippled from the previous QLUT and is used as input into the current QLUT. For QLUT0, the ripple input is from the PFU cin port. The cin data can come from either the fast carry routing or the PFU input b4, or it can be tied to logic 1 or logic 0.

The result output and ripple output are calculated by using generate/propagate circuitry. In ripple mode, the two operands are input into a[3:0] and b[3:0]. The four results bits, one per QLUT, are f[3:0] (see Figure 9). The ripple output from QLUT3 can be routed to dedicated carry-out circuitry into any of four adjacent PLCs, or it can be placed on the o4 PFU output, or both. This allows for cascading PLCs in the ripple mode so that nibble-wide ripple functions can be easily expanded to any length. If an up/down counter or adder/subtracter is needed, the control signal is input on a4.

Programmable Logic Cells (continued)

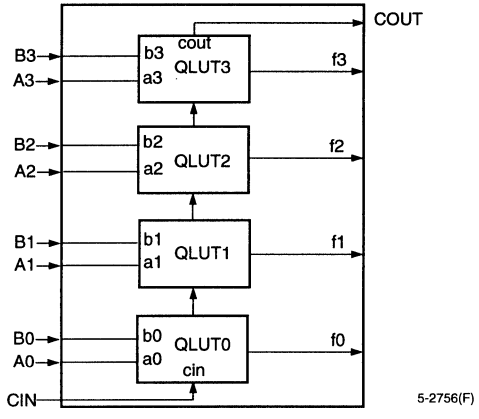


Figure 9. Ripple Mode

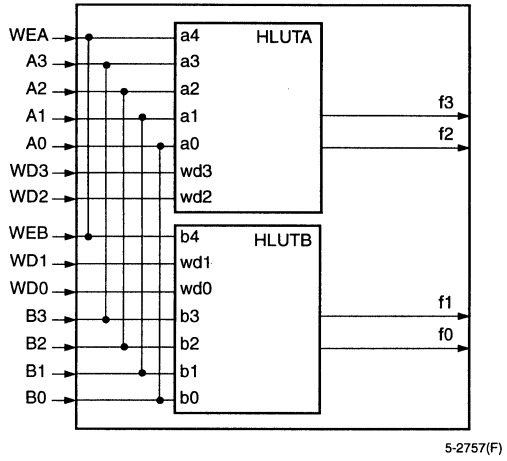


Figure 10. MA/MB Mode—16 x 4 RAM

Each QLUT generates two separate outputs. One of the two outputs selects whether the carry-in is to be propagated to the carry-out of the current QLUT or if the carry-out needs to be generated. The resulting output is placed on the QLUT output. The result bit is created in one half of the QLUT from a single bit from each input bus along with the ripple input bit. These inputs are also used to create the programmable propagate.

Memory Modes — MA and MB Modes

The LUT in the PFU can be configured as either read/write or read-only memory. A read/write address (a[3:0],b[3:0]), write data (wd[1:0], wd[3:2]), and two write enable (wea, web) ports are used for memory. In memory mode, each HLUT can be used as a 16 x 2 memory. Each HLUT is configured independently, allowing functions such as 16 x 4 memory or a 16 x 2 memory in one HLUT and a logic function of five input variables or less in the other HLUT.

Figure 10 illustrates the use of the LUT for a 16 x 4 memory. When the LUTs are used as memory, there are independent address, input data, and output data buses. If the LUT is used as a 16 x 4 read/write memory, the a[3:0] and b[3:0] ports are address inputs. The a4 and b4 ports are write-enable (we) signals. The wd[3:0] inputs are the data inputs. The f[3:0] data outputs can be routed out on the o[4:0] PFU outputs or to the latch/FFs d[3:0] inputs.

To increase memory address locations (e.g., 32 x 4), two or more PLCs can be used. The address and write data inputs for the two PLCs are tied together (bit by bit) and the data outputs are routed through a 3-statable BIDI and then tied together (bit by bit).

The write enable and read enable for each PLC is created from an extended address. The read enable is connected to the 3-state enable input to the BIDIs for a given PLC and then used to enable the 4 bits of data from a PLC onto the read data bus.

To increase the memory's word size (e.g., 16 x 8), two or more PLCs are used again. The address and write enable of the PLCs are tied together, and the data is different for each PLC. Increasing both the address locations and word size is done by using a combination of these two techniques.

The LUT can be used for both memory and a combinatorial logic function simultaneously. Figure 11 shows the use of a LUT implementing a 16 x 2 RAM (HLUTA) and any function of up to five input variables (HLUTB).

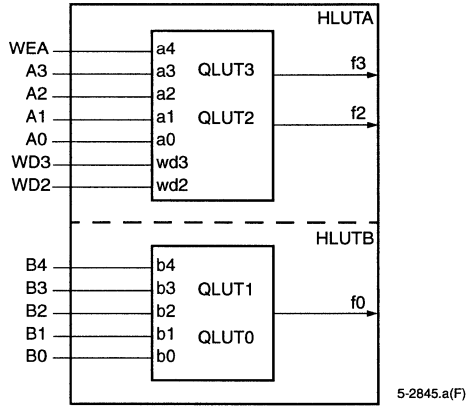


Figure 11. MA/F5 Mode—16 x 2 Memory and One Function of Five Input Variables

Programmable Logic Cells (continued)

Latches/Flip-Flops

The four latches/FFs in the PFU can be used in a variety of configurations. In some cases, the configuration options apply to all four latches/FFs in the PFU. For other options, each latch/FF is independently programmable.

Table 3 summarizes these latch/FF options. The latches/FFs can be configured as either positive or negative level sensitive latches, or positive or negative edge-triggered flip-flops. All latches/FFs in a given PFU share the same clock, and the clock to these latches/FFs can be inverted. The input into each latch/FF is from either the corresponding QLUT output (f[3:0]) or the direct data input (wd[3:0]). For latches/FFs located in the two outer rings of PLCs, additional inputs are possible. These additional inputs are fast paths from I/O pads located in PICs perpendicular to the PLCs. If the latch/FF is not located in the two outer rings of the PLCs, the latch/FF input can also be tied to logic 0, which is the default. The four latch/FF outputs, q[3:0], can be placed on the five PFU outputs, o[4:0].

Table 3. Configuration RAM Controlled Latch/Flip-Flop Operation

Function	Options
Functionality Common to All Latch/FFs in PFU	
LSR Operation	Asynchronous or Synchronous
Clock Polarity	Noninverted or Inverted
Front-End Select	Direct (wd[3:0]) or from LUT (f[3:0])
Functionality Set Individually in Each Latch/FF in PFU	
Latch/FF Mode	Latch or Flip-Flop
Set/Reset Mode	Set or Reset

The four latches/FFs in a PFU share the clock (ck), clock enable (ce), and local set/reset (lsr) inputs. When ce is disabled, each latch/FF retains its previous value when clocked, unless there is an asynchronous set/reset. Both the clock enable and lsr inputs can be inverted to be active-low.

The set/reset operation of the latch/FF is controlled by two parameters: reset mode and set/reset value. When the global (gsrn) or local set/reset (lsr) are active, the storage element operates normally as a latch or FF. The reset mode is used to select a synchronous or asynchronous lsr operation. If synchronous, lsr is enabled if clock enable (ce) is active. The clock enable is supported on FFs, not latches. The clock enable function is implemented by using a two-input multiplexer on the FF input, with one input being the previous state of the FF and the other input being the new data applied to the FF. The select of this two-input multiplexer is clock enable (ce), which selects either the new data or the previous state. When ce is inactive, the FF output does not change when the clock edge arrives.

The global reset (gsrn) is only asynchronous, and it sets/resets all latches/FFs in the FPGA based upon the set/reset configuration bit for each latch/FF. The set/reset value determines whether gsrn and lsr are set or reset inputs. The set/reset value is independent for each latch/FF.

If the local set/reset is not needed, the latch/FF can be configured to have a data front-end select. Two data inputs are possible in the front-end select mode, with the lsr signal used to select which data input is used. The data input into each latch/FF is from the output of its associated QLUT f[3:0] or direct from wd[3:0], bypassing the LUT. In the front-end data select mode, both signals are available to the latches/FFs.

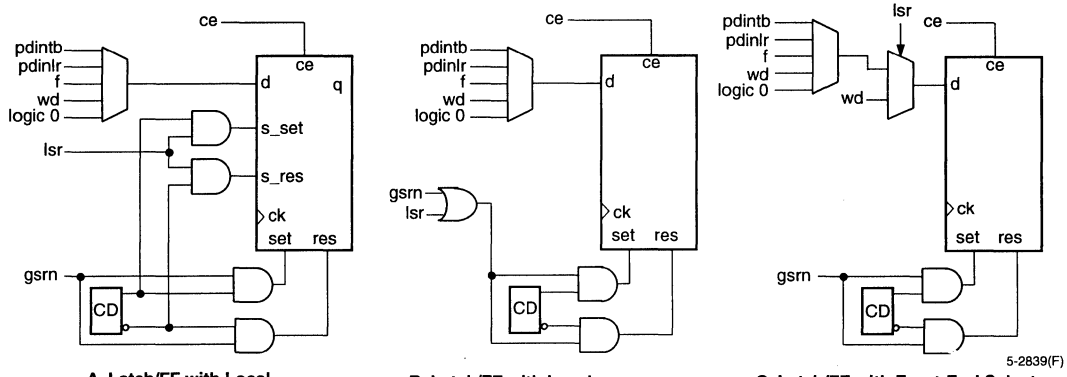
For PLCs that are in the two outside rows or columns of the array, the latch/FFs can have two inputs in addition to the f and wd inputs mentioned above. One input is from an I/O pad located at the PIC closest to either the left or right of the given PLC (if the PLC is in the left two columns or right two columns of the array). The other input is from an I/O pad located at the closest PIC either above or below the given PLC (if the PLC is in the top or the bottom two rows). It should be noted that both inputs are available for a 2 x 2 array of PLCs in each corner of the array. For the entire array of PLCs, if either or both of these inputs is unavailable, the latch/FF can be tied to a logic 0 instead.

To speed up the interface between signals external to the FPGA and the latches/FFs, there are direct paths from latch/FF outputs to the I/O pads. This is done for each PLC that is adjacent to a PIC. The latches/FFs can be configured in three modes:

1. Local synchronous set/reset: the input into the PFU's lsr port is used to synchronously set or reset each latch/FF.
2. Local asynchronous set/reset: the input into lsr asynchronously sets or resets each latch/FF.
3. Latch/FF with front-end select: the data select signal (actually lsr) selects the input into the latches/FFs between the LUT output and direct data in.

For all three modes, each latch/FF can be independently programmed as either set or reset. Each latch/FF in the PFU is independently configured to operate as either a latch or flip-flop. Figure 12 provides the logic functionality of the front-end select, global set/reset, and local set/reset operations.

Programmable Logic Cells (continued)



A. Latch/FF with Local Synchronous Set or Reset

B. Latch/FF with Local Asynchronous Set or Reset

C. Latch/FF with Front-End Select

Note:CD = configuration data.

Figure 12. Latch/FF Set/Reset Configurations

PLC Routing Resources

Generally, the *ORCA* Foundry Development System is used to automatically route interconnections. Interactive routing with the *ORCA* Foundry design editor (EPIC) is also available for design optimization. To use EPIC for interactive layout, an understanding of the routing resources is needed and is provided in this section.

The routing resources consist of switching circuitry and metal interconnect segments. Generally, the metal lines which carry the signals are designated as routing nodes (R-nodes). The switching circuitry connects the routing nodes, providing one or more of three basic functions: signal switching, amplification, and isolation. A net running from a PFU or PIC output (source) to a PLC or PIC input (destination) consists of one or more R-nodes, connected by switching circuitry designated as configurable interconnect points (CIPs).

The following sections discuss PLC, PIC, and interquad routing resources. This section discusses the PLC switching circuitry, intra-PLC routing, inter-PLC routing, and clock distribution.

Configurable Interconnect Points

The process of connecting R-nodes uses three basic types of switching circuits: two types of configurable interconnect points (CIPs) and bidirectional buffers (BIDIs). The basic element in CIPs is one or more pass transistors, each controlled by a configuration RAM bit. The two types of CIPs are the mutually exclusive (or multiplexed) CIP and the independent CIP.

A mutually exclusive set of CIPs contains two or more CIPs, only one of which can be on at a time. An independent CIP has no such restrictions and can be on independent of the state of other CIPs. Figure 13 shows an example of both types of CIPs.

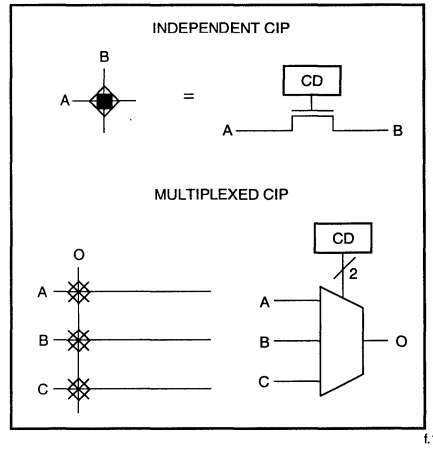


Figure 13. Configurable Interconnect Point

Programmable Logic Cells (continued)

3-Statable Bidirectional Buffers

Bidirectional buffers provide isolation as well as amplification for signals routed a long distance. Bidirectional buffers are also used to drive signals directly onto either vertical or horizontal xL and xH R-nodes (to be described later in the inter-PLC routing section). BIDIs are also used to indirectly route signals through the switching R-nodes. Any number from zero to eight BIDIs can be used in a given PLC.

The BIDIs in a PLC are divided into two nibble-wide sets of four (BIDI and BIDIH). Each of these sets has a separate BIDI controller that can have an application net connected to its TRI input, which is used to 3-state enable the BIDIs. Although only one application net can be connected to both BIDI controllers, the sense of this signal (active-high, active-low, or ignored) can be configured independently. Therefore, one set can be used for driving signals, the other set can be used to create 3-state buses, both sets can be used for 3-state buses, and so forth.

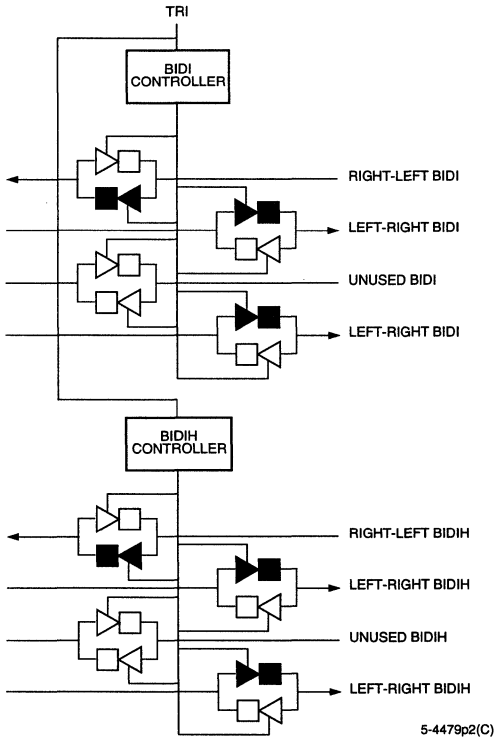


Figure 14. 3-Statable Bidirectional Buffers

Intra-PLC Routing

The function of the intra-PLC routing resources is to connect the PFU's input and output ports to the routing resources used for entry to and exit from the PLC. These are nets for providing PFU feedback, turning corners, or switching from one type of routing resource to another.

PFU Input and Output Ports. There are nineteen input ports to each PFU. The PFU input ports are labelled a[4:0], b[4:0], wd[3:0], c0, ck, lsr, cin, and ce. The six output ports are o[4:0] and cout. These ports correspond to those described in the PFU section.

Switching R-Nodes. There are four sets of switching R-nodes in each PLC, one in each corner. Each set consists of five switching elements, labelled sul[4:0], sur[4:0], sll[4:0], and slr[4:0], for the upper-left, upper-right, lower-left, and lower-right sections of the PFUs, respectively. The switching R-nodes connect to the PFU inputs and outputs as well as the BIDI and BIDIH R-nodes, to be described later. They also connect to both the horizontal and vertical x1 and x4 R-nodes (inter-PLC routing resources, described below) in their specific corner.

One of the four sets of switching R-nodes can be connected to a set of switching R-nodes in each of the four adjacent PLCs or PICs. This allows direct routing of up to five signals without using inter-PLC routing.

BIDI/BIDIH R-Nodes. There are two sets of bidirectional R-nodes in the PLC, each set consisting of four bidirectional buffers. They are designated BIDI and BIDIH and have similar functionality. The BIDI R-nodes are used in conjunction with the xL R-nodes, and the BIDIH R-nodes are used in conjunction with the xH R-nodes. Each side of the four BIDIs in the PLC is connected to a BIDI R-node on the left (BL[3:0]) and on the right (BR[3:0]). These R-nodes can be connected to the xL R-nodes through CIPs, with BL[3:0] connected to the vertical xL R-nodes and BR[3:0] connected to the horizontal xL R-nodes. Both BL[3:0] and BR[3:0] have CIPs which connect to the switching R-nodes.

Similarly, each side of the four BIDIHs is connected to a BIDIH R-node: BLH[3:0] on the left and BRH[3:0] on the right. These R-nodes can also be connected to the xH R-nodes through CIPs, with BLH[3:0] connected to the vertical xH R-nodes and BRH[3:0] connected to the horizontal xH R-nodes. Both BLH[3:0] and BRH[3:0] have CIPs which connect to the switching R-nodes.

CIPs are also provided to connect the BIDIH and BIDIL R-nodes together on each side of the BIDIs. For example, BLH3 can connect to BL3, while BRH3 can connect to BR3.

Programmable Logic Cells (continued)

Inter-PLC Routing Resources

The inter-PLC routing is used to route signals between PLCs. The R-nodes occur in groups of four, and differ in the numbers of PLCs spanned. The x1 R-nodes span one PLC, the x4 R-nodes span four PLCs, the xH R-nodes span one-half the width (height) of the PLC array, and the xL R-nodes span the width (height) of the PLC array. All types of R-nodes run in both horizontal and vertical directions.
Table 4 shows the groups of inter-PLC R-nodes in each PLC. In the table, there are two rows/columns each for x1 and x4 lines. In the design editor, the horizontal x1 and x4 R-nodes are located above and below the PFU. Similarly, the vertical segments are located on each side. The xL and xH R-nodes only run below and to the left of the PFU. The indexes specify individual R-nodes within a group. For example, the vx4[2] R-node runs vertically to the left of the PFU, spans four PLCs, and is the third line in the 4-bit wide bus.

Table 4. Inter-PLC Routing Resources

Horizontal R-Nodes	Vertical R-Nodes	Distance Spanned
hx1[3:0]	vx1[3:0]	One PLC
hx1[7:4]	vx1[7:4]	One PLC
hx4[3:0]	vx4[3:0]	Four PLCs
hx4[7:4]	vx4[7:4]	Four PLCs
hxL[3:0]	vxL[3:0]	PLC Array
hxH[3:0]	vxH[3:0]	1/2 PLC Array
ckl, ckr	ckt, ckb	PLC Array

Figure 15 shows the inter-PLC routing within one PLC. Figure 16 provides a global view of inter-PLC routing resources across multiple PLCs.

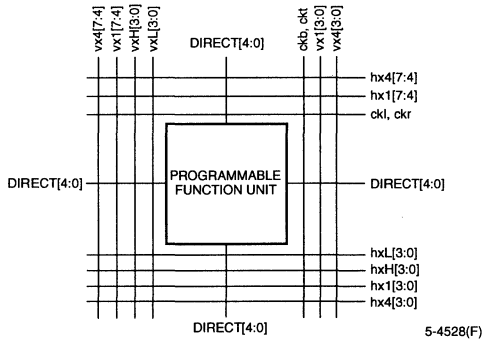


Figure 15. Single PLC View of Inter-PLC R-Nodes

x1 R-Nodes. There are a total of 16 x1 R-nodes per PLC: eight vertical and eight horizontal. Each of these is subdivided into nibble-wide buses: hx1[3:0], hx1[7:4], vx1[3:0], and vx1[7:4]. An x1 line is one PLC long. If a net is longer than one PLC, an x1 R-node can be lengthened to n times its length by turning on n – 1 CIPs. A signal is routed onto an x1 R-node via the switching R-nodes.

x4 R-Nodes. There are four sets of four x4 R-nodes, for a total of 16 x4 R-nodes per PLC. They are hx4[3:0], hx4[7:4], vx4[3:0], and vx4[7:4]. Each set of x4 R-nodes is twisted each time it passes through a PLC, and one of the four is broken with a CIP. This allows a signal to be routed for a length of four cells in any direction on a single line without additional CIPs. The x4 R-nodes can be used to route any nets that require minimum delay. A longer net is routed by connecting two x4 R-nodes together by a CIP. The x4 R-nodes are accessed via the switching R-nodes.

xL R-Nodes. The long xL R-nodes run vertically and horizontally the height and width of the array, respectively. There are a total of eight xL R-nodes per PLC: four horizontal (hxL[3:0]) and four vertical (vxL[3:0]). Each PLC column has four xL lines, and each PLC row has four xL R-nodes. Each of the xL R-nodes connects to the two PICs at either end. The ATT2C12, which consists of a 18 x 18 array of PLCs, contains 72 vxL and 72 hxL R-nodes. They are intended primarily for global signals which must travel long distances and require minimum delay and/or skew, such as clocks.

There are three methods for routing signals onto the xL R-nodes. In each PLC, there are two long line drivers: one for a horizontal xL R-node, and one for a vertical xL R-node. Using the long line drivers produces the least delay. The xL R-nodes can also be driven directly by PFU outputs using the BIDI R-nodes. In the third method, the xL R-nodes are accessed by the bidirectional buffers, again using the BIDI R-nodes.

Programmable Logic Cells (continued)

xH R-nodes. Four by half (xH) R-nodes run horizontally and four xH R-nodes run vertically in each row and column in the array. These R-nodes travel a distance of one-half the PLC array before being broken in the middle of the array, where they connect to the interquad block (discussed later). They also connect at the periphery of the FPGA to the PICs, like the xL R-nodes. The xH R-nodes do not twist like xL R-nodes, allowing nibble-wide buses to be routed easily.

Two of the three methods of routing signals onto the xL R-nodes can also be used for the xH R-nodes. A special xH line driver is not supplied for the xH R-nodes.

Clock R-Nodes. For a very fast and low-skew clock (or other global signal tree), clock R-nodes run the entire height and width of the PLC array. There are two horizontal clock R-nodes per PLC row (CKL, CKR) and two vertical clock R-nodes per PLC column (CKT, CKB). The source for these clock R-nodes can be any of the four I/O buffers in the PIC. The horizontal clock R-nodes in a row (CKL and CKR) are driven by the left and right PICs, respectively. The vertical clock R-nodes in a column (CKT, CKB) are driven by the top and bottom PICs, respectively.

The clock R-nodes are designed to be a clock spine. In each PLC, there is a fast connection available from the clock R-node to the long-line driver (described earlier). With this connection, one of the clock R-nodes in each PLC can be used to drive one of the four xL R-nodes perpendicular to it, which, in turn, creates a clock tree. This feature is discussed in detail in the clock distribution section.

Minimizing Routing Delay

The CIP is an active element used to connect two R-nodes. As an active element, it adds significantly to the resistance and capacitance of a net, thus increasing the net's delay. The advantage of the x1 R-node over a x4 R-node is routing flexibility. A net from PLC db to PLC cb is easily routed by using x1 R-nodes. As more CIPs are added to a net, the delay increases. To increase speed, routes that are greater than two PLCs away are routed on the x4 R-nodes because a CIP is located only in every fourth PLC. A net which spans eight PLCs requires seven x1 R-nodes and six CIPs. Using x4 R-nodes, the same net uses two R-nodes and one CIP.

All routing resources in the PLC can carry 4-bit buses. In order for data to be used at a destination PLC that is in data path mode, the data must arrive unscrambled. For example, in data path operation, the least significant bit 0 must arrive at either a[0] or b[0]. If the bus is to be routed by using either x4 or xL R-nodes (both of which twist as they propagate), the bus must be placed on the appropriate lines at the source PLC so that the data arrives at the destination unscrambled. The switching R-nodes provide the most efficient means of connecting adjacent PLCs. Signals routed with these R-nodes have minimum propagation delay.

Programmable Logic Cells (continued)

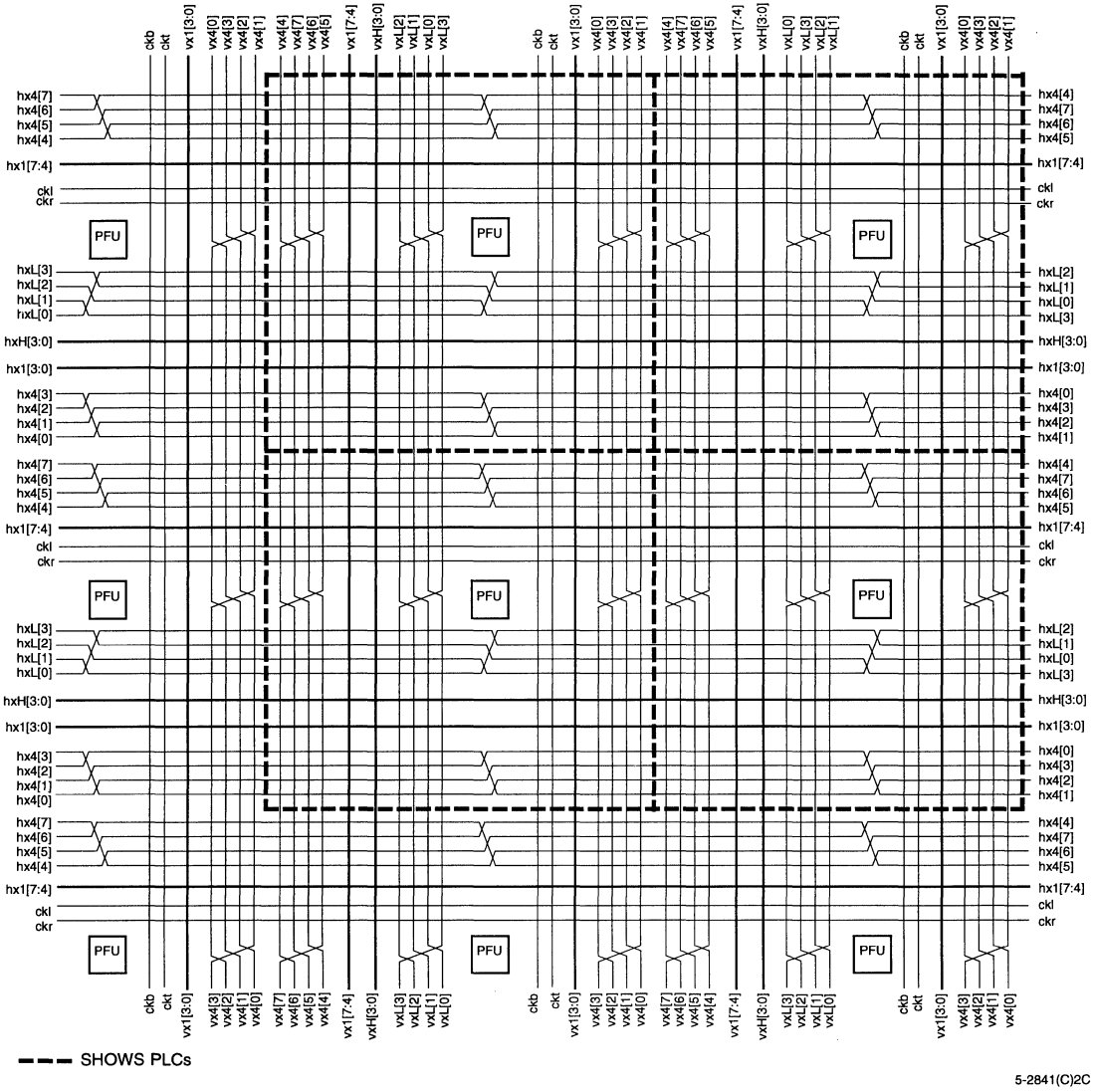


Figure 16. Multiple PLC View of Inter-PLC Routing

Programmable Logic Cells (continued)

PLC Architectural Description

Figure 17 is an architectural drawing of the PLC which reflects the PFU, the R-nodes, and the CIPs. A discussion of each of the letters in the drawing follows.

A. These are switching R-nodes which give the router flexibility. In general switching theory, the more levels of indirection there are in the routing, the more routable the network is. The switching R-nodes can also connect to adjacent PLCs.

The switching R-nodes provide direct connections to PLCs directly to the top, bottom, left, and right, without using other routing resources. The ability to disable this connection between PLCs is provided so that each side of these connections can be used exclusively as switching R-nodes in their respective PLC.

B. These CIPs connect the x1 routing. These are located in the middle of the PLC to allow the block to connect to either the left end of the horizontal x1 R-node from the right or the right end of the horizontal x1 R-node from the left, or both. By symmetry, the same principle is used in the vertical direction. The x1 lines are not twisted, making them suitable for data paths.

C. This set of CIPs is used to connect the x1 and x4 nets to the switching R-nodes or to other x1 and x4 nets. The CIPs on the major diagonal allow data to be transmitted from x1 nets to the switching R-nodes without being scrambled. The CIPs on the major diagonal also allow unscrambled data to be passed between the x1 and x4 nets.

In addition to the major diagonal CIPs for the x1 lines, other CIPs provide an alternative entry path into the PLC in case the first one is already used. The other CIPs are arrayed in two patterns, as shown. Both of these patterns start with the main diagonal, but the extra CIPs are arrayed on either a parallel diagonal shifted by one or shifted by two (modulo the size of the vertical bus (5)). This allows any four application nets incident to the PLC corner to be transferred to the five switching R-nodes in that corner. Many patterns of five nets can also be transferred.

D. The x4 R-nodes are twisted at each PLC. One of the four x4 lines is broken with a CIP, which allows a signal to be routed a distance of four PLCs in any direction on a single R-node without an intermediate CIP. The x4 R-nodes are less populated with CIPs than the x1 lines to increase their speed. A CIP can be enabled to extend an x4 R-node four more PLCs, and so on.

For example, if an application signal is routed onto hx4[4] in a PLC, it appears on hx4[5] in the PLC to the right. This signal step-up continues until it reaches hx4[7], two PLCs later. At this point, the user can break the connection or continue the signal for another four PLCs.

E. These symbols are bidirectional buffers (BIDIs). There are four BIDIs per PLC, and they provide signal amplification as needed to decrease signal delay. The BIDIs are also used to transmit signals on xL lines.

F. These are the BIDI and BIDIH controllers. The 3-state control signal can be disabled. They can be configured as active-high or active-low independently of each other.

G. This set of CIPs allows a BIDI to get or put a signal from one set of switching R-nodes on each side. The BIDIs can be accessed by the switching R-nodes. These CIPs allow a nibble of data to be routed though the BIDIs and continue to a subsequent block. They also provide an alternative routing resource to improve routability.

H. These CIPs are used to take data from/to the BIDIs to/from the xL R-nodes. These CIPs have been optimized to allow the BIDI buffers to drive the large load usually seen when using xL R-nodes.

I. Each latch/FF can accept data: from a LUT output; from a direct data input signal from general routing; or, as in the case of PLCs located in the two rows (columns) adjacent to PICs, directly from the pad. In addition, the LUT outputs can bypass the latches/FFs completely and output data on the general routing resources. The four inputs shown are used as the direct input to the latches/FFs from general routing resources. If the LUT is in memory mode, the four inputs wd[3:0] are the data input to the memory.

Programmable Logic Cells (continued)

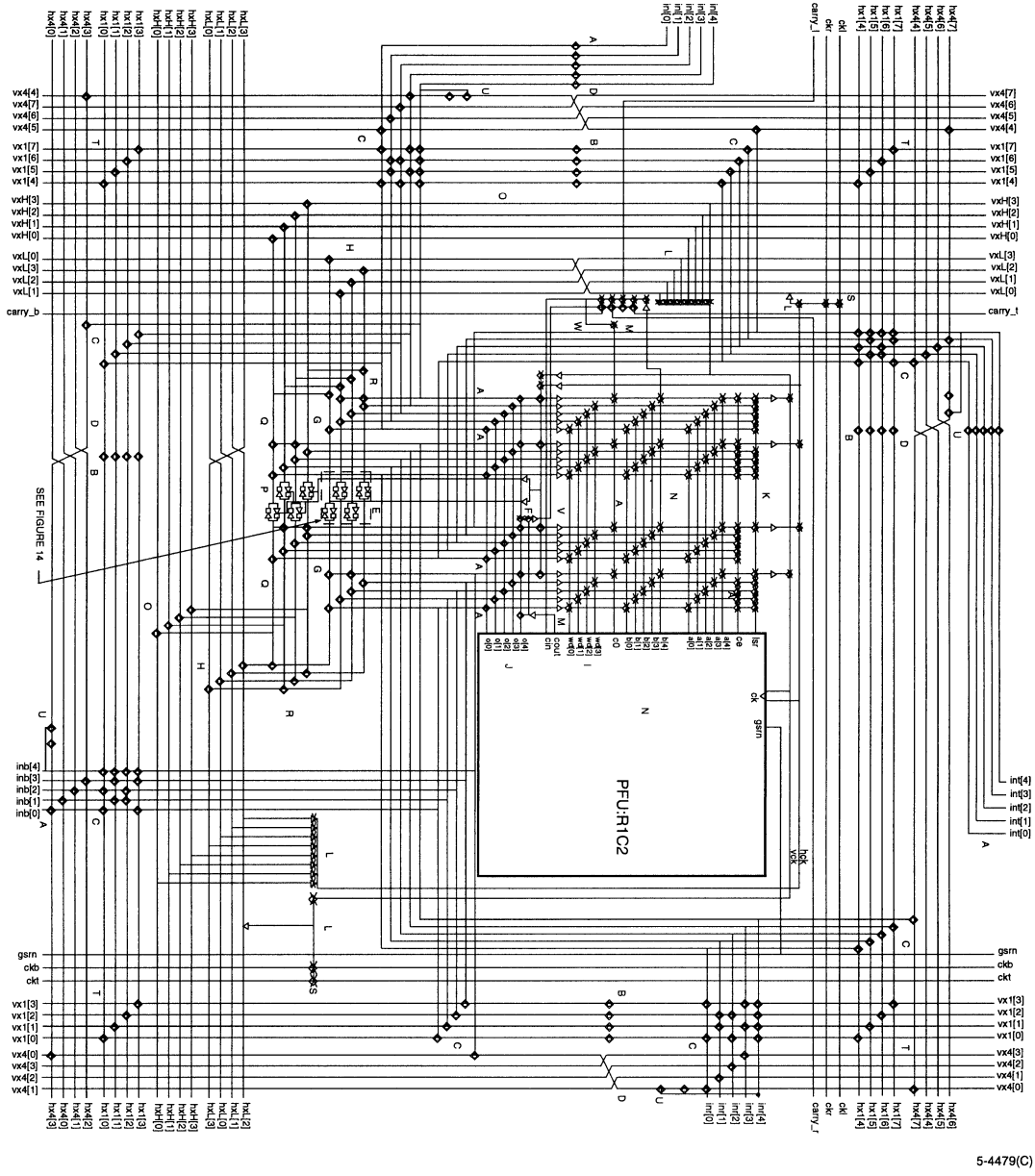


Figure 17. PLC Architecture

Programmable Logic Cells (continued)

J. Any five of the eight output signals can be routed out of the PLC. The eight signals are the four LUT outputs (f0, f1, f2, and f3) and the four latch/FF outputs (q0, q1, q2, and q3). This allows the user to access all four latch/FF outputs, read the present state and next state of a latch/FF, build a 4-bit shift register, etc. Each of the outputs can drive any number of the five PFU outputs. The speed of a signal can be increased by dividing its load among multiple PFU output drivers.

K. These lines deliver the auxiliary signals clock enable and set/reset to the latches/FFs. All four of the latches/FFs share these signals.

L. This is the clock input to the latches/FFs. Any of the horizontal and vertical xH or xL lines can drive the clock of the PLC latches/FFs. Long line drivers are provided so that a PLC can drive one xL R-node in the horizontal direction and one xL R-node in the vertical direction. The xL lines in each direction exhibit the same properties as x4 lines, except there are no CIPs. The clock R-nodes (ckl, ckr, ckt, and ckb) and multiplexers/drivers are used to connect to the xL R-nodes for low-skew, low-delay global signals.

The long lines run the length or width of the PLC array. They rotate to allow four PLCs in one row or column to generate four independent global signals. These lines do not have to be used for clock routing. Any highly used application net can use this resource, especially one requiring low skew.

M. These R-nodes are used to route the fast carry signal to/from the neighboring four PLCs. The carry-out (cout) of the PFU can also be routed out of the PFU onto the fifth output (o4). The carry-in (cin) signal can also be supplied by the b4 input to the PFU.

N. These are the 11 logic inputs to the LUT. The a[4:0] inputs are provided into HLUTA, and the b[4:0] inputs are provided into HLUTB. The c0 input bypasses the main LUT and is used in the pfumux, pfuxor, and pfunand functions (F5M, F5X modes). Since this input bypasses the LUT, it can be used as a fast path around the LUT, allowing the implementation of fast, wide combinatorial functions. The c0 input can be disabled or inverted.

O. The xH R-nodes run one-half the length (width) of the array before being broken by a CIP.

P. The BIDIHs are used to access the xH R-nodes.

Q. The BIDIH R-nodes are used to connect the BIDIHs to the xsw R-nodes, the xH R-nodes, or the BIDI R-nodes.

R. These CIPs connect the BIDI R-nodes and the BIDIH R-nodes.

S. These are clock R-nodes (ckt, ckb, ckl, and ckr) with the multiplexers and drivers to connect to the xL R-nodes.

T. These CIPs connect x1 R-nodes which cross in each corner to allow turns on the x1 R-nodes without using the xsw R-nodes.

U. These CIPs connect x4 R-nodes and xsw R-nodes, allowing nets that run a distance that is not divisible by four to be routed more efficiently.

V. This routing structure allows any PFU output, including LUT and latch/FF outputs, to be placed on o4 and be routed onto the fast carry routing.

W. This routing structure allows the fast carry routing to be routed onto the c0 PFU input.

Programmable Input/Output Cells

The programmable input/output cells (PICs) are located along the perimeter of the device. Each PIC interfaces to four bond pads and contains the necessary routing resources to provide an interface between I/O pads and the PLCs. Each PIC is composed of input buffers, output buffers, and routing resources as described below. Table 5 provides an overview of the programmable functions in an I/O cell. Figure 18 is a simplified diagram of the functionality of the *ORCA* series I/O cells.

Table 5. Input/Output Cell Options

Input	Option
Input Levels	TTL/CMOS
Input Speed	Fast/Delayed
Float Value	Pull-up/Pull-down/None
Direct-in to FF	Fast/Delayed
Output	**Option**
Output Drive	12 mA/6 mA or 6 mA/3 mA
Output Speed	Fast/Slewlim/Sinklim
Output Source	FF Direct-out/General Routing
Output Sense	Active-high/-low
3-State Sense	Active-high/-low (3-state)

Inputs

Each I/O can be configured to be either an input, an output, or bidirectional I/O. Inputs can be configured as either TTL or CMOS compatible. To allow zero hold time on PLC latches/FFs, the input signal can be delayed. Pull-up or pull-down resistors are available on inputs to minimize power consumption.

A fast path from the input buffer to the clock R-nodes is also provided. Any one of the four I/O pads on any PIC can be used to drive the clock R-node generated in that PIC.

To reduce the time required to input a signal into the FPGA, a dedicated path (pdin) from the I/O pads to the PFU flip-flops is provided. Like general input signals, this signal can be configured as normal or delayed. The delayed direct input can be selected independently from the delayed general input. If the fast clock routing is selected from a given I/O pad, then the direct input signal is automatically delayed, decreasing the delay of the fast clock.

Inputs should have transition times of less than 500 ns and should not be left floating. If an input can float, a pull-up or pull-down should be enabled. Floating inputs increase power consumption, produce oscillations, and increase system noise. The inputs have a typical hysteresis of approximately 280 mV to reduce sensitivity to input noise. The PIC contains input circuitry which provides protection against latch-up and electrostatic discharge.

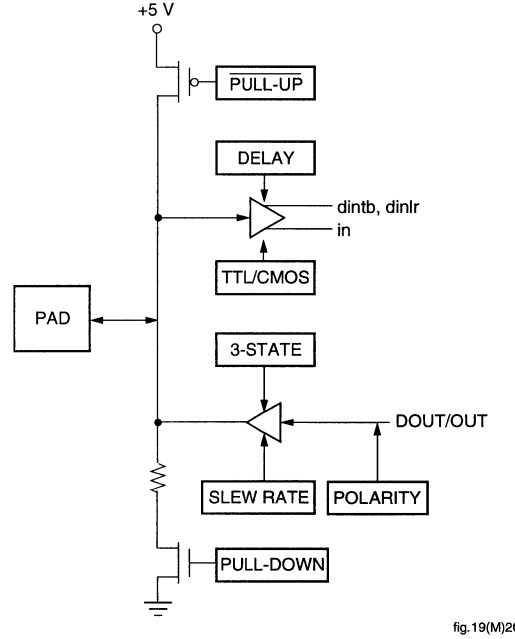


Figure 18. Simplified Diagram of Programmable I/O Cell

Programmable Input/Output Cells

(continued)

Outputs

The PIC's output drivers have programmable drive capability and slew rates. Three propagation delays (fast, slewlim, sinklim) are available on output drivers. The sinklim mode has the longest propagation delay and is used to minimize system noise and minimize power consumption. The fast and slewlim modes allow critical timing to be met.

The drive current is 12 mA sink/6 mA source for the slewlim and fast output speed selections and 6 mA sink/3 mA source for the sinklim output. Two adjacent outputs can be interconnected to increase the output sink current to 24 mA.

All outputs that are not speed critical should be configured as sinklim to minimize power and noise. The number of outputs that switch simultaneously in the same direction should be limited to minimize ground bounce. To minimize ground bounce problems, locate heavily loaded output buffers near the ground pads. Ground bounce is generally a function of the driving circuits, traces on the PCB, and loads and is best determined with a circuit simulation.

Outputs can be inverted, and 3-state control signals can be active-high or active-low. An open-drain output may be obtained by using the same signal for driving the output and 3-state signal nets so that the buffer output is enabled only by a low. At powerup, the output drivers are in slewlim mode, and the input buffers are configured as TTL-level compatible with a pull-up. If an output is not to be driven in the selected configuration mode, it is 3-stated.

Global 3-State Functionality

To increase the testability of the *ORCA* Series FPGAs, the global 3-state function (ts_all) disables the device. The ts_all signal is driven from either an external pin or an internal signal. Before and during configuration, the ts_all signal is driven by the input pad $\overline{\text{RD\_CFGN}}$. After configuration, the ts_all signal can be disabled, driven from the $\overline{\text{RD\_CFGN}}$ input pad, or driven by a general routing signal in the upper-right corner. Before configuration, ts_all is active-low; after configuration, the sense of ts_all can be inverted. The following occur when ts_all is activated:

1. All of the user I/O output buffers are 3-stated, the user I/O input buffers are pulled up (with the pull-down disabled), and the input buffers are configured with TTL input thresholds.
2. The TDO/RD_DATA output buffer is 3-stated.
3. The $\overline{\text{RD\_CFGN}}$, $\overline{\text{RESET}}$, and $\overline{\text{PRGM}}$ input buffers remain active with a pull-up.
4. The DONE output buffer is 3-stated and the input buffer is pulled-up.

PIC Routing Resources

The PIC routing is designed to route 4-bit wide buses efficiently. For example, any four consecutive I/O pads can have both their input and output signals routed into one PLC. Using only PIC routing, either the input or output data can be routed to/from a single PLC from/to any eight pads in a row.

The connections between PLCs and the I/O pad are provided by two basic types of routing resources. These are routing resources internal to the PIC and routing resources used for PIC-PLC connection. Figure 19 and Figure 20 show a high-level and detailed view of these routing resources.

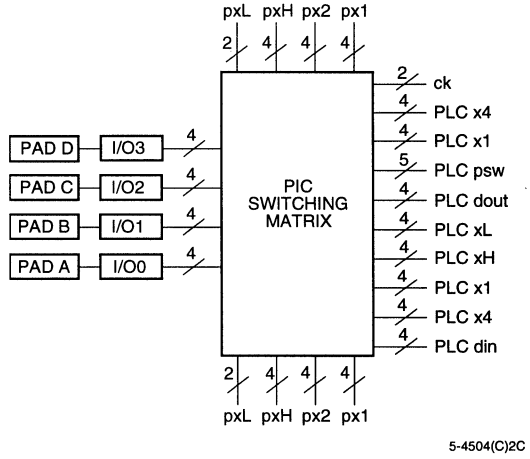


Figure 19. Simplified PIC Routing Diagram

Programmable Input/Output Cells

(continued)

The PIC's name is represented by a two-letter designation to indicate on which side of the device it is located followed by a number to indicate in which row or column it is located. The first letter, P, designates that the cell is a PIC and not a PLC. The second letter indicates the side of the array where the PIC is located. The four sides are left (L), right (R), top (T), and bottom (B). The individual I/O pad is indicated by a single letter (either A, B, C, or D) placed at the end of the PIC name. As an example, PL10A indicates a pad located on the left side of the array in the tenth row.

Each PIC has four pads and each pad can be configured as an input, an output (3-statable), a direct output, or a bidirectional I/O. When the pads are used as inputs, the external signals are provided to the internal circuitry at in[3:0]. When the pads are used to provide direct inputs to the latches/FFs, they are connected through din[3:0]. When the pads are used as outputs, the internal signals connect to the pads through out[3:0]. When the pads are used as direct outputs, the output from the latches/flip-flops in the PLCs to the PIC is designated dout[3:0]. When the outputs are 3-statable, the 3-state enable signals are ts[3:0].

Routing Resources Internal to the PIC

For inter-PIC routing, the PIC contains fourteen R-nodes used to route signals around the perimeter of the FPGA. Figure 19 shows these lines running vertically for a PIC located on the left side. Figure 20 shows the R-nodes running horizontally for a PIC located at the top of the FPGA.

pxL R-Nodes. Each PIC has two pxL R-nodes, labelled pxL[1:0]. Like the xL R-nodes of the PLC, the pxL R-nodes span the entire edge of the FPGA.

pxH R-Nodes. Each PIC has four pxH R-nodes, labelled pxH[3:0]. Like the xH R-nodes of the PLC, the pxH R-nodes span 1/2 the edge of the FPGA.

px2 R-Nodes. There are four px2 R-nodes in each PIC, labelled px2[3:0]. The px2 R-nodes pass through two adjacent PICs before being broken. These are used to route nets around the perimeter a distance of two or more PICs.

px1 R-Nodes. Each PIC has four px1 R-nodes, labelled px1[3:0]. The px1 R-nodes are one PIC long and are extended to adjacent PICs by enabling CIPs.

Programmable Input/Output Cells

(continued)

PIC Architectural Description

The PIC architecture given in Figure 20 is described using the following letter references. The figure depicts a PIC at the top of the array, so inter-PIC routing is horizontal and the indirect PIC-PLC routing is horizontal to vertical. In some cases, letters are provided in more than one location to indicate the path of an R-node.

A. As in the PLCs, the PIC contains a set of R-nodes which run the length (width) of the array. The pxL R-nodes connect in the corners of the array to other pxL R-nodes. The pxL R-nodes also connect to the PIC BIDI, PIC BIDIH, and LLDRV R-nodes. As in the PLC xL R-nodes, the pXH R-nodes twist as they propagate through the PICs.

B. As in the PLCs, the PIC contains a set of R-nodes which run one-half the length (width) of the array. The pxH R-nodes connect in the corners and in the middle of the array perimeter to other pxH R-nodes. The pxH R-nodes also connect to the PIC BIDI, PIC BIDIH, and LLDRV R-nodes. As in the PLC xH R-nodes, the pxH R-nodes do not twist as they propagate through the PICs.

C. The px2[3:0] R-nodes span a length of two PICs before intersecting with a CIP. The CIP allows the length of a path using px2 R-nodes to be extended two PICs.

D. The px1[3:0] R-nodes span a single PIC before intersecting with a CIP. The CIP allows the length of a path using px1 R-nodes to be extended by one PIC.

E. These are four dedicated direct output R-nodes connected to the output buffers. The dout[3:0] signals go directly from a PLC latch/FF to an output buffer, minimizing the latch/FF to pad propagation delay.

F. This is a direct path from the input pad to the PLC latch/flip-flops in the two rows (columns) adjacent to PICs. This input allows a reduced setup time. Direct inputs from the top and bottom PIC rows are pdintb[3:0]. Direct inputs from the left and right PIC columns are pdinlr[3:0].

G. The out[3:0], ts[3:0], and in[3:0] signals for each I/O pad can be routed directly to the adjacent PLC's switching R-nodes.

H. The four TRIDI buffers allow connections from the pads to the PLC xL R-nodes. The TRIDIs also allow connections between the PLC xL R-nodes and the pBIDI R-nodes, which are described in **J** below.

I. The four TRIDIH buffers allow connections from the pads to the PLC xH R-nodes. The TRIDIHs also allow connections between the PLC xH R-nodes and the pBIDIH R-nodes, which are described in **K** below.

J. The pBIDI R-nodes (bidi[3:0]) connect the pxL R-nodes, pxH R-nodes, and the px1 R-nodes. These are bidirectional in that the path can be from the pxL, pxH, or px1 R-nodes to the xL R-nodes, or from the xL R-nodes to the pxL, pxH, or px1 R-nodes.

K. The pBIDIH R-nodes (bidih[3:0]) connect the pxL R-nodes, pxH R-nodes, and the px1 R-nodes. These are bidirectional in that the path can be from the pxL, pxH, or px1 R-nodes to the xH R-nodes, or from the xH R-nodes to the pxL, pxH, or px1 R-nodes.

L. The llin[3:0] R-nodes provide a fast connection from the I/O pads to the xL and xH R-nodes.

M. This set of CIPs allows the eight x1 R-nodes (four on each side) of the PLC perpendicular to the PIC to be connected to either the px1 or px2 R-nodes in the PIC.

N. This set of CIPs allows the eight x4 R-nodes (four on each side) of the PLC perpendicular to the PIC to be connected to the px1 R-nodes. This allows fast access to/from the I/O pads from/to the PLCs.

O. All four of the PLC x4 R-nodes in a group connect to all four of the PLC x4 R-nodes in the adjacent PLC through a CIP. (This differs from the Lucent *ORCA* 1C Series in which two of the x4 R-nodes in adjacent PLCs are directly connected without any CIPs.)

P. The long line driver (LLDRV) R-node can be driven by the xsw4 switching R-node of the adjacent PLC. To provide connectivity to the pads, the LLDRV R-node can also connect to any of the four pxH or to one of the pxL R-nodes. The 3-state enable (ts[i]) for all four I/O pads can be driven by xsw4, pxH, or pxL R-nodes.

Q. For fast clock routing, one of the four I/O pads in each PIC can be selected to be driven onto a dedicated clock R-node. The clock R-node spans the length (width) of the PLC array. This dedicated clock R-node is typically used as a clock spine. In the PLCs, the spine is connected to an xL R-node to provide a clock branch in the perpendicular direction. Since there is another clock R-node in the PIC on the opposite side of the array, only one of the I/O pads in a given row (column) can be used to generate a global signal in this manner, if all PLCs are driven by the signal.

Programmable Input/Output Cells (continued)

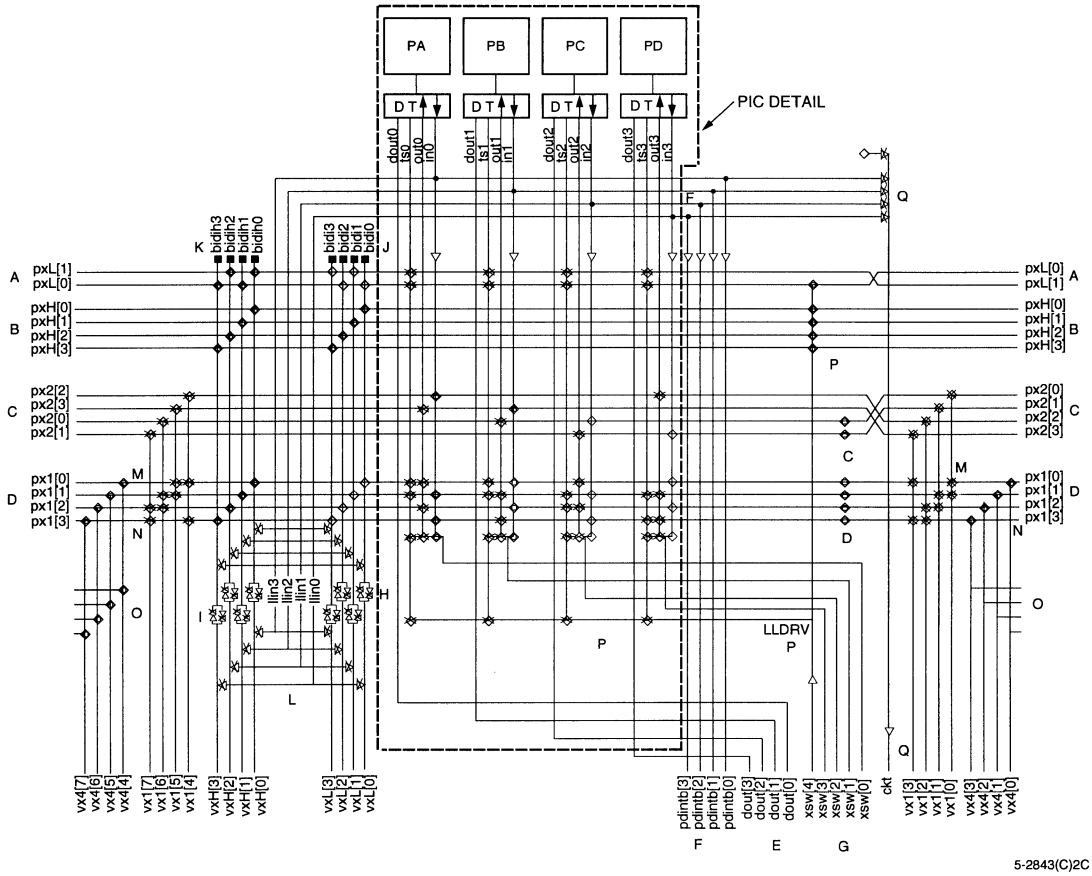


Figure 20. PIC Architecture

Programmable Input/Output Cells

(continued)

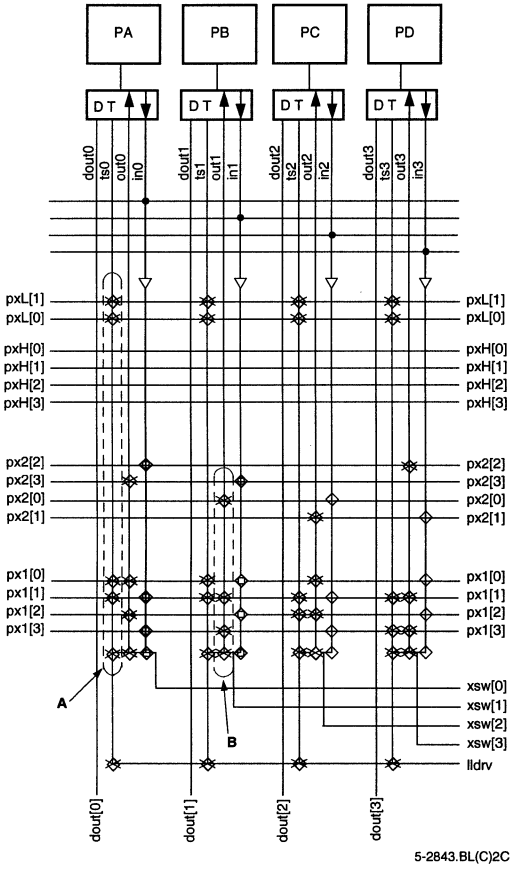


Figure 21. PIC Detail

PLC-PIC Routing Resources

There is no direct connection between the inter-PIC R-nodes and the PLC R-nodes. All connections to/from the PLC must be done through the connecting R-nodes which are perpendicular to the R-nodes in the PIC. The use of perpendicular and parallel R-nodes will be clearer if the PLC and PIC architectures (Figure 17 and Figure 20) are placed side by side. Twenty-nine R-nodes in the PLC can be connected to the fifteen R-nodes in the PIC.

Multiple connections between the PIC px1 R-nodes and the PLC x1 R-nodes are available. These allow buses placed in any arbitrary order on the I/O pads to be unscrambled when placed on the PLC x1 R-nodes. Connections are also available between the PIC px2 R-nodes and the PLC x1 R-nodes.

There are eight tridirectional (four TRIDI/four TRIDIH) buffers in each PIC; they can do the following:

- Drive a signal from an I/O pad onto one of the adjacent PLC's xL or xH R-nodes
- Drive a signal from an I/O pad onto one of the two pxL or four pxH R-nodes in the PIC
- Drive a signal from the PLC xL or xH R-nodes onto one of the two pxL or four pxH R-nodes in the PIC
- Drive a signal from the PIC pxL or pxH R-nodes onto one of the PLC xL or xH R-nodes

Figure 21 shows paths to and from pads and the use of MUX CIPs to connect R-nodes. Detail A shows six MUX CIPs for the pad P0 used to construct the net for the 3-state signal. In the MUX CIP, one of six R-nodes is connected to an R-node to form the net. In this case, the ts0 signal can be driven by either of the two pxLs, px1[0], px1[1], xsw[0], or the lldrv R-nodes. Detail B shows the four MUX CIPs used to drive the P1 output. The source R-node for out1 is either xsw[1], px1[1], px1[3], or px2[2].

Interquad Routing

In the *ORCA* 2C Series devices, the PLC array is split into four equal quadrants. In between these quadrants, routing has been added to route signals between the quadrants, especially to the quadrant in the opposite corner. The two types of interquad blocks, vertical and horizontal, are pitch matched to PICs. Vertical interquad blocks (vIQ) run between quadrants on the left and right, while horizontal interquad blocks (hIQ) run between top and bottom quadrants. Since hIQ and vIQ blocks have the same logic, only the hIQ block is described below.

The interquad routing connects xL and xH R-nodes. It does not affect local routing (xsw, x1, x4, fast carry), so local routing is the same, whether PLC-PLC connections cross quadrants or not. There are no connections to the local R-nodes in the interquad blocks. Figure 22 presents a (not to scale) view of interquad routing.

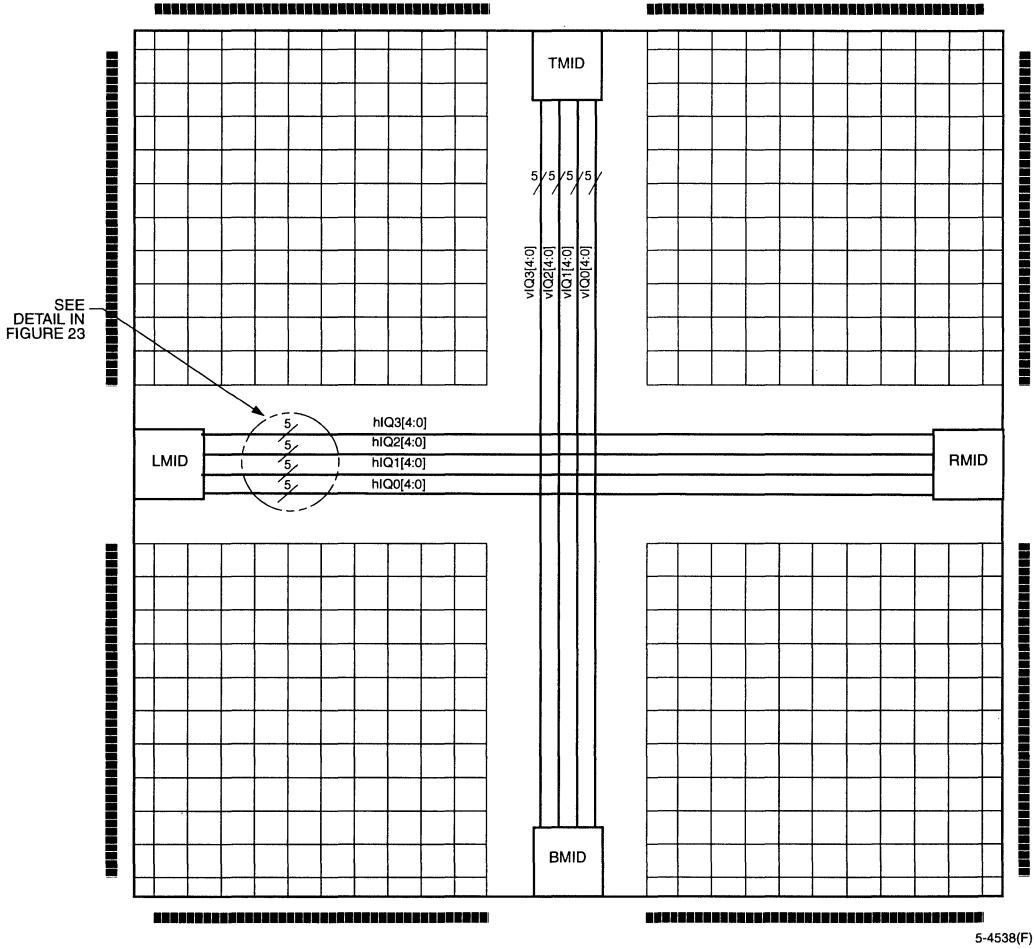


Figure 22. Interquad Routing

Interquad Routing (continued)

In the hIQ block in Figure 23, the xH R-nodes from one quadrant connect through a CIP to its counterpart in the opposite quadrant, creating a path that spans the PLC array. Since a passive CIP is used to connect the two xH R-nodes, a 3-state signal can be routed on the two xH R-nodes in the opposite quadrants, and then they can be connected through this CIP.

In the hIQ block, the 20 hIQ R-nodes span the array in a horizontal direction. The 20 hIQ R-nodes consist of four groups of five R-nodes each. To effectively route nibble-wide buses, each of these sets of five R-nodes can connect to only one of the bits of the nibble for both the xH and xL. For example, hIQ0 R-nodes can only connect to the xH0 and xL0 R-nodes, and the hIQ1 R-nodes can connect only to the xH1 and xL1 R-nodes, etc. Buffers are provided for routing signals from the xH and xL R-nodes onto the hIQ R-nodes and from the hIQ R-nodes onto the xH and xL R-nodes. Therefore, a connection from one quadrant to another can be made using only two xH R-nodes (one in each quadrant) and one interquad R-node.

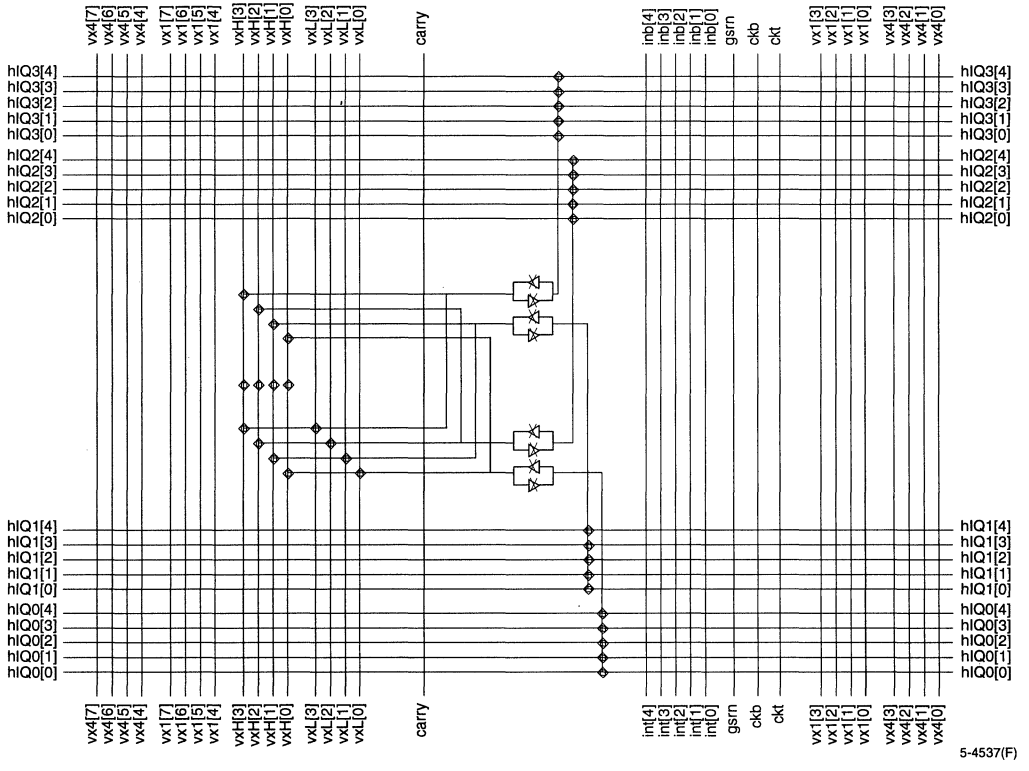


Figure 23. hIQ Block Detail

Interquad Routing (continued)

ATT2C40 Subquad Routing

In the *ORCA* ATT2C40, each quadrant of the device is split into smaller arrays of PLCs called subquads. Each of these subquads is made of a 4 x 4 array of PLCs (for a total of 16 per subquadrant), except at the outer edges of array, which have less than 16 PLCs per subquad. New routing resources, called subquad R-nodes, have been added between each adjacent pair of subquads to enhance the routability of the ATT2C40. A portion of the center of the ATT2C40 array is shown in Figure 24, including the subquad blocks containing a 4 x 4 array of PLCs, the interquad routing R-nodes, and the subquad routing R-nodes.

All of the inter-PLC routing resources discussed previously continue to be routed between a PLC and its adjacent PLC, even if the two adjacent PLCs are in different subquad blocks. Since the PLC routing has not been modified for the ATT2C40 architecture, this means that all of the same routing connections are possible for the ATT2C40 as for any other *ORCA* 2C Series device. In this way, the ATT2C40 is upwardly compatible when compared with the other 2C Series devices. As the inter-PLC routing runs between subquad blocks, it crosses the new subquad R-nodes. When this happens, CIPs are used to connect the subquad R-nodes to the x4 and/or the xH R-nodes which lie along the other axis of the PLC array.

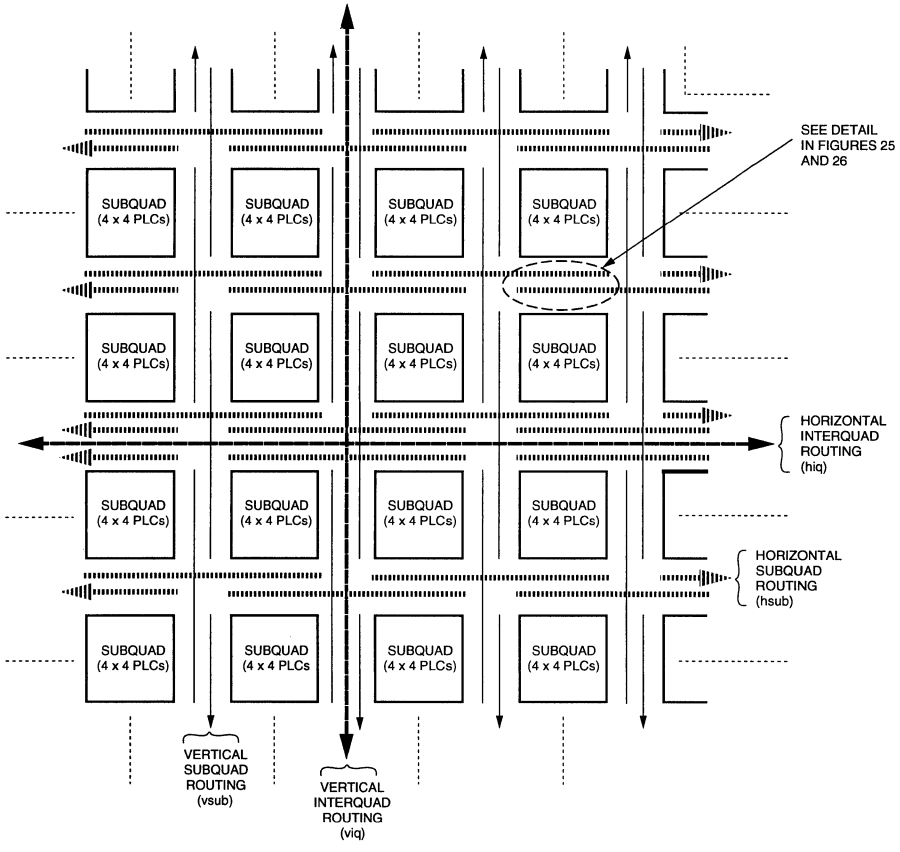


Figure 24. Subquad Blocks and Subquad Routing

Interquad Routing (continued)

The x4 and xH R-nodes make the only connections to the subquad R-nodes; therefore, the array remains symmetrical and homogeneous. Since each subquad is made from a 4 x 4 array of PLCs, the distance between sets of subquad R-nodes is four PLCs, which is also the distance between the breaks of the x4 R-nodes. Therefore, each x4 R-node will cross exactly one set of subquad R-nodes. Since all x4 R-nodes make the same connections to the subquad R-nodes that they cross, all x4 R-nodes in the array have the same connectivity, and the symmetry of the routing is preserved. Since all xH R-nodes cross the same number of subquad blocks, the symmetry is maintained for the xH R-nodes as well.

The new subquad R-nodes travel a length of eight PLCs (seven PLCs on the outside edge) before they are broken. Unlike other inter-PLC R-nodes, they cannot be connected end-to-end. As shown in Figure 24, some of the horizontal (vertical) subquad R-nodes have connectivity to the subquad to the left of (above) the current subquad, while others have connectivity to the subquad to the right (below). This allows connections to/from the current subquad from/to the PLCs in all subquads that surround it.

Between all subquads, including in the center of the array, there are three groups of subquad R-nodes where each group contains four R-nodes. Figure 25 shows the connectivity of these three groups of subquad R-nodes (hsub) to the vx4 and vxH R-nodes running between a vertical pair of PLCs. Between each vertical pair of subquad blocks, four of the blocks shown in Figure 25 are used, one for each pair of vertical PLCs.

The first two groups, depicted as A and B, have connectivity to only one of the two sets of x4 R-nodes between pairs of PLCs. Since they are very lightly loaded, they are very fast. The third group, C, connects to both groups of x4 R-nodes between pairs of PLCs, as well as all of the xH R-nodes between pairs of PLCs, providing high flexibility. The connectivity for the vertical subquad routing (vsub) is the same as described above for the horizontal subquad routing, when rotated onto the other axis.

At the center row and column of each quadrant, a fourth group of subquad R-nodes has been added. These subquad R-nodes only have connectivity to the xH R-nodes. The xH R-nodes are also broken at this point, which means that each xH R-node travels one-half of the quadrant (i.e., one-quarter of the device) before it is broken by a CIP. Since the xH R-nodes can be connected end-to-end, the resulting line can be either one-quarter, one-half, three-quarters, or the entire length of the array. The connectivity of the xH R-nodes and this fourth group of subquad R-nodes, indicated as D, are detailed in Figure 26. Again, the connectivity for the vertical subquad routing (vsub) is the same as the horizontal subquad routing, when rotated onto the other axis.

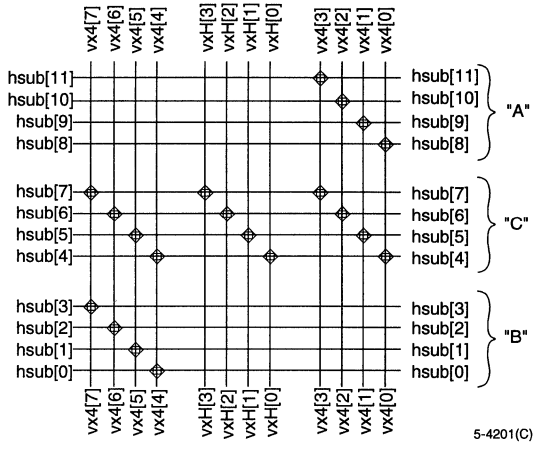


Figure 25. Horizontal Subquad Routing Connectivity

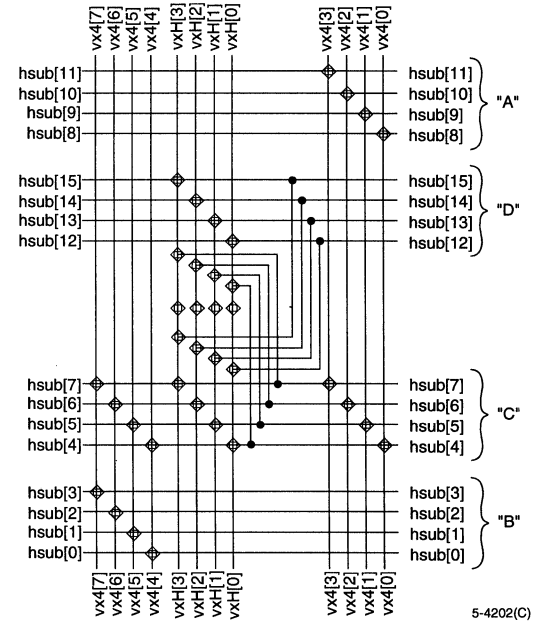


Figure 26. Horizontal Subquad Routing Connectivity (Half Quad)

Interquad Routing (continued)

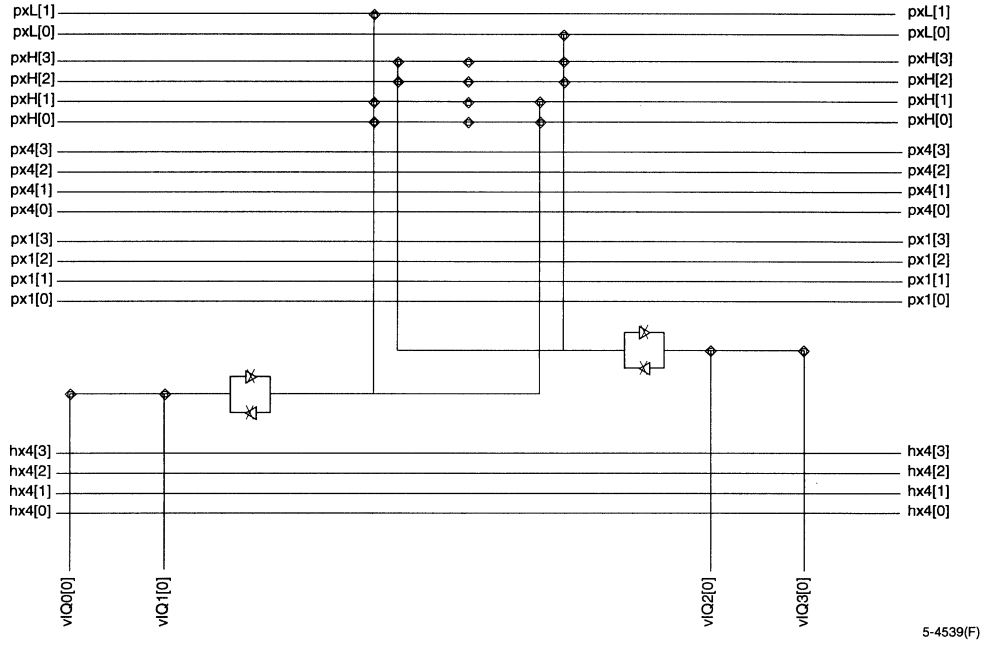


Figure 27. Top (TMID) Routing

PIC Interquad (MID) Routing

Between the PICs in each quadrant, there is also connectivity between the PIC routing and the interquad routing. These blocks are called LMID (left), TMID (top), RMID (right), and BMID (bottom). The TMID routing is shown in Figure 27. As with the hIQ and vIQ blocks, the only connectivity to the PIC routing is to the global pxH and pxL R-nodes.

The pxH R-nodes from the one quadrant can be connected through a CIP to its counterpart in the opposite quadrant, providing a path that spans the array of PICs. Since a passive CIP is used to connect the two pxH R-nodes, a 3-state signal can be routed on the two pxH R-nodes in the opposite quadrants, and then connected through this CIP. As with the hIQ and vIQ blocks, CIPs and buffers allow nibble-wide connections between the interquad R-nodes, the xH R-nodes, and the xL R-nodes.

Programmable Corner Cells

Programmable Routing

The programmable corner cell (PCC) contains the circuitry to connect the routing of the two PICs in each corner of the device. The PIC px1 and px2 R-nodes are directly connected together from one PIC to another. The PIC pxL R-nodes are connected from one block to another through tridirectional buffers. Four CIPs in each corner connect the four pxH R-nodes from each side of the device.

Special-Purpose Functions

In addition to routing functions, special-purpose functions are located in each FPGA corner. The upper-left PCC contains connections to the boundary-scan logic. The upper-right PCC contains connections to the readback logic and the connectivity to the global 3-state signal (ts_all). The lower-left PCC contains connections to the internal oscillator.

The lower-right PCC contains connections to the start-up and global reset logic. During configuration, the $\overline{\text{RESET}}$ input pad always initiates a configuration abort, as described in the FPGA States of Operation section. After configuration, the global set/reset signal (gsrn) can either be disabled (the default), directly connected to the $\overline{\text{RESET}}$ input pad, or sourced by a lower-right corner signal. If the $\overline{\text{RESET}}$ input pad is not used as a global reset after configuration, this pad can be used as a normal input pad. During start-up, the release of the global set/reset, the release of the I/Os, and the release of the external DONE signal can each be timed individually based upon the start-up clock. The start-up clock can come from CCLK or it can be routed into the start-up block using the lower-right corner routing resources. More details on start-up can be found in the FPGA States of Operation section.

Clock Distribution Network

The *ORCA* 2C series clock distribution scheme uses primary and secondary clocks. This provides the system designer with additional flexibility in assigning clock input pins.

One advantage is that board-level clock traces routed to the FPGA are shorter. On a PC board, the added length of high-speed clock traces routed to dedicated clock input pins can significantly increase the parasitic impedances. The primary advantage of the *ORCA* clock distribution is the availability of a large number of clocks, since all I/O pins are configurable as clocks.

Primary Clock

The primary clock distribution is shown in Figure 28. If the clock signal is from an I/O pad, it can be driven onto a clock R-node. The clock R-nodes do not provide clock signals directly to the PFU; they act as clock spines from which clocks are branched to xL R-nodes. The xL R-nodes then feed the clocks to PFUs. A multiplexer in each PLC is used to transition from the clock spine to the branch.

For a clock spine in the horizontal direction, the inputs into the multiplexer are the two R-nodes from the left and right PICs (ckl and ckr) and the local clock R-node from the perpendicular direction (hck). This signal is then buffered and driven onto one of the vertical xL R-nodes, forming the branches. The same structure is used for a clock spine in the vertical direction. In this case, the multiplexer selects from R-nodes from the top and bottom PICs (ckt, ckb, and vck) and drives the signal onto one of the horizontal xL R-nodes.

Figure 28 illustrates the distribution of the low-skew primary clock to a large number of loads using a main spine and branches. Each row (column) has two dedicated clock R-nodes originating from PICs on opposite sides of the array. The clock is input from the pads to the dedicated clock R-node ckt to form the clock spine (see Figure 28, Detail A). From the clock spine, net branches are routed using horizontal xL lines. Clocks into PLCs are tapped from the xL R-nodes, as shown in Figure 28, Detail B.

Clock Distribution Network (continued)

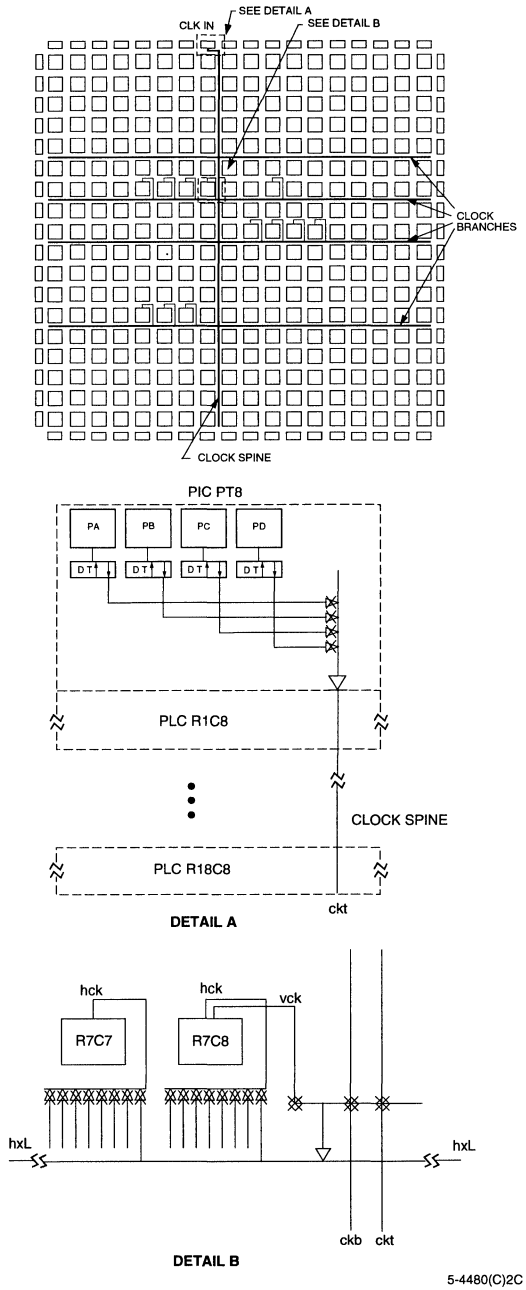


Figure 28. Primary Clock Distribution

Secondary Clock

There are times when a primary clock is either not available or not desired, and a secondary clock is needed. For example:

- Only one input pad per PIC can be placed on the clock routing. If a second input pad in a given PIC requires global signal routing, a secondary clock route must be used.
- Since there is only one branch driver in each PLC for either direction (vertical and horizontal), both clock R-nodes in a particular row or column (ckl and ckr, for example) cannot drive a branch. Therefore, two clocks should not be placed into I/O pads in PICs on the opposite sides of the same row or column if global clocks are to be used.
- Since the clock R-nodes can only be driven from input pads, internally generated clocks should use secondary clock routing.

Figure 29 illustrates the secondary clock distribution. If the clock signal originates from either the left or right side of the FPGA, it can be routed through the TRIDI buffers in the PIC onto one of the adjacent PLC's horizontal xL R-nodes. If the clock signal originates from the top or bottom of the FPGA, the vertical xL R-nodes are used for routing. In either case, an xL R-node is used as the clock spine. In the same manner, if a clock is only going to be used in one quadrant, the xH R-nodes can be used as a clock spine. The routing of the clock spine from the input pads to the vxL (vxH) using the BIDIs (BIDIHs) is shown in Figure 29, Detail A.

In each PLC, a low-skew connection through a long line driver can be used to connect a horizontal xL R-node to a vertical xL R-node or vice versa. As shown in Figure 29, Detail B, this is used to route the branches from the clock spine. If the clock spine is a vertical xL R-node, then the branches are horizontal xL R-nodes and vice versa. The clock is then routed into each PLC from the xL R-node clock branches.

Clock Distribution Network (continued)

To minimize skew, the PLC clock input for all PLCs must be connected to the branch xL R-nodes, not the spine xL R-node. Even in PLCs where the clock is routed from the spine to the branches, the clock should be routed back into the PLC from the clock branch.

If the clock is to drive only a limited number of loads, the PFUs can be connected directly to the clock spine. In this case, all flip-flops driven by the clock must be located in the same row or column.

Alternatively, the clock can be routed from the spine to the branches by using the BIDIs instead of the long line drivers. This results in added delay in the clock net, but the clock skew is approximately equal to the clock routed using the long line drivers. This method can be used to create a clock that is used in only one quadrant. The xH R-nodes act as a clock spine, which is then routed to perpendicular xH R-nodes (the branches) using the BIDIHs.

Clock signals, such as the output of a counter, can also be generated in PLCs and routed onto an xL R-node, which then acts as a clock spine. Although the clock can be generated in any PLC, it is recommended that the clock be located as close to the center of the FPGA as possible to minimize clock skew.

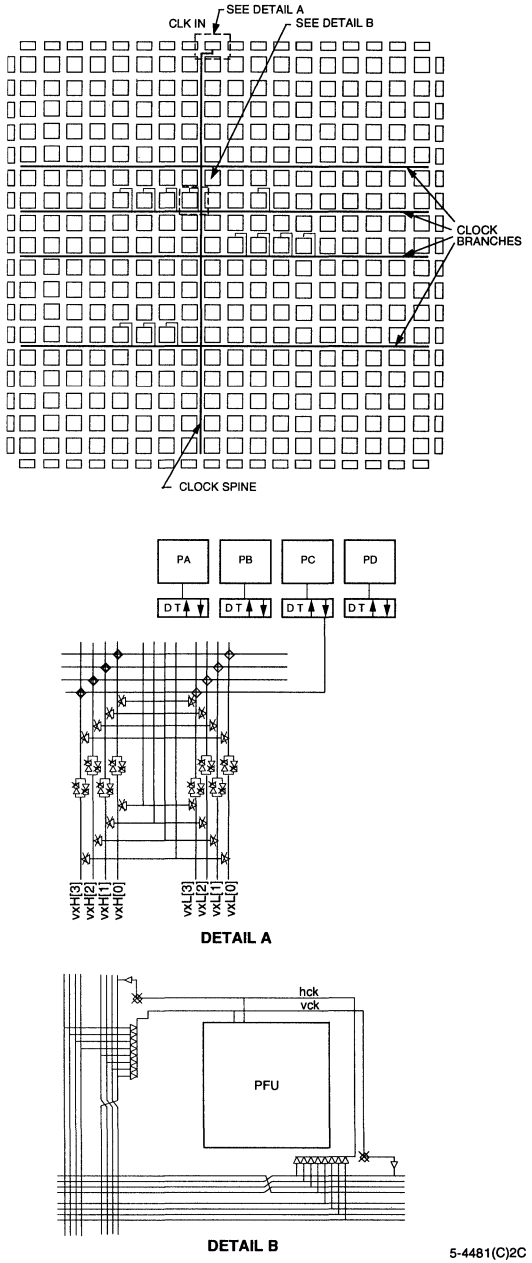


Figure 29. Secondary Clock Distribution

FPGA States of Operation

Prior to becoming operational, the FPGA goes through a sequence of states, including initialization, configuration, and start-up. This section discusses these three states. Figure 30 outlines the FPGA states.

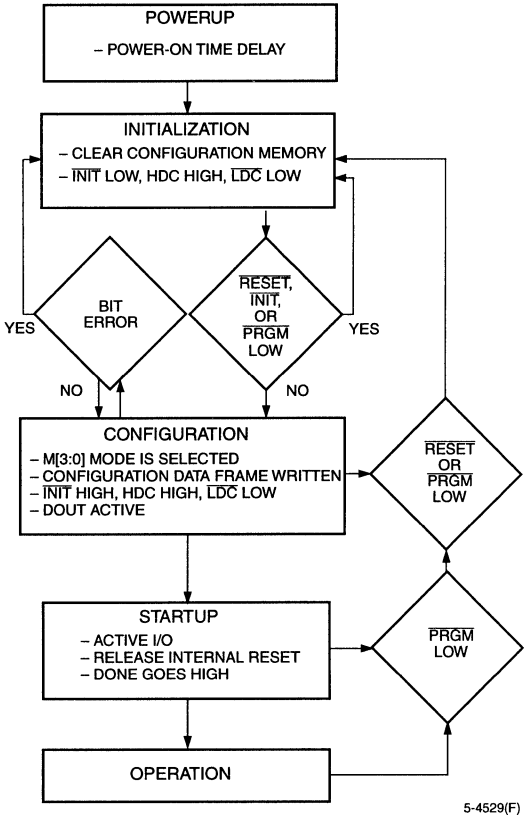


Figure 30. FPGA States of Operation

Initialization

Upon powerup, the device goes through an initialization process. First, an internal power-on-reset circuit is triggered when power is applied. When VDD reaches the voltage at which portions of the FPGA begin to operate (2.5 V to 3 V), the I/Os are configured based on the configuration mode, as determined by the mode select inputs M[2:0]. A time-out delay is initiated when VDD reaches between 3.0 V and 4.0 V to allow the power supply voltage to stabilize. The $\overline{\text{INIT}}$ and DONE outputs are low. At powerup, if VDD does not rise from 2.0 V to VDD in less than 25 ms, the user should delay configuration by inputting a low into $\overline{\text{INIT}}$, $\overline{\text{PRGM}}$, or $\overline{\text{RESET}}$ until VDD is greater than the recommended minimum operating voltage (4.75 V for commercial devices).

At the end of initialization, the default configuration option is that the configuration RAM is written to a low state. This prevents shorts prior to configuration. As a configuration option, after the first configuration (i.e., at reconfiguration), the user can reconfigure without clearing the internal configuration RAM first.

The active-low, open-drain initialization signal $\overline{\text{INIT}}$ is released and must be pulled high by an external resistor when initialization is complete. To synchronize the configuration of multiple FPGAs, one or more $\overline{\text{INIT}}$ pins should be wire-ANDed. If $\overline{\text{INIT}}$ is held low by one or more FPGAs or an external device, the FPGA remains in the initialization state. $\overline{\text{INIT}}$ can be used to signal that the FPGAs are not yet initialized. After $\overline{\text{INIT}}$ goes high for two internal clock cycles, the mode lines are sampled and the FPGA enters the configuration state.

The high during configuration (HDC), low during configuration ($\overline{\text{LDC}}$), and DONE signals are active outputs in the FPGA's initialization and configuration states. HDC, $\overline{\text{LDC}}$, and DONE can be used to provide control of external logic signals such as reset, bus enable, or PROM enable during configuration. For parallel master configuration modes, these signals provide PROM enable control and allow the data pins to be shared with user logic signals.

If configuration has begun, an assertion of $\overline{\text{RESET}}$ or $\overline{\text{PRGM}}$ initiates an abort, returning the FPGA to the initialization state. The $\overline{\text{PRGM}}$ and $\overline{\text{RESET}}$ pins must be pulled back high before the FPGA will enter the configuration state. During the start-up and operating states, only the assertion of $\overline{\text{PRGM}}$ causes a reconfiguration.

FPGA States of Operation (continued)

In the master configuration modes, the FPGA is the source of configuration clock (CCLK). In this mode, the initialization state is extended to ensure that, in daisy-chain operation, all daisy-chained slave devices are ready. Independent of differences in clock rates, master mode devices remain in the initialization state an additional six internal clock cycles after $\overline{\text{INIT}}$ goes high.

When configuration is initiated, a counter in the FPGA is set to 0 and begins to count configuration clock cycles applied to the FPGA. As each configuration data frame is supplied to the FPGA, it is internally assembled into data words. Each data word is loaded into the internal configuration memory. The configuration loading process is complete when the internal length count equals the loaded length count in the length count field, and the required end of configuration frame is written.

All I/Os operate as TTL inputs during configuration. All I/Os that are not used during the configuration process are 3-stated with internal pull-ups. During configuration, the PLC latch/FFs are held set/reset and the internal BIDI buffers are 3-stated. The TRIDIs in the PICs are not 3-stated. The combinatorial logic begins to function as the FPGA is configured. Figure 31 shows the general waveform of the initialization, configuration, and start-up states.

Configuration

The *ORCA* Series FPGA functionality is determined by the state of internal configuration RAM. This configuration RAM can be loaded in a number of different modes. In these configuration modes, the FPGA can act as a master or a slave of other devices in the system. The decision as to which configuration mode to use is a system design issue. The next section discusses configuration in detail, including the configuration data format and the configuration modes used to load the configuration data in the FPGA.

Start-Up

After configuration, the FPGA enters the start-up phase. This phase is the transition between the configuration and operational states.

This begins when the number of CCLKs received after $\overline{\text{INIT}}$ goes high is equal to the value of the length count field in the configuration frame and when the end of configuration frame has been written. The system design issue in the start-up phase is to ensure the user I/Os become active without inadvertently activating devices in the system or causing bus contention. A second system design concern is the timing of the release of global set/reset of the PLC latches/FFs.

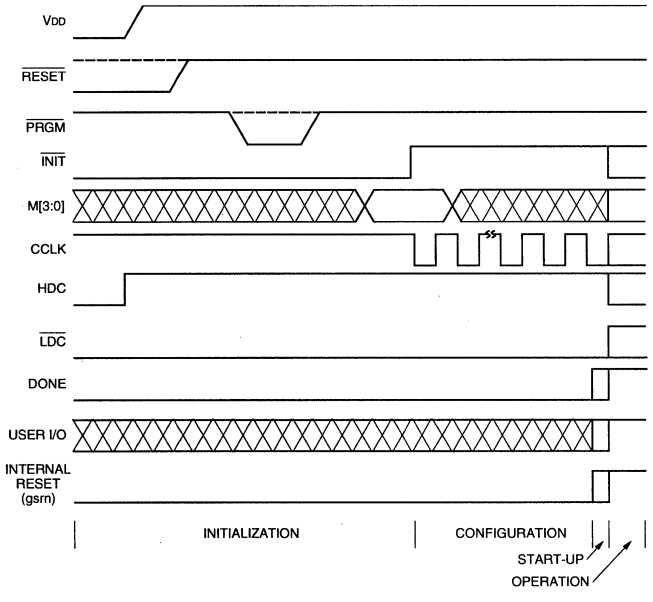


Figure 31. Initialization/Configuration/Start-Up Waveforms

FPGA States of Operation (continued)

There are configuration options which control the relative timing of three events: DONE going high, release of the set/reset of internal FFs, and user I/Os becoming active. Figure 32 shows the start-up timing for both the *ORCA* and ATT3000 Series FPGAs. The system designer determines the relative timing of the I/Os becoming active, DONE going high, and the release of the set/reset of internal FFs. In the *ORCA* Series FPGA, the three events can occur in any arbitrary sequence. This means that they can occur before or after each other, or they can occur simultaneously. The default is for DONE to go high first. This allows configuration sources to be disconnected so that there is no bus contention when the I/Os become active in later cycles. The FFs are set/reset one cycle after DONE goes high so that operation begins in a known state. The DONE output is an open drain and may include an optional internal pull-up resistor to accommodate wired ANDing. The open-drain DONE outputs from multiple FPGAs can be ANDed and used as an active-high ready signal, an active-low PROM enable, or a reset to other portions of the system.

There is also a synchronous start-up mode where start-up does not begin until DONE goes high. The enabling of the FPGA outputs and the set/reset of the internal flip-flops can be triggered or delayed from the rising edge of DONE. Start-up can be delayed by holding the DONE signal low in the synchronous start-up mode. If the DONE signals of multiple FPGAs are tied together, with all in the synchronous start-up mode, start-up does not begin until all of the FPGAs are configured. Normally, the three events are triggered by CCLK. As a configuration option, the three events can be triggered by a user clock, UCLK. This allows start-up to be synchronized by a known system clock. When the user clock option is enabled, the user can still hold DONE low to delay start-up. This allows the synchronization of the start-up of multiple FPGAs. In addition to controlling the FPGA during start-up, additional start-up techniques to avoid contention include using isolation devices between the FPGA and other circuits in the system, reassigning I/O locations, and maintaining I/Os as 3-stated outputs until contentions are resolved.

Reconfiguration

To reconfigure the FPGA when the device is operating in the system, a low pulse is input into $\overline{\text{PRGM}}$. The configuration data in the FPGA is cleared, and the I/Os not used for configuration are 3-stated. The FPGA then samples the mode select inputs and begins reconfiguration. When reconfiguration is complete, DONE is released, allowing it to be pulled high.

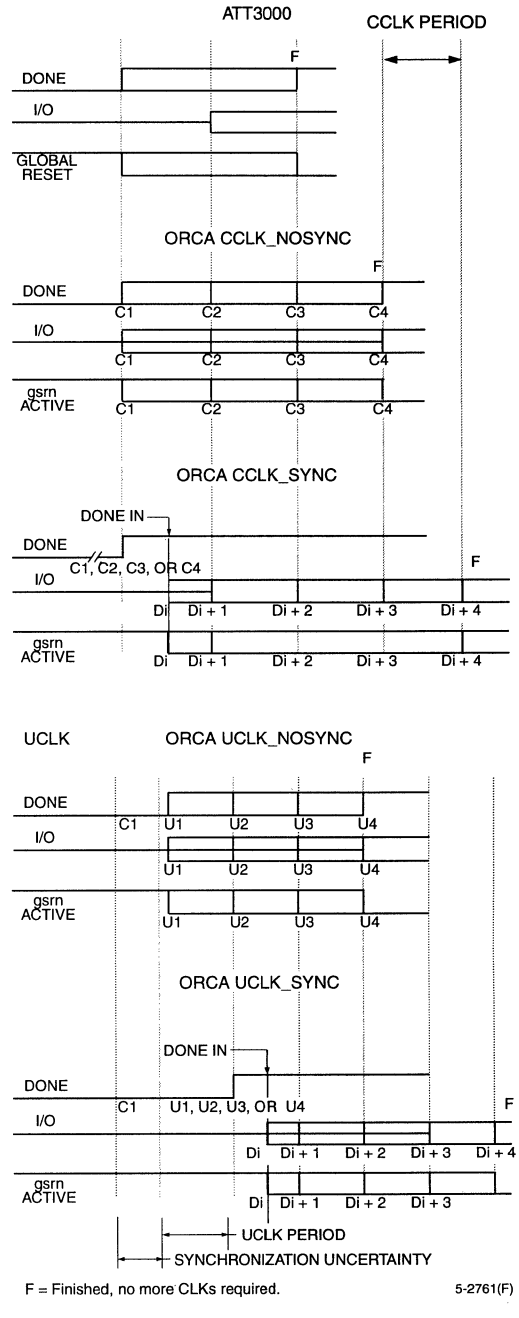


Figure 32. Start-Up Waveform

Configuration Data Format

This section discusses using the *ORCA* Foundry Development System to generate configuration RAM data and then provides the details of the configuration frame format.

Using *ORCA* Foundry to Generate Configuration RAM Data

The configuration data defines the I/O functionality, logic, and interconnections. The bit stream is generated by the development system. The bit stream created by the bit stream generation tool is a series of 1s and 0s used to write the FPGA configuration RAM. The bit stream can be loaded into the FPGA using one of the configuration modes discussed later. In the bit stream generator, the designer selects options which affect the FPGA's functionality. Using the output of bit stream generator, circuit.bit, the development system's download tool can load the configuration data into the *ORCA* series FPGA evaluation board from a PC or workstation. Alternatively, a user can program a PROM (such as the ATT1700 Series Serial ROMs or standard EPROMs) and load the FPGA from the PROM. The development system's PROM programming tool produces a file in .mks or .exo format.

Configuration Data Frame

A detailed description of the frame format is shown in Figure 33. The header frame begins with a series of 1s and a preamble of 0010, followed by a 24-bit length count field representing the total number of configuration clocks needed to complete the loading of the FPGAs. Following the header frame is an optional ID frame. This frame contains data used to determine if the bit stream is being loaded to the correct type of *ORCA* FPGA (e.g., is a bit stream generated for an ATT2C15 actually being sent to an ATT2C15?). It has a secondary function of optionally enabling the parity checking logic for the rest of the data frames.

The configuration data frames follow, with each frame starting with a 0 start bit and ending with three or more 1 stop bits. Following the start bit of each frame are four control bits: program bit, set to 1 if this is a data frame; compress bit, set to 1 if this is a compressed frame; and the opar and epar parity bits, to be discussed in the Bit Stream Error Checking section. An 11-bit address field (that determines which column in the FPGA is to be written) is followed by alignment and write control bits. For uncompressed frames, the data bits needed to write one column in the FPGA are next. For compressed frames, the data bits from the previous frame are sent to a different FPGA column, as specified by the new address bits; therefore, new data bits are not required. When configuration of the current FPGA is finished, an end-of-configuration frame (where the program bit is set to 0) is sent to the FPGA. The length and number of data frames and information about the PROM size for the 2C series FPGAs are given in Table 6.

Configuration Data Format (continued)

Table 6. Configuration Frame Size

Device	2C04	2C06	2C08	2C10	2C12	2C15	2C26	2C40
# of Frames	480	568	656	744	832	920	1096	1378
Data Bits/Frame	110	130	150	170	190	210	250	316
Configuration Data (# of frames x # of data bits/frame)	52,800	73,840	98,400	126,480	158,080	193,200	274,000	435,448
Maximum Total # Bits/Frame (align bits, 1 write bit, 8 stop bits)	136	160	176	200	216	240	280	344
Maximum Configuration Data (# bits x # of frames)	65,280	90,880	115,456	148,800	179,712	220,800	306,880	474,032
Maximum PROM Size (bits) (add 48-bit header, ID frame, and 40-bit end of configuration frame)	65,504	91,128	115,720	149,088	180,016	221,128	307,248	474,464

The data frames for all the 2C series devices are given in Table 7. An alignment field is required in the slave parallel mode for the uncompressed format. The alignment field (shown by [A]) is a series of 0s: five for the 2C06, 2C10, 2C15, and 2C26; three for the 2C40; and one for the 2C04, 2C08, and 2C12. The alignment field is not required in any other mode.

Table 7. Configuration Data Frames

ATT2C04	
Uncompressed	010 opar epar [addr10:0] [A]1[Data109:0]111
Compressed	011 opar epar [addr10:0] 111
ATT2C06	
Uncompressed	010 opar epar [addr10:0] [A]1[Data129:0]111
Compressed	011 opar epar [addr10:0] 111
ATT2C08	
Uncompressed	010 opar epar [addr10:0] [A]1[Data149:0]111
Compressed	011 opar epar [addr10:0] 111
ATT2C10	
Uncompressed	010 opar epar [addr10:0] [A]1[Data169:0]111
Compressed	011 opar epar [addr10:0] 111
ATT2C12	
Uncompressed	010 opar epar [addr10:0] [A]1[Data189:0]111
Compressed	011 opar epar [addr10:0] 111
ATT2C15	
Uncompressed	010 opar epar [addr10:0] [A]1[Data209:0]111
Compressed	011 opar epar [addr10:0] 111
ATT2C26	
Uncompressed	010 opar epar [addr10:0] [A]1[Data249:0]111
Compressed	011 opar epar [addr10:0] 111
ATT2C40	
Uncompressed	010 opar epar [addr10:0] [A]1[Data315:0]111
Compressed	011 opar epar [addr10:0] 111

Configuration Data Format (continued)

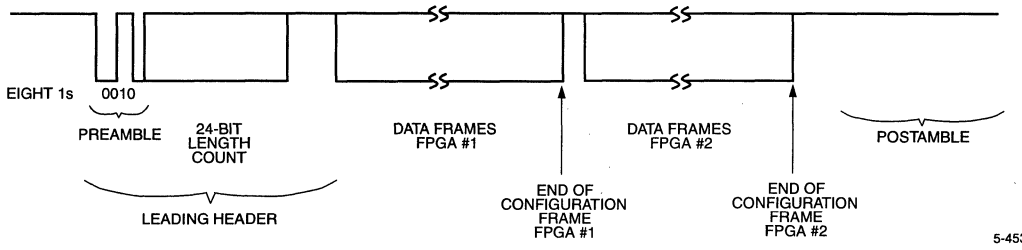


Figure 33. Serial Configuration Data Format

Header	11111111	Leading header — 4 bits minimum dummy bits
	0010	Preamble
	24-Bit Length Count	Configuration frame length
	1111	Trailing header — 4 bits minimum dummy bits
ID Frame (Optional)	0	Frame start
	P—1	Must be set to 1 to indicate data frame
	C—0	Must be set to 0 to indicate uncompressed
	Opar, Epar	Frame parity bits
	Addr[10:0] = 11111111111	ID frame address
	Prty_En	Set to 1 to enable parity
	Reserved [42:0]	Reserved bits set to 0
	ID	20-bit part ID
	111	Three or more stop bits (high) to separate frames
Configuration Data Frame (repeated for each data frame)	0	Frame start
	P—1 or 0	1 indicates data frame; 0 indicates all frames are written
	C—1 or 0	Uncompressed — 0 indicates data and address are supplied; Compressed — 1 indicates only address is supplied
	Opar, Epar	Frame parity bits
	Addr[10:0]	Column address in FPGA to be written
	A	Alignment bit (different number of 0s needed for each part)
	1	Write bit — used in uncompressed data frame
	Data Bits	Needed only in an uncompressed data frame
	.	.
	.	.
	111	One or more stop bits (high) to separate frames
End of Configuration	0010011111111111	16 bits—00 indicates all frames are written
Postamble	111111	Additional 1s

Note: For slave parallel mode, the byte containing the preamble must be 11110010. The number of leading header dummy bits must be (n * 8) + 4, where n is any nonnegative integer and the number of trailing dummy bits must be (n * 8), where n is any positive integer. The number of stop bits/frame for slave parallel mode must be (x * 8), where x is a positive integer. Note also that the bit stream generator tool supplies a bit stream which is compatible with all configuration modes, including slave parallel mode.

Figure 34. Configuration Frame Format and Contents

Bit Stream Error Checking

There are three different types of bit stream error checking performed in the *ORCA* 2C FPGAs: ID frame, frame alignment, and parity checking.

An optional ID data frame can be sent to a specified address in the FPGA. This ID frame contains a unique code for the part it was generated for which is compared within the FPGA. Any differences are flagged as an ID error.

Every data frame in the FPGA begins with a start bit set to 0 and three or more stop bits set to 1. If any of the three previous bits were a 0 when a start bit is encountered, it is flagged as a frame alignment error.

Parity checking is also done on the FPGA for each frame, if it has been enabled by setting the prty_en bit to 1 in the ID frame. Two parity bits, opar and epar, are used to check the parity of bits in alternating bit positions to even parity in each data frame. If an odd number of ones is found for either the even bits (starting with the start bit) or the odd bits (starting with the program bit), then a parity error is flagged.

When any of the three possible errors occur, the FPGA is forced into the INIT state, forcing $\overline{\text{INIT}}$ low. The FPGA will remain in this state until either the $\overline{\text{RESET}}$ or $\overline{\text{PRGM}}$ pins are asserted.

FPGA Configuration Modes

There are eight methods for configuring the FPGA. Seven of the configuration modes are selected on the M0, M1, and M2 inputs. The eighth configuration mode is accessed through the boundary-scan interface. A fourth input, M3, is used to select the frequency of the internal oscillator, which is the source for CCLK in some configuration modes. The nominal frequencies of the internal oscillator are 1.25 MHz and 10 MHz. The 1.25 MHz frequency is selected when the M3 input is unconnected or driven to a high state.

There are three basic FPGA configuration modes: master, slave, and peripheral. The configuration data can be transmitted to the FPGA serially or in parallel bytes. As a master, the FPGA provides the control signals out to strobe data in. As a slave device, a clock is generated externally and provided into CCLK. In the peripheral mode, the FPGA acts as a microprocessor peripheral. Table 8 lists the functions of the configuration mode pins.

Table 8. Configuration Modes

M2	M1	M0	CCLK	Configuration Mode	Data
0	0	0	Output	Master	Serial
0	0	1	Input	Slave Parallel	Parallel
0	1	0	Reserved		
0	1	1	Input	Sync Peripheral	Parallel
1	0	0	Output	Master (up)	Parallel
1	0	1	Output	Async Peripheral	Parallel
1	1	0	Output	Master (down)	Parallel
1	1	1	Input	Slave	Serial

Master Parallel Mode

The master parallel configuration mode is generally used to interface to industry-standard byte-wide memory such as the 2764 and larger EPROMs. Figure 35 provides the connections for master parallel mode. The FPGA outputs an 18-bit address on A[17:0] to memory and reads one byte of configuration data on the rising edge of RCLK. The parallel bytes are internally serialized starting with the least significant bit, D0.

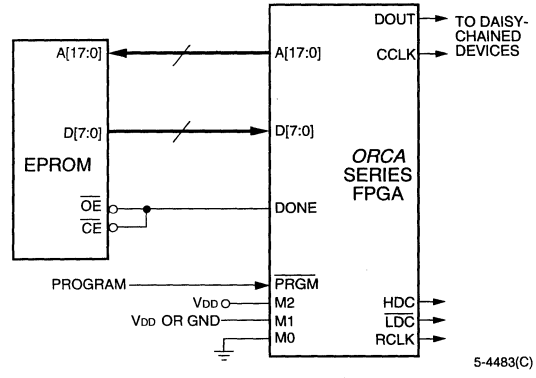


Figure 35. Master Parallel Configuration Schematic

There are two parallel master modes: master up and master down. In master up, the starting memory address is 00000 Hex and the FPGA increments the address for each byte loaded. In master down, the starting memory address is 3FFFF Hex and the FPGA decrements the address.

One master mode FPGA can interface to the memory and provide configuration data on DOUT to additional FPGAs in a daisy chain. The configuration data on DOUT is provided synchronously with the falling edge of CCLK. The frequency of the CCLK output is eight times that of RCLK.

FPGA Configuration Modes (continued)

Master Serial Mode

In the master serial mode, the FPGA loads the configuration data from an external serial ROM. The configuration data is either loaded automatically at start-up or on a $\overline{\text{PRGM}}$ command to reconfigure. The ATT1700 and ATT1700A Series can be used to configure the FPGA in the master serial mode. This provides a simple four-pin interface in an eight-pin package. The ATT1736, ATT1765, and ATT17128 serial ROMs store 32K, 64K, and 128K bits, respectively.

Configuration in the master serial mode can be done at powerup and/or upon a configure command. The system or the FPGA must activate the serial ROM's RESET/OE and $\overline{\text{CE}}$ inputs. At powerup, the FPGA and serial ROM each contain internal power-on reset circuitry that allows the FPGA to be configured without the system providing an external signal. The power-on reset circuitry causes the serial ROM's internal address pointer to be reset. After powerup, the FPGA automatically enters its initialization phase.

The serial ROM/FPGA interface used depends on such factors as the availability of a system reset pulse, availability of an intelligent host to generate a configure command, whether a single serial ROM is used or multiple serial ROMs are cascaded, whether the serial ROM contains a single or multiple configuration programs, etc. Because of differing system requirements and capabilities, a single FPGA/serial ROM interface is generally not appropriate for all applications.

Data is read in the FPGA sequentially from the serial ROM. The DATA output from the serial ROM is connected directly into the DIN input of the FPGA. The CCLK output from the FPGA is connected to the CLOCK input of the serial ROM. During the configuration process, CCLK clocks one data bit on each rising edge.

Since the data and clock are direct connects, the FPGA/serial ROM design task is to use the system or FPGA to enable the RESET/OE and $\overline{\text{CE}}$ of the serial ROM(s). There are several methods for enabling the serial ROM's RESET/OE and $\overline{\text{CE}}$ inputs. The serial ROM's RESET/OE is programmable to function with RESET active-high and $\overline{\text{OE}}$ active-low or $\overline{\text{RESET}}$ active-low and OE active-high.

In Figure 36, serial ROMs are cascaded to configure multiple daisy-chained FPGAs. The host generates a 500 ns low pulse into the FPGA's $\overline{\text{PRGM}}$ input. The FPGA's $\overline{\text{INIT}}$ input is connected to the serial ROMs' $\overline{\text{RESET}}$/OE input, which has been programmed to function with $\overline{\text{RESET}}$ active-low and OE active-high. The FPGA DONE is routed to the $\overline{\text{CE}}$ pin. The low on DONE enables the serial ROMs. At the completion of configuration, the high on the FPGA's DONE disables the serial ROM.

Serial ROMs can also be cascaded to support the configuration of multiple FPGAs or to load a single FPGA when configuration data requirements exceed the capacity of a single serial ROM. After the last bit from the first serial ROM is read, the serial ROM outputs $\overline{\text{CEO}}$ low and 3-states the DATA output. The next serial ROM recognizes the low on $\overline{\text{CE}}$ input and outputs configuration data on the DATA output. After configuration is complete, the FPGA's DONE output into $\overline{\text{CE}}$ disables the serial ROMs.

This FPGA/serial ROM interface is not used in applications in which a serial ROM stores multiple configuration programs. In these applications, the next configuration program to be loaded is stored at the ROM location that follows the last address for the previous configuration program. The reason the interface in Figure 36 will not work in this application is that the low output on the $\overline{\text{INIT}}$ signal would reset the serial ROM address pointer, causing the first configuration to be reloaded.

In some applications, there can be contention on the FPGA's DIN pin. During configuration, DIN receives configuration data, and after configuration, it is a user I/O. If there is contention, an early DONE at start-up (selected in *ORCA* Foundry) may correct the problem. An alternative is to use $\overline{\text{LDC}}$ to drive the serial ROM's $\overline{\text{CE}}$ pin. In order to reduce noise, it is generally better to run the master serial configuration at 1.25 MHz (M3 pin tied high), rather than 10 MHz, if possible.

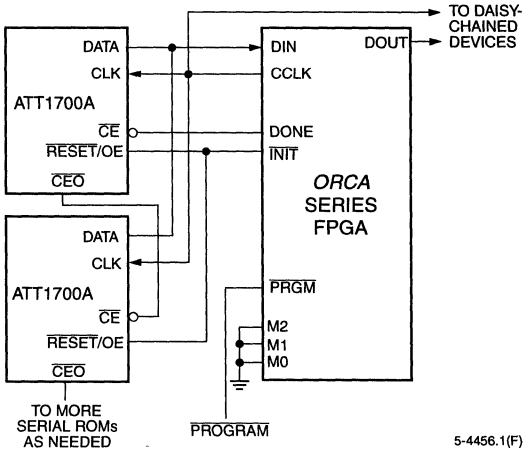


Figure 36. Master Serial Configuration Schematic

FPGA Configuration Modes (continued)

Asynchronous Peripheral Mode

Figure 37 shows the connections needed for the asynchronous peripheral mode. In this mode, the FPGA system interface is similar to that of a microprocessor-peripheral interface. The microprocessor generates the control signals to write an 8-bit byte into the FPGA. The FPGA control inputs include active-low $\overline{CS0}$ and active-high CS1 chip selects and a write $\overline{WR}$ input. The chip selects can be cycled or maintained at a static level during the configuration cycle. Each byte of data is written into the FPGA's D[7:0] input pins.

The FPGA provides a RDY/$\overline{BUSY}$ status output to indicate that another byte can be loaded. A low on RDY/$\overline{BUSY}$ indicates that the double-buffered hold/shift registers are not ready to receive data. The shortest time RDY/$\overline{BUSY}$ is low occurs when a byte is loaded into the hold register and the shift register is empty, in which case the byte is immediately transferred to the shift register. The longest time for RDY/$\overline{BUSY}$ to remain low occurs when a byte is loaded into the holding register and the shift register has just started shifting configuration data into configuration RAM. The RDY/$\overline{BUSY}$ status is also available on the D7 pin by enabling the chip selects, setting $\overline{WR}$ high, and setting $\overline{RD}$ low.

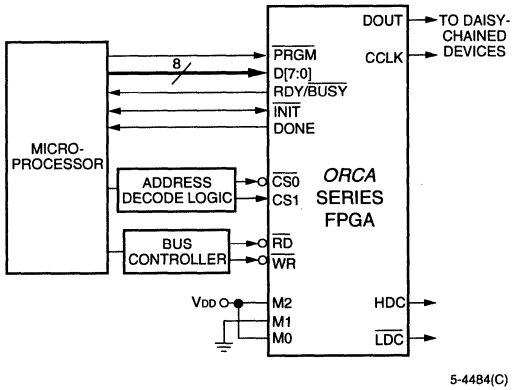


Figure 37. Asynchronous Peripheral Configuration Schematic

Synchronous Peripheral Mode

In the synchronous peripheral mode, byte-wide data is input into D[7:0] on the rising edge of the CCLK input. The first data byte is clocked in on the second CCLK after $\overline{INIT}$ goes high. Subsequent data bytes are clocked in on every eighth rising edge of CCLK. The RDY/$\overline{BUSY}$ signal is an output which acts as an acknowledge. RDY/$\overline{BUSY}$ goes high one CCLK after data is clocked and, after one CCLK cycle, returns low. The process repeats until all of the data is loaded into the FPGA. The data begins shifting on DOUT 1.5 cycles after it is loaded in parallel. It requires additional CCLKs after the last byte is loaded to complete the shifting. Figure 38 shows the connections for synchronous peripheral mode.

As with master modes, the peripheral modes can be used as the lead FPGA for a daisy chain of slave FPGAs.

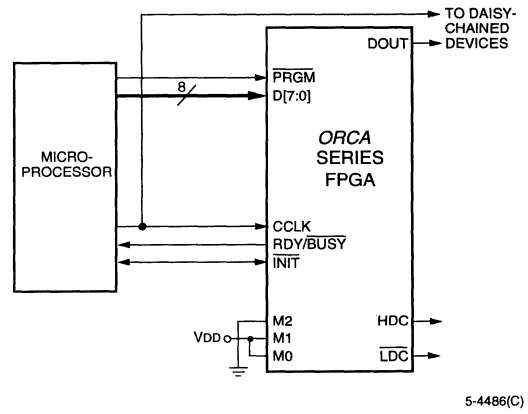


Figure 38. Synchronous Peripheral Configuration Schematic

FPGA Configuration Modes (continued)

Slave Serial Mode

The slave serial mode is primarily used when multiple FPGAs are configured in a daisy chain. The serial slave serial mode is also used on the FPGA evaluation board which interfaces to the download cable. A device in the slave serial mode can be used as the lead device in a daisy chain. Figure 39 shows the connections for the slave serial configuration mode.

The configuration data is provided into the FPGA's DIN input synchronous with the configuration clock CCLK input. After the FPGA has loaded its configuration data, it retransmits the incoming configuration data on DOUT. CCLK is routed into all slave serial mode devices in parallel.

Multiple slave FPGAs can be loaded with identical configurations simultaneously. This is done by loading the configuration data into the DIN inputs in parallel.

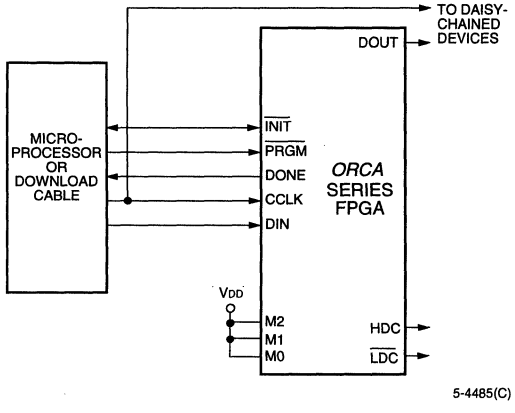


Figure 39. Slave Serial Configuration Schematic

Slave Parallel Mode

The slave parallel mode is essentially the same as the slave serial mode except that 8 bits of data are input on pins D[7:0] for each CCLK cycle. Due to 8 bits of data being input per CCLK cycle, the DOUT pin does not contain a valid bit stream for slave parallel mode. As a result, the lead device cannot be used in the slave parallel mode in a daisy-chain configuration.

Figure 40 is a schematic of the connections for the slave parallel configuration mode. $\overline{WR}$ and $\overline{CS0}$ are active-low chip select signals, and CS1 is an active-high chip select signal. These chip selects allow the user to configure multiple FPGAs in slave parallel mode using an 8-bit data bus common to all of the FPGAs. These chip selects can then be used to select the FPGA(s) to be configured with a given bit stream, but once an FPGA has been selected, it cannot be deselected until it has been completely programmed.

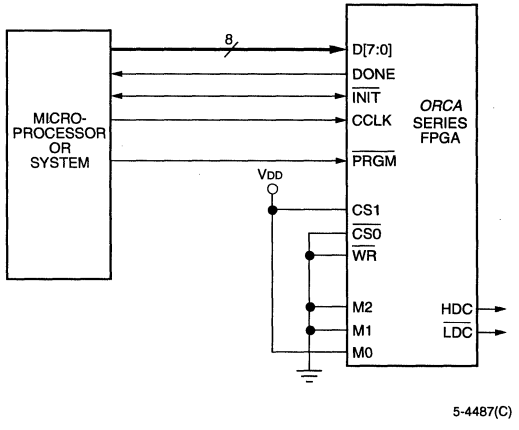


Figure 40. Slave Parallel Configuration Schematic

FPGA Configuration Modes (continued)

Daisy Chain

Multiple FPGAs can be configured by using a daisy chain of the FPGAs. Daisy chaining uses a lead FPGA and one or more FPGAs configured in slave serial mode. The lead FPGA can be configured in any mode except slave parallel mode. (Daisy chaining is not available with the boundary-scan ram_w instruction, discussed later.)

All daisy-chained FPGAs are connected in series. Each FPGA reads and shifts the preamble and length count in on positive CCLK and out on negative CCLK edges.

An upstream FPGA which has received the preamble and length count outputs a high on DOUT until it has received the appropriate number of data frames so that downstream FPGAs do not receive frame start bits (0s). After loading and retransmitting the preamble and length count to a daisy chain of slave devices, the lead device loads its configuration data frames. The loading of configuration data continues after the lead device has received its configuration data if its internal frame bit counter has not reached the length count. When the configuration RAM is full and the number of bits received is less than the length count field, the FPGA shifts any additional data out on DOUT.

The configuration data is read into DIN of slave devices on the positive edge of CCLK, and shifted out DOUT on the negative edge of CCLK. Figure 41 shows the connections for loading multiple FPGAs in a daisy-chain configuration.

The generation of CCLK for the daisy-chained devices which are in slave serial mode differs depending on the configuration mode of the lead device. A master parallel mode device uses its internal timing generator to produce an internal CCLK at eight times its memory address rate (RCLK). The asynchronous peripheral mode device outputs eight CCLKs for each write cycle. If the lead device is configured in either synchronous peripheral or a slave mode, CCLK is routed to the lead device and to all of the daisy-chained devices.

The development system can create a composite configuration bit stream for configuring daisy-chained FPGAs. The frame format is a preamble, a length count for the total bit stream, multiple concatenated data frames, an end-of-configuration frame per device, a postamble, and an additional fill bit per device in the serial chain.

As seen in Figure 41, the $\overline{\text{INIT}}$ pins for all of the FPGAs are connected together. This is required to guarantee that powerup and initialization will work correctly. In general, the DONE pins for all of the FPGAs are also connected together as shown to guarantee that all of the FPGAs enter the start-up state simultaneously. This may not be required, depending upon the start-up sequence desired.

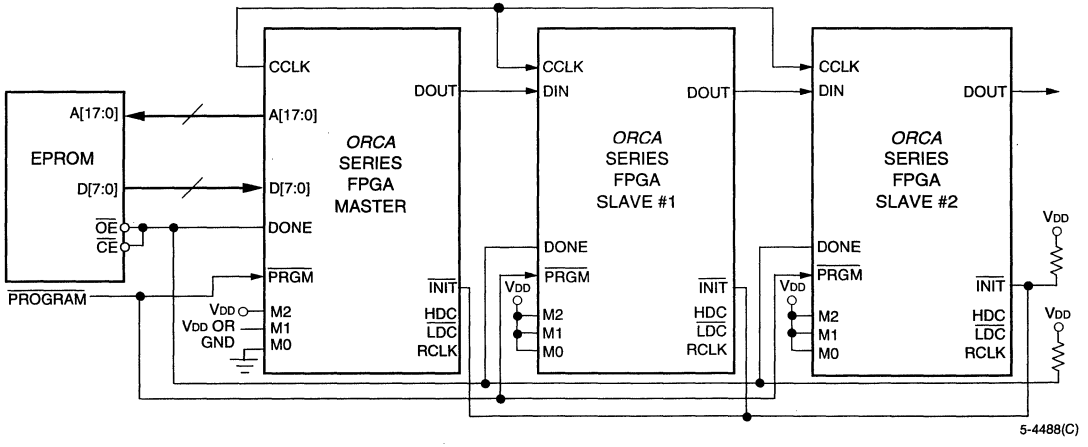


Figure 41. Daisy-Chain Configuration Schematic

Readback

Readback is used to read back the configuration data and, optionally, the state of the PFU outputs. A readback operation can be done while the FPGA is in normal system operation. The readback operation cannot be daisy chained. To use readback, the user selects options in the bit stream generator in the *ORCA* Foundry development system.

Table 9 provides readback options selected in the bit stream generator tool. The table provides the number of times that the configuration data can be read back. This is intended primarily to give the user control over the security of the FPGA's configuration program. The user can prohibit readback (0), allow a single readback (1), or allow unrestricted readback (U).

The pins used for readback are readback data (RD_DATA), read configuration ($\overline{\text{RD\_CFGN}}$), and configuration clock (CCLK). A readback operation is initiated by a high-to-low transition on $\overline{\text{RD\_CFGN}}$. The $\overline{\text{RD\_CFGN}}$ input must remain low during the readback operation. The readback operation can be restarted at frame 0 by setting the $\overline{\text{RD\_CFGN}}$ pin high, applying at least two rising edges of CCLK, and then applying $\overline{\text{RD\_CFGN}}$ low again. One bit of data is shifted out on RD_DATA on the rising edge of CCLK. The first start bit of the readback frame is transmitted out on the first rising edge of CCLK after $\overline{\text{RD\_CFGN}}$ is input low.

It should be noted that the RD_DATA output pin is also used as the dedicated boundary-scan output pin, TDO. If this pin is being used as TDO, the RD_DATA output from readback can be routed internally to any other pin desired. The $\overline{\text{RD\_CFGN}}$ input pin is also used to control the global 3-state (ts_all) function. Before and during configuration, the ts_all signal is always driven by the RD_CFGN input and readback is disabled. After configuration, the selection as to whether this input drives the readback or global 3-state function is determined by a set of bit stream options. If used as the $\overline{\text{RD\_CFGN}}$ input for readback, the internal ts_all input can be routed internally to be driven by any input pin.

The readback frame contains the configuration data and the state of the internal logic. During readback, the value of all five PFU outputs can be captured. The following options are allowed when doing a capture of the PFU outputs.

1. Do not capture data (the data written to the RAMs, usually 0, will be read back).
2. Capture data upon entering readback.
3. Capture data based upon a configurable signal internal to the FPGA. If this signal is tied to logic 0, capture RAMs are written continuously, which is equivalent to ATT3000 Series capture.
4. Capture data on either options 2 or 3 above.

The readback frame has a similar, but not identical, format to the configuration frame. This eases a bitwise comparison between the configuration and readback data. The readback data is not inverted. Every data frame has one low start bit and one high stop bit. The preamble, including the length count field, is not part of the readback frame. The readback frame contains states in locations not used in the configuration. These locations need to be masked out when comparing the configuration and readback frames. The development system optionally provides a readback bit steam to compare to readback data from the FPGA.

Table 9. Readback Options

Option	Function
0	Inhibit Readback
1	Allow One Readback Only
U	Allow Unrestricted Number of Readbacks

Boundary Scan

The increasing complexity of integrated circuits (ICs) and IC packages has increased the difficulty of testing printed-circuit boards (PCBs). To address this testing problem, the *IEEE* standard 1149.1 - 1990 (*IEEE* Standard Test Access Port and Boundary-Scan Architecture) is implemented in the *ORCA* series of FPGAs. It allows users to efficiently test the interconnection between integrated circuits on a PCB as well as test the integrated circuit itself. The *IEEE* 1149.1 standard is a well-defined protocol that ensures interoperability among boundary-scan (BSCAN) equipped devices from different vendors.

The *IEEE* 1149.1 standard defines a test access port (TAP) that consists of a four-pin interface with an optional reset pin for boundary-scan testing of integrated circuits in a system. The *ORCA* Series FPGA provides four interface pins: test data in (TDI), test mode select (TMS), test clock (TCK), and test data out (TDO). The $\overline{\text{PRGM}}$ pin used to reconfigure the device also resets the boundary-scan logic.

The user test host serially loads test commands and test data into the FPGA through these pins to drive outputs and examine inputs. In the configuration shown in Figure 42, where boundary scan is used to test ICs, test data is transmitted serially into TDI of the first BSCAN device (U1), through TDO/TDI connections between BSCAN devices (U2 and U3), and out TDO of the last BSCAN device (U4). In this configuration, the TMS and TCK signals are routed to all boundary-scan ICs in parallel so that all boundary-scan components operate in the same state. In other configurations, multiple scan paths are used instead of a single ring. When multiple scan paths are used, each ring is independently controlled by its own TMS and TCK signals.

Figure 43 provides a system interface for components used in the boundary-scan testing of PCBs. The three major components shown are the test host, boundary-scan support circuit, and the devices under test (DUTs). The DUTs shown here are *ORCA* Series FPGAs with dedicated boundary-scan circuitry. The test host is normally one of the following: automatic test equipment (ATE), a workstation, a PC, or a microprocessor.

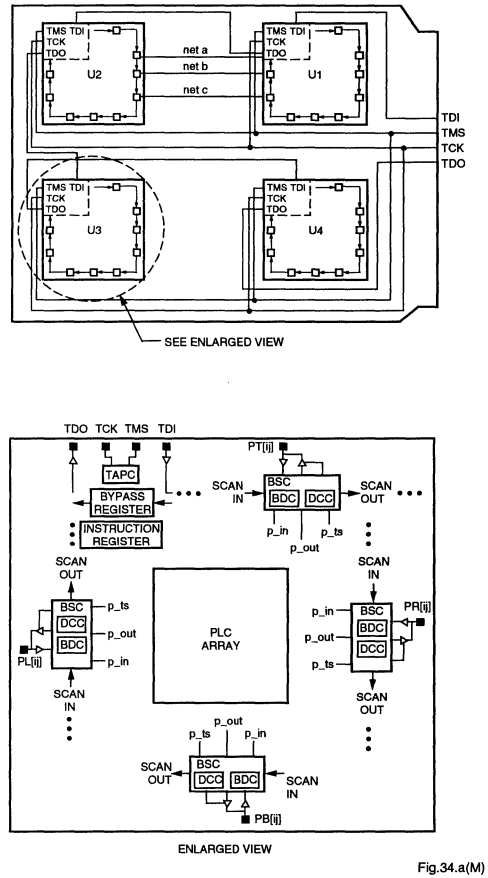


Fig.34.a(M)

Key: BSC = boundary-scan cell, BDC = bidirectional data cell, and DCC = data control cell.

Figure 42. Printed-Circuit Board with Boundary-Scan Circuitry

Boundary Scan (continued)

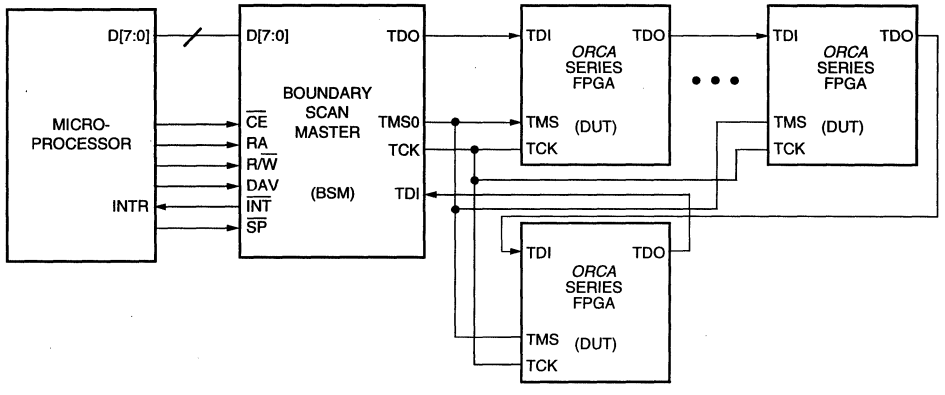


Figure 43. Boundary-Scan Interface

The boundary-scan support circuit shown in Figure 43 is the Lucent 497AA Boundary-Scan Master (BSM). The BSM off-loads tasks from the test host to increase test throughput. To interface between the test host and the DUTs, the BSM has a general microprocessor interface and provides parallel-to-serial/serial-to-parallel conversion, as well as three 8K data buffers. The BSM also increases test throughput with a dedicated automatic test pattern generator and with compression of the test response with a signature analysis register. The PC-based Lucent boundary-scan test card/software allows a user to quickly prototype a boundary-scan test setup.

Boundary-Scan Instructions

The *ORCA* Series boundary-scan circuitry is used for three mandatory *IEEE* 1149.1 tests (EXTEST, SAMPLE/PRELOAD, BYPASS) and four Lucent-defined instructions. The 3-bit wide instruction register supports the eight instructions listed in Table 10.

Table 10. Boundary-Scan Instructions

Code	Instruction
000	EXTEST
001	PLC Scan Ring 1
010	RAM Write (RAM_W)
011	Reserved
100	SAMPLE/PRELOAD
101	PLC Scan Ring 2
110	RAM Read (RAM_R)
111	BYPASS

The external test (EXTEST) instruction allows the interconnections between ICs in a system to be tested for opens and stuck-at faults. If an EXTEST instruction is performed for the system shown in Figure 42, the connections between U1 and U2 (shown by nets a, b, and c) can be tested by driving a value onto the given nets from one device and then determining whether the same value is seen at the other device. This is determined by shifting 2 bits of data for each pin (one for the output value and one for the 3-state value) through the BSR until each one aligns to the appropriate pin. Then, based upon the value of the 3-state signal, either the I/O pad is driven to the value given in the BSR, or the BSR is updated with the input value from the I/O pad, which allows it to be shifted out TDO.

The SAMPLE instruction is useful for system debugging and fault diagnosis by allowing the data at the FPGA's I/Os to be observed during normal operation. The data for all of the I/Os is captured simultaneously into the BSR, allowing them to be shifted-out TDO to the test host. Since each I/O buffer in the PICs is bidirectional, two pieces of data are captured for each I/O pad: the value at the I/O pad and the value of the 3-state control signal.

There are four Lucent-defined instructions. The PLC scan rings 1 and 2 (PSR1, PSR2) allow user-defined internal scan paths using the PLC latches/FFs. The RAM_Write Enable (RAM_W) instruction allows the user to serially configure the FPGA through TDI. The RAM_Read Enable (RAM_R) allows the user to read back RAM contents on TDO after configuration.

Boundary Scan (continued)

ORCA Boundary-Scan Circuitry

The *ORCA* Series boundary-scan circuitry includes a test access port controller (TAPC), instruction register (IR), boundary-scan register (BSR), and bypass register. It also includes circuitry to support the four Lucent-defined instructions.

Figure 44 shows a functional diagram of the boundary-scan circuitry that is implemented in the *ORCA* series. The input pins' (TMS, TCK, and TDI) locations vary depending on the part, and the output pin is the dedicated TDO/RD_DATA output pad. Test data in (TDI) is the serial input data. Test mode select (TMS) controls the boundary-scan test access port controller (TAPC). Test clock (TCK) is the test clock on the board.

The BSR is a series connection of boundary-scan cells (BSCs) around the periphery of the IC. Each I/O pad on the FPGA, except for CCLK, DONE, and the boundary-scan pins (TCK, TDI, TMS, and TDO), is included in the BSR. The first BSC in the BSR (connected to TDI) is located in the first PIC I/O pad on the left of the top side of the FPGA (PTA PIC). The BSR proceeds clockwise around the top, right, bottom, and left sides of the array. The last BSC in the BSR (connected to TDO) is located on the top of the left side of the array (PLA3).

The bypass instruction uses a single FF which resynchronizes test data that is not part of the current scan operation. In a bypass instruction, test data received on TDI is shifted out of the bypass register to TDO. Since the BSR (which requires a two FF delay for each pad) is bypassed, test throughput is increased when devices that are not part of a test operation are bypassed.

The boundary-scan logic is enabled before and during configuration. After configuration, a configuration option determines whether or not boundary-scan logic is used.

The 32-bit boundary-scan identification register contains the manufacturer's ID number, unique part number, and version, but is not implemented in the *ORCA* series of FPGAs. If boundary scan is not used, TMS, TDI, and TCK become user I/Os, and TDO is 3-stated or used in the readback operation.

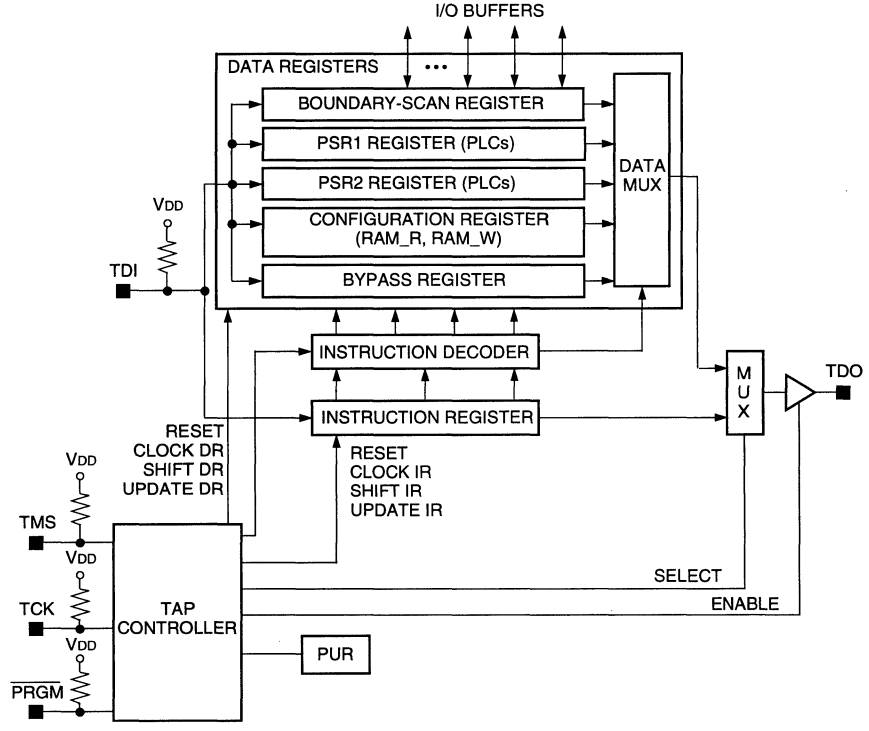


Figure 44. *ORCA* Series Boundary-Scan Circuitry Functional Diagram

Boundary Scan (continued)

ORCA Series TAP Controller (TAPC)

The *ORCA* Series TAP controller (TAPC) is a 1149.1 compatible test access port controller. The 16 JTAG state assignments from the *IEEE* 1149.1 specification are used. The TAPC is controlled by TCK and TMS. The TAPC states are used for loading the IR to allow three basic functions in testing: providing test stimuli (Update DR), test execution (Run Test/Idle), and obtaining test responses (Capture DR). The TAPC allows the test host to shift in and out both instructions and test data/results. The inputs and outputs of the TAPC are provided in the table below. The outputs are primarily the control signals to the instruction register and the data register.

Table 11. TAP Controller Input/Outputs

Symbol	I/O	Function
TMS	I	Test Mode Select
TCK	I	Test Clock
PUR	I	Powerup Reset
$\overline{\text{PRGM}}$	I	BSCAN Reset
TRESET	O	Test Logic Reset
Select	O	Select IR (High); Select DR (Low)
Enable	O	Test Data Out Enable
Capture DR	O	Capture/Parallel Load DR
Capture IR	O	Capture/Parallel Load IR
Shift DR	O	Shift Data Register
Shift IR	O	Shift Instruction Register
Update DR	O	Update/Parallel Load DR
Update IR	O	Update/Parallel Load IR

The TAPC generates control signals which allow capture, shift, and update operations on the instruction and data registers. In the capture operation, data is loaded into the register. In the shift operation, the captured data is shifted out while new data is shifted in. In the update operation, either the instruction register is loaded for instruction decode, or the boundary-scan register is updated for control of outputs.

The test host generates a test by providing input into the *ORCA* Series TMS input synchronous with TCK. This sequences the TAPC through states in order to perform the desired function on the instruction register or a data register. Figure 45 provides a diagram of the state transitions for the TAPC. The next state is determined by the TMS input value.

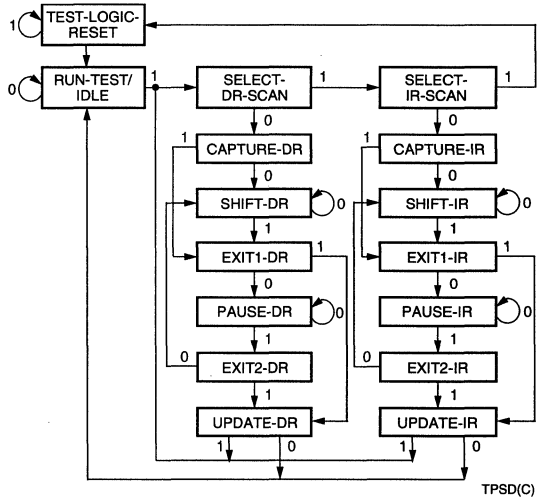


Figure 45. TAP Controller State Transition Diagram

Boundary Scan (continued)

Boundary-Scan Cells

Figure 46 is a diagram of the boundary-scan cell (BSC) in the *ORCA* series PICs. There are four BSCs in each PIC: one for each pad, except as noted above. The BSCs are connected serially to form the BSR. The BSC controls the functionality of the in, out, and 3-state signals for each pad.

The BSC allows the I/O to function in either the normal or test mode. Normal mode is defined as when an output buffer receives input from the PLC array and provides output at the pad or when an input buffer provides input from the pad to the PLC array. In the test mode, the BSC executes a boundary-scan operation, such as shifting in scan data from an upstream BSC in the BSR, providing test stimuli to the pad, capturing test data at the pad, etc.

The primary functions of the BSC are shifting scan data serially in the BSR and observing input (p_in), output (p_out), and 3-state (p_ts) signals at the pads. The BSC consists of two circuits: the bidirectional data cell is used to access the input and output data, and the direction control cell is used to access the 3-state value. Both cells consist of a flip-flop used to shift scan data which feeds a flip-flop to control the I/O buffer. The bidirectional data cell is connected serially to the direction control cell to form a boundary-scan shift register.

The TAPC signals (capture, update, shiftn, treset, and TCK) and the MODE signal control the operation of the BSC. The bidirectional data cell is also controlled by the high out/low in (HOLI) signal generated by the direction control cell. When HOLI is low, the bidirectional data cell receives input buffer data into the BSC. When HOLI is high, the BSC is loaded with functional data from the PLC.

The MODE signal is generated from the decode of the instruction register. When the MODE signal is high (EXTEST), the scan data is propagated to the output buffer. When the MODE signal is low (BYPASS or SAMPLE), functional data from the FPGA's internal logic is propagated to the output buffer.

The boundary-scan description language (BSDL) is provided for each device in the *ORCA* series of FPGAs. The BSDL is generated from a device profile, pinout, and other boundary-scan information.

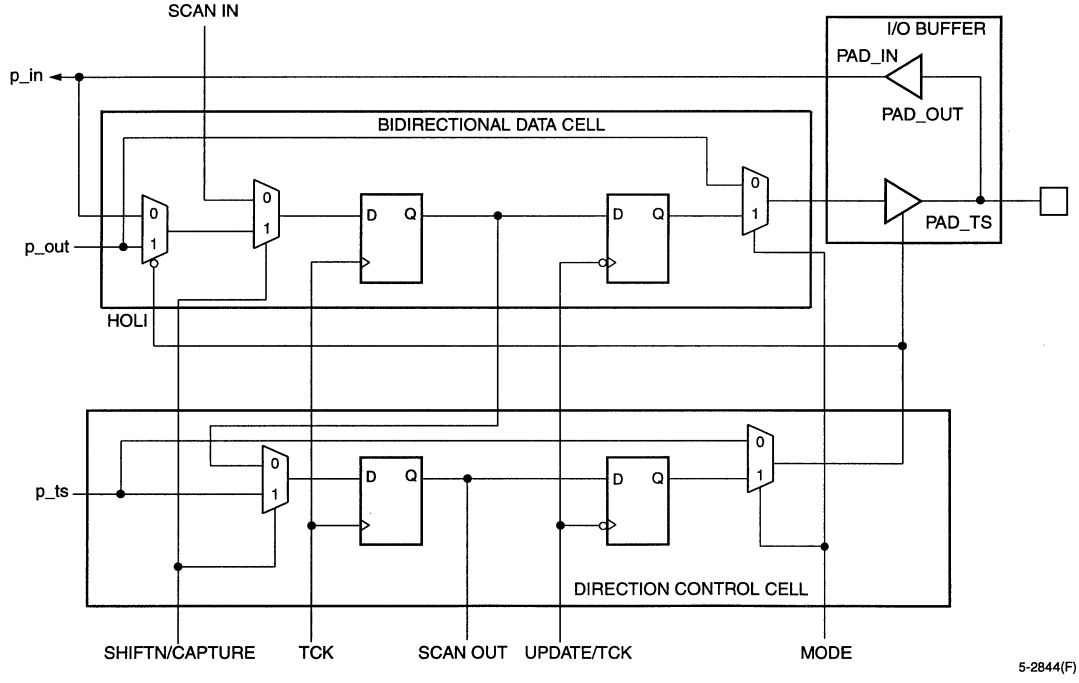


Figure 46. Boundary-Scan Cell

Boundary Scan (continued)

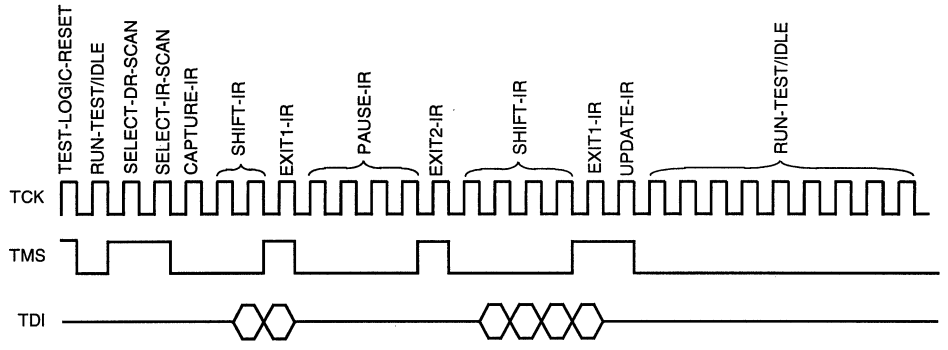


Figure 47. Instruction Register Scan Timing Diagram

Boundary-Scan Timing

To ensure race-free operation, data changes on specific clock edges. The TMS and TDI inputs are clocked in on the rising edge of TCK, while changes on TDO occur on the falling edge of TCK. In the execution of an EXTEST instruction, parallel data is output from the BSR to the FPGA pads on the falling edge of TCK. The maximum frequency allowed for TCK is 10 MHz.

Figure 47 shows timing waveforms for an instruction scan operation. The diagram shows the use of TMS to sequence the TAPC through states. The test host (or BSM) changes data on the falling edge of TCK, and it is clocked into the DUT on the rising edge.

ORCA Timing Characteristics

To define speed grades, the *ORCA* Series part number designation (see Table 45) uses a single-digit number to designate a speed grade. This number is not related to any single ac parameter. Higher numbers indicate a faster set of timing parameters. The actual speed sorting is based on testing the delay in a path consisting of an input buffer, combinatorial delay through all PLCs in a row, and an output buffer. Other tests are then done to verify other delay parameters, such as routing delays, setup times to FFs, etc.

The most accurate timing characteristics are reported by the timing analyzer in the *ORCA* Foundry Development System. A timing report provided by the development system after layout divides path delays into logic and routing delays. The timing analyzer can also provide logic delays prior to layout. While this allows routing budget estimates, there is wide variance in routing delays associated with different layouts.

The logic timing parameters noted in the Electrical Characteristics section of this data sheet are the same as those in the design tools. In the PFU timing given in Table 30, symbol names are generally a concatenation of the PFU operating mode (as defined in Table 2) and the parameter type. The wildcard character (*) is used in symbol names to indicate that the parameter applies to any sub-LUT. The setup, hold, and propagation delay parameters, defined below, are designated in the symbol name by the SET, HLD, and DEL characters, respectively.

ORCA Timing Characteristics

(continued)

The values given for the parameters are the same as those used during production testing and speed binning of the devices. The junction temperature and supply voltage used to characterize the devices are listed in the delay tables. Actual delays at nominal temperature and voltage for best-case processes can be much better than the values given.

Table 12 and Table 13 provide approximate power supply and junction temperature derating for commercial and industrial devices. The delay values in this data sheet and reported by *ORCA* Foundry are shown as **1.00** in the tables. The method for determining the maximum junction temperature is defined in the Thermal Characteristics section. Taken cumulatively, the range of parameter values for best-case vs. worst-case processing, supply voltage, and junction temperature can approach 3 to 1.

Table 12. Derating for Commercial Devices

TJ (°C)	Power Supply Voltage		
	4.75 V	5.0 V	5.25 V
0	0.79	0.77	0.75
25	0.83	0.81	0.79
85	**1.00**	0.97	0.95
100	1.05	1.03	1.00
125	1.14	1.11	1.08

Table 13. Derating for Industrial Devices

TJ (°C)	Power Supply Voltage				
	4.5 V	4.75 V	5.0 V	5.25 V	5.5 V
–40	0.70	0.69	0.67	0.65	0.64
0	0.79	0.77	0.75	0.73	0.72
25	0.83	0.81	0.79	0.77	0.76
85	**1.00**	0.97	0.95	0.92	0.91
100	1.05	1.02	1.00	0.97	0.95
125	1.14	1.11	1.08	1.05	1.03

In addition to supply voltage, process variation, and operating temperature, circuit and process improvements of the *ORCA* series FPGAs over time will result in significant improvement of the actual performance over those listed for a speed grade. Even though lower speed grades may still be available, the distribution of yield to timing parameters may be several speed bins higher than that designated on a product brand. Design practices need to consider best-case timing parameters (e.g., delays = 0), as well as worst-case timing.

The routing delays are a function of fan-out and the capacitance associated with the CIPs and metal interconnect in the path. The number of logic elements that can be driven (or fan-out) by PFUs is unlimited, although the delay to reach a valid logic level can exceed timing requirements. It is difficult to make accurate routing delay estimates prior to design compilation based on fan-out. This is because the CAE software may delete redundant logic inserted by the designer to reduce fan-out, and/or it may also automatically reduce fan-out by net splitting.
The waveform test points are given in the Measurement Conditions section of this data sheet. The timing parameters given in the electrical characteristics tables in this data sheet follow industry practices, and the values they reflect are described below.

Propagation Delay — the time between the specified reference points. The delays provided are the worst case of the tphh and tpll delays for noninverting functions, tplh and tphl for inverting functions, and tphz and tplz for 3-state enable.

Setup Time — the interval immediately preceding the transition of a clock or latch enable signal, during which the data must be stable to ensure it is recognized as the intended value.

Hold Time — the interval immediately following the transition of a clock or latch enable signal, during which the data must be held stable to ensure it is recognized as the intended value.

3-state Enable — the time from when a ts[3:0] signal becomes active and the output pad reaches the high-impedance state.

Estimating Power Dissipation

The total operating power dissipated is estimated by summing the standby (IDDSB), internal, and external power dissipated. The internal and external power is the power consumed in the PLCs and PICs, respectively. In general, the standby power is small and may be neglected. The total operating power is as follows:

$$P_T = \Sigma P_{PLC} + \Sigma P_{PIC}$$

The internal operating power is made up of two parts: clock generation and PFU output power. The PFU output power can be estimated based upon the number of PFU outputs switching when driving an average fan-out of two:

$$P_{PFU} = 0.19 \text{ mW/MHz}$$

For each PFU output that switches, 0.19 mW/MHz needs to be multiplied times the frequency (in MHz) that the output switches. Generally, this can be estimated by using one-half the clock rate, multiplied by some activity factor; for example, 20%.

The power dissipated by the clock generation circuitry is based upon three parts: the fixed clock power, the power/clock branch row or column, and the clock power dissipated in each PFU that uses this particular clock. Therefore, the clock power can be calculated for the three parts using the following equations:

2C04 Clock Power

P = [0.64 mW/MHz
+ (0.22 mW/MHz – Branch) (# Branches)
+ (0.025 mW/MHz – PFU) (# PFUs)] fCLK

For a quick estimate, the worst-case (typical circuit) 2C04 Clock Power ≈ 4.1 mW/MHz.

2C06 Clock Power

P = [0.65 mW/MHz
+ (0.26 mW/MHz – Branch) (# Branches)
+ (0.025 mW/MHz – PFU) (# PFUs)] fCLK

For a quick estimate, the worst-case (typical circuit) 2C06 Clock Power ≈ 5.6 mW/MHz.

2C08 Clock Power

P = [0.66 mW/MHz
+ (0.29 mW/MHz – Branch) (# Branches)
+ (0.025 mW/MHz – PFU) (# PFUs)] fCLK

For a quick estimate, the worst-case (typical circuit) 2C08 Clock Power ≈ 7.2 mW/MHz.

2C10 Clock Power

P = [0.67 mW/MHz
+ (0.33 mW/MHz – Branch) (# Branches)
+ (0.025 mW/MHz – PFU) (# PFUs)] fCLK

For a quick estimate, the worst-case (typical circuit) 2C10 Clock Power ≈ 9.2 mW/MHz.

2C12 Clock Power

P = [0.69 mW/MHz
+ (0.37 mW/MHz – Branch) (# Branches)
+ (0.025 mW/MHz – PFU) (# PFUs)] fCLK

For a quick estimate, the worst-case (typical circuit) 2C12 Clock Power ≈ 11.4 mW/MHz.

2C15 Clock Power

P = [0.70 mW/MHz
+ (0.40 mW/MHz – Branch) (# Branches)
+ (0.025 mW/MHz – PFU) (# PFUs)] fCLK

For a quick estimate, the worst-case (typical circuit) 2C15 Clock Power ≈ 13.7 mW/MHz.

2C26 Clock Power

P = [0.71 mW/MHz
+ (0.47 mW/MHz – Branch) (# Branches)
+ (0.025 mW/MHz – PFU) (# PFUs)] fCLK

For a quick estimate, the worst-case (typical circuit) 2C26 Clock Power ≈ 19.2 mW/MHz.

2C40 Clock Power

P = [0.75 mW/MHz
+ (0.57 mW/MHz – Branch) (# Branches)
+ (0.025 mW/MHz – PFU) (# PFUs)] fCLK

For a quick estimate, the worst-case (typical circuit) 2C40 Clock Power ≈ 29.1 mW/MHz.

The power dissipated in a PIC is the sum of the power dissipated in the four I/Os in the PIC. This consists of power dissipated by inputs and ac power dissipated by outputs. The power dissipated in each I/O depends on whether it is configured as an input, output, or input/output. If an I/O is operating as an output, then there is a power dissipation component for PIN, as well as POUT. This is because the output feeds back to the input.

Estimating Power Dissipation

(continued)

The power dissipated by a TTL input buffer is estimated as:

$$P_{TTL} = 1.8 \text{ mW} + 0.20 \text{ mW/MHz}$$

The power dissipated by a CMOS input buffer is estimated as:

$$P_{CMOS} = 0.20 \text{ mW/MHz}$$

The ac power dissipation from an output or bidirectional is estimated by the following:

$$P_{OUT} = (C_L + 9 \text{ pF}) \times V_{DD}^2 \times F \text{ Watts}$$

where the unit for C_L is farads, and the unit for F is Hz.

As an example of estimating power dissipation, suppose that a fully utilized 2C15 has an average of three outputs for each of the 400 PFUs, that all 20 clock branches are used, that 150 of the 400 PFUs have FFs clocked at 40 MHz, and that the PFU outputs have an average activity factor of 20%.

Twenty TTL-configured inputs, 20 CMOS-configured inputs, 32 outputs driving 30 pF loads, and 16 bidirectional I/Os driving 50 pF loads are also generated from the 40 MHz clock with an average activity factor of 20%. The worst-case power dissipation is estimated as follows:

P_{PFU} = 400 x 3 (0.19 mW/MHz x 20 MHz x 20%)
= 912 mW

P_{CLK} = [0.70 mW/MHz + (0.40 mW/MHz – Branch) (20 Branches)
+ (0.025 mW/MHz – PFU) (150 PFUs) [40 MHz]]
= 498 mW

P_{TTL} = 20 x [1.8 mW + (0.20 mW/MHz x 20 MHz x 20%)]
= 52 mW

P_{CMOS} = 20 x [0.20 mW x 20 MHz x 20%]
= 16 mW

P_{OUT} = 30 x [(30 pF + 9 pF) x 5.5252 x 20 MHz x 20%]
= 129 mW

P_{BID} = 16 x [(50 pF + 9 pF) x 5.5252 x 20 MHz x 20%]
= 104 mW

TOTAL = 1.71 W

Pin Information

Table 14. Pin Descriptions

Symbol	I/O	Description
Dedicated Pins		
VDD	—	Positive power supply.
GND	—	Ground supply.
$\overline{\text{RESET}}$	I	During configuration, $\overline{\text{RESET}}$ forces the restart of configuration and a pull-up is enabled. After configuration, $\overline{\text{RESET}}$ can be used as a general FPGA input or as a direct input, which causes all PLC latches/FFs to be asynchronously set/reset.
CCLK	I	In the master and asynchronous peripheral modes, CCLK is an output which strobes configuration data in. In the slave or synchronous peripheral mode, CCLK is input synchronous with the data on DIN or D[7:0].
DONE	I/O	DONE is a bidirectional pin with an optional pull-up resistor. As an active-low, open-drain output, it indicates that configuration is complete. As an input, a low level on DONE delays FPGA start-up after configuration.
$\overline{\text{PRGM}}$	I	$\overline{\text{PRGM}}$ is an active-low input that forces the restart of configuration and resets the boundary-scan circuitry. This pin always has an active pull-up.
$\overline{\text{RD\_CFGN}}$	I	If readback is enabled, after configuration, a high-to-low transition on $\overline{\text{RD\_CFGN}}$ initiates a readback of configuration data, including PFU output states, starting with frame address 0. During configuration, this is an active-low input that activates the TS_ALL function and 3-states all the I/O. This same functionality can be selected after configuration as well. This pin always has an active pull-up.
RD_DATA/TDO	O	RD_DATA/TDO is a dual-function pin. If used for readback, RD_DATA provides configuration data out. If used in boundary scan, TDO is test data out.
Special-Purpose Pins		
RDY/$\overline{\text{BUSY}}$	O	During configuration in peripheral mode, RDY/$\overline{\text{BUSY}}$ indicates another byte can be written to the FPGA. If a read operation is done when the device is selected, the same status is also available on D7 in asynchronous peripheral mode. After configuration, the pin is a user-programmable I/O.
RCLK	O	During the master parallel configuration mode, RCLK is a read output signal to an external memory. This output is not normally used. After configuration, this pin is a user-programmable I/O pin.
DIN	I	During slave serial or master serial configuration modes, DIN accepts serial configuration data synchronous with CCLK. During parallel configuration modes, DIN is the D0 input. During configuration, a pull-up is enabled, and after configuration, this pin is a user-programmable I/O pin.
M0, M1, M2	I	M0—M2 are used to select the configuration mode. See Table 8 for the configuration modes. During configuration, a pull-up is enabled, and after configuration, the pins are user-programmable I/O.
M3	I	M3 is used to select the speed of the internal oscillator during configuration. When M3 is low, the oscillator frequency is 10 MHz. When M3 is high, the oscillator is 1.25 MHz. During configuration, a pull-up is enabled, and after configuration, this pin is a user-programmable I/O pin.

Pin Information (continued)

Table 14. Pin Descriptions (continued)

Symbol	I/O	Description
TDI, TCK, TMS	I	If boundary scan is used, these pins are Test Data In, Test Clock, and Test Mode Select inputs. If boundary scan is not selected, all boundary-scan functions are inhibited once configuration is complete, and these pins are user-programmable I/O pins. Even if boundary scan is not used, either TCK or TMS must be held at logic 1 during configuration. Each pin has a pull-up enabled during configuration.
HDC	O	High During Configuration is output high until configuration is complete. It is used as a control output indicating that configuration is not complete. After configuration, this pin is a user-programmable I/O pin.
$\overline{\text{LDC}}$	O	$\overline{\text{Low During Configuration}}$ is output low until configuration is complete. It is used as a control output indicating that configuration is not complete. After configuration, this pin is a user-programmable I/O pin.
$\overline{\text{INIT}}$	I/O	$\overline{\text{INIT}}$ is a bidirectional signal before and during configuration. During configuration, a pull-up is enabled, but an external pull-up resistor is recommended. As an active-low open-drain output, $\overline{\text{INIT}}$ is held low during power stabilization and internal clearing of memory. As an active-low input, $\overline{\text{INIT}}$ holds the FPGA in the wait-state before the start of configuration. After configuration, the pin is a user-programmable I/O pin.
$\overline{\text{CS0}}$, CS1, $\overline{\text{WR}}$, $\overline{\text{RD}}$	I	$\overline{\text{CS0}}$, CS1, $\overline{\text{WR}}$, $\overline{\text{RD}}$ are used in the asynchronous peripheral configuration modes. The FPGA is selected when $\overline{\text{CS0}}$ is low and CS1 is high. When selected, a low on the write strobe, $\overline{\text{WR}}$, loads the data on D[7:0] inputs into an internal data buffer. $\overline{\text{WR}}$, $\overline{\text{CS0}}$, and CS1 are also used as chip selects in the slave parallel mode. A low on $\overline{\text{RD}}$ changes D7 into a status output. As a status indication, a high indicates ready and a low indicates busy. $\overline{\text{WR}}$ and $\overline{\text{RD}}$ should not be used simultaneously. If they are, the write strobe overrides. During configuration, a pull-up is enabled, and after configuration, the pins are user-programmable I/O pins.
A[17:0]	O	During master parallel configuration mode, A[17:0] address the configuration EPROM. During configuration, a pull-up is enabled, and after configuration, the pins are user-programmable I/O pins.
D[7:0]	I	During master parallel, peripheral, and slave parallel configuration modes, D[7:0] receive configuration data and each pin has a pull-up enabled. After configuration, the pins are user-programmable I/O pins.
DOUT	O	During configuration, DOUT is the serial data output that can drive the DIN of daisy-chained slave LCA devices. Data out on DOUT changes on the falling edge of CCLK. After configuration, DOUT is a user-programmable I/O pin.

Pin Information (continued)

Package Compatibility

The package pinouts are consistent across *ORCA* Series FPGAs. This allows a designer to select a package based on I/O requirements and change the FPGA without relaying out the printed-circuit board. The change might be to a larger FPGA, if additional functionality is needed, or a smaller FPGA to decrease unit cost.

Table 15 provides the number of user I/Os available for Lucent *ORCA* Series FPGAs for each available package. Each package has six dedicated configuration pins.

Tables 16—25 provide the package pin and pin function for the *ORCA* 2C Series FPGAs and packages. The bond pad name is identified in the PIC nomenclature used in the *ORCA* Foundry design editor.

When the number of FPGA bond pads exceeds the number of package pins, bond pads are unused. When the number of of package pins exceeds the number of bond pads, package pins are left unconnected (no connects).

For each package in the 2C Series, Tables 16—25 provide package pin functionality and the bond pad connection. When a package pin is to be left as a no connect for a specific die, it is indicated as a note in the bond pad column for the FPGA. The tables provide no information on unused pads.

Table 15. *ORCA* 2C Series FPGA I/Os Summary

Device	84-Pin PLCC	100-Pin TQFP	144-Pin TQFP	160-Pin MQFP	208-Pin SQFP/ SQFP-PQ2	240-Pin SQFP/ SQFP-PQ2	256-Pin PBGA	304-Pin SQFP/ SQFP-PQ2	364-Pin CPGA	428-Pin CPGA
ATT2C04										
User I/Os	64	77	114	130	160	—	—	—	—	—
VDD/VSS	14	17	24	24	31	—	—	—	—	—
ATT2C06										
User I/Os	64	77	114	130	171	192	—	—	—	—
VDD/VSS	14	17	24	24	31	42	—	—	—	—
ATT2C08										
User I/Os	64	—	—	130	171	192	221	224	—	—
VDD/VSS	14	—	—	24	31	40	27	46	—	—
ATT2C10										
User I/Os	64	—	—	130	171	192	223	252	—	—
VDD/VSS	14	—	—	24	31	40	27	46	—	—
ATT2C12										
User I/Os	—	—	—	—	171	192	223	252	288	—
VDD/VSS	—	—	—	—	31	42	27	46	38	—
ATT2C15										
User I/Os	—	—	—	—	171	192	—	252	320	—
VDD/VSS	—	—	—	—	31	42	—	46	38	—
ATT2C26										
User I/Os	—	—	—	—	171	192	—	252	—	384
VDD/VSS	—	—	—	—	31	42	—	46	—	38
ATT2C40										
User I/Os	—	—	—	—	171	192	—	252	—	384
VDD/VSS	—	—	—	—	31	42	—	46	—	38

Pin Information (continued)

Table 16. ATT2C04, ATT2C06, ATT2C08, and ATT2C10 84-Pin PLCC Pinout

Pin	2C04 Pad	2C06 Pad	2C08 Pad	2C10 Pad	Function	Pin	2C04 Pad	2C06 Pad	2C08 Pad	2C10 Pad	Function
1	VSS	VSS	VSS	VSS	VSS	43	VSS	VSS	VSS	VSS	VSS
2	PT5A	PT6A	PT7A	PT8A	I/O-D2	44	PB6A	PB7A	PB8A	PB9A	I/O
3	VSS	VSS	VSS	VSS	VSS	45	VSS	VSS	VSS	VSS	VSS
4	PT4D	PT5D	PT6D	PT7D	I/O-D1	46	PB7A	PB8A	PB9A	PB10A	I/O
5	PT4A	PT5A	PT6A	PT7A	I/O-D0/DIN	47	PB7D	PB8D	PB9D	PB10D	I/O
6	PT3A	PT4A	PT5A	PT6A	I/O-DOUT	48	PB8A	PB9A	PB10A	PB11A	I/O-HDC
7	PT2D	PT3D	PT4D	PT5D	I/O	49	PB9A	PB10A	PB11A	PB12A	I/O-$\overline{\text{LDC}}$
8	PT2A	PT3A	PT4A	PT4A	I/O-TDI	50	PB9D	PB10D	PB11D	PB13A	I/O
9	PT1D	PT2A	PT3A	PT3A	I/O-TMS	51	PB10A	PB11A	PB12C	PB13D	I/O-$\overline{\text{INIT}}$
10	PT1A	PT1A	PT1A	PT1A	I/O-TCK	52	PB10D	PB12A	PB13D	PB15D	I/O
11	RD_DATA /TDO	RD_DATA /TDO	RD_DATA /TDO	RD_DATA /TDO	RD_DATA /TDO	53	DONE	DONE	DONE	DONE	DONE
12	VDD	VDD	VDD	VDD	VDD	54	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$
13	VSS	VSS	VSS	VSS	VSS	55	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$
14	PL1C	PL1A	PL2D	PL2D	I/O-A0	56	PR10A	PR12A	PR14A	PR16A	I/O-M0
15	PL1A	PL2A	PL3A	PL3A	I/O-A1	57	PR10D	PR11A	PR12A	PR14A	I/O
16	PL2D	PL3D	PL4D	PL4A	I/O-A2	58	PR9A	PR10A	PR11A	PR13B	I/O-M1
17	PL2A	PL3A	PL4A	PL5A	I/O-A3	59	PR9D	PR10D	PR11D	PR12B	I/O
18	PL3A	PL4A	PL5A	PL6A	I/O-A4	60	PR8A	PR9A	PR10A	PR11A	I/O-M2
19	PL4D	PL5D	PL6D	PL7D	I/O-A5	61	PR7A	PR8A	PR9A	PR10A	I/O-M3
20	PL4A	PL5A	PL6A	PL7A	I/O-A6	62	PR7D	PR8D	PR9D	PR10D	I/O
21	PL5A	PL6A	PL7A	PL8A	I/O-A7	63	PR6A	PR7A	PR8D	PR9D	I/O
22	VDD	VDD	VDD	VDD	VDD	64	VDD	VDD	VDD	VDD	VDD
23	PL6A	PL7A	PL8A	PL9A	I/O-A8	65	PR5A	PR6A	PR7A	PR8A	I/O
24	VSS	VSS	VSS	VSS	VSS	66	VSS	VSS	VSS	VSS	VSS
25	PL7D	PL8D	PL9D	PL10D	I/O-A9	67	PR4A	PR5A	PR6A	PR7A	I/O
26	PL7A	PL8A	PL9A	PL10A	I/O-A10	68	PR4D	PR5D	PR6D	PR7D	I/O
27	PL8A	PL9A	PL10A	PL11A	I/O-A11	69	PR3A	PR4A	PR5A	PR6A	I/O-CS1
28	PL9D	PL10D	PL11D	PL12D	I/O-A12	70	PR2A	PR3A	PR4A	PR5A	I/O-$\overline{\text{CS0}}$
29	PL9A	PL10A	PL11A	PL13D	I/O-A13	71	PR2D	PR3D	PR4D	PR4D	I/O
30	PL10D	PL11A	PL12A	PL14C	I/O-A14	72	PR1A	PR2A	PR3A	PR3A	I/O-$\overline{\text{RD}}$
31	PL10A	PL12A	PL14A	PL16A	I/O-A15	73	PR1D	PR1A	PR2A	PR2A	I/O-$\overline{\text{WR}}$
32	CCLK	CCLK	CCLK	CCLK	CCLK	74	$\overline{\text{RD\_CFGN}}$	$\overline{\text{RD\_CFGN}}$	$\overline{\text{RD\_CFGN}}$	$\overline{\text{RD\_CFGN}}$	$\overline{\text{RD\_CFGN}}$
33	VDD	VDD	VDD	VDD	VDD	75	VDD	VDD	VDD	VDD	VDD
34	VSS	VSS	VSS	VSS	VSS	76	VSS	VSS	VSS	VSS	VSS
35	PB1A	PB1A	PB1A	PB1A	I/O-A16	77	PT10C	PT12A	PT13D	PT15D	I/O-RDY/RCLK
36	PB1D	PB2A	PB3A	PB3B	I/O-A17	78	PT9D	PT11A	PT12C	PT13D	I/O-D7
37	PB2A	PB3A	PB3D	PB4D	I/O	79	PT9C	PT10D	PT11D	PT13A	I/O
38	PB2D	PB3D	PB4D	PB5D	I/O	80	PT9A	PT10A	PT11B	PT12B	I/O-D6
39	PB3A	PB4A	PB5A	PB6A	I/O	81	PT8A	PT9A	PT10A	PT11A	I/O-D5
40	PB4A	PB5A	PB6A	PB7A	I/O	82	PT7D	PT8D	PT9D	PT10D	I/O
41	PB4D	PB5D	PB6D	PB7D	I/O	83	PT7A	PT8A	PT9A	PT10A	I/O-D4
42	PB5A	PB6A	PB7A	PB8A	I/O	84	PT6A	PT7A	PT8A	PT9A	I/O-D3

Pin Information (continued)

Table 17. ATT2C04 and ATT2C06 100-Pin TQFP Pinout

Pin	2C04 Pad	2C06 Pad	Function	Pin	2C04 Pad	2C06 Pad	Function
1	VDD	VDD	VDD	43	PB8C	PB9C	I/O
2	VSS	VSS	VSS	44	PB8D	PB9D	I/O
3	PL1C	PL1A	I/O-A0	45	PB9A	PB10A	I/O-$\overline{LDC}$
4	PL1A	PL2A	I/O-A1	46	PB9D	PB10D	I/O
5	PL2D	PL3D	I/O-A2	47	PB10A	PB11A	I/O-$\overline{INIT}$
6	PL2A	PL3A	I/O-A3	48	PB10D	PB12A	I/O
7	PL3D	PL4D	I/O	49	DONE	DONE	DONE
8	PL3A	PL4A	I/O-A4	50	VDD	VDD	VDD
9	PL4D	PL5D	I/O-A5	51	$\overline{RESET}$	$\overline{RESET}$	$\overline{RESET}$
10	PL4A	PL5A	I/O-A6	52	$\overline{PRGM}$	$\overline{PRGM}$	$\overline{PRGM}$
11	PL5D	PL6D	I/O	53	PR10A	PR12A	I/O-M0
12	PL5A	PL6A	I/O-A7	54	PR10D	PR11A	I/O
13	VDD	VDD	VDD	55	PR9A	PR10A	I/O-M1
14	PL6A	PL7A	I/O-A8	56	PR9D	PR10D	I/O
15	VSS	VSS	VSS	57	PR8A	PR9A	I/O-M2
16	PL7D	PL8D	I/O-A9	58	PR8D	PR9D	I/O
17	PL7A	PL8A	I/O-A10	59	PR7A	PR8A	I/O-M3
18	PL8A	PL9A	I/O-A11	60	PR7D	PR8D	I/O
19	PL9D	PL10D	I/O-A12	61	VSS	VSS	VSS
20	PL9C	PL10C	I/O	62	PR6A	PR7A	I/O
21	PL9A	PL10A	I/O-A13	63	VDD	VDD	VDD
22	PL10D	PL11A	I/O-A14	64	PR5A	PR6A	I/O
23	PL10A	PL12A	I/O-A15	65	VSS	VSS	VSS
24	VSS	VSS	VSS	66	PR4A	PR5A	I/O
25	CCLK	CCLK	CCLK	67	PR4D	PR5D	I/O
26	VDD	VDD	VDD	68	PR3A	PR4A	I/O-CS1
27	VSS	VSS	VSS	69	PR3D	PR4D	I/O
28	PB1A	PB1A	I/O-A16	70	PR2A	PR3A	I/O-$\overline{CS0}$
29	PB1C	PB1D	I/O	71	PR2D	PR3D	I/O
30	PB1D	PB2A	I/O-A17	72	PR1A	PR2A	I/O-$\overline{RD}$
31	PB2A	PB3A	I/O	73	PR1C	PR2D	I/O
32	PB2D	PB3D	I/O	74	PR1D	PR1A	I/O-$\overline{WR}$
33	PB3A	PB4A	I/O	75	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$
34	PB4A	PB5A	I/O	76	VDD	VDD	VDD
35	PB4D	PB5D	I/O	77	VSS	VSS	VSS
36	PB5A	PB6A	I/O	78	PT10C	PT12A	I/O-RDY/RCLK
37	VSS	VSS	VSS	79	PT9D	PT11A	I/O-D7
38	PB6A	PB7A	I/O	80	PT9C	PT10D	I/O
39	VSS	VSS	VSS	81	PT9A	PT10A	I/O-D6
40	PB7A	PB8A	I/O	82	PT8D	PT9D	I/O
41	PB7D	PB8D	I/O	83	PT8A	PT9A	I/O-D5
42	PB8A	PB9A	I/O-HDC	84	PT7D	PT8D	I/O

Pin Information (continued)

Table 17. ATT2C04 and ATT2C06 100-Pin TQFP Pinout (continued)

Pin	2C04 Pad	2C06 Pad	Function	Pin	2C04 Pad	2C06 Pad	Function
85	PT7A	PT8A	I/O-D4	93	PT3D	PT4D	I/O
86	PT6D	PT7D	I/O	94	PT3A	PT4A	I/O-DOUT
87	PT6A	PT7A	I/O-D3	95	PT2D	PT3D	I/O
88	VSS	VSS	VSS	96	PT2A	PT3A	I/O-TDI
89	PT5A	PT6A	I/O-D2	97	PT1D	PT2A	I/O-TMS
90	VSS	VSS	VSS	98	PT1C	PT1D	I/O
91	PT4D	PT5D	I/O-D1	99	PT1A	PT1A	I/O-TCK
92	PT4A	PT5A	I/O-D0/DIN	100	RD_DATA/TDO	RD_DATA/TDO	RD_DATA/TDO

Pin Information (continued)

Table 18. ATT2C04 and ATT2C06 144-Pin TQFP Pinout

Pin	2C04 Pad	2C06 Pad	Function	Pin	2C04 Pad	2C06 Pad	Function
1	VDD	VDD	VDD	43	PB2B	PB3B	I/O
2	VSS	VSS	VSS	44	PB2D	PB3D	I/O
3	PL1C	PL1A	I/O-A0	45	VDD	VDD	VDD
4	PL1B	PL2D	I/O	46	PB3A	PB4A	I/O
5	PL1A	PL2A	I/O-A1	47	PB3D	PB4D	I/O
6	PL2D	PL3D	I/O-A2	48	PB4A	PB5A	I/O
7	PL2A	PL3A	I/O-A3	49	PB4C	PB5C	I/O
8	PL3D	PL4D	I/O	50	PB4D	PB5D	I/O
9	PL3C	PL4C	I/O	51	PB5A	PB6A	I/O
10	PL3A	PL4A	I/O-A4	52	PB5C	PB6C	I/O
11	PL4D	PL5D	I/O-A5	53	PB5D	PB6D	I/O
12	PB4C	PB5C	I/O	54	VSS	VSS	VSS
13	PL4A	PL5A	I/O-A6	55	PB6A	PB7A	I/O
14	VSS	VSS	VSS	56	PB6C	PB7C	I/O
15	PL5D	PL6D	I/O	57	PB6D	PB7D	I/O
16	PL5C	PL6C	I/O	58	PB7A	PB8A	I/O
17	PL5A	PL6A	I/O-A7	59	PB7D	PB8D	I/O
18	VDD	VDD	VDD	60	PB8A	PB9A	I/O-HDC
19	PL6D	PL7D	I/O	61	PB8C	PB9C	I/O
20	PL6C	PL7C	I/O	62	PB8D	PB9D	I/O
21	PL6A	PL7A	I/O-A8	63	VDD	VDD	VDD
22	VSS	VSS	VSS	64	PB9A	PB10A	I/O-$\overline{\text{LDC}}$
23	PL7D	PL8D	I/O-A9	65	PB9C	PB10C	I/O
24	PL7A	PL8A	I/O-A10	66	PB9D	PB10D	I/O
25	PL8D	PL9D	I/O	67	PB10A	PB11A	I/O-$\overline{\text{INIT}}$
26	PL8C	PL9C	I/O	68	PB10C	PB11D	I/O
27	PL8A	PL9A	I/O-A11	69	PB10D	PB12A	I/O
28	PL9D	PL10D	I/O-A12	70	VSS	VSS	VSS
29	PB9C	PB10C	I/O	71	DONE	DONE	DONE
30	PL9A	PL10A	I/O-A13	72	VDD	VDD	VDD
31	PL10D	PL11A	I/O-A14	73	VSS	VSS	VSS
32	PL10C	PL12D	I/O	74	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$
33	PL10B	PL12B	I/O	75	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$
34	PL10A	PL12A	I/O-A15	76	PR10A	PR12A	I/O-M0
35	VSS	VSS	VSS	77	PR10B	PR12D	I/O
36	CCLK	CCLK	CCLK	78	PR10D	PR11A	I/O
37	VDD	VDD	VDD	79	PR9A	PR10A	I/O-M1
38	VSS	VSS	VSS	80	PR9C	PR10C	I/O
39	PB1A	PB1A	I/O-A16	81	PR9D	PR10D	I/O
40	PB1C	PB1D	I/O	82	PR8A	PR9A	I/O-M2
41	PB1D	PB2A	I/O-A17	83	PR8B	PR9B	I/O
42	PB2A	PB3A	I/O	84	PR8D	PR9D	I/O

Pin Information (continued)

Table 18. ATT2C04 and ATT2C06 144-Pin TQFP Pinout (continued)

Pin	2C04 Pad	2C06 Pad	Function	Pin	2C04 Pad	2C06 Pad	Function
85	PR7A	PR8A	I/O-M3	115	PT9C	PT10D	I/O
86	PR7D	PR8D	I/O	116	PT9B	PT10C	I/O
87	VSS	VSS	VSS	117	PT9A	PT10A	I/O-D6
88	PR6A	PR7A	I/O	118	VDD	VDD	VDD
89	PR6C	PR7C	I/O	119	PT8D	PT9D	I/O
90	PR6D	PR7D	I/O	120	PT8A	PT9A	I/O-D5
91	VDD	VDD	VDD	121	PT7D	PT8D	I/O
92	PR5A	PR6A	I/O	122	PT7B	PT8B	I/O
93	PR5C	PR6C	I/O	123	PT7A	PT8A	I/O-D4
94	PR5D	PR6D	I/O	124	PT6D	PT7D	I/O
95	VSS	VSS	VSS	125	PT6C	PT7C	I/O
96	PR4A	PR5A	I/O	126	PT6A	PT7A	I/O-D3
97	PR4C	PR5C	I/O	127	VSS	VSS	VSS
98	PR4D	PR5D	I/O	128	PT5D	PT6D	I/O
99	PR3A	PR4A	I/O-CS1	129	PT5C	PT6C	I/O
100	PR3D	PR4D	I/O	130	PT5A	PT6A	I/O-D2
101	PR2A	PR3A	I/O-$\overline{CS0}$	131	PT4D	PT5D	I/O-D1
102	PR2D	PR3D	I/O	132	PT4C	PT5C	I/O
103	PR1A	PR2A	I/O-$\overline{RD}$	133	PT4A	PT5A	I/O-D0/DIN
104	PR1B	PR2B	I/O	134	PT3D	PT4D	I/O
105	PR1C	PR2D	I/O	135	PT3A	PT4A	I/O-DOUT
106	PR1D	PR1A	I/O-$\overline{WR}$	136	VDD	VDD	VDD
107	VSS	VSS	VSS	137	PT2D	PT3D	I/O
108	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	138	PT2C	PT3C	I/O
109	VDD	VDD	VDD	139	PT2A	PT3A	I/O-TDI
110	VSS	VSS	VSS	140	PT1D	PT2A	I/O-TMS
111	PT10D	PT10D	I/O	141	PT1C	PT1D	I/O
112	PT10C	PT12A	I/O-RDY/RCLK	142	PT1A	PT1A	I/O-TCK
113	PT10B	PT11D	I/O	143	VSS	VSS	VSS
114	PT9D	PT11A	I/O-D7	144	RD_DATA/ TDO	RD_DATA/ TDO	RD_DATA/TDO

Pin Information (continued)

Table 19. ATT2C04, ATT2C06, ATT2C08, and ATT2C10 160-Pin QFP Pinout

Pin	2C04 Pad	2C06 Pad	2C08 Pad	2C10 Pad	Function
1	VDD	VDD	VDD	VDD	VDD
2	VSS	VSS	VSS	VSS	VSS
3	PL1D	PL1D	PL1D	PL1D	I/O
4	PL1C	PL1A	PL2D	PL2D	I/O-A0
5	PL1B	PL2D	PL3D	PL3D	I/O
6	PL1A	PL2A	PL3A	PL3A	I/O-A1
7	PL2D	PL3D	PL4D	PL4A	I/O-A2
8	PL2C	PL3C	PL4C	PL5C	I/O
9	PL2A	PL3A	PL4A	PL5A	I/O-A3
10	PL3D	PL4D	PL5D	PL6D	I/O
11	PL3C	PL4C	PL5C	PL6C	I/O
12	PL3A	PL4A	PL5A	PL6A	I/O-A4
13	PL4D	PL5D	PL6D	PL7D	I/O-A5
14	PL4C	PL5C	PL6C	PL7C	I/O
15	PL4A	PL5A	PL6A	PL7A	I/O-A6
16	VSS	VSS	VSS	VSS	VSS
17	PL5D	PL6D	PL7D	PL8D	I/O
18	PL5C	PL6C	PL7C	PL8C	I/O
19	PL5A	PL6A	PL7A	PL8A	I/O-A7
20	VDD	VDD	VDD	VDD	VDD
21	PL6D	PL7D	PL8D	PL9D	I/O
22	PL6C	PL7C	PL8C	PL9C	I/O
23	PL6A	PL7A	PL8A	PL9A	I/O-A8
24	VSS	VSS	VSS	VSS	VSS
25	PL7D	PL8D	PL9D	PL10D	I/O-A9
26	PL7B	PL8B	PL9B	PL10B	I/O
27	PL7A	PL8A	PL9A	PL10A	I/O-A10
28	PL8D	PL9D	PL10D	PL11D	I/O
29	PL8C	PL9C	PL10C	PL11C	I/O
30	PL8A	PL9A	PL10A	PL11A	I/O-A11
31	PL9D	PL10D	PL11D	PL12D	I/O-A12
32	PL9C	PL10C	PL11C	PL12C	I/O
33	PL9B	PL10B	PL11B	PL12B	I/O
34	PL9A	PL10A	PL11A	PL13D	I/O-A13
35	PL10D	PL11A	PL12A	PL14C	I/O-A14
36	PL10C	PL12D	PL13D	PL15D	I/O
37	PL10B	PL12B	PL14D	PL16D	I/O
38	PL10A	PL12A	PL14A	PL16A	I/O-A15
39	CCLK	CCLK	CCLK	CCLK	CCLK
40	VSS	VSS	VSS	VSS	VSS
41	VDD	VDD	VDD	VDD	VDD
42	VSS	VSS	VSS	VSS	VSS
43	PB1A	PB1A	PB1A	PB1A	I/O-A16

Pin Information (continued)

Table 19. ATT2C04, ATT2C06, ATT2C08, and ATT2C10 160-Pin QFP Pinout (continued)

Pin	2C04 Pad	2C06 Pad	2C08 Pad	2C10 Pad	Function
44	PB1B	PB1C	PB2A	PB2A	I/O
45	PB1C	PB1D	PB2D	PB2D	I/O
46	PB1D	PB2A	PB3A	PB3B	I/O-A17
47	PB2A	PB3A	PB3D	PB4D	I/O
48	PB2B	PB3B	PB4A	PB5A	I/O
49	PB2C	PB3C	PB4C	PB5C	I/O
50	PB2D	PB3D	PB4D	PB5D	I/O
51	VDD	VDD	VDD	VDD	VDD
52	PB3A	PB4A	PB5A	PB6A	I/O
53	PB3D	PB4D	PB5D	PB6D	I/O
54	PB4A	PB5A	PB6A	PB7A	I/O
55	PB4C	PB5C	PB6C	PB7C	I/O
56	PB4D	PB5D	PB6D	PB7D	I/O
57	PB5A	PB6A	PB7A	PB8A	I/O
58	PB5C	PB6C	PB7C	PB8C	I/O
59	PB5D	PB6D	PB7D	PB8D	I/O
60	VSS	VSS	VSS	VSS	VSS
61	PB6A	PB7A	PB8A	PB9A	I/O
62	PB6C	PB7C	PB8C	PB9C	I/O
63	PB6D	PB7D	PB8D	PB9D	I/O
64	PB7A	PB8A	PB9A	PB10A	I/O
65	PB7D	PB8D	PB9D	PB10D	I/O
66	PB8A	PB9A	PB10A	PB11A	I/O-HDC
67	PB8C	PB9C	PB10C	PB11C	I/O
68	PB8D	PB9D	PB10D	PB11D	I/O
69	VDD	VDD	VDD	VDD	VDD
70	PB9A	PB10A	PB11A	PB12A	I/O-$\overline{\text{LDC}}$
71	PB9B	PB10B	PB11D	PB13A	I/O
72	PB9C	PB10C	PB12A	PB13B	I/O
73	PB9D	PB10D	PB12B	PB13C	I/O
74	PB10A	PB11A	PB12C	PB13D	I/O-$\overline{\text{INIT}}$
75	PB10B	PB11C	PB12D	PB14A	I/O
76	PB10C	PB11D	PB13D	PB15D	I/O
77	PB10D	PB12A	PB14D	PB16D	I/O
78	VSS	VSS	VSS	VSS	VSS
79	DONE	DONE	DONE	DONE	DONE
80	VDD	VDD	VDD	VDD	VDD
81	VSS	VSS	VSS	VSS	VSS
82	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$
83	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$
84	PR10A	PR12A	PR14A	PR16A	I/O-M0
85	PR10B	PR12D	PR13A	PR15A	I/O

Pin Information (continued)

Table 19. ATT2C04, ATT2C06, ATT2C08, and ATT2C10 160-Pin QFP Pinout (continued)

Pin	2C04 Pad	2C06 Pad	2C08 Pad	2C10 Pad	Function
86	PR10C	PR11A	PR13D	PR15D	I/O
87	PR10D	PR11B	PR12A	PR14A	I/O
88	PR9A	PR10A	PR11A	PR13B	I/O-M1
89	PR9B	PR10B	PR11B	PR13C	I/O
90	PR9C	PR10C	PR11C	PR12A	I/O
91	PR9D	PR10D	PR11D	PR12B	I/O
92	PR8A	PR9A	PR10A	PR11A	I/O-M2
93	PR8B	PR9B	PR10B	PR11B	I/O
94	PR8D	PR9D	PR10D	PR11D	I/O
95	PR7A	PR8A	PR9A	PR10A	I/O-M3
96	PR7D	PR8D	PR9D	PR10D	I/O
97	VSS	VSS	VSS	VSS	VSS
98	PR6A	PR7A	PR8A	PR9A	I/O
99	PR6C	PR7C	PR8C	PR9C	I/O
100	PR6D	PR7D	PR8D	PR9D	I/O
101	VDD	VDD	VDD	VDD	VDD
102	PR5A	PR6A	PR7A	PR8A	I/O
103	PR5C	PR6C	PR7C	PR8C	I/O
104	PR5D	PR6D	PR7D	PR8D	I/O
105	VSS	VSS	VSS	VSS	VSS
106	PR4A	PR5A	PR6A	PR7A	I/O
107	PR4C	PR5C	PR6C	PR7C	I/O
108	PR4D	PR5D	PR6D	PR7D	I/O
109	PR3A	PR4A	PR5A	PR6A	I/O-CS1
110	PR3B	PR4B	PR5B	PR6B	I/O
111	PR3D	PR4D	PR5D	PR6D	I/O
112	PR2A	PR3A	PR4A	PR5A	I/O-$\overline{CS0}$
113	PR2C	PR3C	PR4B	PR4B	I/O
114	PR2D	PR3D	PR4D	PR4D	I/O
115	PR1A	PR2A	PR3A	PR3A	I/O-$\overline{RD}$
116	PR1B	PR2C	PR3C	PR3C	I/O
117	PR1C	PR2D	PR3D	PR3D	I/O
118	PR1D	PR1A	PR2A	PR2A	I/O-$\overline{WR}$
119	VSS	VSS	VSS	VSS	VSS
120	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$
121	VDD	VDD	VDD	VDD	VDD
122	VSS	VSS	VSS	VSS	VSS
123	PT10D	PT12D	PT14D	PT16D	I/O
124	PT10C	PT12A	PT13D	PT15D	I/O-RDY/RCLK
125	PT10B	PT11D	PT13A	PT15A	I/O
126	PT10A	PT11C	PT12D	PT14D	I/O
127	PT9D	PT11A	PT12C	PT13D	I/O-D7

Pin Information (continued)

Table 19. ATT2C04, ATT2C06, ATT2C08, and ATT2C10 160-Pin QFP Pinout (continued)

Pin	2C04 Pad	2C06 Pad	2C08 Pad	2C10 Pad	Function
128	PT9C	PT10D	PT12A	PT13B	I/O
129	PT9B	PT10C	PT11D	PT13A	I/O
130	PT9A	PT10A	PT11B	PT12B	I/O-D6
131	VDD	VDD	VDD	VDD	VDD
132	PT8D	PT9D	PT10D	PT11D	I/O
133	PT8A	PT9A	PT10A	PT11A	I/O-D5
134	PT7D	PT8D	PT9D	PT10D	I/O
135	PT7B	PT8B	PT9B	PT10B	I/O
136	PT7A	PT8A	PT9A	PT10A	I/O-D4
137	PT6D	PT7D	PT8D	PT9D	I/O
138	PT6C	PT7C	PT8C	PT9C	I/O
139	PT6A	PT7A	PT8A	PT9A	I/O-D3
140	VSS	VSS	VSS	VSS	VSS
141	PT5D	PT6D	PT7D	PT8D	I/O
142	PT5C	PT6C	PT7C	PT8C	I/O
143	PT5A	PT6A	PT7A	PT8A	I/O-D2
144	PT4D	PT5D	PT6D	PT7D	I/O-D1
145	PT4C	PT5C	PT6C	PT7C	I/O
146	PT4A	PT5A	PT6A	PT7A	I/O-D0/DIN
147	PT3D	PT4D	PT5D	PT6D	I/O
148	PT3C	PT4C	PT5C	PT6C	I/O
149	PT3A	PT4A	PT5A	PT6A	I/O-DOUT
150	VDD	VDD	VDD	VDD	VDD
151	PT2D	PT3D	PT4D	PT5D	I/O
152	PT2C	PT3C	PT4C	PT5A	I/O
153	PT2B	PT3B	PT4B	PT4D	I/O
154	PT2A	PT3A	PT4A	PT4A	I/O-TDI
155	PT1D	PT2A	PT3A	PT3A	I/O-TMS
156	PT1C	PT1D	PT2A	PT2A	I/O
157	PT1B	PT1C	PT1D	PT1D	I/O
158	PT1A	PT1A	PT1A	PT1A	I/O-TCK
159	VSS	VSS	VSS	VSS	VSS
160	RD_DATA/TDO	RD_DATA/TDO	RD_DATA/TDO	RD_DATA/TDO	RD_DATA/TDO

Pin Information (continued)

Table 20. ATT2C04, ATT2C06, ATT2C08, ATT2C10, ATT2C12, ATT2C15, ATT2C26, and ATT2C40 208-Pin SQFP/SQFP-PQ2 Pinout

Pin	2C04 Pad	2C06 Pad	2C08 Pad	2C10 Pad	2C12 Pad	2C15 Pad	2C26 Pad	2C40 Pad	Function
1	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
2	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
3	PL1D	PL1D	PL1D	PL1D	PL1D	PL1D	PL1D	PL1D	I/O
4	PL1C	PL1A	PL2D	PL2D	PL2D	PL2D	PL2D	PL3D	I/O-A0
5	PL1B	PL2D	PL3D	PL3D	PL3D	PL4D	PL4D	PL5D	I/O
6	See Note	PL2C	PL3C	PL3C	PL3A	PL4A	PL4A	PL6D	I/O
7	PL1A	PL2A	PL3A	PL3A	PL4A	PL5A	PL5A	PL8D	I/O-A1
8	PL2D	PL3D	PL4D	PL4A	PL5A	PL6A	PL6A	PL9A	I/O-A2
9	PL2C	PL3C	PL4C	PL5C	PL6D	PL7D	PL7D	PL10D	I/O
10	PL2B	PL3B	PL4B	PL5B	PL6B	PL7B	PL7B	PL10B	I/O
11	PL2A	PL3A	PL4A	PL5A	PL6A	PL7A	PL7A	PL10A	I/O-A3
12	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
13	PL3D	PL4D	PL5D	PL6D	PL7D	PL8D	PL8D	PL11D	I/O
14	PL3C	PL4C	PL5C	PL6C	PL7C	PL8C	PL8A	PL11A	I/O
15	PL3B	PL4B	PL5B	PL6B	PL7B	PL8B	PL9D	PL12D	I/O
16	PL3A	PL4A	PL5A	PL6A	PL7A	PL8A	PL9A	PL12A	I/O-A4
17	PL4D	PL5D	PL6D	PL7D	PL8D	PL9D	PL10D	PL13D	I/O-A5
18	PL4C	PL5C	PL6C	PL7C	PL8C	PL9C	PL10A	PL13A	I/O
19	PL4B	PL5B	PL6B	PL7B	PL8B	PL9B	PL11D	PL14D	I/O
20	PL4A	PL5A	PL6A	PL7A	PL8A	PL9A	PL11A	PL14A	I/O-A6
21	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
22	PL5D	PL6D	PL7D	PL8D	PL9D	PL10D	PL12D	PL15D	I/O
23	PL5C	PL6C	PL7C	PL8C	PL9C	PL10C	PL12C	PL15C	I/O
24	PL5B	PL6B	PL7B	PL8B	PL9B	PL10B	PL12B	PL15B	I/O
25	PL5A	PL6A	PL7A	PL8A	PL9A	PL10A	PL12A	PL15A	I/O-A7
26	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
27	PL6D	PL7D	PL8D	PL9D	PL10D	PL11D	PL13D	PL16D	I/O
28	PL6C	PL7C	PL8C	PL9C	PL10C	PL11C	PL13C	PL16C	I/O
29	PL6B	PL7B	PL8B	PL9B	PL10B	PL11B	PL13B	PL16B	I/O
30	PL6A	PL7A	PL8A	PL9A	PL10A	PL11A	PL13A	PL16A	I/O-A8
31	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
32	PL7D	PL8D	PL9D	PL10D	PL11D	PL12D	PL14D	PL17D	I/O-A9
33	PL7C	PL8C	PL9C	PL10C	PL11C	PL12C	PL14A	PL17A	I/O
34	PL7B	PL8B	PL9B	PL10B	PL11B	PL12B	PL15D	PL18D	I/O
35	PL7A	PL8A	PL9A	PL10A	PL11A	PL12A	PL15A	PL18A	I/O-A10
36	PL8D	PL9D	PL10D	PL11D	PL12D	PL13D	PL16D	PL19D	I/O
37	PL8C	PL9C	PL10C	PL11C	PL12C	PL13C	PL16A	PL19A	I/O
38	PL8B	PL9B	PL10B	PL11B	PL12B	PL13B	PL17D	PL20D	I/O

Note: The ATT2C04 does not have bond pads connected to 208-pin SQFP package pin numbers 6, 45, 47, 56, 60, 102, 153, 154, 166, 201, and 203.

Pin Information (continued)

Table 20. ATT2C04, ATT2C06, ATT2C08, ATT2C10, ATT2C12, ATT2C15, ATT2C26, and ATT2C40 208-Pin SQFP/SQFP-PQ2 Pinout (continued)

Pin	2C04 Pad	2C06 Pad	2C08 Pad	2C10 Pad	2C12 Pad	2C15 Pad	2C26 Pad	2C40 Pad	Function
39	PL8A	PL9A	PL10A	PL11A	PL12A	PL13A	PL17A	PL20A	I/O-A11
40	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
41	PL9D	PL10D	PL11D	PL12D	PL13D	PL14D	PL18D	PL21D	I/O-A12
42	PL9C	PL10C	PL11C	PL12C	PL13B	PL14B	PL18B	PL21B	I/O
43	PL9B	PL10B	PL11B	PL12B	PL14D	PL15D	PL19D	PL22D	I/O
44	PL9A	PL10A	PL11A	PL13D	PL14B	PL15B	PL19B	PL22B	I/O-A13
45	See Note	PL11D	PL12D	PL13B	PL15D	PL16D	PL20D	PL23D	I/O
46	PL10D	PL11A	PL12A	PL14C	PL16D	PL17D	PL21D	PL25A	I/O-A14
47	See Note	PL12D	PL13D	PL15D	PL17D	PL18D	PL22D	PL27D	I/O
48	PL10C	PL12C	PL13A	PL15A	PL17A	PL19D	PL23D	PL28D	I/O
49	PL10B	PL12B	PL14D	PL16D	PL18C	PL19A	PL23A	PL28A	I/O
50	PL10A	PL12A	PL14A	PL16A	PL18A	PL20A	PL24A	PL30A	I/O-A15
51	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
52	CCLK	CCLK	CCLK	CCLK	CCLK	CCLK	CCLK	CCLK	CCLK
53	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
54	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
55	PB1A	PB1A	PB1A	PB1A	PB1A	PB1A	PB1A	PB1A	I/O-A16
56	See Note	PB1B	PB1D	PB1D	PB1D	PB2A	PB2A	PB3A	I/O
57	PB1B	PB1C	PB2A	PB2A	PB2A	PB2D	PB2D	PB3D	I/O
58	PB1C	PB1D	PB2D	PB2D	PB2D	PB3D	PB3D	PB4D	I/O
59	PB1D	PB2A	PB3A	PB3B	PB3D	PB4D	PB4D	PB5D	I/O-A17
60	See Note	PB2D	PB3D	PB4D	PB4D	PB5D	PB5D	PB6D	I/O
61	PB2A	PB3A	PB4A	PB5A	PB5B	PB6B	PB6B	PB7D	I/O
62	PB2B	PB3B	PB4B	PB5B	PB5D	PB6D	PB6D	PB8D	I/O
63	PB2C	PB3C	PB4C	PB5C	PB6B	PB7B	PB7B	PB9D	I/O
64	PB2D	PB3D	PB4D	PB5D	PB6D	PB7D	PB7D	PB10D	I/O
65	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
66	PB3A	PB4A	PB5A	PB6A	PB7A	PB8A	PB8A	PB11A	I/O
67	PB3B	PB4B	PB5B	PB6B	PB7B	PB8B	PB8D	PB11D	I/O
68	PB3C	PB4C	PB5C	PB6C	PB7C	PB8C	PB9A	PB12A	I/O
69	PB3D	PB4D	PB5D	PB6D	PB7D	PB8D	PB9D	PB12D	I/O
70	PB4A	PB5A	PB6A	PB7A	PB8A	PB9A	PB10A	PB13A	I/O
71	PB4B	PB5B	PB6B	PB7B	PB8B	PB9B	PB10D	PB13D	I/O
72	PB4C	PB5C	PB6C	PB7C	PB8C	PB9C	PB11A	PB14A	I/O
73	PB4D	PB5D	PB6D	PB7D	PB8D	PB9D	PB11D	PB14D	I/O
74	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS

Note: The ATT2C04 does not have bond pads connected to 208-pin SQFP package pin numbers 6, 45, 47, 56, 60, 102, 153, 154, 166, 201, and 203.

Pin Information (continued)

Table 20. ATT2C04, ATT2C06, ATT2C08, ATT2C10, ATT2C12, ATT2C15, ATT2C26, and ATT2C40 208-Pin SQFP/SQFP-PQ2 Pinout (continued)

Pin	2C04 Pad	2C06 Pad	2C08 Pad	2C10 Pad	2C12 Pad	2C15 Pad	2C26 Pad	2C40 Pad	Function
75	PB5A	PB6A	PB7A	PB8A	PB9A	PB10A	PB12A	PB15A	I/O
76	PB5B	PB6B	PB7B	PB8B	PB9B	PB10B	PB12B	PB15B	I/O
77	PB5C	PB6C	PB7C	PB8C	PB9C	PB10C	PB12C	PB15C	I/O
78	PB5D	PB6D	PB7D	PB8D	PB9D	PB10D	PB12D	PB15D	I/O
79	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
80	PB6A	PB7A	PB8A	PB9A	PB10A	PB11A	PB13A	PB16A	I/O
81	PB6B	PB7B	PB8B	PB9B	PB10B	PB11B	PB13B	PB16B	I/O
82	PB6C	PB7C	PB8C	PB9C	PB10C	PB11C	PB13C	PB16C	I/O
83	PB6D	PB7D	PB8D	PB9D	PB10D	PB11D	PB13D	PB16D	I/O
84	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
85	PB7A	PB8A	PB9A	PB10A	PB11A	PB12A	PB14A	PB17A	I/O
86	PB7B	PB8B	PB9B	PB10B	PB11B	PB12B	PB14D	PB17D	I/O
87	PB7C	PB8C	PB9C	PB10C	PB11C	PB12C	PB15A	PB18A	I/O
88	PB7D	PB8D	PB9D	PB10D	PB11D	PB12D	PB15D	PB18D	I/O
89	PB8A	PB9A	PB10A	PB11A	PB12A	PB13A	PB16A	PB19A	I/O-HDC
90	PB8B	PB9B	PB10B	PB11B	PB12B	PB13B	PB16D	PB19D	I/O
91	PB8C	PB9C	PB10C	PB11C	PB12C	PB13C	PB17A	PB20A	I/O
92	PB8D	PB9D	PB10D	PB11D	PB12D	PB13D	PB17D	PB20D	I/O
93	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
94	PB9A	PB10A	PB11A	PB12A	PB13A	PB14A	PB18A	PB21A	I/O-$\overline{\text{LDC}}$
95	PB9B	PB10B	PB11D	PB13A	PB13D	PB14D	PB18D	PB22D	I/O
96	PB9C	PB10C	PB12A	PB13B	PB14A	PB15A	PB19A	PB23A	I/O
97	PB9D	PB10D	PB12B	PB13C	PB14D	PB15D	PB19D	PB24D	I/O
98	PB10A	PB11A	PB12C	PB13D	PB15A	PB16A	PB20A	PB25A	I/O-$\overline{\text{INIT}}$
99	PB10B	PB11C	PB12D	PB14A	PB16A	PB17A	PB21A	PB26A	I/O
100	PB10C	PB11D	PB13A	PB15A	PB17A	PB18A	PB22A	PB1A	I/O
101	PB10D	PB12A	PB13D	PB15D	PB18A	PB19D	PB23D	PB28D	I/O
102	See Note	PB12D	PB14D	PB16D	PB18D	PB20D	PB24D	PB30D	I/O
103	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
104	DONE	DONE	DONE	DONE	DONE	DONE	DONE	DONE	DONE
105	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
106	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$
107	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$
108	PR10A	PR12A	PR14A	PR16A	PR18A	PR20A	PR24A	PR30A	I/O-M0
109	PR10B	PR12D	PR13A	PR15A	PR18D	PR19A	PR23A	PR28A	I/O
110	PR10C	PR11A	PR13D	PR15D	PR17B	PR18A	PR22A	PR27A	I/O

Note: The ATT2C04 does not have bond pads connected to 208-pin SQFP package pin numbers 6, 45, 47, 56, 60, 102, 153, 154, 166, 201, and 203.

Pin Information (continued)

Table 20. ATT2C04, ATT2C06, ATT2C08, ATT2C10, ATT2C12, ATT2C15, ATT2C26, and ATT2C40 208-Pin SQFP/SQFP-PQ2 Pinout (continued)

Pin	2C04 Pad	2C06 Pad	2C08 Pad	2C10 Pad	2C12 Pad	2C15 Pad	2C26 Pad	2C40 Pad	Function
111	PR10D	PR11B	PR12A	PR14A	PR16A	PR17A	PR21A	PR26A	I/O
112	PR9A	PR10A	PR11A	PR13B	PR15D	PR16D	PR20D	PR23D	I/O-M1
113	PR9B	PR10B	PR11B	PR13C	PR14A	PR15A	PR19A	PR22A	I/O
114	PR9C	PR10C	PR11C	PR12A	PR14D	PR15D	PR19D	PR22D	I/O
115	PR9D	PR10D	PR11D	PR12B	PR13A	PR14A	PR18A	PR21A	I/O
116	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
117	PR8A	PR9A	PR10A	PR11A	PR12A	PR13A	PR17A	PR20A	I/O-M2
118	PR8B	PR9B	PR10B	PR11B	PR12B	PR13B	PR17D	PR20D	I/O
119	PR8C	PR9C	PR10C	PR11C	PR12C	PR13C	PR16A	PR19A	I/O
120	PR8D	PR9D	PR10D	PR11D	PR12D	PR13D	PR16D	PR19D	I/O
121	PR7A	PR8A	PR9A	PR10A	PR11A	PR12A	PR15A	PR18A	I/O-M3
122	PR7B	PR8B	PR9B	PR10B	PR11B	PR12B	PR15D	PR18D	I/O
123	PR7C	PR8C	PR9C	PR10C	PR11C	PR12C	PR14A	PR17A	I/O
124	PR7D	PR8D	PR9D	PR10D	PR11D	PR12D	PR14D	PR17D	I/O
125	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
126	PR6A	PR7A	PR8A	PR9A	PR10A	PR11A	PR13A	PR16A	I/O
127	PR6B	PR7B	PR8B	PR9B	PR10B	PR11B	PR13B	PR16B	I/O
128	PR6C	PR7C	PR8C	PR9C	PR10C	PR11C	PR13C	PR16C	I/O
129	PR6D	PR7D	PR8D	PR9D	PR10D	PR11D	PR13D	PR16D	I/O
130	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
131	PR5A	PR6A	PR7A	PR8A	PR9A	PR10A	PR12A	PR15A	I/O
132	PR5B	PR6B	PR7B	PR8B	PR9B	PR10B	PR12B	PR15B	I/O
133	PR5C	PR6C	PR7C	PR8C	PR9C	PR10C	PR12C	PR15C	I/O
134	PR5D	PR6D	PR7D	PR8D	PR9D	PR10D	PR12D	PR15D	I/O
135	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
136	PR4A	PR5A	PR6A	PR7A	PR8A	PR9A	PR11A	PR14A	I/O
137	PR4B	PR5B	PR6B	PR7B	PR8B	PR9B	PR11D	PR14D	I/O
138	PR4C	PR5C	PR6C	PR7C	PR8C	PR9C	PR10A	PR13A	I/O
139	PR4D	PR5D	PR6D	PR7D	PR8D	PR9D	PR10D	PR13D	I/O
140	PR3A	PR4A	PR5A	PR6A	PR7A	PR8A	PR9A	PR12A	I/O-CS1
141	PR3B	PR4B	PR5B	PR6B	PR7B	PR8B	PR9D	PR12D	I/O
142	PR3C	PR4C	PR5C	PR6C	PR7C	PR8C	PR8A	PR11A	I/O
143	PR3D	PR4D	PR5D	PR6D	PR7D	PR8D	PR8D	PR11D	I/O
144	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
145	PR2A	PR3A	PR4A	PR5A	PR6A	PR7A	PR7A	PR10A	I/O-$\overline{CS0}$
146	PR2B	PR3B	PR4B	PR4B	PR6B	PR7B	PR7B	PR10B	I/O

Note: The ATT2C04 does not have bond pads connected to 208-pin SQFP package pin numbers 6, 45, 47, 56, 60, 102, 153, 154, 166, 201, and 203.

Pin Information (continued)

Table 20. ATT2C04, ATT2C06, ATT2C08, ATT2C10, ATT2C12, ATT2C15, ATT2C26, and ATT2C40 208-Pin SQFP/SQFP-PQ2 Pinout (continued)

Pin	2C04 Pad	2C06 Pad	2C08 Pad	2C10 Pad	2C12 Pad	2C15 Pad	2C26 Pad	2C40 Pad	Function
147	PR2C	PR3C	PR4C	PR4C	PR5B	PR6B	PR6B	PR9B	I/O
148	PR2D	PR3D	PR4D	PR4D	PR5D	PR6D	PR6D	PR9D	I/O
149	PR1A	PR2A	PR3A	PR3A	PR4A	PR5A	PR5A	PR8A	I/O-$\overline{RD}$
150	PR1B	PR2C	PR3C	PR3C	PR4D	PR5D	PR5D	PR6A	I/O
151	PR1C	PR2D	PR3D	PR3D	PR3A	PR4A	PR4A	PR5A	I/O
152	PR1D	PR1A	PR2A	PR2A	PR2A	PR3A	PR3A	PR4A	I/O-$\overline{WR}$
153	See Note	PR1C	PR2D	PR2D	PR2C	PR2A	PR2A	PR3A	I/O
154	See Note	PR1D	PR1A	PR1A	PR1A	PR1A	PR1A	PR2A	I/O
155	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
156	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$
157	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
158	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
159	PT10D	PT12D	PT14D	PT16D	PT18D	PT20D	PT24D	PT30D	I/O
160	PT10C	PT12A	PT13D	PT15D	PT17D	PT19A	PT23A	PT28A	I/O-RDY/RCLK
161	PT10B	PT11D	PT13A	PT15A	PT16D	PT17D	PT21D	PT26D	I/O
162	PT10A	PT11C	PT12D	PT14D	PT16A	PT17A	PT21A	PT26A	I/O
163	PT9D	PT11A	PT12C	PT13D	PT15D	PT16D	PT20D	PT25D	I/O-D7
164	PT9C	PT10D	PT12A	PT13B	PT14D	PT15D	PT19D	PT24D	I/O
165	PT9B	PT10C	PT11D	PT13A	PT14A	PT15A	PT19A	PT23D	I/O
166	See Note	PT10B	PT11C	PT12D	PT13D	PT14D	PT18D	PT22D	I/O
167	PT9A	PT10A	PT11B	PT12B	PT13B	PT14B	PT18B	PT21D	I/O-D6
168	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
169	PT8D	PT9D	PT10D	PT11D	PT12D	PT13D	PT17D	PT20D	I/O
170	PT8C	PT9C	PT10C	PT11C	PT12C	PT13C	PT17A	PT20A	I/O
171	PT8B	PT9B	PT10B	PT11B	PT12B	PT13B	PT16D	PT19D	I/O
172	PT8A	PT9A	PT10A	PT11A	PT12A	PT13A	PT16A	PT19A	I/O-D5
173	PT7D	PT8D	PT9D	PT10D	PT11D	PT12D	PT15D	PT18D	I/O
174	PT7C	PT8C	PT9C	PT10C	PT11C	PT12C	PT15A	PT18A	I/O
175	PT7B	PT8B	PT9B	PT10B	PT11B	PT12B	PT14D	PT17D	I/O
176	PT7A	PT8A	PT9A	PT10A	PT11A	PT12A	PT14A	PT17A	I/O-D4
177	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
178	PT6D	PT7D	PT8D	PT9D	PT10D	PT11D	PT13D	PT16D	I/O
179	PT6C	PT7C	PT8C	PT9C	PT10C	PT11C	PT13C	PT16C	I/O
180	PT6B	PT7B	PT8B	PT9B	PT10B	PT11B	PT13B	PT16B	I/O
181	PT6A	PT7A	PT8A	PT9A	PT10A	PT11A	PT13A	PT16A	I/O-D3
182	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS

Note: The ATT2C04 does not have bond pads connected to 208-pin SQFP package pin numbers 6, 45, 47, 56, 60, 102, 153, 154, 166, 201, and 203.

Pin Information (continued)

Table 20. ATT2C04, ATT2C06, ATT2C08, ATT2C10, ATT2C12, ATT2C15, ATT2C26, and ATT2C40 208-Pin SQFP/SQFP-PQ2 Pinout (continued)

Pin	2C04 Pad	2C06 Pad	2C08 Pad	2C10 Pad	2C12 Pad	2C15 Pad	2C26 Pad	2C40 Pad	Function
183	PT5D	PT6D	PT7D	PT8D	PT9D	PT10D	PT12D	PT15D	I/O
184	PT5C	PT6C	PT7C	PT8C	PT9C	PT10C	PT12C	PT15C	I/O
185	PT5B	PT6B	PT7B	PT8B	PT9B	PT10B	PT12B	PT15B	I/O
186	PT5A	PT6A	PT7A	PT8A	PT9A	PT10A	PT12A	PT15A	I/O-D2
187	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
188	PT4D	PT5D	PT6D	PT7D	PT8D	PT9D	PT11D	PT14D	I/O-D1
189	PT4C	PT5C	PT6C	PT7C	PT8C	PT9C	PT11A	PT14A	I/O
190	PT4B	PT5B	PT6B	PT7B	PT8B	PT9B	PT10D	PT13D	I/O
191	PT4A	PT5A	PT6A	PT7A	PT8A	PT9A	PT10A	PT13A	I/O-D0/DIN
192	PT3D	PT4D	PT5D	PT6D	PT7D	PT8D	PT9D	PT12D	I/O
193	PT3C	PT4C	PT5C	PT6C	PT7C	PT8C	PT9A	PT12A	I/O
194	PT3B	PT4B	PT5B	PT6B	PT7B	PT8B	PT8D	PT11D	I/O
195	PT3A	PT4A	PT5A	PT6A	PT7A	PT8A	PT8A	PT11A	I/O-DOUT
196	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
197	PT2D	PT3D	PT4D	PT5D	PT6D	PT7D	PT7D	PT10D	I/O
198	PT2C	PT3C	PT4C	PT5A	PT6A	PT7A	PT7A	PT9A	I/O
199	PT2B	PT3B	PT4B	PT4D	PT5C	PT6C	PT6C	PT8A	I/O
200	PT2A	PT3A	PT4A	PT4A	PT5A	PT6A	PT6A	PT7A	I/O-TDI
201	See Note	PT2D	PT3D	PT3D	PT4A	PT5A	PT5A	PT6A	I/O
202	PT1D	PT2A	PT3A	PT3A	PT3A	PT4A	PT4A	PT5A	I/O-TMS
203	See Note	PT1D	PT2D	PT2D	PT2C	PT3A	PT3A	PT4A	I/O
204	PT1C	PT1C	PT2A	PT2A	PT2A	PT2A	PT2A	PT3A	I/O
205	PT1B	PT1B	PT1D	PT1D	PT1D	PT1D	PT1D	PT2D	I/O
206	PT1A	PT1A	PT1A	PT1A	PT1A	PT1A	PT1A	PT1A	I/O-TCK
207	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
208	RD_DATA/ TDO	RD_DATA/ TDO	RD_DATA/ TDO	RD_DATA/ TDO	RD_DATA/ TDO	RD_DATA/ TDO	RD_DATA/ TDO	RD_DATA/ TDO	RD_DATA/TDO

Note: The ATT2C04 does not have bond pads connected to 208-pin SQFP package pin numbers 6, 45, 47, 56, 60, 102, 153, 154, 166, 201, and 203.

Pin Information (continued)

Table 21. ATT2C06, ATT2C08, ATT2C10, ATT2C12, ATT2C15, ATT2C26, ATT2C40 and 240-Pin SQFP/SQFP-PQ2 Pinout

Pin	2C06 Pad	2C08 Pad	2C10 Pad	2C12 Pad	2C15 Pad	2C26 Pad	2C40 Pad	Function
1	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
2	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
3	PL1D	PL1D	PL1D	PL1D	PL1D	PL1D	PL1D	I/O
4	PL1C	PL1B	PL1B	PL1C	PL1C	PL1C	PL1A	I/O
5	PL1B	PL1A	PL1A	PL1B	PL1B	PL1B	PL2D	I/O
6	PL1A	PL2D	PL2D	PL2D	PL2D	PL2D	PL3D	I/O-A0
7	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
8	PL2D	PL3D	PL3D	PL3D	PL4D	PL4D	PL5D	I/O
9	PL2C	PL3C	PL3C	PL3A	PL4A	PL4A	PL6D	I/O
10	PL2B	PL3B	PL3B	PL4D	PL5D	PL5D	PL7D	I/O
11	PL2A	PL3A	PL3A	PL4A	PL5A	PL5A	PL8D	I/O-A1
12	PL3D	PL4D	PL4A	PL5A	PL6A	PL6A	PL9A	I/O-A2
13	PL3C	PL4C	PL5C	PL6D	PL7D	PL7D	PL10D	I/O
14	PL3B	PL4B	PL5B	PL6B	PL7B	PL7B	PL10B	I/O
15	PL3A	PL4A	PL5A	PL6A	PL7A	PL7A	PL10A	I/O-A3
16	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
17	PL4D	PL5D	PL6D	PL7D	PL8D	PL8D	PL11D	I/O
18	PL4C	PL5C	PL6C	PL7C	PL8C	PL8A	PL11A	I/O
19	PL4B	PL5B	PL6B	PL7B	PL8B	PL9D	PL12D	I/O
20	PL4A	PL5A	PL6A	PL7A	PL8A	PL9A	PL12A	I/O-A4
21	PL5D	PL6D	PL7D	PL8D	PL9D	PL10D	PL13D	I/O-A5
22	PL5C	PL6C	PL7C	PL8C	PL9C	PL10A	PL13A	I/O
23	PL5B	PL6B	PL7B	PL8B	PL9B	PL11D	PL14D	I/O
24	PL5A	PL6A	PL7A	PL8A	PL9A	PL11A	PL14A	I/O-A6
25	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
26	PL6D	PL7D	PL8D	PL9D	PL10D	PL12D	PL15D	I/O
27	PL6C	PL7C	PL8C	PL9C	PL10C	PL12C	PL15C	I/O
28	PL6B	PL7B	PL8B	PL9B	PL10B	PL12B	PL15B	I/O
29	PL6A	PL7A	PL8A	PL9A	PL10A	PL12A	PL15A	I/O-A7
30	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
31	PL7D	PL8D	PL9D	PL10D	PL11D	PL13D	PL16D	I/O
32	PL7C	PL8C	PL9C	PL10C	PL11C	PL13C	PL16C	I/O
33	PL7B	PL8B	PL9B	PL10B	PL11B	PL13B	PL16B	I/O
34	PL7A	PL8A	PL9A	PL10A	PL11A	PL13A	PL16A	I/O-A8
35	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
36	PL8D	PL9D	PL10D	PL11D	PL12D	PL14D	PL17D	I/O-A9
37	PL8C	PL9C	PL10C	PL11C	PL12C	PL14A	PL17A	I/O
38	PL8B	PL9B	PL10B	PL11B	PL12B	PL15D	PL18D	I/O
39	PL8A	PL9A	PL10A	PL11A	PL12A	PL15A	PL18A	I/O-A10
40	PL9D	PL10D	PL11D	PL12D	PL13D	PL16D	PL19D	I/O
41	PL9C	PL10C	PL11C	PL12C	PL13C	PL16A	PL19A	I/O
42	PL9B	PL10B	PL11B	PL12B	PL13B	PL17D	PL20D	I/O
43	PL9A	PL10A	PL11A	PL12A	PL13A	PL17A	PL20A	I/O-A11
44	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD

Note: The ATT2C08 and ATT2C10 do not have bond pads connected to 240-pin SQFP package pin numbers 113 and 188.

Pin Information (continued)

Table 21. ATT2C06, ATT2C08, ATT2C10, ATT2C12, ATT2C15, ATT2C26, ATT2C40 and 240-Pin SQFP/SQFP-PQ2 Pinout (continued)

Pin	2C06 Pad	2C08 Pad	2C10 Pad	2C12 Pad	2C15 Pad	2C26 Pad	2C40 Pad	Function
45	PL10D	PL11D	PL12D	PL13D	PL14D	PL18D	PL21D	I/O-A12
46	PL10C	PL11C	PL12C	PL13B	PL14B	PL18B	PL21B	I/O
47	PL10B	PL11B	PL12B	PL14D	PL15D	PL19D	PL22D	I/O
48	PL10A	PL11A	PL13D	PL14B	PL15B	PL19B	PL22B	I/O-A13
49	PL11D	PL12D	PL13B	PL14A	PL15A	PL19A	PL22A	I/O
50	PL11C	PL12C	PL13A	PL15D	PL16D	PL20D	PL23D	I/O
51	PL11B	PL12B	PL14D	PL15B	PL16B	PL20B	PL24D	I/O
52	PL11A	PL12A	PL14C	PL16D	PL17D	PL21D	PL25A	I/O-A14
53	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
54	PL12D	PL13D	PL15D	PL17D	PL18D	PL22D	PL27D	I/O
55	PL12C	PL13A	PL15A	PL17A	PL19D	PL23D	PL28D	I/O
56	PL12B	PL14D	PL16D	PL18C	PL19A	PL23A	PL28A	I/O
57	PL12A	PL14A	PL16A	PL18A	PL20A	PL24A	PL30A	I/O-A15
58	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
59	CCLK	CCLK	CCLK	CCLK	CCLK	CCLK	CCLK	CCLK
60	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
61	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
62	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
63	PB1A	PB1A	PB1A	PB1A	PB1A	PB1A	PB1A	I/O-A16
64	PB1B	PB1D	PB1D	PB1D	PB2A	PB2A	PB3A	I/O
65	PB1C	PB2A	PB2A	PB2A	PB2D	PB2D	PB3D	I/O
66	PB1D	PB2D	PB2D	PB2D	PB3D	PB3D	PB4D	I/O
67	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
68	PB2A	PB3A	PB3B	PB3D	PB4D	PB4D	PB5D	I/O-A17
69	PB2B	PB3B	PB4B	PB4D	PB5D	PB5D	PB6D	I/O
70	PB2C	PB3C	PB4C	PB5A	PB6A	PB6A	PB7A	I/O
71	PB2D	PB3D	PB4D	PB5B	PB6B	PB6B	PB7D	I/O
72	PB3A	PB4A	PB5A	PB5D	PB6D	PB6D	PB8D	I/O
73	PB3B	PB4B	PB5B	PB6A	PB7A	PB7A	PB9A	I/O
74	PB3C	PB4C	PB5C	PB6B	PB7B	PB7B	PB9D	I/O
75	PB3D	PB4D	PB5D	PB6D	PB7D	PB7D	PB10D	I/O
76	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
77	PB4A	PB5A	PB6A	PB7A	PB8A	PB8A	PB11A	I/O
78	PB4B	PB5B	PB6B	PB7B	PB8B	PB8D	PB11D	I/O
79	PB4C	PB5C	PB6C	PB7C	PB8C	PB9A	PB12A	I/O
80	PB4D	PB5D	PB6D	PB7D	PB8D	PB9D	PB12D	I/O
81	PB5A	PB6A	PB7A	PB8A	PB9A	PB10A	PB13A	I/O
82	PB5B	PB6B	PB7B	PB8B	PB9B	PB10D	PB13D	I/O
83	PB5C	PB6C	PB7C	PB8C	PB9C	PB11A	PB14A	I/O
84	PB5D	PB6D	PB7D	PB8D	PB9D	PB11D	PB14D	I/O
85	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
86	PB6A	PB7A	PB8A	PB9A	PB10A	PB12A	PB15A	I/O
87	PB6B	PB7B	PB8B	PB9B	PB10B	PB12B	PB15B	I/O
88	PB6C	PB7C	PB8C	PB9C	PB10C	PB12C	PB15C	I/O

Note: The ATT2C08 and ATT2C10 do not have bond pads connected to 240-pin SQFP package pin numbers 113 and 188.

Pin Information (continued)

Table 21. ATT2C06, ATT2C08, ATT2C10, ATT2C12, ATT2C15, ATT2C26, ATT2C40 and 240-Pin SQFP/SQFP-PQ2 Pinout (continued)

Pin	2C06 Pad	2C08 Pad	2C10 Pad	2C12 Pad	2C15 Pad	2C26 Pad	2C40 Pad	Function
89	PB6D	PB7D	PB8D	PB9D	PB10D	PB12D	PB15D	I/O
90	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
91	PB7A	PB8A	PB9A	PB10A	PB11A	PB13A	PB16A	I/O
92	PB7B	PB8B	PB9B	PB10B	PB11B	PB13B	PB16B	I/O
93	PB7C	PB8C	PB9C	PB10C	PB11C	PB13C	PB16C	I/O
94	PB7D	PB8D	PB9D	PB10D	PB11D	PB13D	PB16D	I/O
95	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
96	PB8A	PB9A	PB10A	PB11A	PB12A	PB14A	PB17A	I/O
97	PB8B	PB9B	PB10B	PB11B	PB12B	PB14D	PB17D	I/O
98	PB8C	PB9C	PB10C	PB11C	PB12C	PB15A	PB18A	I/O
99	PB8D	PB9D	PB10D	PB11D	PB12D	PB15D	PB18D	I/O
100	PB9A	PB10A	PB11A	PB12A	PB13A	PB16A	PB19A	I/O-HDC
101	PB9B	PB10B	PB11B	PB12B	PB13B	PB16D	PB19D	I/O
102	PB9C	PB10C	PB11C	PB12C	PB13C	PB17A	PB20A	I/O
103	PB9D	PB10D	PB11D	PB12D	PB13D	PB17D	PB20D	I/O
104	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
105	PB10A	PB11A	PB12A	PB13A	PB14A	PB18A	PB21A	I/O-$\overline{\text{LDC}}$
106	PB10B	PB11D	PB13A	PB13D	PB14D	PB18D	PB22D	I/O
107	PB10C	PB12A	PB13B	PB14A	PB15A	PB19A	PB23A	I/O
108	PB10D	PB12B	PB13C	PB14D	PB15D	PB19D	PB24D	I/O
109	PB11A	PB12C	PB13D	PB15A	PB16A	PB20A	PB25A	I/O-$\overline{\text{INIT}}$
110	PB11B	PB12D	PB14A	PB15D	PB16D	PB20D	PB25D	I/O
111	PB11C	PB13A	PB15A	PB16A	PB17A	PB21A	PB26A	I/O
112	PB11D	PB13B	PB15B	PB16D	PB17D	PB21D	PB26D	I/O
113	VSS	See Note	See Note	VSS	VSS	VSS	VSS	VSS
114	PB12A	PB13D	PB15D	PB17A	PB18A	PB22A	PB27A	I/O
115	PB12B	PB14A	PB16A	PB17D	PB19A	PB23A	PB28A	I/O
116	PB12C	PB14B	PB16B	PB18A	PB19D	PB23D	PB28D	I/O
117	PB12D	PB14D	PB16D	PB18D	PB20D	PB24D	PB30D	I/O
118	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
119	DONE	DONE	DONE	DONE	DONE	DONE	DONE	DONE
120	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
121	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
122	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$
123	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$
124	PR12A	PR14A	PR16A	PR18A	PR20A	PR24A	PR30A	I/O-M0
125	PR12B	PR14D	PR16D	PR18C	PR20D	PR24D	PR29D	I/O
126	PR12C	PR13A	PR15A	PR18D	PR19A	PR23A	PR28A	I/O
127	PR12D	PR13D	PR15D	PR17B	PR18A	PR22A	PR27A	I/O
128	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
129	PR11A	PR12A	PR14A	PR16A	PR17A	PR21A	PR26A	I/O
130	PR11B	PR12B	PR14C	PR16D	PR17D	PR21D	PR25A	I/O
131	PR11C	PR12C	PR14D	PR15A	PR16A	PR20A	PR24A	I/O
132	PR11D	PR12D	PR13A	PR15C	PR16C	PR20C	PR24D	I/O

Note: The ATT2C08 and ATT2C10 do not have bond pads connected to 240-pin SQFP package pin numbers 113 and 188.

Pin Information (continued)

Table 21. ATT2C06, ATT2C08, ATT2C10, ATT2C12, ATT2C15, ATT2C26, ATT2C40 and 240-Pin SQFP/SQFP-PQ2 Pinout (continued)

Pin	2C06 Pad	2C08 Pad	2C10 Pad	2C12 Pad	2C15 Pad	2C26 Pad	2C40 Pad	Function
133	PR10A	PR11A	PR13B	PR15D	PR16D	PR20D	PR23D	I/O-M1
134	PR10B	PR11B	PR13C	PR14A	PR15A	PR19A	PR22A	I/O
135	PR10C	PR11C	PR12A	PR14D	PR15D	PR19D	PR22D	I/O
136	PR10D	PR11D	PR12B	PR13A	PR14A	PR18A	PR21A	I/O
137	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
138	PR9A	PR10A	PR11A	PR12A	PR13A	PR17A	PR20A	I/O-M2
139	PR9B	PR10B	PR11B	PR12B	PR13B	PR17D	PR20D	I/O
140	PR9C	PR10C	PR11C	PR12C	PR13C	PR16A	PR19A	I/O
141	PR9D	PR10D	PR11D	PR12D	PR13D	PR16D	PR19D	I/O
142	PR8A	PR9A	PR10A	PR11A	PR12A	PR15A	PR18A	I/O-M3
143	PR8B	PR9B	PR10B	PR11B	PR12B	PR15D	PR18D	I/O
144	PR8C	PR9C	PR10C	PR11C	PR12C	PR14A	PR17A	I/O
145	PR8D	PR9D	PR10D	PR11D	PR12D	PR14D	PR17D	I/O
146	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
147	PR7A	PR8A	PR9A	PR10A	PR11A	PR13A	PR16A	I/O
148	PR7B	PR8B	PR9B	PR10B	PR11B	PR13B	PR16B	I/O
149	PR7C	PR8C	PR9C	PR10C	PR11C	PR13C	PR16C	I/O
150	PR7D	PR8D	PR9D	PR10D	PR11D	PR13D	PR16D	I/O
151	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
152	PR6A	PR7A	PR8A	PR9A	PR10A	PR12A	PR15A	I/O
153	PR6B	PR7B	PR8B	PR9B	PR10B	PR12B	PR15B	I/O
154	PR6C	PR7C	PR8C	PR9C	PR10C	PR12C	PR15C	I/O
155	PR6D	PR7D	PR8D	PR9D	PR10D	PR12D	PR15D	I/O
156	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
157	PR5A	PR6A	PR7A	PR8A	PR9A	PR11A	PR14A	I/O
158	PR5B	PR6B	PR7B	PR8B	PR9B	PR11D	PR14D	I/O
159	PR5C	PR6C	PR7C	PR8C	PR9C	PR10A	PR13A	I/O
160	PR5D	PR6D	PR7D	PR8D	PR9D	PR10D	PR13D	I/O
161	PR4A	PR5A	PR6A	PR7A	PR8A	PR9A	PR12A	I/O-CS1
162	PR4B	PR5B	PR6B	PR7B	PR8B	PR9D	PR12D	I/O
163	PR4C	PR5C	PR6C	PR7C	PR8C	PR8A	PR11A	I/O
164	PR4D	PR5D	PR6D	PR7D	PR8D	PR8D	PR11D	I/O
165	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
166	PR3A	PR4A	PR5A	PR6A	PR7A	PR7A	PR10A	I/O-$\overline{CS0}$
167	PR3B	PR4B	PR4B	PR6B	PR7B	PR7B	PR10B	I/O
168	PR3C	PR4C	PR4C	PR5B	PR6B	PR6B	PR9B	I/O
169	PR3D	PR4D	PR4D	PR5D	PR6D	PR6D	PR9D	I/O
170	PR2A	PR3A	PR3A	PR4A	PR5A	PR5A	PR8A	I/O-$\overline{RD}$
171	PR2B	PR3B	PR3B	PR4B	PR5B	PR5B	PR7A	I/O
172	PR2C	PR3C	PR3C	PR4D	PR5D	PR5D	PR6A	I/O
173	PR2D	PR3D	PR3D	PR3A	PR4A	PR4A	PR5A	I/O
174	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
175	PR1A	PR2A	PR2A	PR2A	PR3A	PR3A	PR4A	I/O-$\overline{WR}$
176	PR1B	PR2D	PR2D	PR2C	PR2A	PR2A	PR3A	I/O

Note: The ATT2C08 and ATT2C10 do not have bond pads connected to 240-pin SQFP package pin numbers 113 and 188.

Pin Information (continued)

Table 21. ATT2C06, ATT2C08, ATT2C10, ATT2C12, ATT2C15, ATT2C26, ATT2C40 and 240-Pin SQFP/SQFP-PQ2 Pinout (continued)

Pin	2C06 Pad	2C08 Pad	2C10 Pad	2C12 Pad	2C15 Pad	2C26 Pad	2C40 Pad	Function
177	PR1C	PR1A	PR1A	PR1A	PR1A	PR1A	PR2A	I/O
178	PR1D	PR1D	PR1D	PR1D	PR1D	PR1D	PR1D	I/O
179	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
180	$\overline{\text{RD\_CFGN}}$	$\overline{\text{RD\_CFGN}}$	$\overline{\text{RD\_CFGN}}$	$\overline{\text{RD\_CFGN}}$	$\overline{\text{RD\_CFGN}}$	$\overline{\text{RD\_CFGN}}$	$\overline{\text{RD\_CFGN}}$	$\overline{\text{RD\_CFGN}}$
181	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
182	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
183	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
184	PT12D	PT14D	PT16D	PT18D	PT20D	PT24D	PT30D	I/O
185	PT12C	PT14C	PT16C	PT18B	PT20A	PT24A	PT29A	I/O
186	PT12B	PT14A	PT16A	PT18A	PT19D	PT23D	PT28D	I/O
187	PT12A	PT13D	PT15D	PT17D	PT19A	PT23A	PT28A	I/O-RDY/RCLK
188	VSS	See Note	See Note	VSS	VSS	VSS	VSS	VSS
189	PT11D	PT13B	PT15B	PT16D	PT17D	PT21D	PT26D	I/O
190	PT11C	PT13A	PT15A	PT16C	PT17C	PT21C	PT26C	I/O
191	PT11B	PT12D	PT14D	PT16A	PT17A	PT21A	PT26A	I/O
192	PT11A	PT12C	PT13D	PT15D	PT16D	PT20D	PT25D	I/O-D7
193	PT10D	PT12A	PT13B	PT14D	PT15D	PT19D	PT24D	I/O
194	PT10C	PT11D	PT13A	PT14A	PT15A	PT19A	PT23D	I/O
195	PT10B	PT11C	PT12D	PT13D	PT14D	PT18D	PT22D	I/O
196	PT10A	PT11B	PT12B	PT13B	PT14B	PT18B	PT21D	I/O-D6
197	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
198	PT9D	PT10D	PT11D	PT12D	PT13D	PT17D	PT20D	I/O
199	PT9C	PT10C	PT11C	PT12C	PT13C	PT17A	PT20A	I/O
200	PT9B	PT10B	PT11B	PT12B	PT13B	PT16D	PT19D	I/O
201	PT9A	PT10A	PT11A	PT12A	PT13A	PT16A	PT19A	I/O-D5
202	PT8D	PT9D	PT10D	PT11D	PT12D	PT15D	PT18D	I/O
203	PT8C	PT9C	PT10C	PT11C	PT12C	PT15A	PT18A	I/O
204	PT8B	PT9B	PT10B	PT11B	PT12B	PT14D	PT17D	I/O
205	PT8A	PT9A	PT10A	PT11A	PT12A	PT14A	PT17A	I/O-D4
206	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
207	PT7D	PT8D	PT9D	PT10D	PT11D	PT13D	PT16D	I/O
208	PT7C	PT8C	PT9C	PT10C	PT11C	PT13C	PT16C	I/O
209	PT7B	PT8B	PT9B	PT10B	PT11B	PT13B	PT16B	I/O
210	PT7A	PT8A	PT9A	PT10A	PT11A	PT13A	PT16A	I/O-D3
211	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
212	PT6D	PT7D	PT8D	PT9D	PT10D	PT12D	PT15D	I/O
213	PT6C	PT7C	PT8C	PT9C	PT10C	PT12C	PT15C	I/O
214	PT6B	PT7B	PT8B	PT9B	PT10B	PT12B	PT15B	I/O
215	PT6A	PT7A	PT8A	PT9A	PT10A	PT12A	PT15A	I/O-D2
216	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
217	PT5D	PT6D	PT7D	PT8D	PT9D	PT11D	PT14D	I/O-D1
218	PT5C	PT6C	PT7C	PT8C	PT9C	PT11A	PT14A	I/O
219	PT5B	PT6B	PT7B	PT8B	PT9B	PT10D	PT13D	I/O
220	PT5A	PT6A	PT7A	PT8A	PT9A	PT10A	PT13A	I/O-D0/DIN

Note: The ATT2C08 and ATT2C10 do not have bond pads connected to 240-pin SQFP package pin numbers 113 and 188.

Pin Information (continued)

Table 21. ATT2C06, ATT2C08, ATT2C10, ATT2C12, ATT2C15, ATT2C26, ATT2C40 and 240-Pin SQFP/SQFP-PQ2 Pinout (continued)

Pin	2C06 Pad	2C08 Pad	2C10 Pad	2C12 Pad	2C15 Pad	2C26 Pad	2C40 Pad	Function
221	PT4D	PT5D	PT6D	PT7D	PT8D	PT9D	PT12D	I/O
222	PT4C	PT5C	PT6C	PT7C	PT8C	PT9A	PT12A	I/O
223	PT4B	PT5B	PT6B	PT7B	PT8B	PT8D	PT11D	I/O
224	PT4A	PT5A	PT6A	PT7A	PT8A	PT8A	PT11A	I/O-DOUT
225	VDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD
226	PT3D	PT4D	PT5D	PT6D	PT7D	PT7D	PT10D	I/O
227	PT3C	PT4C	PT5A	PT6A	PT7A	PT7A	PT9A	I/O
228	PT3B	PT4B	PT4D	PT5C	PT6C	PT6C	PT8A	I/O
229	PT3A	PT4A	PT4A	PT5A	PT6A	PT6A	PT7A	I/O-TDI
230	PT2D	PT3D	PT3D	PT4D	PT5D	PT5D	PT6D	I/O
231	PT2C	PT3C	PT3C	PT4A	PT5A	PT5A	PT6A	I/O
232	PT2B	PT3B	PT3B	PT3D	PT4D	PT4D	PT5D	I/O
233	PT2A	PT3A	PT3A	PT3A	PT4A	PT4A	PT5A	I/O-TMS
234	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
235	PT1D	PT2D	PT2D	PT2C	PT3A	PT3A	PT4A	I/O
236	PT1C	PT2A	PT2A	PT2A	PT2A	PT2A	PT3A	I/O
237	PT1B	PT1D	PT1D	PT1D	PT1D	PT1D	PT2D	I/O
238	PT1A	PT1A	PT1A	PT1A	PT1A	PT1A	PT1A	I/O-TCK
239	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS
240	RD_DATA/ TDO	RD_DATA/ TDO	RD_DATA/ TDO	RD_DATA/ TDO	RD_DATA/ TDO	RD_DATA/ TDO	RD_DATA/ TDO	RD_DATA/TDO

Note: The ATT2C08 and ATT2C10 do not have bond pads connected to 240-pin SQFP package pin numbers 113 and 188.

Pin Information (continued)

Table 22. ATT2C08, ATT2C10, and ATT2C12 256-Pin PBGA Pinout

Pin	2C08 Pad	2C10 Pad	2C12 Pad	Function
A1	VSS	VSS	VSS	VSS
B1	VDD	VDD	VDD	VDD
C2	PL1D	PL1D	PL1D	I/O
D2	PL1B	PL1B	PL1C	I/O
D3	PL1A	PL1A	PL1B	I/O
E4	PL2D	PL2D	PL2D	I/O-A0
C1	PL2C	PL2C	PL2C	I/O
D1	PL2B	PL2B	PL2B	I/O
E3	PL2A	PL2A	PL2A	I/O
D4	VSS	VSS	VSS	VSS
E2	PL3D	PL3D	PL3D	I/O
E1	PL3C	PL3C	PL3A	I/O
F3	PL3B	PL3B	PL4D	I/O
G4	PL3A	PL3A	PL4A	I/O-A1
D8	VSS	VSS	VSS	VSS
F2	—	PL4D	PL5D	I/O
F1	PL4D	PL4A	PL5A	I/O-A2
G3	PL4C	PL5C	PL6D	I/O
G2	PL4B	PL5B	PL6B	I/O
G1	PL4A	PL5A	PL6A	I/O-A3
D6	VDD	VDD	VDD	VDD
H3	PL5D	PL6D	PL7D	I/O
H2	PL5C	PL6C	PL7C	I/O
H1	PL5B	PL6B	PL7B	I/O
J4	PL5A	PL6A	PL7A	I/O-A4
J3	PL6D	PL7D	PL8D	I/O-A5
J2	PL6C	PL7C	PL8C	I/O
J1	PL6B	PL7B	PL8B	I/O
K2	PL6A	PL7A	PL8A	I/O-A6
D13	VSS	VSS	VSS	VSS
K3	PL7D	PL8D	PL9D	I/O
K1	PL7C	PL8C	PL9C	I/O
L1	PL7B	PL8B	PL9B	I/O
L2	PL7A	PL8A	PL9A	I/O-A7
D11	VDD	VDD	VDD	VDD
L3	PL8D	PL9D	PL10D	I/O
L4	PL8C	PL9C	PL10C	I/O
M1	PL8B	PL9B	PL10B	I/O
M2	PL8A	PL9A	PL10A	I/O-A8
D17	VSS	VSS	VSS	VSS
M3	PL9D	PL10D	PL11D	I/O-A9
M4	PL9C	PL10C	PL11C	I/O
N1	PL9B	PL10B	PL11B	I/O
N2	PL9A	PL10A	PL11A	I/O-A10
N3	PL10D	PL11D	PL12D	I/O
P1	PL10C	PL11C	PL12C	I/O
P2	PL10B	PL11B	PL12B	I/O
R1	PL10A	PL11A	PL12A	I/O-A11
D15	VDD	VDD	VDD	VDD
P3	PL11D	PL12D	PL13D	I/O-A12
R2	PL11C	PL12C	PL13B	I/O
T1	PL11B	PL12B	PL14D	I/O
P4	PL11A	PL13D	PL14B	I/O-A13
R3	PL12D	PL13B	PL14A	I/O
H4	VSS	VSS	VSS	VSS
T2	PL12C	PL13A	PL15D	I/O
W3	—	—	—	No Connect
U1	PL12B	PL14D	PL15B	I/O
T3	PL12A	PL14C	PL16D	I/O-A14
H17	VSS	VSS	VSS	VSS
U2	PL13D	PL15D	PL17D	I/O
V1	PL13C	PL15C	PL17C	I/O
T4	PL13B	PL15B	PL17B	I/O
U3	PL13A	PL15A	PL17A	I/O
V2	PL14D	PL16D	PL18D	I/O
W1	PL14C	PL16C	PL18C	I/O
V3	PL14B	PL16B	PL18B	I/O
W2	PL14A	PL16A	PL18A	I/O-A15
N4	VSS	VSS	VSS	VSS
Y1	CCLK	CCLK	CCLK	CCLK
F4	VDD	VDD	VDD	VDD
N17	VSS	VSS	VSS	VSS
F17	VDD	VDD	VDD	VDD
Y2	PB1A	PB1A	PB1A	I/O-A16
W4	PB1C	PB1C	PB1C	I/O
V4	PB1D	PB1D	PB1D	I/O
U5	PB2A	PB2A	PB2A	I/O
Y3	PB2B	PB2B	PB2B	I/O
Y4	PB2C	PB2C	PB2C	I/O
V5	PB2D	PB2D	PB2D	I/O
U4	VSS	VSS	VSS	VSS
W5	PB3A	PB3B	PB3D	I/O-A17
U8	VSS	VSS	VSS	VSS
Y5	PB3B	PB4B	PB4D	I/O
V6	PB3C	PB4C	PB5A	I/O
U7	PB3D	PB4D	PB5B	I/O

Notes:

The W3 pin on the 256-pin PBGA package is unconnected for all devices listed in this table.

The ATT2C08 does not have bond pads connected to the 256-pin PBGA package pins F2 and Y17.

A 4 x 4 array of thermal balls is found at the center of the package. The balls can be attached to the ground plane of the board for enhanced thermal capability (see Table 26) or left unconnected.

Pin Information (continued)

Table 22. ATT2C08, ATT2C10, and ATT2C12 256-Pin PBGA Pinout (continued)

Pin	2C08 Pad	2C10 Pad	2C12 Pad	Function
W6	PB4A	PB5A	PB5D	I/O
Y6	PB4B	PB5B	PB6A	I/O
V7	PB4C	PB5C	PB6B	I/O
W7	PB4D	PB5D	PB6D	I/O
K4	VDD	VDD	VDD	VDD
Y7	PB5A	PB6A	PB7A	I/O
V8	PB5B	PB6B	PB7B	I/O
W8	PB5C	PB6C	PB7C	I/O
Y8	PB5D	PB6D	PB7D	I/O
U9	PB6A	PB7A	PB8A	I/O
V9	PB6B	PB7B	PB8B	I/O
W9	PB6C	PB7C	PB8C	I/O
Y9	PB6D	PB7D	PB8D	I/O
U13	VSS	VSS	VSS	VSS
W10	PB7A	PB8A	PB9A	I/O
V10	PB7B	PB8B	PB9B	I/O
Y10	PB7C	PB8C	PB9C	I/O
Y11	PB7D	PB8D	PB9D	I/O
U17	VSS	VSS	VSS	VSS
W11	PB8A	PB9A	PB10A	I/O
V11	PB8B	PB9B	PB10B	I/O
U11	PB8C	PB9C	PB10C	I/O
Y12	PB8D	PB9D	PB10D	I/O
W12	PB9A	PB10A	PB11A	I/O
V12	PB9B	PB10B	PB11B	I/O
U12	PB9C	PB10C	PB11C	I/O
Y13	PB9D	PB10D	PB11D	I/O
W13	PB10A	PB11A	PB12A	I/O-HDC
V13	PB10B	PB11B	PB12B	I/O
Y14	PB10C	PB11C	PB12C	I/O
W14	PB10D	PB11D	PB12D	I/O
L17	VDD	VDD	VDD	VDD
Y15	PB11A	PB12A	PB13A	I/O-$\overline{\text{LDC}}$
V14	PB11B	PB12C	PB13B	I/O
W15	PB11C	PB12D	PB13C	I/O
Y16	PB11D	PB13A	PB13D	I/O
U14	PB12A	PB13B	PB14A	I/O
V15	PB12B	PB13C	PB14D	I/O
W16	PB12C	PB13D	PB15A	I/O-$\overline{\text{INIT}}$
Y17	—	PB14A	PB15D	I/O
V16	PB12D	PB14B	PB16A	I/O
W17	PB13A	PB15A	PB16D	I/O
Y18	PB13B	PB15B	PB17A	I/O

Pin	2C08 Pad	2C10 Pad	2C12 Pad	Function
U16	PB13C	PB15C	PB17C	I/O
V17	PB13D	PB15D	PB17D	I/O
W18	PB14A	PB16A	PB18A	I/O
Y19	PB14B	PB16B	PB18B	I/O
V18	PB14C	PB16C	PB18C	I/O
W19	PB14D	PB16D	PB18D	I/O
Y20	DONE	DONE	DONE	DONE
R4	VDD	VDD	VDD	VDD
W20	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$
V19	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$
U19	PR14A	PR16A	PR18A	I/O-M0
U18	PR14C	PR16C	PR18C	I/O
T17	PR14D	PR16D	PR18D	I/O
V20	PR13A	PR15A	PR17A	I/O
U20	PR13B	PR15B	PR17B	I/O
T18	PR13C	PR15C	PR17C	I/O
T19	PR13D	PR15D	PR17D	I/O
T20	PR12A	PR14A	PR16A	I/O
R18	PR12B	PR14C	PR16D	I/O
P17	PR12C	PR14D	PR15A	I/O
R19	PR12D	PR13A	PR15C	I/O
R20	PR11A	PR13B	PR15D	I/O-M1
P18	PR11B	PR13C	PR14A	I/O
P19	PR11C	PR12A	PR14D	I/O
P20	PR11D	PR12B	PR13A	I/O
R17	VDD	VDD	VDD	VDD
N18	PR10A	PR11A	PR12A	I/O-M2
N19	PR10B	PR11B	PR12B	I/O
N20	PR10C	PR11C	PR12C	I/O
M17	PR10D	PR11D	PR12D	I/O
M18	PR9A	PR10A	PR11A	I/O-M3
M19	PR9B	PR10B	PR11B	I/O
M20	PR9C	PR10C	PR11C	I/O
L19	PR9D	PR10D	PR11D	I/O
L18	PR8A	PR9A	PR10A	I/O
L20	PR8B	PR9B	PR10B	I/O
K20	PR8C	PR9C	PR10C	I/O
K19	PR8D	PR9D	PR10D	I/O
U6	VDD	VDD	VDD	VDD
K18	PR7A	PR8A	PR9A	I/O
K17	PR7B	PR8B	PR9B	I/O
J20	PR7C	PR8C	PR9C	I/O
J19	PR7D	PR8D	PR9D	I/O

Notes:

The W3 pin on the 256-pin PBGA package is unconnected for all devices listed in this table.

The ATT2C08 does not have bond pads connected to the 256-pin PBGA package pins F2 and Y17.

A 4 x 4 array of thermal balls is found at the center of the package. The balls can be attached to the ground plane of the board for enhanced thermal capability (see Table 26) or left unconnected.

Pin Information (continued)

Table 22. ATT2C08, ATT2C10, and ATT2C12 256-Pin PBGA Pinout (continued)

Pin	2C08 Pad	2C10 Pad	2C12 Pad	Function
J18	PR6A	PR7A	PR8A	I/O
J17	PR6B	PR7B	PR8B	I/O
H20	PR6C	PR7C	PR8C	I/O
H19	PR6D	PR7D	PR8D	I/O
H18	PR5A	PR6A	PR7A	I/O-CS1
G20	PR5B	PR6B	PR7B	I/O
G19	PR5C	PR6C	PR7C	I/O
F20	PR5D	PR6D	PR7D	I/O
U10	VDD	VDD	VDD	VDD
G18	PR4A	PR5A	PR6A	I/O-$\overline{CS0}$
F19	PR4B	PR4B	PR6B	I/O
E20	PR4C	PR4C	PR5B	I/O
G17	PR4D	PR4D	PR5D	I/O
F18	PR3A	PR3A	PR4A	I/O-$\overline{RD}$
E19	PR3B	PR3B	PR4B	I/O
D20	PR3C	PR3C	PR4D	I/O
E18	PR3D	PR3D	PR3A	I/O
D19	PR2A	PR2A	PR2A	I/O-$\overline{WR}$
C20	PR2B	PR2B	PR2B	I/O
E17	PR2C	PR2C	PR2C	I/O
D18	PR2D	PR2D	PR2D	I/O
C19	PR1A	PR1A	PR1A	I/O
B20	PR1B	PR1B	PR1B	I/O
C18	PR1C	PR1C	PR1C	I/O
B19	PR1D	PR1D	PR1D	I/O
A20	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$
U15	VDD	VDD	VDD	VDD
A19	PT14D	PT16D	PT18D	I/O
B18	PT14C	PT16C	PT18C	I/O
B17	PT14B	PT16B	PT18B	I/O
C17	PT14A	PT16A	PT18A	I/O
D16	PT13D	PT15D	PT17D	I/O-RDY/ RCLK
A18	PT13C	PT15C	PT17A	I/O
A17	PT13B	PT15B	PT16D	I/O
C16	PT13A	PT15A	PT16C	I/O
B16	PT12D	PT14D	PT16A	I/O
A16	PT12C	PT13D	PT15D	I/O-D7
C15	PT12B	PT13C	PT15A	I/O
D14	PT12A	PT13B	PT14D	I/O
B15	PT11D	PT13A	PT14A	I/O
A15	PT11C	PT12D	PT13D	I/O
C14	PT11B	PT12B	PT13B	I/O-D6
B14	PT11A	PT12A	PT13A	I/O
A14	PT10D	PT11D	PT12D	I/O
C13	PT10C	PT11C	PT12C	I/O
B13	PT10B	PT11B	PT12B	I/O
A13	PT10A	PT11A	PT12A	I/O-D5
D12	PT9D	PT10D	PT11D	I/O
C12	PT9C	PT10C	PT11C	I/O
B12	PT9B	PT10B	PT11B	I/O
A12	PT9A	PT10A	PT11A	I/O-D4
B11	PT8D	PT9D	PT10D	I/O
C11	PT8C	PT9C	PT10C	I/O
A11	PT8B	PT9B	PT10B	I/O
A10	PT8A	PT9A	PT10A	I/O-D3
B10	PT7D	PT8D	PT9D	I/O
C10	PT7C	PT8C	PT9C	I/O
D10	PT7B	PT8B	PT9B	I/O
A9	PT7A	PT8A	PT9A	I/O-D2
B9	PT6D	PT7D	PT8D	I/O-D1
C9	PT6C	PT7C	PT8C	I/O
D9	PT6B	PT7B	PT8B	I/O
A8	PT6A	PT7A	PT8A	I/O-D0/DIN
B8	PT5D	PT6D	PT7D	I/O
C8	PT5C	PT6C	PT7C	I/O
A7	PT5B	PT6B	PT7B	I/O
B7	PT5A	PT6A	PT7A	I/O-DOUT
A6	PT4D	PT5D	PT6D	I/O
C7	PT4C	PT5A	PT6A	I/O
B6	PT4B	PT4D	PT5C	I/O
A5	PT4A	PT4A	PT5A	I/O-TDI
D7	PT3D	PT3D	PT4D	I/O
C6	PT3C	PT3C	PT4A	I/O
B5	PT3B	PT3B	PT3D	I/O
A4	PT3A	PT3A	PT3A	I/O-TMS
C5	PT2D	PT2D	PT2D	I/O
B4	PT2C	PT2C	PT2C	I/O
A3	PT2B	PT2B	PT2B	I/O
D5	PT2A	PT2A	PT2A	I/O
C4	PT1D	PT1D	PT1D	I/O
B3	PT1C	PT1C	PT1C	I/O
B2	PT1B	PT1B	PT1B	I/O
A2	PT1A	PT1A	PT1A	I/O-TCK
C3	RD_DATA /TDO	RD_DATA /TDO	RD_DATA /TDO	RD_DATA/ TDO

Notes:
The W3 pin on the 256-pin PBGA package is unconnected for all devices listed in this table.
The ATT2C08 does not have bond pads connected to the 256-pin PBGA package pins F2 and Y17.
A 4 x 4 array of thermal balls is found at the center of the package. The balls can be attached to the ground plane of the board for enhanced thermal capability (see Table 26) or left unconnected.

Pin Information (continued)

Table 23. ATT2C08, ATT2C10, ATT2C12, ATT2C15, ATT2C26, and ATT2C40 304-Pin SQFP/SQFP-PQ2 Pinout

Pin	2C08 Pad	2C10 Pad	2C12 Pad	2C15 Pad	2C26 Pad	2C40 Pad	Function
1	VSS	VSS	VSS	VSS	VSS	VSS	VSS
2	VDD	VDD	VDD	VDD	VDD	VDD	VDD
3	VSS	VSS	VSS	VSS	VSS	VSS	VSS
4	PL1D	PL1D	PL1D	PL1D	PL1D	PL1D	I/O
5	PL1C	PL1C	PL1C	PL1C	PL1C	PL1A	I/O
6	PL1B	PL1B	PL1B	PL1B	PL1B	PL2D	I/O
7	PL1A	PL1A	PL1A	PL1A	PL1A	PL2A	I/O
8	PL2D	PL2D	PL2D	PL2D	PL2D	PL3D	I/O-A0
9	PL2C	PL2C	PL2C	PL2A	PL2A	PL3A	I/O
10	PL2B	PL2B	PL2B	PL3D	PL3D	PL4D	I/O
11	PL2A	PL2A	PL2A	PL3A	PL3A	PL4A	I/O
12	VSS	VSS	VSS	VSS	VSS	VSS	VSS
13	PL3D	PL3D	PL3D	PL4D	PL4D	PL5D	I/O
14	PL3C	PL3C	PL3A	PL4A	PL4A	PL6D	I/O
15	PL3B	PL3B	PL4D	PL5D	PL5D	PL7D	I/O
16	PL3A	PL3A	PL4A	PL5A	PL5A	PL8D	I/O-A1
17	See Note	PL4D	PL5D	PL6D	PL6D	PL9D	I/O
18	See Note	PL4C	PL5C	PL6C	PL6C	PL9C	I/O
19	See Note	PL4B	PL5B	PL6B	PL6B	PL9B	I/O
20	PL4D	PL4A	PL5A	PL6A	PL6A	PL9A	I/O-A2
21	See Note	PL5D	PL6D	PL7D	PL7D	PL10D	I/O
22	PL4C	PL5C	PL6C	PL7C	PL7C	PL10C	I/O
23	PL4B	PL5B	PL6B	PL7B	PL7B	PL10B	I/O
24	PL4A	PL5A	PL6A	PL7A	PL7A	PL10A	I/O-A3
25	VDD	VDD	VDD	VDD	VDD	VDD	VDD
26	PL5D	PL6D	PL7D	PL8D	PL8D	PL11D	I/O
27	PL5C	PL6C	PL7C	PL8C	PL8A	PL11A	I/O
28	PL5B	PL6B	PL7B	PL8B	PL9D	PL12D	I/O
29	PL5A	PL6A	PL7A	PL8A	PL9A	PL12A	I/O-A4
30	PL6D	PL7D	PL8D	PL9D	PL10D	PL13D	I/O-A5
31	PL6C	PL7C	PL8C	PL9C	PL10A	PL13A	I/O
32	PL6B	PL7B	PL8B	PL9B	PL11D	PL14D	I/O
33	PL6A	PL7A	PL8A	PL9A	PL11A	PL14A	I/O-A6
34	VSS	VSS	VSS	VSS	VSS	VSS	VSS
35	PL7D	PL8D	PL9D	PL10D	PL12D	PL15D	I/O
36	PL7C	PL8C	PL9C	PL10C	PL12C	PL15C	I/O
37	PL7B	PL8B	PL9B	PL10B	PL12B	PL15B	I/O
38	PL7A	PL8A	PL9A	PL10A	PL12A	PL15A	I/O-A7
39	VDD	VDD	VDD	VDD	VDD	VDD	VDD
40	PL8D	PL9D	PL10D	PL11D	PL13D	PL16D	I/O
41	PL8C	PL9C	PL10C	PL11C	PL13C	PL16C	I/O

Note: The ATT2C08 does not have bond pads connected to 304-pin SQFP package pin numbers 17, 18, 19, 21, 57, 59, 64, 89, 91—93, 131, 138, 140, 171, 174, 175, 206—209, 242, 243, 249, 282, 283, 286, and 287.

Pin Information (continued)

Table 23. ATT2C08, ATT2C10, ATT2C12, ATT2C15, ATT2C26, and ATT2C40 304-Pin SQFP/SQFP-PQ2 Pinout (continued)

Pin	2C08 Pad	2C10 Pad	2C12 Pad	2C15 Pad	2C26 Pad	2C40 Pad	Function
42	PL8B	PL9B	PL10B	PL11B	PL13B	PL16B	I/O
43	PL8A	PL9A	PL10A	PL11A	PL13A	PL16A	I/O-A8
44	VSS	VSS	VSS	VSS	VSS	VSS	VSS
45	PL9D	PL10D	PL11D	PL12D	PL14D	PL17D	I/O-A9
46	PL9C	PL10C	PL11C	PL12C	PL14A	PL17A	I/O
47	PL9B	PL10B	PL11B	PL12B	PL15D	PL18D	I/O
48	PL9A	PL10A	PL11A	PL12A	PL15A	PL18A	I/O-A10
49	PL10D	PL11D	PL12D	PL13D	PL16D	PL19D	I/O
50	PL10C	PL11C	PL12C	PL13C	PL16A	PL19A	I/O
51	PL10B	PL11B	PL12B	PL13B	PL17D	PL20D	I/O
52	PL10A	PL11A	PL12A	PL13A	PL17A	PL20A	I/O-A11
53	VDD	VDD	VDD	VDD	VDD	VDD	VDD
54	PL11D	PL12D	PL13D	PL14D	PL18D	PL21D	I/O-A12
55	PL11C	PL12C	PL13B	PL14B	PL18B	PL21B	I/O
56	PL11B	PL12B	PL13A	PL14A	PL18A	PL21A	I/O
57	See Note	PL12A	PL14D	PL15D	PL19D	PL22D	I/O
58	PL11A	PL13D	PL14B	PL15B	PL19B	PL22B	I/O-A13
59	See Note	PL13C	PL14A	PL15A	PL19A	PL22A	I/O
60	PL12D	PL13B	PL15D	PL16D	PL20D	PL23D	I/O
61	PL12C	PL13A	PL15B	PL16B	PL20B	PL24D	I/O
62	PL12B	PL14D	PL15A	PL16A	PL20A	PL25D	I/O
63	PL12A	PL14C	PL16D	PL17D	PL21D	PL25A	I/O-A14
64	See Note	PL14A	PL16A	PL17A	PL21A	PL26A	I/O
65	VSS	VSS	VSS	VSS	VSS	VSS	VSS
66	PL13D	PL15D	PL17D	PL18D	PL22D	PL27D	I/O
67	PL13C	PL15C	PL17C	PL18C	PL22C	PL27C	I/O
68	PL13B	PL15B	PL17B	PL18A	PL22A	PL27A	I/O
69	PL13A	PL15A	PL17A	PL19D	PL23D	PL28D	I/O
70	PL14D	PL16D	PL18D	PL19C	PL23C	PL28C	I/O
71	PL14C	PL16C	PL18C	PL19A	PL23A	PL28A	I/O
72	PL14B	PL16B	PL18B	PL20D	PL24D	PL29A	I/O
73	PL14A	PL16A	PL18A	PL20A	PL24A	PL30A	I/O-A15
74	VSS	VSS	VSS	VSS	VSS	VSS	VSS
75	CCLK	CCLK	CCLK	CCLK	CCLK	CCLK	CCLK
76	VDD	VDD	VDD	VDD	VDD	VDD	VDD
77	VSS	VSS	VSS	VSS	VSS	VSS	VSS
78	VDD	VDD	VDD	VDD	VDD	VDD	VDD
79	VSS	VSS	VSS	VSS	VSS	VSS	VSS
80	PB1A	PB1A	PB1A	PB1A	PB1A	PB1A	I/O-A16
81	PB1B	PB1B	PB1B	PB1C	PB1C	PB2A	I/O
82	PB1C	PB1C	PB1C	PB1D	PB1D	PB2D	I/O

Note: The ATT2C08 does not have bond pads connected to 304-pin SQFP package pin numbers 17, 18, 19, 21, 57, 59, 64, 89, 91—93, 131, 138, 140, 171, 174, 175, 206—209, 242, 243, 249, 282, 283, 286, and 287.

Pin Information (continued)

Table 23. ATT2C08, ATT2C10, ATT2C12, ATT2C15, ATT2C26, and ATT2C40 304-Pin SQFP/SQFP-PQ2 Pinout (continued)

Pin	2C08 Pad	2C10 Pad	2C12 Pad	2C15 Pad	2C26 Pad	2C40 Pad	Function
83	PB1D	PB1D	PB1D	PB2A	PB2A	PB3A	I/O
84	PB2A	PB2A	PB2A	PB2D	PB2D	PB3D	I/O
85	PB2B	PB2B	PB2B	PB3A	PB3A	PB4A	I/O
86	PB2C	PB2C	PB2C	PB3C	PB3C	PB4C	I/O
87	PB2D	PB2D	PB2D	PB3D	PB3D	PB4D	I/O
88	VSS	VSS	VSS	VSS	VSS	VSS	VSS
89	See Note	PB3A	PB3A	PB4A	PB4A	PB5A	I/O
90	PB3A	PB3B	PB3D	PB4D	PB4D	PB5D	I/O-A17
91	See Note	PB3C	PB4A	PB5A	PB5A	PB6A	I/O
92	See Note	PB3D	PB4D	PB5D	PB5D	PB6D	I/O
93	See Note	PB4A	PB5A	PB6A	PB6A	PB7A	I/O
94	PB3B	PB4B	PB5B	PB6B	PB6B	PB7D	I/O
95	PB3C	PB4C	PB5C	PB6C	PB6C	PB8A	I/O
96	PB3D	PB4D	PB5D	PB6D	PB6D	PB8D	I/O
97	PB4A	PB5A	PB6A	PB7A	PB7A	PB9A	I/O
98	PB4B	PB5B	PB6B	PB7B	PB7B	PB9D	I/O
99	PB4C	PB5C	PB6C	PB7C	PB7C	PB10A	I/O
100	PB4D	PB5D	PB6D	PB7D	PB7D	PB10D	I/O
101	VDD	VDD	VDD	VDD	VDD	VDD	VDD
102	PB5A	PB6A	PB7A	PB8A	PB8A	PB11A	I/O
103	PB5B	PB6B	PB7B	PB8B	PB8D	PB11D	I/O
104	PB5C	PB6C	PB7C	PB8C	PB9A	PB12A	I/O
105	PB5D	PB6D	PB7D	PB8D	PB9D	PB12D	I/O
106	PB6A	PB7A	PB8A	PB9A	PB10A	PB13A	I/O
107	PB6B	PB7B	PB8B	PB9B	PB10D	PB13D	I/O
108	PB6C	PB7C	PB8C	PB9C	PB11A	PB14A	I/O
109	PB6D	PB7D	PB8D	PB9D	PB11D	PB14D	I/O
110	VSS	VSS	VSS	VSS	VSS	VSS	VSS
111	PB7A	PB8A	PB9A	PB10A	PB12A	PB15A	I/O
112	PB7B	PB8B	PB9B	PB10B	PB12B	PB15B	I/O
113	PB7C	PB8C	PB9C	PB10C	PB12C	PB15C	I/O
114	PB7D	PB8D	PB9D	PB10D	PB12D	PB15D	I/O
115	VSS	VSS	VSS	VSS	VSS	VSS	VSS
116	PB8A	PB9A	PB10A	PB11A	PB13A	PB16A	I/O
117	PB8B	PB9B	PB10B	PB11B	PB13B	PB16B	I/O
118	PB8C	PB9C	PB10C	PB11C	PB13C	PB16C	I/O
119	PB8D	PB9D	PB10D	PB11D	PB13D	PB16D	I/O
120	VSS	VSS	VSS	VSS	VSS	VSS	VSS
121	PB9A	PB10A	PB11A	PB12A	PB14A	PB17A	I/O
122	PB9B	PB10B	PB11B	PB12B	PB14D	PB17D	I/O
123	PB9C	PB10C	PB11C	PB12C	PB15A	PB18A	I/O

Note: The ATT2C08 does not have bond pads connected to 304-pin SQFP package pin numbers 17, 18, 19, 21, 57, 59, 64, 89, 91—93, 131, 138, 140, 171, 174, 175, 206—209, 242, 243, 249, 282, 283, 286, and 287.

Pin Information (continued)

Table 23. ATT2C08, ATT2C10, ATT2C12, ATT2C15, ATT2C26, and ATT2C40 304-Pin SQFP/SQFP-PQ2 Pinout (continued)

Pin	2C08 Pad	2C10 Pad	2C12 Pad	2C15 Pad	2C26 Pad	2C40 Pad	Function
124	PB9D	PB10D	PB11D	PB12D	PB15D	PB18D	I/O
125	PB10A	PB11A	PB12A	PB13A	PB16A	PB19A	I/O-HDC
126	PB10B	PB11B	PB12B	PB13B	PB16D	PB19D	I/O
127	PB10C	PB11C	PB12C	PB13C	PB17A	PB20A	I/O
128	PB10D	PB11D	PB12D	PB13D	PB17D	PB20D	I/O
129	VDD	VDD	VDD	VDD	VDD	VDD	VDD
130	PB11A	PB12A	PB13A	PB14A	PB18A	PB21A	I/O-$\overline{\text{LDC}}$
131	See Note	PB12B	PB13B	PB14B	PB18B	PB21D	I/O
132	PB11B	PB12C	PB13C	PB14C	PB18C	PB22A	I/O
133	PB11C	PB12D	PB13D	PB14D	PB18D	PB22D	I/O
134	PB11D	PB13A	PB14A	PB15A	PB19A	PB23A	I/O
135	PB12A	PB13B	PB14B	PB15B	PB19B	PB24A	I/O
136	PB12B	PB13C	PB14D	PB15D	PB19D	PB24D	I/O
137	PB12C	PB13D	PB15A	PB16A	PB20A	PB25A	I/O-$\overline{\text{INIT}}$
138	See Note	PB14A	PB15D	PB16D	PB20D	PB25D	I/O
139	PB12D	PB14B	PB16A	PB17A	PB21A	PB26A	I/O
140	See Note	PB14D	PB16D	PB17D	PB21D	PB26D	I/O
141	VSS	VSS	VSS	VSS	VSS	VSS	VSS
142	PB13A	PB15A	PB17A	PB18A	PB22A	PB27A	I/O
143	PB13B	PB15B	PB17B	PB18B	PB22B	PB27B	I/O
144	PB13C	PB15C	PB17C	PB18D	PB22D	PB27D	I/O
145	PB13D	PB15D	PB17D	PB19A	PB23A	PB28A	I/O
146	PB14A	PB16A	PB18A	PB19D	PB23D	PB28D	I/O
147	PB14B	PB16B	PB18B	PB20A	PB24A	PB29A	I/O
148	PB14C	PB16C	PB18C	PB20B	PB24B	PB29D	I/O
149	PB14D	PB16D	PB18D	PB20D	PB24D	PB30D	I/O
150	VSS	VSS	VSS	VSS	VSS	VSS	VSS
151	DONE	DONE	DONE	DONE	DONE	DONE	DONE
152	VDD	VDD	VDD	VDD	VDD	VDD	VDD
153	VSS	VSS	VSS	VSS	VSS	VSS	VSS
154	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$
155	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$
156	PR14A	PR16A	PR18A	PR20A	PR24A	PR30A	I/O-M0
157	PR14B	PR16B	PR18B	PR20C	PR24C	PR29A	I/O
158	PR14C	PR16C	PR18C	PR20D	PR24D	PR29D	I/O
159	PR14D	PR16D	PR18D	PR19A	PR23A	PR28A	I/O
160	PR13A	PR15A	PR17A	PR19D	PR23D	PR28D	I/O
161	PR13B	PR15B	PR17B	PR18A	PR22A	PR27A	I/O
162	PR13C	PR15C	PR17C	PR18B	PR22B	PR27B	I/O
163	PR13D	PR15D	PR17D	PR18D	PR22D	PR27D	I/O
164	VSS	VSS	VSS	VSS	VSS	VSS	VSS

Note: The ATT2C08 does not have bond pads connected to 304-pin SQFP package pin numbers 17, 18, 19, 21, 57, 59, 64, 89, 91—93, 131, 138, 140, 171, 174, 175, 206—209, 242, 243, 249, 282, 283, 286, and 287.

Pin Information (continued)

Table 23. ATT2C08, ATT2C10, ATT2C12, ATT2C15, ATT2C26, and ATT2C40 304-Pin SQFP/SQFP-PQ2 Pinout (continued)

Pin	2C08 Pad	2C10 Pad	2C12 Pad	2C15 Pad	2C26 Pad	2C40 Pad	Function
165	PR12A	PR14A	PR16A	PR17A	PR21A	PR26A	I/O
166	PR12B	PR14C	PR16D	PR17D	PR21D	PR25A	I/O
167	PR12C	PR14D	PR15A	PR16A	PR20A	PR24A	I/O
168	PR12D	PR13A	PR15C	PR16C	PR20C	PR24D	I/O
169	PR11A	PR13B	PR15D	PR16D	PR20D	PR23D	I/O-M1
170	PR11B	PR13C	PR14A	PR15A	PR19A	PR22A	I/O
171	See Note	PR13D	PR14C	PR15C	PR19C	PR22C	I/O
172	PR11C	PR12A	PR14D	PR15D	PR19D	PR22D	I/O
173	PR11D	PR12B	PR13A	PR14A	PR18A	PR21A	I/O
174	See Note	PR12C	PR13C	PR14C	PR18C	PR21C	I/O
175	See Note	PR12D	PR13D	PR14D	PR18D	PR21D	I/O
176	VDD	VDD	VDD	VDD	VDD	VDD	VDD
177	PR10A	PR11A	PR12A	PR13A	PR17A	PR20A	I/O-M2
178	PR10B	PR11B	PR12B	PR13B	PR17D	PR20D	I/O
179	PR10C	PR11C	PR12C	PR13C	PR16A	PR19A	I/O
180	PR10D	PR11D	PR12D	PR13D	PR16D	PR19D	I/O
181	PR9A	PR10A	PR11A	PR12A	PR15A	PR18A	I/O-M3
182	PR9B	PR10B	PR11B	PR12B	PR15D	PR18D	I/O
183	PR9C	PR10C	PR11C	PR12C	PR14A	PR17A	I/O
184	PR9D	PR10D	PR11D	PR12D	PR14D	PR17D	I/O
185	VSS	VSS	VSS	VSS	VSS	VSS	VSS
186	PR8A	PR9A	PR10A	PR11A	PR13A	PR16A	I/O
187	PR8B	PR9B	PR10B	PR11B	PR13B	PR16B	I/O
188	PR8C	PR9C	PR10C	PR11C	PR13C	PR16C	I/O
189	PR8D	PR9D	PR10D	PR11D	PR13D	PR16D	I/O
190	VDD	VDD	VDD	VDD	VDD	VDD	VDD
191	PR7A	PR8A	PR9A	PR10A	PR12A	PR15A	I/O
192	PR7B	PR8B	PR9B	PR10B	PR12B	PR15B	I/O
193	PR7C	PR8C	PR9C	PR10C	PR12C	PR15C	I/O
194	PR7D	PR8D	PR9D	PR10D	PR12D	PR15D	I/O
195	VSS	VSS	VSS	VSS	VSS	VSS	VSS
196	PR6A	PR7A	PR8A	PR9A	PR11A	PR14A	I/O
197	PR6B	PR7B	PR8B	PR9B	PR11D	PR14D	I/O
198	PR6C	PR7C	PR8C	PR9C	PR10A	PR13A	I/O
199	PR6D	PR7D	PR8D	PR9D	PR10D	PR13D	I/O
200	PR5A	PR6A	PR7A	PR8A	PR9A	PR12A	I/O-CS1
201	PR5B	PR6B	PR7B	PR8B	PR9D	PR12D	I/O
202	PR5C	PR6C	PR7C	PR8C	PR8A	PR11A	I/O
203	PR5D	PR6D	PR7D	PR8D	PR8D	PR11D	I/O
204	VDD	VDD	VDD	VDD	VDD	VDD	VDD
205	PR4A	PR5A	PR6A	PR7A	PR7A	PR10A	I/O-$\overline{CS0}$

Note: The ATT2C08 does not have bond pads connected to 304-pin SQFP package pin numbers 17, 18, 19, 21, 57, 59, 64, 89, 91—93, 131, 138, 140, 171, 174, 175, 206—209, 242, 243, 249, 282, 283, 286, and 287.

Pin Information (continued)

Table 23. ATT2C08, ATT2C10, ATT2C12, ATT2C15, ATT2C26, and ATT2C40 304-Pin SQFP/SQFP-PQ2 Pinout (continued)

Pin	2C08 Pad	2C10 Pad	2C12 Pad	2C15 Pad	2C26 Pad	2C40 Pad	Function
206	See Note	PR5B	PR6B	PR7B	PR7B	PR10B	I/O
207	See Note	PR5C	PR6C	PR7C	PR7C	PR10C	I/O
208	See Note	PR5D	PR6D	PR7D	PR7D	PR10D	I/O
209	See Note	PR4A	PR5A	PR6A	PR6A	PR9A	I/O
210	PR4B	PR4B	PR5B	PR6B	PR6B	PR9B	I/O
211	PR4C	PR4C	PR5C	PR6C	PR6C	PR9C	I/O
212	PR4D	PR4D	PR5D	PR6D	PR6D	PR9D	I/O
213	PR3A	PR3A	PR4A	PR5A	PR5A	PR8A	I/O-$\overline{RD}$
214	PR3B	PR3B	PR4B	PR5B	PR5B	PR7A	I/O
215	PR3C	PR3C	PR4D	PR5D	PR5D	PR6A	I/O
216	PR3D	PR3D	PR3A	PR4A	PR4A	PR5A	I/O
217	VSS	VSS	VSS	VSS	VSS	VSS	VSS
218	PR2A	PR2A	PR2A	PR3A	PR3A	PR4A	I/O-$\overline{WR}$
219	PR2B	PR2B	PR2B	PR3B	PR3B	PR4B	I/O
220	PR2C	PR2C	PR2C	PR2A	PR2A	PR3A	I/O
221	PR2D	PR2D	PR2D	PR2D	PR2D	PR3D	I/O
222	PR1A	PR1A	PR1A	PR1A	PR1A	PR2A	I/O
223	PR1B	PR1B	PR1B	PR1B	PR1B	PR2D	I/O
224	PR1C	PR1C	PR1C	PR1C	PR1C	PR1A	I/O
225	PR1D	PR1D	PR1D	PR1D	PR1D	PR1D	I/O
226	VSS	VSS	VSS	VSS	VSS	VSS	VSS
227	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$
228	VDD	VDD	VDD	VDD	VDD	VDD	VDD
229	VSS	VSS	VSS	VSS	VSS	VSS	VSS
230	VDD	VDD	VDD	VDD	VDD	VDD	VDD
231	VSS	VSS	VSS	VSS	VSS	VSS	VSS
232	PT14D	PT16D	PT18D	PT20D	PT24D	PT30D	I/O
233	PT14C	PT16C	PT18C	PT20C	PT24C	PT30A	I/O
234	PT14B	PT16B	PT18B	PT20A	PT24A	PT29A	I/O
235	PT14A	PT16A	PT18A	PT19D	PT23D	PT28D	I/O
236	PT13D	PT15D	PT17D	PT19A	PT23A	PT28A	I/O-RDY/RCLK
237	PT13C	PT15C	PT17C	PT18D	PT22D	PT27D	I/O
238	PT13B	PT15B	PT17B	PT18C	PT22C	PT27C	I/O
239	PT13A	PT15A	PT17A	PT18A	PT22A	PT27A	I/O
240	VSS	VSS	VSS	VSS	VSS	VSS	VSS
241	PT12D	PT14D	PT16D	PT17D	PT21D	PT26D	I/O
242	See Note	PT14B	PT16C	PT17C	PT21C	PT26C	I/O
243	See Note	PT14A	PT16A	PT17A	PT21A	PT26A	I/O
244	PT12C	PT13D	PT15D	PT16D	PT20D	PT25D	I/O-D7
245	PT12B	PT13C	PT15A	PT16A	PT20A	PT25A	I/O
246	PT12A	PT13B	PT14D	PT15D	PT19D	PT24D	I/O

Note: The ATT2C08 does not have bond pads connected to 304-pin SQFP package pin numbers 17, 18, 19, 21, 57, 59, 64, 89, 91—93, 131, 138, 140, 171, 174, 175, 206—209, 242, 243, 249, 282, 283, 286, and 287.

Pin Information (continued)

Table 23. ATT2C08, ATT2C10, ATT2C12, ATT2C15, ATT2C26, and ATT2C40 304-Pin SQFP/SQFP-PQ2 Pinout (continued)

Pin	2C08 Pad	2C10 Pad	2C12 Pad	2C15 Pad	2C26 Pad	2C40 Pad	Function
247	PT11D	PT13A	PT14A	PT15A	PT19A	PT23D	I/O
248	PT11C	PT12D	PT13D	PT14D	PT18D	PT22D	I/O
249	See Note	PT12C	PT13C	PT14C	PT18C	PT22A	I/O
250	PT11B	PT12B	PT13B	PT14B	PT18B	PT21D	I/O-D6
251	PT11A	PT12A	PT13A	PT14A	PT18A	PT21A	I/O
252	VDD	VDD	VDD	VDD	VDD	VDD	VDD
253	PT10D	PT11D	PT12D	PT13D	PT17D	PT20D	I/O
254	PT10C	PT11C	PT12C	PT13C	PT17A	PT20A	I/O
255	PT10B	PT11B	PT12B	PT13B	PT16D	PT19D	I/O
256	PT10A	PT11A	PT12A	PT13A	PT16A	PT19A	I/O-D5
257	PT9D	PT10D	PT11D	PT12D	PT15D	PT18D	I/O
258	PT9C	PT10C	PT11C	PT12C	PT15A	PT18A	I/O
259	PT9B	PT10B	PT11B	PT12B	PT14D	PT17D	I/O
260	PT9A	PT10A	PT11A	PT12A	PT14A	PT17A	I/O-D4
261	VSS	VSS	VSS	VSS	VSS	VSS	VSS
262	PT8D	PT9D	PT10D	PT11D	PT13D	PT16D	I/O
263	PT8C	PT9C	PT10C	PT11C	PT13C	PT16C	I/O
264	PT8B	PT9B	PT10B	PT11B	PT13B	PT16B	I/O
265	PT8A	PT9A	PT10A	PT11A	PT13A	PT16A	I/O-D3
266	VSS	VSS	VSS	VSS	VSS	VSS	VSS
267	PT7D	PT8D	PT9D	PT10D	PT12D	PT15D	I/O
268	PT7C	PT8C	PT9C	PT10C	PT12C	PT15C	I/O
269	PT7B	PT8B	PT9B	PT10B	PT12B	PT15B	I/O
270	PT7A	PT8A	PT9A	PT10A	PT12A	PT15A	I/O-D2
271	VSS	VSS	VSS	VSS	VSS	VSS	VSS
272	PT6D	PT7D	PT8D	PT9D	PT11D	PT14D	I/O-D1
273	PT6C	PT7C	PT8C	PT9C	PT11A	PT14A	I/O
274	PT6B	PT7B	PT8B	PT9B	PT10D	PT13D	I/O
275	PT6A	PT7A	PT8A	PT9A	PT10A	PT13A	I/O-D0/DIN
276	PT5D	PT6D	PT7D	PT8D	PT9D	PT12D	I/O
277	PT5C	PT6C	PT7C	PT8C	PT9A	PT12A	I/O
278	PT5B	PT6B	PT7B	PT8B	PT8D	PT11D	I/O
279	PT5A	PT6A	PT7A	PT8A	PT8A	PT11A	I/O-DOUT
280	VDD	VDD	VDD	VDD	VDD	VDD	VDD
281	PT4D	PT5D	PT6D	PT7D	PT7D	PT10D	I/O
282	See Note	PT5C	PT6C	PT7C	PT7C	PT10A	I/O
283	See Note	PT5B	PT6B	PT7B	PT7B	PT9D	I/O
284	PT4C	PT5A	PT6A	PT7A	PT7A	PT9A	I/O
285	PT4B	PT4D	PT5D	PT6D	PT6D	PT8D	I/O
286	See Note	PT4C	PT5C	PT6C	PT6C	PT8A	I/O
287	See Note	PT4B	PT5B	PT6B	PT6B	PT7D	I/O

Note: The ATT2C08 does not have bond pads connected to 304-pin SQFP package pin numbers 17, 18, 19, 21, 57, 59, 64, 89, 91—93, 131, 138, 140, 171, 174, 175, 206—209, 242, 243, 249, 282, 283, 286, and 287.

Pin Information (continued)

Table 23. ATT2C08, ATT2C10, ATT2C12, ATT2C15, ATT2C26, and ATT2C40 304-Pin SQFP/SQFP-PQ2 Pinout (continued)

Pin	2C08 Pad	2C10 Pad	2C12 Pad	2C15 Pad	2C26 Pad	2C40 Pad	Function
288	PT4A	PT4A	PT5A	PT6A	PT6A	PT7A	I/O-TDI
289	PT3D	PT3D	PT4D	PT5D	PT5D	PT6D	I/O
290	PT3C	PT3C	PT4A	PT5A	PT5A	PT6A	I/O
291	PT3B	PT3B	PT3D	PT4D	PT4D	PT5D	I/O
292	PT3A	PT3A	PT3A	PT4A	PT4A	PT5A	I/O-TMS
293	VSS	VSS	VSS	VSS	VSS	VSS	VSS
294	PT2D	PT2D	PT2D	PT3D	PT3D	PT4D	I/O
295	PT2C	PT2C	PT2C	PT3A	PT3A	PT4A	I/O
296	PT2B	PT2B	PT2B	PT2D	PT2D	PT3D	I/O
297	PT2A	PT2A	PT2A	PT2A	PT2A	PT3A	I/O
298	PT1D	PT1D	PT1D	PT1D	PT1D	PT2D	I/O
299	PT1C	PT1C	PT1C	PT1C	PT1C	PT1C	I/O
300	PT1B	PT1B	PT1B	PT1B	PT1B	PT1D	I/O
301	PT1A	PT1A	PT1A	PT1A	PT1A	PT1A	I/O-TCK
302	VSS	VSS	VSS	VSS	VSS	VSS	VSS
303	RD_DATA/ TDO	RD_DATA/ TDO	RD_DATA/TDO	RD_DATA/TDO	RD_DATA/TDO	RD_DATA/TDO	RD_DATA/TDO
304	VDD	VDD	VDD	VDD	VDD	VDD	VDD

Note: The ATT2C08 does not have bond pads connected to 304-pin SQFP package pin numbers 17, 18, 19, 21, 57, 59, 64, 89, 91—93, 131, 138, 140, 171, 174, 175, 206—209, 242, 243, 249, 282, 283, 286, and 287.

Pin Information (continued)

Table 24. ATT2C12 and ATT2C15 364-Pin CPGA Pinout

Pin	2C12 Pad	2C15 Pad	Function	Pin	2C12 Pad	2C15 Pad	Function
G25	VSS	VSS	VSS	A19	PL9D	PL10D	I/O
J27	VDD	VDD	VDD	D18	PL9C	PL10C	I/O
G23	VSS	VSS	VSS	B18	PL9B	PL10B	I/O
E29	PL1D	PL1D	I/O	C17	PL9A	PL10A	I/O-A7
B32	PL1C	PL1C	I/O	G7	VDD	VDD	VDD
D28	PL1B	PL1B	I/O	E17	PL10D	PL11D	I/O
A33	PL1A	PL1A	I/O	A17	PL10C	PL11C	I/O
C29	PL2D	PL2D	I/O-A0	D16	PL10B	PL11B	I/O
E27	See Note	PL2C	I/O	B16	PL10A	PL11A	I/O-A8
B30	See Note	PL2B	I/O	G15	VSS	VSS	VSS
F26	PL2C	PL2A	I/O	A15	PL11D	PL12D	I/O-A9
A31	PL2B	PL3D	I/O	F16	PL11C	PL12C	I/O
D26	See Note	PL3C	I/O	B14	PL11B	PL12B	I/O
C27	See Note	PL3B	I/O	C15	PL11A	PL12A	I/O-A10
E25	PL2A	PL3A	I/O	A13	PL12D	PL13D	I/O
G21	VSS	VSS	VSS	E15	PL12C	PL13C	I/O
B28	PL3D	PL4D	I/O	A11	PL12B	PL13B	I/O
F24	PL3C	PL4C	I/O	D14	PL12A	PL13A	I/O-A11
A29	PL3B	PL4B	I/O	C13	PL13D	PL14D	I/O-A12
D24	PL3A	PL4A	I/O	B12	PL13C	PL14C	I/O
C25	PL4D	PL5D	I/O	F14	PL13B	PL14B	I/O
E23	PL4C	PL5C	I/O	A9	PL13A	PL14A	I/O
B26	PL4B	PL5B	I/O	E13	PL14D	PL15D	I/O
F22	PL4A	PL5A	I/O-A1	B10	PL14C	PL15C	I/O
G19	VSS	VSS	VSS	D12	PL14B	PL15B	I/O-A13
D22	PL5D	PL6D	I/O	C11	PL14A	PL15A	I/O
A27	PL5C	PL6C	I/O	G13	VSS	VSS	VSS
E21	PL5B	PL6B	I/O	F12	PL15D	PL16D	I/O
C23	PL5A	PL6A	I/O-A2	A7	PL15C	PL16C	I/O
F20	PL6D	PL7D	I/O	E11	PL15B	PL16B	I/O
B24	PL6C	PL7C	I/O	B8	PL15A	PL16A	I/O
C21	PL6B	PL7B	I/O	D10	PL16D	PL17D	I/O-A14
A25	PL6A	PL7A	I/O-A3	C9	PL16C	PL17C	I/O
G27	VDD	VDD	VDD	F10	PL16B	PL17B	I/O
B22	PL7D	PL8D	I/O	A5	PL16A	PL17A	I/O
D20	PL7C	PL8C	I/O	G11	VSS	VSS	VSS
A23	PL7B	PL8B	I/O	E9	PL17D	PL18D	I/O
E19	PL7A	PL8A	I/O-A4	B6	PL17C	PL18C	I/O
A21	PL8D	PL9D	I/O-A5	D8	See Note	PL18B	I/O
C19	PL8C	PL9C	I/O	C7	PL17B	PL18A	I/O
B20	PL8B	PL9B	I/O	F8	PL17A	PL19D	I/O
F18	PL8A	PL9A	I/O-A6	A3	PL18D	PL19C	I/O
G17	VSS	VSS	VSS	E7	See Note	PL19B	I/O

Notes: The ATT2C12 does not have bond pads connected to 364-pin CPGA package pin numbers E27, B30, D26, C27, D8, E7, D6, C3, B2, G5, D2, H4, AF4, AL1, AG5, AL3, AK6, AM4, AH8, AJ9, AN31, AK26, AH26, AL29, AH30, AG29, AK32, AG31, F32, H30, C33, and G29.The J9 pin is used for package orientation only.The ceramic PGA contains single large VDD and VSS planes to which all VDD and VSS bond pads are connected.

Pin Information (continued)

Table 24. ATT2C12 and ATT2C15 364-Pin CPGA Pinout (continued)

Pin	2C12 Pad	2C15 Pad	Function	Pin	2C12 Pad	2C15 Pad	Function
B4	PL18C	PL19A	I/O	L1	PB7C	PB8C	I/O
C5	PL18B	PL20D	I/O	R5	PB7D	PB8D	I/O
D6	See Note	PL20C	I/O	N1	PB8A	PB9A	I/O
C3	See Note	PL20B	I/O	R3	PB8B	PB9B	I/O
F6	PL18A	PL20A	I/O-A15	P2	PB8C	PB9C	I/O
G9	VSS	VSS	VSS	T6	PB8D	PB9D	I/O
D4	CCLK	CCLK	CCLK	R1	PB9A	PB10A	I/O
J7	VDD	VDD	VDD	T4	PB9B	PB10B	I/O
L7	VDD	VDD	VDD	T2	PB9C	PB10C	I/O
R7	VSS	VSS	VSS	U3	PB9D	PB10D	I/O
E5	PB1A	PB1A	I/O-A16	U7	VSS	VSS	VSS
B2	See Note	PB1B	I/O	U5	PB10A	PB11A	I/O
F4	PB1B	PB1C	I/O	U1	PB10B	PB11B	I/O
A1	PB1C	PB1D	I/O	V4	PB10C	PB11C	I/O
E3	PB1D	PB2A	I/O	V2	PB10D	PB11D	I/O
G5	See Note	PB2B	I/O	W1	PB11A	PB12A	I/O
D2	See Note	PB2C	I/O	V6	PB11B	PB12B	I/O
H6	PB2A	PB2D	I/O	Y2	PB11C	PB12C	I/O
C1	PB2B	PB3A	I/O	W3	PB11D	PB12D	I/O
H4	See Note	PB3B	I/O	AA1	PB12A	PB13A	I/O-HDC
G3	PB2C	PB3C	I/O	W5	PB12B	PB13B	I/O
J5	PB2D	PB3D	I/O	AC1	PB12C	PB13C	I/O
F2	PB3A	PB4A	I/O	Y4	PB12D	PB13D	I/O
K6	PB3B	PB4B	I/O	AA7	VDD	VDD	VDD
E1	PB3C	PB4C	I/O	AA3	PB13A	PB14A	I/O-$\overline{\text{LDC}}$
K4	PB3D	PB4D	I/O-A17	AB2	PB13B	PB14B	I/O
J3	PB4A	PB5A	I/O	Y6	PB13C	PB14C	I/O
L5	PB4B	PB5B	I/O	AE1	PB13D	PB14D	I/O
H2	PB4C	PB5C	I/O	AA5	PB14A	PB15A	I/O
M6	PB4D	PB5D	I/O	AD2	PB14B	PB15B	I/O
M4	PB5A	PB6A	I/O	AB4	PB14C	PB15C	I/O
G1	PB5B	PB6B	I/O	AC3	PB14D	PB15D	I/O
N5	PB5C	PB6C	I/O	AB6	PB15A	PB16A	I/O-$\overline{\text{INIT}}$
L3	PB5D	PB6D	I/O	AG1	PB15B	PB16B	I/O
P6	PB6A	PB7A	I/O	AC5	PB15C	PB16C	I/O
K2	PB6B	PB7B	I/O	AF2	PB15D	PB16D	I/O
N3	PB6C	PB7C	I/O	AD4	PB16A	PB17A	I/O
J1	PB6D	PB7D	I/O	AE3	PB16B	PB17B	I/O
N7	VDD	VDD	VDD	AD6	PB16C	PB17C	I/O
M2	PB7A	PB8A	I/O	AJ1	PB16D	PB17D	I/O
P4	PB7B	PB8B	I/O	AE5	PB17A	PB18A	I/O

Notes: The ATT2C12 does not have bond pads connected to 364-pin CPGA package pin numbers E27, B30, D26, C27, D8, E7, D6, C3, B2, G5, D2, H4, AF4, AL1, AG5, AL3, AK6, AM4, AH8, AJ9, AN31, AK26, AH26, AL29, AH30, AG29, AK32, AG31, F32, H30, C33, and G29. The J9 pin is used for package orientation only. The ceramic PGA contains single large VDD and VSS planes to which all VDD and VSS bond pads are connected.

Pin Information (continued)

Table 24. ATT2C12 and ATT2C15 364-Pin CPGA Pinout (continued)

Pin	2C12 Pad	2C15 Pad	Function	Pin	2C12 Pad	2C15 Pad	Function
AH2	PB17B	PB18B	I/O	AM12	PR13C	PR14C	I/O
AF4	See Note	PB18C	I/O	AL13	PR13D	PR14D	I/O
AG3	PB17C	PB18D	I/O	AE7	VDD	VDD	VDD
AF6	PB17D	PB19A	I/O	AK14	PR12A	PR13A	I/O-M2
AL1	See Note	PB19B	I/O	AN11	PR12B	PR13B	I/O
AG5	See Note	PB19C	I/O	AJ15	PR12C	PR13C	I/O
AK2	PB18A	PB19D	I/O	AN13	PR12D	PR13D	I/O
AJ3	PB18B	PB20A	I/O	AL15	PR11A	PR12A	I/O-M3
AH4	PB18C	PB20B	I/O	AM14	PR11B	PR12B	I/O
AL3	See Note	PB20C	I/O	AH16	PR11C	PR12C	I/O
AH6	PB18D	PB20D	I/O	AN15	PR11D	PR12D	I/O
W7	VSS	VSS	VSS	AG15	VSS	VSS	VSS
AK4	DONE	DONE	DONE	AK16	PR10A	PR11A	I/O
AC7	VDD	VDD	VDD	AM16	PR10B	PR11B	I/O
AG11	VSS	VSS	VSS	AJ17	PR10C	PR11C	I/O
AM2	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	AN17	PR10D	PR11D	I/O
AJ5	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	AG7	VDD	VDD	VDD
AN1	PR18A	PR20A	I/O-M0	AL17	PR9A	PR10A	I/O
AK6	See Note	PR20B	I/O	AK18	PR9B	PR10B	I/O
AL5	PR18B	PR20C	I/O	AM18	PR9C	PR10C	I/O
AJ7	PR18C	PR20D	I/O	AN19	PR9D	PR10D	I/O
AG9	PR18D	PR19A	I/O	AG17	VSS	VSS	VSS
AM4	See Note	PR19B	I/O	AH18	PR8A	PR9A	I/O
AH8	See Note	PR19C	I/O	AM20	PR8B	PR9B	I/O
AN3	PR17A	PR19D	I/O	AL19	PR8C	PR9C	I/O
AK8	PR17B	PR18A	I/O	AN21	PR8D	PR9D	I/O
AL7	PR17C	PR18B	I/O	AJ19	PR7A	PR8A	I/O-CS1
AJ9	See Note	PR18C	I/O	AN23	PR7B	PR8B	I/O
AM6	PR17D	PR18D	I/O	AK20	PR7C	PR8C	I/O
AG13	VSS	VSS	VSS	AM22	PR7D	PR8D	I/O
AH10	PR16A	PR17A	I/O	AG27	VDD	VDD	VDD
AN5	PR16B	PR17B	I/O	AN25	PR6A	PR7A	I/O-$\overline{\text{CS0}}$
AK10	PR16C	PR17C	I/O	AL21	PR6B	PR7B	I/O
AL9	PR16D	PR17D	I/O	AM24	PR6C	PR7C	I/O
AJ11	PR15A	PR16A	I/O	AH20	PR6D	PR7D	I/O
AM8	PR15B	PR16B	I/O	AL23	PR5A	PR6A	I/O
AH12	PR15C	PR16C	I/O	AJ21	PR5B	PR6B	I/O
AN7	PR15D	PR16D	I/O-M1	AN27	PR5C	PR6C	I/O
AL11	PR14A	PR15A	I/O	AK22	PR5D	PR6D	I/O
AK12	PR14B	PR15B	I/O	AG19	VSS	VSS	VSS
AM10	PR14C	PR15C	I/O	AM26	PR4A	PR5A	I/O-$\overline{\text{RD}}$
AJ13	PR14D	PR15D	I/O	AH22	PR4B	PR5B	I/O
AN9	PR13A	PR14A	I/O	AL25	PR4C	PR5C	I/O
AH14	PR13B	PR14B	I/O	AJ23	PR4D	PR5D	I/O

Notes: The ATT2C12 does not have bond pads connected to 364-pin CPGA package pin numbers E27, B30, D26, C27, D8, E7, D6, C3, B2, G5, D2, H4, AF4, AL1, AG5, AL3, AK6, AM4, AH8, AJ9, AN31, AK26, AH26, AL29, AH30, AG29, AK32, AG31, F32, H30, C33, and G29.The J9 pin is used for package orientation only.The ceramic PGA contains single large VDD and VSS planes to which all VDD and VSS bond pads are connected.

Pin Information (continued)

Table 24. ATT2C12 and ATT2C15 364-Pin CPGA Pinout (continued)

Pin	2C12 Pad	2C15 Pad	Function	Pin	2C12 Pad	2C15 Pad	Function
AN29	PR3A	PR4A	I/O	AA29	PT14B	PT15B	I/O
AK24	PR3B	PR4B	I/O	AC31	PT14A	PT15A	I/O
AM28	PR3C	PR4C	I/O	Y28	PT13D	PT14D	I/O
AH24	PR3D	PR4D	I/O	AD32	PT13C	PT14C	I/O
AG21	VSS	VSS	VSS	AA31	PT13B	PT14B	I/O-D6
AL27	PR2A	PR3A	I/O-$\overline{WR}$	AE33	PT13A	PT14A	I/O
AJ25	PR2B	PR3B	I/O	AA27	VDD	VDD	VDD
AN31	See Note	PR3C	I/O	AB32	PT12D	PT13D	I/O
AK26	See Note	PR3D	I/O	Y30	PT12C	PT13C	I/O
AM30	PR2C	PR2A	I/O	AC33	PT12B	PT13B	I/O
AH26	See Note	PR2B	I/O	W29	PT12A	PT13A	I/O-D5
AL29	See Note	PR2C	I/O	AA33	PT11D	PT12D	I/O
AG25	PR2D	PR2D	I/O	W31	PT11C	PT12C	I/O
AJ27	PR1A	PR1A	I/O	Y32	PT11B	PT12B	I/O
AL31	PR1B	PR1B	I/O	V28	PT11A	PT12A	I/O-D4
AK28	PR1C	PR1C	I/O	W33	PT10D	PT11D	I/O
AK30	PR1D	PR1D	I/O	V30	PT10C	PT11C	I/O
AG23	VSS	VSS	VSS	V32	PT10B	PT11B	I/O
AH28	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	U31	PT10A	PT11A	I/O-D3
AE27	VDD	VDD	VDD	U27	VSS	VSS	VSS
AC27	VDD	VDD	VDD	U29	PT9D	PT10D	I/O
W27	VSS	VSS	VSS	U33	PT9C	PT10C	I/O
AJ29	PT18D	PT20D	I/O	T30	PT9B	PT10B	I/O
AM32	PT18C	PT20C	I/O	T32	PT9A	PT10A	I/O-D2
AH30	See Note	PT20B	I/O	R33	PT8D	PT9D	I/O-D1
AN33	PT18B	PT20A	I/O	T28	PT8C	PT9C	I/O
AJ31	PT18A	PT19D	I/O	P32	PT8B	PT9B	I/O
AG29	See Note	PT19C	I/O	R31	PT8A	PT9A	I/O-D0/DIN
AK32	See Note	PT19B	I/O	N33	PT7D	PT8D	I/O
AF28	PT17D	PT19A	I/O-RDY/RCLK	R29	PT7C	PT8C	I/O
AL33	PT17C	PT18D	I/O	L33	PT7B	PT8B	I/O
AF30	PT17B	PT18C	I/O	P30	PT7A	PT8A	I/O-DOUT
AG31	See Note	PT18B	I/O	N27	VDD	VDD	VDD
AE29	PT17A	PT18A	I/O	N31	PT6D	PT7D	I/O
AH32	PT16D	PT17D	I/O	M32	PT6C	PT7C	I/O
AD28	PT16C	PT17C	I/O	P28	PT6B	PT7B	I/O
AJ33	PT16B	PT17B	I/O	J33	PT6A	PT7A	I/O
AD30	PT16A	PT17A	I/O	N29	PT5D	PT6D	I/O
AE31	PT15D	PT16D	I/O-D7	K32	PT5C	PT6C	I/O
AC29	PT15C	PT16C	I/O	M30	PT5B	PT6B	I/O
AF32	PT15B	PT16B	I/O	L31	PT5A	PT6A	I/O-TDI
AB28	PT15A	PT16A	I/O	M28	PT4D	PT5D	I/O
AB30	PT14D	PT15D	I/O	G33	PT4C	PT5C	I/O

Notes: The ATT2C12 does not have bond pads connected to 364-pin CPGA package pin numbers E27, B30, D26, C27, D8, E7, D6, C3, B2, G5, D2, H4, AF4, AL1, AG5, AL3, AK6, AM4, AH8, AJ9, AN31, AK26, AH26, AL29, AH30, AG29, AK32, AG31, F32, H30, C33, and G29.The J9 pin is used for package orientation only.The ceramic PGA contains single large VDD and VSS planes to which all VDD and VSS bond pads are connected.

Pin Information (continued)

Table 24. ATT2C12 and ATT2C15 364-Pin CPGA Pinout (continued)

Pin	2C12 Pad	2C15 Pad	Function	Pin	2C12 Pad	2C15 Pad	Function
AG33	PT14C	PT15C	I/O	L29	PT4B	PT5B	I/O
H32	PT4A	PT5A	I/O	C33	See Note	PT2C	I/O
K30	PT3D	PT4D	I/O	G29	See Note	PT2B	I/O
J31	PT3C	PT4C	I/O	D32	PT2A	PT2A	I/O
K28	PT3B	PT4B	I/O	E31	PT1D	PT1D	I/O
E33	PT3A	PT4A	I/O-TMS	F30	PT1C	PT1C	I/O
J29	PT2D	PT3D	I/O	C31	PT1B	PT1B	I/O
F32	See Note	PT3C	I/O	F28	PT1A	PT1A	I/O-TCK
H30	See Note	PT3B	I/O	R27	VSS	VSS	VSS
G31	PT2C	PT3A	I/O	D30	RD_DATA/TDO	RD_DATA/TDO	RD_DATA/TDO
H28	PT2B	PT2D	I/O	L27	VDD	VDD	VDD

Notes: The ATT2C12 does not have bond pads connected to 364-pin CPGA package pin numbers E27, B30, D26, C27, D8, E7, D6, C3, B2, G5, D2, H4, AF4, AL1, AG5, AL3, AK6, AM4, AH8, AJ9, AN31, AK26, AH26, AL29, AH30, AG29, AK32, AG31, F32, H30, C33, and G29.The J9 pin is used for package orientation only.The ceramic PGA contains single large VDD and VSS planes to which all VDD and VSS bond pads are connected.

Pin Information (continued)

Table 25. ATT2C26 and ATT2C40 428-Pin CPGA Pinout

Pin	2C26 Pad	2C40 Pad	Function	Pin	2C26 Pad	2C40 Pad	Function
D34	VSS	VSS	VSS	B22	PL10B	PL13B	I/O
AL33	VDD	VDD	VDD	F20	PL10A	PL13A	I/O
E33	VSS	VSS	VSS	C21	PL11D	PL14D	I/O
C33	PL1D	PL1D	I/O	D20	PL11C	PL14C	I/O
D32	PL1C	PL1A	I/O	A21	PL11B	PL14B	I/O
B32	PL1B	PL2D	I/O	G19	PL11A	PL14A	I/O-A6
E29	PL1A	PL2A	I/O	F32	VSS	VSS	VSS
F28	PL2D	PL3D	I/O-A0	B20	PL12D	PL15D	I/O
C31	PL2C	PL3C	I/O	F18	PL12C	PL15C	I/O
G27	PL2B	PL3B	I/O	C19	PL12B	PL15B	I/O
A31	PL2A	PL3A	I/O	E19	PL12A	PL15A	I/O-A7
H26	PL3D	PL4D	I/O	H18	VDD	VDD	VDD
D30	PL3C	PL4C	I/O	E17	PL13D	PL16D	I/O
D28	PL3B	PL4B	I/O	A19	PL13C	PL16C	I/O
B30	PL3A	PL4A	I/O	D18	PL13B	PL16B	I/O
F26	PL4D	PL5D	I/O	B18	PL13A	PL16A	I/O-A8
C29	PL4C	PL5C	I/O	G31	VSS	VSS	VSS
G25	PL4B	PL5B	I/O	D14	PL14D	PL17D	I/O-A9
A29	PL4A	PL6D	I/O	A17	PL14C	PL17C	I/O
E27	PL5D	PL7D	I/O	G17	PL14B	PL17B	I/O
B28	PL5C	PL7C	I/O	C17	PL14A	PL17A	I/O
H24	PL5B	PL7B	I/O	F16	PL15D	PL18D	I/O
C27	PL5A	PL8D	I/O-A1	B16	PL15C	PL18C	I/O
E25	PL6D	PL9D	I/O	E15	PL15B	PL18B	I/O
A27	PL6C	PL9C	I/O	D16	PL15A	PL18A	I/O-A10
G23	PL6B	PL9B	I/O	E13	PL16D	PL19D	I/O
D26	PL6A	PL9A	I/O-A2	A15	PL16C	PL19C	I/O
F24	PL7D	PL10D	I/O	F14	PL16B	PL19B	I/O
B26	PL7C	PL10C	I/O	C15	PL16A	PL19A	I/O
D24	PL7B	PL10B	I/O	H16	PL17D	PL20D	I/O
C25	PL7A	PL10A	I/O-A3	B14	PL17C	PL20C	I/O
H22	VDD	VDD	VDD	G15	PL17B	PL20B	I/O
A25	PL8D	PL11D	I/O	A13	PL17A	PL20A	I/O-A11
E23	PL8C	PL11C	I/O	H14	VDD	VDD	VDD
B24	PL8B	PL11B	I/O	C13	PL18D	PL21D	I/O-A12
F22	PL8A	PL11A	I/O	D10	PL18C	PL21C	I/O
C23	PL9D	PL12D	I/O	B12	PL18B	PL21B	I/O
G21	PL9C	PL12C	I/O	E11	PL18A	PL21A	I/O
A23	PL9B	PL12B	I/O	D12	PL19D	PL22D	I/O
H20	PL9A	PL12A	I/O-A4	F12	PL19C	PL22C	I/O
D22	PL10D	PL13D	I/O-A5	A11	PL19B	PL22B	I/O-A13
E21	PL10C	PL13C	I/O	G13	PL19A	PL22A	I/O

Note: The ceramic PGA contains single large VDD and VSS planes to which all VDD and VSS bond pads are connected.

Pin Information (continued)

Table 25. ATT2C26 and ATT2C40 428-Pin CPGA Pinout (continued)

Pin	2C26 Pad	2C40 Pad	Function	Pin	2C26 Pad	2C40 Pad	Function
C11	PL20D	PL23D	I/O	L3	PB5C	PB6C	I/O
E9	PL20C	PL23C	I/O	M6	PB5D	PB6D	I/O
B10	PL20B	PL24D	I/O	L1	PB6A	PB7A	I/O
H12	PL20A	PL25D	I/O	N7	PB6B	PB7D	I/O
A9	PL21D	PL25A	I/O-A14	M4	PB6C	PB8A	I/O
F10	PL21C	PL26C	I/O	N5	PB6D	PB8D	I/O
C9	PL21B	PL26B	I/O	M2	PB7A	PB9A	I/O
G11	PL21A	PL26A	I/O	P6	PB7B	PB9D	I/O
B8	PL22D	PL27D	I/O	N3	PB7C	PB10A	I/O
E7	PL22C	PL27C	I/O	P4	PB7D	PB10D	I/O
D8	PL22B	PL27B	I/O	P8	VDD	VDD	VDD
F8	PL22A	PL27A	I/O	R7	PB8A	PB11A	I/O
A7	PL23D	PL28D	I/O	N1	PB8B	PB11B	I/O
G9	PL23C	PL28C	I/O	T8	PB8C	PB11C	I/O
C7	PL23B	PL28B	I/O	P2	PB8D	PB11D	I/O
H10	PL23A	PL28A	I/O	R5	PB9A	PB12A	I/O
D6	PL24D	PL29A	I/O	R3	PB9B	PB12B	I/O
B6	PL24C	PL30C	I/O	T6	PB9C	PB12C	I/O
F4	PL24B	PL30B	I/O	R1	PB9D	PB12D	I/O
C5	PL24A	PL30A	I/O-A15	T4	PB10A	PB13A	I/O
H30	VSS	VSS	VSS	U7	PB10B	PB13B	I/O
G5	CCLK	CCLK	CCLK	T2	PB10C	PB13C	I/O
AM34	VDD	VDD	VDD	U5	PB10D	PB13D	I/O
AN35	VDD	VDD	VDD	U3	PB11A	PB14A	I/O
D4	VSS	VSS	VSS	V4	PB11B	PB14B	I/O
H6	PB1A	PB1A	I/O-A16	U1	PB11C	PB14C	I/O
E3	PB1B	PB1B	I/O	V6	PB11D	PB14D	I/O
J7	PB1C	PB2A	I/O	E5	VSS	VSS	VSS
F2	PB1D	PB2D	I/O	V2	PB12A	PB15A	I/O
G3	PB2A	PB3A	I/O	W5	PB12B	PB15B	I/O
J5	PB2B	PB3B	I/O	W3	PB12C	PB15C	I/O
G1	PB2C	PB3C	I/O	W7	PB12D	PB15D	I/O
K8	PB2D	PB3D	I/O	F6	VSS	VSS	VSS
H4	PB3A	PB4A	I/O	W1	PB13A	PB16A	I/O
K6	PB3B	PB4B	I/O	Y4	PB13B	PB16B	I/O
H2	PB3C	PB4C	I/O	Y2	PB13C	PB16C	I/O
K4	PB3D	PB4D	I/O	Y6	PB13D	PB16D	I/O
J3	PB4A	PB5A	I/O	G7	VSS	VSS	VSS
L7	PB4B	PB5B	I/O	AA1	PB14A	PB17A	I/O
J1	PB4C	PB5C	I/O	Y8	PB14B	PB17B	I/O
M8	PB4D	PB5D	I/O-A17	AA3	PB14C	PB17C	I/O
K2	PB5A	PB6A	I/O	AA5	PB14D	PB17D	I/O
L5	PB5B	PB6B	I/O	AB2	PB15A	PB18A	I/O

Note: The ceramic PGA contains single large VDD and VSS planes to which all VDD and VSS bond pads are connected.

Pin Information (continued)

Table 25. ATT2C26 and ATT2C40 428-Pin CPGA Pinout (continued)

Pin	2C26 Pad	2C40 Pad	Function	Pin	2C26 Pad	2C40 Pad	Function
AA7	PB15B	PB18B	I/O	AR5	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$
AB4	PB15C	PB18C	I/O	AP6	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$	$\overline{\text{PRGM}}$
AB6	PB15D	PB18D	I/O	AT6	PR24A	PR30A	I/O-M0
AC5	PB16A	PB19A	I/O-HDC	AN7	PR24B	PR30B	I/O
AC1	PB16B	PB19B	I/O	AR7	PR24C	PR29A	I/O
AD4	PB16C	PB19C	I/O	AM8	PR24D	PR29D	$\overline{\text{I/O}}$
AC3	PB16D	PB19D	I/O	AK32	VDD	VDD	VDD
AD6	PB17A	PB20A	I/O	AK10	PR23A	PR28A	I/O
AD2	PB17B	PB20B	I/O	AU7	PR23B	PR28B	I/O
AC7	PB17C	PB20C	I/O	AL9	PR23C	PR28C	I/O
AE1	PB17D	PB20D	I/O	AP8	PR23D	PR28D	I/O
V8	VDD	VDD	VDD	AN9	PR22A	PR27A	I/O
AE3	PB18A	PB21A	I/O-$\overline{\text{LDC}}$	AT8	PR22B	PR27B	I/O
AE5	PB18B	PB21D	I/O	AL11	PR22C	PR27C	I/O
AF2	PB18C	PB22A	I/O	AR9	PR22D	PR27D	I/O
AG5	PB18D	PB22D	I/O	AP4	VSS	VSS	VSS
AF4	PB19A	PB23A	I/O	AK12	PR21A	PR26A	I/O
AF6	PB19B	PB24A	I/O	AU9	PR21B	PR26B	I/O
AG1	PB19C	PB24C	I/O	AM10	PR21C	PR26C	I/O
AD8	PB19D	PB24D	I/O	AT10	PR21D	PR25A	I/O
AG3	PB20A	PB25A	I/O-$\overline{\text{INIT}}$	AP10	PR20A	PR24A	I/O
AE7	PB20B	PB25B	I/O	AR11	PR20B	PR24B	I/O
AH2	PB20C	PB25C	I/O	AL13	PR20C	PR24D	I/O
AH4	PB20D	PB25D	I/O	AU11	PR20D	PR23D	I/O-M1
AJ1	PB21A	PB26A	I/O	AK14	PR19A	PR22A	I/O
AH6	PB21B	PB26B	I/O	AP12	PR19B	PR22B	I/O
AJ3	PB21C	PB26C	I/O	AM12	PR19C	PR22C	I/O
AF8	PB21D	PB26D	I/O	AT12	PR19D	PR22D	I/O
AK2	PB22A	PB27A	I/0	AN11	PR18A	PR21A	I/O
AG7	PB22B	PB27B	I/O	AR13	PR18B	PR21B	I/O
AK4	PB22C	PB27C	I/O	AN13	PR18C	PR21C	I/O
AJ5	PB22D	PB27D	I/O	AU13	PR18D	PR21D	I/O
AL1	PB23A	PB28A	I/O	AK16	VDD	VDD	VDD
AJ7	PB23B	PB28B	I/O	AT14	PR17A	PR20A	I/O-M2
AL3	PB23C	PB28C	I/O	AL15	PR17B	PR20B	I/O
AH8	PB23D	PB28D	I/O	AR15	PR17C	PR20C	I/O
AK6	PB24A	PB29A	I/O	AM14	PR17D	PR20D	I/O
AM2	PB24B	PB29D	I/O	AU15	PR16A	PR19A	I/O
AL5	PB24C	PB30C	I/O	AP14	PR16B	PR19B	I/O
AN3	PB24D	PB30D	I/O	AP16	PR16C	PR19C	I/O
H8	VSS	VSS	VSS	AN15	PR16D	PR19D	I/O
AM4	DONE	DONE	DONE	AT16	PR15A	PR18A	I/O-M3
AB8	VDD	VDD	VDD	AM16	PR15B	PR18B	I/O

Note: The ceramic PGA contains single large VDD and VSS planes to which all VDD and VSS bond pads are connected.

Pin Information (continued)

Table 25. ATT2C26 and ATT2C40 428-Pin CPGA Pinout (continued)

Pin	2C26 Pad	2C40 Pad	Function	Pin	2C26 Pad	2C40 Pad	Function
AR17	PR15C	PR18C	I/O	AR27	PR5A	PR8A	I/O-$\overline{RD}$
AL17	PR15D	PR18D	I/O	AL25	PR5B	PR7A	I/O
AU17	PR14A	PR17A	I/O	AT28	PR5C	PR7C	I/O
AN17	PR14B	PR17B	I/O	AP28	PR5D	PR6A	I/O
AT18	PR14C	PR17C	I/O	AU29	PR4A	PR5A	I/O
AK18	PR14D	PR17D	I/O	AM28	PR4B	PR5B	I/O
AN5	VSS	VSS	VSS	AR29	PR4C	PR5C	I/O
AR19	PR13A	PR16A	I/O	AK26	PR4D	PR5D	I/O
AM18	PR13B	PR16B	I/O	AL7	VSS	VSS	VSS
AN19	PR13C	PR16C	I/O	AT30	PR3A	PR4A	I/O-$\overline{WR}$
AP18	PR13D	PR16D	I/O	AL27	PR3B	PR4B	I/O
AK20	VDD	VDD	VDD	AP30	PR3C	PR4C	I/O
AL19	PR12A	PR15A	I/O	AN29	PR3D	PR4D	I/O
AU19	PR12B	PR15B	I/O	AU31	PR2A	PR3A	I/O
AP20	PR12C	PR15C	I/O	AL29	PR2B	PR3B	I/O
AT20	PR12D	PR15D	I/O	AR31	PR2C	PR3C	I/O
AM6	VSS	VSS	VSS	AK28	PR2D	PR3D	I/O
AM20	PR11A	PR14A	I/O	AM30	PR1A	PR2A	I/O
AU21	PR11B	PR14B	I/O	AT32	PR1B	PR2D	I/O
AN21	PR11C	PR14C	I/O	AN31	PR1C	PR1A	I/O
AR21	PR11D	PR14D	I/O	AR33	PR1D	PR1D	I/O
AL21	PR10A	PR13A	I/O	AK8	VSS	VSS	VSS
AT22	PR10B	PR13B	I/O	AP32	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$	$\overline{RD\_CFGN}$
AM22	PR10C	PR13C	I/O	AJ31	VDD	VDD	VDD
AP22	PR10D	PR13D	I/O	AH30	VDD	VDD	VDD
AN23	PR9A	PR12A	I/O-CS1	AP34	VSS	VSS	VSS
AU23	PR9B	PR12B	I/O	AJ33	PT24D	PT30D	I/O
AP24	PR9C	PR12C	I/O	AM36	PT24C	PT30A	I/O
AR23	PR9D	PR12D	I/O	AH32	PT24B	PT29B	I/O
AK22	PR8A	PR11A	I/O	AL35	PT24A	PT29A	I/O
AT24	PR8B	PR11B	I/O	AL37	PT23D	PT28D	I/O
AL23	PR8C	PR11C	I/O	AH34	PT23C	PT28C	$\overline{I/O}$
AU25	PR8D	PR11D	I/O	AK34	PT23B	PT28B	I/O
AK24	VDD	VDD	VDD	AG31	PT23A	PT28A	I/O-RDY/RCLK
AR25	PR7A	PR10A	I/O-$\overline{CS0}$	AK36	PT22D	PT27D	I/O
AM24	PR7B	PR10B	I/O	AF30	PT22C	PT27C	I/O
AT26	PR7C	PR10C	I/O	AJ35	PT22B	PT27B	I/O
AN25	PR7D	PR10D	I/O	AG33	PT22A	PT27A	I/O
AP26	PR6A	PR9A	I/O	AJ37	PT21D	PT26D	I/O
AN27	PR6B	PR9B	I/O	AF32	PT21C	PT26C	I/O
AU27	PR6C	PR9C	I/O	AH36	PT21B	PT26B	I/O
AM26	PR6D	PR9D	I/O	AE31	PT21A	PT26A	I/O

Note: The ceramic PGA contains single large VDD and VSS planes to which all VDD and VSS bond pads are connected.

Pin Information (continued)

Table 25. ATT2C26 and ATT2C40 428-Pin CPGA Pinout (continued)

Pin	2C26 Pad	2C40 Pad	Function	Pin	2C26 Pad	2C40 Pad	Function
AG35	PT20D	PT25D	I/O-D7	T32	PT10C	PT13C	I/O
AE33	PT20C	PT25C	I/O	T34	PT10B	PT13B	I/O
AG37	PT20B	PT25B	I/O	N33	PT10A	PT13A	I/O-D0/DIN
AD32	PT20A	PT25A	I/O	P32	PT9D	PT12D	I/O
AF34	PT19D	PT24D	I/O	R35	PT9C	PT12C	I/O
AD34	PT19C	PT24C	I/O	R31	PT9B	PT12B	I/O
AF36	PT19B	PT24B	I/O	P36	PT9A	PT12A	I/O
AC33	PT19A	PT23D	I/O	M32	PT8D	PT11D	I/O
AE35	PT18D	PT22D	I/O	N37	PT8C	PT11C	I/O
AC31	PT18C	PT22A	I/O	N31	PT8B	PT11B	I/O
AE37	PT18B	PT21D	I/O-D6	M36	PT8A	PT11A	I/O-DOUT
AB32	PT18A	PT21A	I/O	Y30	VDD	VDD	VDD
AD30	VDD	VDD	VDD	N35	PT7D	PT10D	I/O
AB30	PT17D	PT20D	I/O	P30	PT7C	PT10A	I/O
AD36	PT17C	PT20C	I/O	L37	PT7B	PT9D	I/O
Y34	PT17B	PT20B	I/O	L33	PT7A	PT9A	I/O
AC35	PT17A	PT20A	I/O	M34	PT6D	PT8D	I/O
AA33	PT16D	PT19D	I/O	K34	PT6C	PT8A	I/O
AC37	PT16C	PT19C	I/O	L35	PT6B	PT7D	I/O
AA31	PT16B	PT19B	I/O	M30	PT6A	PT7A	I/O-TDI
AB34	PT16A	PT19A	I/O-D5	J37	PT5D	PT6D	I/O
AB36	PT15D	PT18D	I/O	L31	PT5C	PT6C	I/O
V34	PT15C	PT18C	I/O	K36	PT5B	PT6B	I/O
AA35	PT15B	PT18B	I/O	K32	PT5A	PT6A	I/O
Y32	PT15A	PT18A	I/O	H36	PT4D	PT5D	I/O
AA37	PT14D	PT17D	I/O	J33	PT4C	PT5C	I/O
W33	PT14C	PT17C	I/O	J35	PT4B	PT5B	I/O
Y36	PT14B	PT17B	I/O	J31	PT4A	PT5A	I/O-TMS
U33	PT14A	PT17A	I/O-D4	AL31	VSS	VSS	VSS
W35	PT13D	PT16D	I/O	G37	PT3D	PT4D	I/O
W31	PT13C	PT16C	I/O	K30	PT3C	PT4C	I/O
W37	PT13B	PT16B	I/O	H34	PT3B	PT4B	I/O
V32	PT13A	PT16A	I/O-D3	H32	PT3A	PT4A	I/O
AN33	VSS	VSS	VSS	G35	PT2D	PT3D	I/O
V36	PT12D	PT15D	I/O	G33	PT2C	PT3C	I/O
P34	PT12C	PT15C	I/O	F36	PT2B	PT3B	I/O
U37	PT12B	PT15B	I/O	E31	PT2A	PT3A	I/O
V30	PT12A	PT15A	I/O-D2	F30	PT1D	PT2D	I/O
AM32	VSS	VSS	VSS	F34	PT1C	PT2A	I/O
T36	PT11D	PT14D	I/O-D1	G29	PT1B	PT1D	I/O
R33	PT11C	PT14C	I/O	E35	PT1A	PT1A	I/O-TCK
U35	PT11B	PT14B	I/O	AK30	VSS	VSS	VSS
U31	PT11A	PT14A	I/O	H28	RD_DATA/TDO	RD_DATA/TDO	RD_DATA/TDO
R37	PT10D	PT13D	I/O	T30	VDD	VDD	VDD

Note: The ceramic PGA contains single large VDD and VSS planes to which all VDD and VSS bond pads are connected.

Package Thermal Characteristics

When silicon die junction temperature is below the recommended junction temperature of 125 °C, the temperature-activated failure mechanisms are minimized. There are four major factors that affect the thermal resistance value: silicon device size/paddle size, board mounting configuration (board density, multilayer nature of board), package type and size, and system airflow over the package. The values in the table below reflect the capability of the various package types to dissipate heat at given airflow rates. The numbers represent the delta °C/W between the ambient temperature and the device junction temperature.

To test package thermal characteristics, a single package containing a 0.269 in. sq. test IC of each configuration is mounted at the center of a printed-circuit board (PCB) measuring 8 in. x 13 in. x 0.062 in. The assembled PCB is mounted vertically in the center of the rectangular test section of a wind tunnel. The walls of the wind tunnel simulate adjacent boards in the electronic rack and can be adjusted to study the effects of PCB spacing. Forced air at room temperature is supplied by a pair of push-pull blowers which can be regulated to supply the desired air velocities. The air velocity is measured with a hot-wire anemometer at the center of the channel, 3 in. upstream from the package.

A typical test consists of regulating the wind tunnel blowers to obtain the desired air velocity and applying power to the test IC. The power to the IC is adjusted until the maximum junction temperature (as measured by its diodes) reaches 115 °C to 120 °C. The thermal resistance Θ_{JA} (°C/W) is computed by using the power supplied to the IC, junction temperature, ambient temperature, and air velocity:

$$\Theta JA = \frac{TJ - TA}{QC}$$

where:

T_J = peak temperature on the active surface of the IC

T_A = ambient air temperature

Q_C = IC power

The tests are repeated at several velocities from 0 fpm (feet per minute) to 1000 fpm.

The definition of the junction-to-case thermal resistance Θ_{JC} is:

$$\Theta JC = \frac{TJ - TC}{QC}$$

where:

T_C = temperature measured to the thermocouple at the top dead center of the package

The actual Θ_{JC} measurement performed at Lucent, Θ_{J-TDC}, uses a different package mounting arrangement than the one defined for Θ_{JC} in MIL-STD-883D and SEMI standards. Please contact Lucent for a diagram.

The maximum power dissipation for a package is calculated from the maximum allowed junction temperature (T_{Jmax}, 125 °C), the maximum ambient temperature (T_{Amax}), and the junction-to-ambient thermal characteristic for the given package (Θ_{JA}). The maximum power for the package is calculated as follows:

$$\text{Max. Power (Watts)} = (125\ °C - T_{Amax}) \times (1/\Theta_{JA})$$

In Table 26 and Table 26, a maximum power dissipation for each package is shown with T_{Amax} = 70 °C for the commercial temperature range and the Θ_{JA} used is for 0 feet per minute of air flowing over the package. If your application does not correspond to these parameters, the maximum power dissipation should be recalculated using the formula above.

Once the power dissipated by the FPGA has been determined (see the Estimating Power Dissipation section), the maximum junction temperature of the FPGA can be found. This is needed to determine if speed derating of the device from the 85 °C junction temperature used in all of the delay tables is needed. Using the maximum ambient temperature, T_{Amax}, and the power dissipated by the device, P, the maximum junction temperature is given by:

$$T_{Jmax} = T_{Amax} + (P \bullet \Theta_{JA})\ °C$$

Table 26 and Table 26 list the thermal characteristics for all packages used with the ORCA 2C Series of FPGAs.

Package Thermal Characteristics (continued)

Table 26. *ORCA* Plastic Package Thermal Characteristics

Package	ΘJA (°C/W)			ΘJC (°C/W)	Max Power (W) (70 °C—0 fpm)
	0 fpm	200 fpm	400 fpm		
84-Pin PLCC	40	35	32	9	1.38
100-Pin TQFP	61	49	46	6	0.9
144-Pin TQFP	52	39	36	4	1.05
160-Pin QFP	40	36	32	8	1.38
208-Pin SQFP	37	33	29	8	1.49
208-Pin SQFP-PQ2	16	14	12	1.3	3.43
240-Pin SQFP	35	31	28	7	1.57
240-Pin SQFP-PQ2	15	12	10	1.3	3.66
256-Pin PBGA[1]	21	17 (est.)	14 (est.)	TBD	2.62
256-Pin PBGA[2]	28	24 (est.)	22 (est.)	TBD	1.97
304-Pin SQFP	33	30	27	6	1.67
304-Pin SQFP-PQ2	12	10	8	1.3	4.58

Notes:
1. With thermal balls connected to board ground plane.
2. Without thermal balls connected to board ground plane.

Table 27. *ORCA* Ceramic Package Thermal Characteristics

Package	ΘJA (°C/W)			ΘJC (°C/W)	Max Power (W) (70 °C—0 fpm)
	0 fpm	200 fpm	400 fpm		
364-Pin CPGA	18	16	14	2.3	3.05
428-Pin CPGA	18	16	14	2.3	3.05

Package Coplanarity

The coplanarity of Lucent Technologies' postmolded packages is 4 mils. The coplanarity of selected packages is scheduled to be reduced to 3.1 mils. All Lucent *ORCA* Series FPGA ceramic packages are through-hole mount.

Package Parasitics

The electrical performance of an IC package, such as signal quality and noise sensitivity, is directly affected by the package parasitics. Table 28 lists eight parasitics associated with the *ORCA* packages. These parasitics represent the contributions of all components of a package, which include the bond wires, all internal package routing, and the external leads.

Four inductances in nH are listed: L_W and L_L, the self-inductance of the lead; and L_{MW} and L_{ML}, the mutual inductance to the nearest neighbor lead. These parameters are important in determining ground bounce noise and inductive crosstalk noise. Three capacitances in pF are listed: C_M, the mutual capacitance of the lead to the nearest neighbor lead; and C_1 and C_2, the total capacitance of the lead to all other leads (all other leads are assumed to be grounded). These parameters are important in determining capacitive crosstalk and the capacitive loading effect of the lead.

The parasitic values in Table 28 are for the circuit model of bond wire and package lead parasitics. If the mutual capacitance value is not used in the designer's model, then the value listed as mutual capacitance should be added to each of the C_1 and C_2 capacitors. The PGAs contain power and ground planes that will make the inductance value for power and ground leads the minimum value listed. The PGAs also have a significant range of parasitic values. This is due to the large variation in internal trace lengths and is also due to two signal metal layers that are separated from the ground plane by different distances. The upper signal layer is more inductive but less capacitive than the closer, lower signal layer.

Package Parasitics (continued)

Table 28. *ORCA* 2C Package Parasitics

Package Type	L_W	M_W	R_W	C_1	C_2	C_M	L_L	M_L
84-Pin PLCC	3	1	160	1	1	0.5	7—11	3—6
100-Pin TQFP	3	1	160	0.7	0.7	0.94	3—4	1.5—2
144-Pin TQFP	3.5	1.5	175	1	1	0.6	4—6	2—2.5
160-Pin QFP	4	2	200	1	1	1	13—17	8—11
208-Pin SQFP	4	2	200	1	1	1	7—10	4—6
208-Pin SQFP-PQ2	4	2	200	1	1	1	6—9	4—6
240-Pin SQFP	4	2	200	1	1	1	8—12	5—8
240-Pin SQFP-PQ2	4	2	200	1	1	1	7—11	4—7
256-Pin PBGA	5	2	220	1	1	1	5—13	2—6
304-Pin SQFP	5	2	220	1	1	1	12—18	7—12
304-Pin SQFP-PQ2	5	2	220	1	1	1	11—17	7—12
364-Pin CPGA	2	1	1000	1—2	1—2	0.5—1	2—11*	1—4
428-Pin CPGA	2	1	1000	1—2	1—2	0.6—1.2	2—11*	1—4

* Leads designated as ground (power) can be connected to the ground plane, reducing the trace inductance to the minimum value listed.

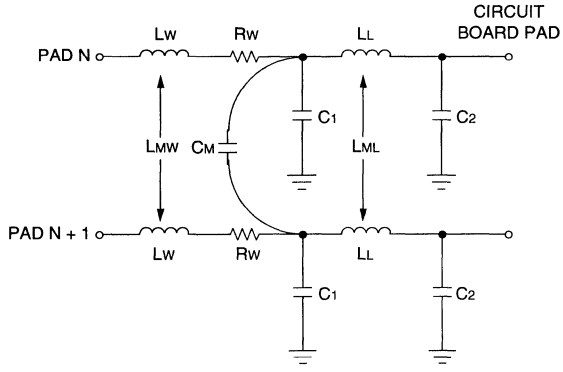


Figure 48. Package Parasitics

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 operations sections of this data sheet. Exposure to Absolute Maximum Ratings for extended periods can adversely affect device reliability.

The Lucent *ORCA* Series FPGAs include circuitry designed to protect the chips from damaging substrate injection currents and prevent accumulations of static charge. Nevertheless, conventional precautions should be observed during storage, handling, and use to avoid exposure to excessive electrical stress.

Parameter	Symbol	Min	Max	Unit
Storage Temperature	T_{stg}	–65	150	°C
Supply Voltage with Respect to Ground	V_{DD}	–0.5	7.0	V
Input Signal with Respect to Ground	—	–0.5	V_{DD} + 0.3	V
Signal Applied to High-impedance Output	—	–0.5	V_{DD} + 0.3	V
Maximum Soldering Temperature	—	—	260	°C

Recommended Operating Conditions

Mode	Temperature Range (Ambient)	Supply Voltage
Commercial	0 °C to 70 °C	5 V ± 5%
Industrial	–40 °C to +85 °C	5 V ± 10%

Note: The maximum recommended junction temperature, T_J, during operation is 125 °C.

Electrical Characteristics

Table 29. Electrical Characteristics

Commercial: V_{DD} = 5.0 V ± 5%, 0 °C ≤ T_A ≤ 70 °C; Industrial: V_{DD} = 5.0 V ± 10%, –40 °C ≤ T_A ≤ +85 °C.

Parameter	Symbol	Test Conditions	Min	Max	Unit
Input Voltage:		Input configured as CMOS			
High	V_{IH}		70% V_{DD}	V_{DD} + 0.3	V
Low	V_{IL}		GND – 0.5	20% V_{DD}	V
Input Voltage:		Input configured as TTL			
High	V_{IH}		2.0	V_{DD} + 0.3	V
Low	V_{IL}		–0.5	0.8	V
Output Voltage:					
High	V_{OH}	V_{DD} = Min, I_{OH} = 6 mA or 3 mA	2.4	—	V
Low	V_{OL}	V_{DD} = Min, I_{OL} = 12 mA or 6 mA	—	0.4	V
Input Leakage Current	I_L	V_{DD} = Max, V_{IN} = V_{SS} or V_{DD}	–10	10	µA
Standby Current:	I_{DDSB}	T_A = 25 °C, V_{DD} = 5.0 V,			
ATT2C04		internal oscillator running, no	—	6.5	mA
ATT2C06		output loads, inputs at	—	7.0	mA
ATT2C08		V_{DD} or GND	—	7.7	mA
ATT2C10			—	8.4	mA
ATT2C12			—	9.2	mA
ATT2C15			—	10.0	mA
ATT2C26			—	12.2	mA
ATT2C40			—	16.3	mA
Standby Current:	I_{DDSB}	T_A = 25 °C, V_{DD} = 5.0 V,			
ATT2C04		internal oscillator stopped, no	—	1.5	mA
ATT2C06		output loads, inputs at	—	2.0	mA
ATT2C08		V_{DD} or GND	—	2.7	mA
ATT2C10			—	3.4	mA
ATT2C12			—	4.2	mA
ATT2C15			—	5.0	mA
ATT2C26			—	7.2	mA
ATT2C40			—	11.3	mA
Data Retention Voltage	V_{DR}	T_A = 25 °C	2.3	—	V
Input Capacitance	C_{IN}	T_A = 25 °C, V_{DD} = 5.0 V Test frequency = 1 MHz	—	10	pF
Output Capacitance	C_{OUT}	T_A = 25 °C, V_{DD} = 5.0 V Test frequency = 1 MHz	—	10	pF
DONE Pull-up Resistor	R_{DONE}	—	100K	—	Ω
M3, M2, M1, and M0 Pull-up Resistors	R_M	—	100K	—	Ω
I/O Pad Static Pull-up Current	I_{PU}	V_{DD} = 5.25 V, V_{IN} = V_{SS}, T_A = 0 °C	14.4	50.9	µA
I/O Pad Static Pull-down Current	I_{PD}	V_{DD} = 5.25 V, V_{IN} = V_{DD}, T_A = 0 °C	26	103	µA
I/O Pad Pull-up Resistor	R_{PU}	V_{DD} = 5.25 V, V_{IN} = V_{SS}, T_A = 0 °C	100K	—	Ω
I/O Pad Pull-down Resistor	R_{PD}	V_{DD} = 5.25 V, V_{IN} = V_{DD}, T_A = 0 °C	50K	—	Ω

Timing Characteristics

Table 30. PFU Timing Characteristics

Commercial: V_{DD} = 5.0 V ± 5%, 0 °C ≤ T_A ≤ 70 °C; Industrial: V_{DD} = 5.0 V ± 10%, –40 °C ≤ T_A ≤ +85 °C.

Parameter	Symbol	Speed						Unit
		-2		-3				
		Min	Max	Min	Max	Min	Max	
Input Requirements								
Clock Low Time	TCL	3.2	—	2.5	—			ns
Clock High Time	TCH	3.2	—	2.5	—			ns
Global S/R Pulse Width (gsrn)	TRW	2.8	—	2.5	—			ns
Local S/R Pulse Width	TPW	3.0	—	2.5	—			ns
Combinatorial Setup Times (T_J = +85 °C, V_{DD} = Min):								
Four Input Variables to Clock (a[4:0], b[4:0] to ck)	F4*_SET	2.4	—	1.7	—			ns
Five Input Variables to Clock (a[4:0], b[4:0] to ck)	F5*_SET	2.5	—	1.9	—			ns
PFUMUX to Clock (a[4:0], b[4:0] to ck)	MUX_SET	3.9	—	2.9	—			ns
PFUMUX to Clock (c0 to ck)	COMUX_SET	1.5	—	1.2	—			ns
PFUNAND to Clock (a[4:0], b[4:0] to ck)	ND_SET	3.9	—	2.9	—			ns
PFUNAND to Clock (c0 to ck)	COND_SET	1.7	—	1.2	—			ns
PFUXOR to Clock (a[4:0], b[4:0] to ck)	XOR_SET	4.8	—	3.6	—			ns
PFUXOR to Clock (c0 to ck)	COXOR_SET	1.6	—	1.2	—			ns
Data In to Clock (wd[3:0] to ck)	D*_SET	0.5	—	0.1	—			ns
Clock Enable to Clock (ce to ck)	CKEN_SET	1.6	—	1.2	—			ns
Local Set/Reset (synchronous) (lsr to ck)	LSR_SET	1.7	—	1.4	—			ns
Data Select to Clock (sel to ck)	SELECT_SET	1.9	—	1.5	—			ns
Pad Direct In	PDIN_SET	0.0	—	0.0	—			ns
Combinatorial Hold Times (T_J = All, V_{DD} = All):								
Data In (wd[3:0] from ck)	D*_HLD	0.6	—	0.4	—			ns
Clock Enable (ce from ck)	CKEN_HLD	0.6	—	0.4	—			ns
Local Set/Reset (synchronous) (lsr from ck)	LSR_HLD	0.0	—	0.0	—			ns
Data Select (sel from ck)	SELECT_HLD	0.0	—	0.0	—			ns
Pad Direct In Hold (dia[3:0], dib[3:0] to ck)	PDIN_HLD	1.5	—	1.4	—			ns
All Others	—	0	—	0	—			ns
Output Characteristics								
Combinatorial Delays (T_J = +85 °C, V_{DD} = Min):								
Four Input Variables (a[4:0], b[4:0] to o[4:0])	F4*_DEL	—	5.1	—	3.6			ns
Five Input Variables (a[4:0], b[4:0] to o[4:0])	F5*_DEL	—	5.2	—	3.7			ns
PFUMUX (a[4:0], b[4:0] to o[4:0])	MUX_DEL	—	5.8	—	4.6			ns
PFUMUX (c0 to o[4:0])	COMUX_DEL	—	4.1	—	3.0			ns
PFUNAND (a[4:0], b[4:0] to o[4:0])	ND_DEL	—	5.8	—	4.8			ns
PFUNAND (c0 to o[4:0])	COND_DEL	—	3.8	—	3.0			ns
PFUXOR (a[4:0], b[4:0] to o[4:0])	XOR_DEL	—	6.7	—	5.3			ns
PFUXOR (c0 to o[4:0])	COXOR_DEL	—	4.2	—	3.0			ns
Sequential Delays (T_J = +85 °C, V_{DD} = Min):								
Local S/R (async) to PFU Out (lsr to o[4:0])	LSR_DEL	—	5.6	—	4.2			ns
Global S/R to PFU Out (gsrn to o[4:0])	GSR_DEL	—	4.0	—	3.1			ns
Clock to PFU Out (ck to o[4:0]) — Register	REG_DEL	—	3.9	—	2.8			ns
Clock to PFU Out (ck to o[4:0]) — Latch	LTCH_DEL	—	4.0	—	2.8			ns
Transparent Latch (wd[3:0] to 0[4:0])	LTCH_DDEL	—	5.0	—	3.5			ns

Timing Characteristics (continued)

Table 30. PFU Timing Characteristics (continued)

Commercial: V_{DD} = 5.0 V ± 5%, 0 °C ≤ T_A ≤ 70 °C; Industrial: V_{DD} = 5.0 V ± 10%, –40 °C ≤ T_A ≤ +85 °C.

Parameter	Symbol	Speed						Unit
		-2		-3				
		Min	Max	Min	Max	Min	Max	
Ripple Mode Characteristics								
Ripple Setup Times (T_J = +85 °C, V_{DD} = Min):								
Operands to Clock (a[3:0], b[3:0] to ck)	RIP_SET	6.7	—	5.0	—			ns
Carry-In to Clock (cin to ck)	CIN_SET	4.0	—	3.2	—			ns
Add/Subtract to Clock (a4 to ck)	AS_SET	8.2	—	5.6	—			ns
Ripple Hold Times (T_J = All, V_{DD} = All): All	T_H	0	—	0	—			ns
Ripple Delays (T_J = +85 °C, V_{DD} = Min):								
Operands to Carry-Out (a[3:0], b[3:0] to cout)	RIP_CODEL	—	5.4	—	3.8			ns
Operands to Carry-Out (o4) (a[3:0], b[3:0] to o4)	RIP_O4DEL	—	6.9	—	4.8			ns
Operands to PFU Out (a[3:0], b[3:0] to o[4:0])	RIP_DEL	—	9.3	—	6.8			ns
Carry-In to Carry-Out (cin to cout)	CIN_CODEL	—	1.9	—	1.6			ns
Carry-In to Carry-Out (o4) (cin to o4)	CIN_O4DEL	—	3.5	—	2.6			ns
Carry-In to PFU Out (cin to o[4:0])	CIN_DEL	—	6.7	—	5.0			ns
Add/Subtract to Carry-Out (a4 to cout)	AS_CODEL	—	6.1	—	4.5			ns
Add/Subtract to Carry-Out (o4) (a4 to o4)	AS_O4DEL	—	7.6	—	5.6			ns
Add/Subtract to PFU Out (a4 to o[4:0])	AS_DEL	—	10.8	—	7.6			ns
Read/Write Memory Characteristics								
Read Operation (T_J = +85 °C, V_{DD} = Min):								
Read Cycle Time	T_{RC}	5.1	—	3.6	—			ns
Data Valid after Address (a[3:0], b[3:0] to o[4:0])	MEM*_ADEL	—	5.1	—	3.6			ns
Read Operation, Clocking Data into Latch/Flip-Flop (T_J = +85 °C, V_{DD} = Min):								
Address to Clock Setup Time (a[3:0], b[3:0] to ck)	MEM*_ASET	2.4	—	1.8	—			ns
Clock to PFU Out (ck to o[4:0]) — Register	REG_DEL	—	3.9	—	2.8			ns
Write Operation (T_J = +85 °C, V_{DD} = Min):								
Write Cycle Time	T_{WC}	5.5	—	4.5	—			ns
Write Enable Pulse Width (a4/b4)	T_{PW}	3.0	—	2.5	—			ns
Setup Time (T_J = +85 °C, V_{DD} = Min):								
Address to wren (a[3:0]/b[3:0] to a4/b4)	MEM*_AWRSET	0.1	—	0.1	—			ns
Data to wren (wd[3:0] to a4/b4)	MEM*_DWRSET	0.0	—	0.0	—			ns
Hold Time (T_J = All, V_{DD} = All):								
Address from wren (a[3:0]/b[3:0] from a4/b4)	MEM*_WRAHLD	2.4	—	1.7	—			ns
Data from wren (wd[3:0] from a4/b4)	MEM*_WRDHLD	2.4	—	2.0	—			ns

Timing Characteristics (continued)

Table 30. PFU Timing Characteristics (continued)

Commercial: V_{DD} = 5.0 V ± 5%, 0 °C ≤ T_A ≤ 70 °C; Industrial: V_{DD} = 5.0 V ± 10%, –40 °C ≤ T_A ≤ +85 °C.

Parameter	Symbol	Speed						Unit
		-2		-3				
		Min	Max	Min	Max	Min	Max	
Read During Write Operation (T_J = +85 °C, V_{DD} = Min)								
Write Enable to PFU Output Delay (a4/b4 to o[4:0])	MEM*_WRDEL	—	8.1	—	5.7			ns
Data to PFU Output Delay (wd[3:0] to o[4:0])	MEM*_DDEL	—	6.1	—	4.4			ns
Read During Write, Clocking Data into Latch/Flip-Flop								
Setup Time (T_J = +85 °C, V_{DD} = Min):								
Write Enable to Clock (a4/b4 to ck)	MEM*_WRSET	5.4	—	4.4	—			ns
Data (wd[3:0] to ck)	MEM*_DSET	3.5	—	2.6	—			ns
Hold Time (T_J = All, V_{DD} = All): All	T_H	0	—	0	—			ns
Clock to PFU Out (ck to o[4:0]) — Register	REG_DEL	—	3.9	—	2.8			ns

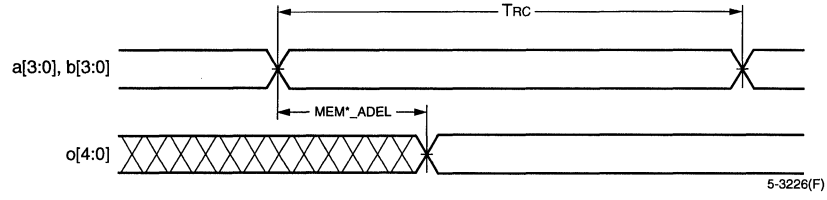


Figure 49. Read Operation—Flip-Flop Bypass

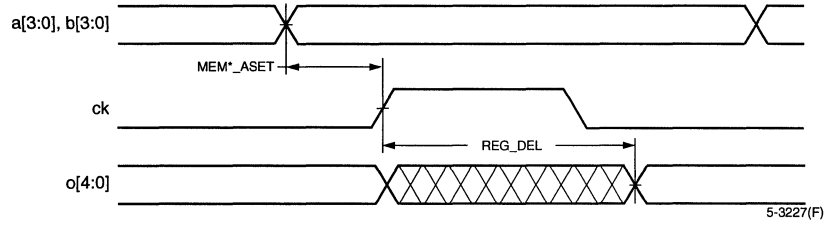


Figure 50. Read Operation—LUT Memory Loading Flip-Flops

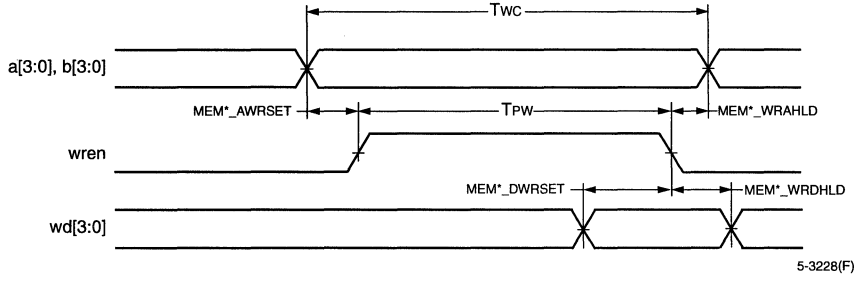


Figure 51. Write Operation

Timing Characteristics (continued)

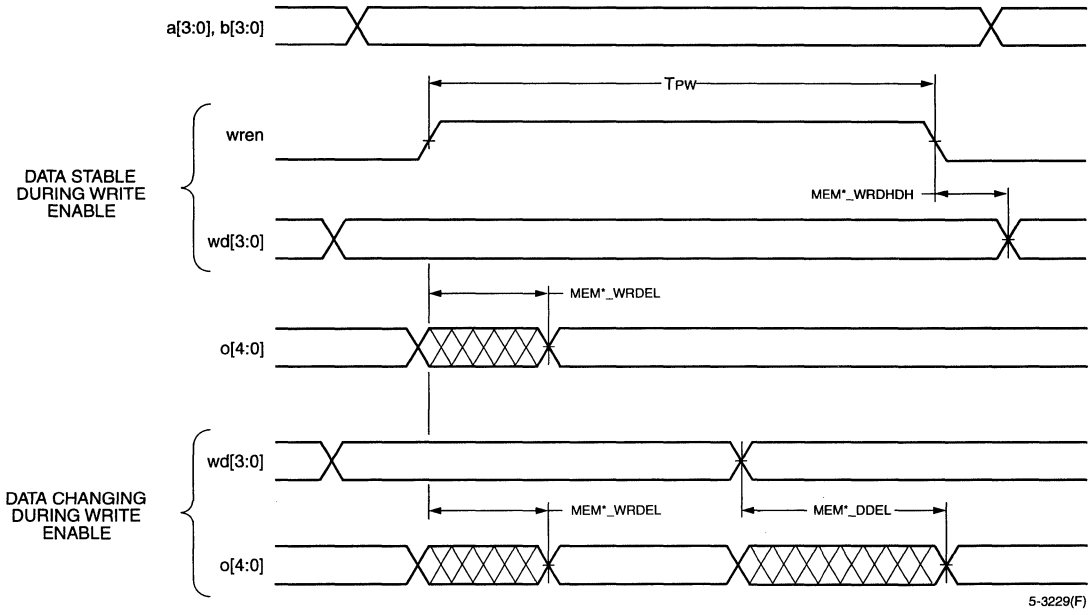


Figure 52. Read During Write

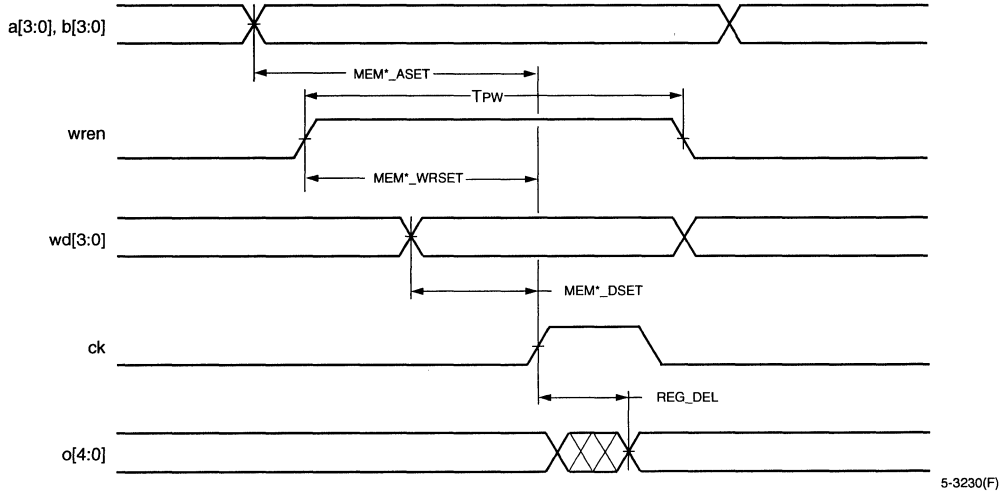


Figure 53. Read During Write—Clocking Data into Flip-Flop

Timing Characteristics (continued)

Table 31. PLC BIDI and Direct Routing Timing Characteristics

Commercial: V_{DD} = 5.0 V ± 5%, 0 °C ≤ T_A ≤ 70 °C; Industrial: V_{DD} = 5.0 V ± 10%, –40 °C ≤ T_A ≤ +85 °C.

Parameter	Symbol	Speed						Unit
		-2		-3				
		Min	Max	Min	Max	Min	Max	
PLC 3-Statable BIDIs (T_J = +85 °C, V_{DD} = Min)								
BIDI Propagation Delay	TRI_DEL	—	1.2	—	1.0			ns
BIDI 3-State Enable/Disable Delay	TRIEN_DEL	—	1.7	—	1.3			ns
Direct Routing (T_J = +85 °C, V_{DD} = Min)								
PFU to PFU Delay (xSW)	DIR_DEL	—	1.4	—	1.1			ns
PFU Feedback (xSW)	FDBK_DEL	—	1.0	—	0.8			ns

Table 32. Clock Delay

Commercial: V_{DD} = 5.0 V ± 5%, 0 °C ≤ T_A ≤ 70 °C; Industrial: V_{DD} = 5.0 V ± 10%, –40 °C ≤ T_A ≤ +85 °C.

Device (T_J = +85 °C, V_{DD} = Min)	Symbol	Speed						Unit
		-2		-3				
		Min	Max	Min	Max	Min	Max	
ATT2C04	CLK_DEL	—	5.5	—	4.4			
ATT2C06	CLK_DEL	—	5.6	—	4.5			ns
ATT2C08	CLK_DEL	—	5.8	—	4.6			ns
ATT2C10	CLK_DEL	—	5.9	—	4.7			ns
ATT2C12	CLK_DEL	—	6.1	—	4.9			ns
ATT2C15	CLK_DEL	—	6.2	—	5.0			ns
ATT2C26	CLK_DEL	—	6.4	—	5.2			ns
ATT2C40	CLK_DEL	—	6.9	—	5.8			ns

Note: This clock delay is for a fully routed clock tree that uses the primary clock network. It includes both the input buffer delay and the clock routing to the PFU CLK input. The delay will be reduced if any of the clock branches are not used.

Timing Characteristics (continued)

Table 33. Programmable I/O Cell Timing Characteristics

Commercial: V_{DD} = 5.0 V ± 5%, 0 °C ≤ T_A ≤ 70 °C; Industrial: V_{DD} = 5.0 V ± 10%, –40 °C ≤ T_A ≤ +85 °C; C_L = 50 pF.

Parameter	Symbol	Speed						Unit
		-2		-3				
		Min	Max	Min	Max	Min	Max	
Inputs (T_J = +85 °C, V_{DD} = Min)								
Input Rise Time	T_R	—	500	—	500			ns
Input Fall Time	T_F	—	500	—	500			ns
Pad to In Delay	FASTIN_G_DEL	—	3.1	—	2.3			ns
Pad to TRIDI Delay	FASTIN_L_DEL	—	2.7	—	1.9			ns
Pad to In Delay (delay mode)	DLYIN_G_DEL	—	7.8	—	6.2			ns
Pad to TRIDI Delay (delay mode)	DLYIN_L_DEL	—	2.5	—	1.9			ns
Pad to Nearest PFU Latch Output	CHIP_LATCH	—	6.8	—	5.1			ns
Setup Time:								
Pad to Nearest PFU ck	CHIP_SET	2.8	—	2.1	—			ns
Pad to Nearest PFU ck (delay mode)*	DLY_CHIP_SET	8.7	—	6.8	—			ns
Outputs (T_J = +85 °C, V_{DD} = Min)								
PFU ck to Pad Delay (dout[3:0] to pad):								
Fast	DOUT_DEL(F)	—	7.6	—	5.7			ns
Slewlim	DOUT_DEL(SL)	—	9.3	—	6.9			ns
Sinklim	DOUT_DEL(SI)	—	12.4	—	8.9			ns
Output to Pad Delay (out[3:0] to pad):								
Fast	OUT_DEL(F)	—	5.0	—	4.0			ns
Slewlim	OUT_DEL(SL)	—	6.7	—	5.2			ns
Sinklim	OUT_DEL(SI)	—	9.8	—	7.2			ns
3-state Enable Delay (ts[3:0] to pad):								
Fast	TS_DEL(F)	—	5.8	—	4.7			ns
Slewlim	TS_DEL(SL)	—	7.5	—	5.9			ns
Sinklim	TS_DEL(SI)	—	10.6	—	7.9			ns

* If the input buffer is placed in delay mode, the chip hold time to the nearest PFU latch is guaranteed to be 0 if the clock is routed using the primary clock network; (T_J = All, V_{DD} = All). For the ATT2C40, this is only valid for input buffers on the same half of the device as the clock pin.

Note: The delays for all input buffers assume an input rise/fall time of ≤1 V/ns.

Timing Characteristics (continued)

Table 34. General Configuration Mode Timing Characteristics

Commercial: V_{DD} = 5.0 V ± 5%, 0 °C ≤ T_A ≤ 70 °C; Industrial: V_{DD} = 5.0 V ± 10%, –40 °C ≤ T_A ≤ +85 °C; C_L = 50 pF.

Parameter	Symbol	Min	Max	Unit
All Configuration Modes				
M[3:0] Setup Time to $\overline{INIT}$ High	TSMODE	50.0	—	ns
M[3:0] Hold Time from $\overline{INIT}$ High	THMODE	600.0	—	ns
$\overline{RESET}$ Pulse Width Low	TRW	50.0	—	ns
$\overline{PRGM}$ Pulse Width Low	TPGW	50.0	—	ns
Master and Asynchronous Peripheral Modes				
Power-on Reset Delay	TPO	16.24	43.80	ms
CCLK Period (M3 = 0)	TCCLK	62.00	167.00	ns
(M3 = 1)		496.00	1336.00	ns
Configuration Latency (noncompressed)	TCL			
ATT2C04 (M3 = 0)		4.05	10.90*	ms
(M3 = 1)		32.38	87.21*	ms
ATT2C06 (M3 = 0)		5.63	15.18*	ms
(M3 = 1)		45.08	121.42*	ms
ATT2C08 (M3 = 0)		7.16	19.28*	ms
(M3 = 1)		57.27	154.25*	ms
ATT2C10 (M3 = 0)		9.23	24.85*	ms
(M3 = 1)		73.80	198.80*	ms
ATT2C12 (M3 = 0)		11.14	30.01*	ms
(M3 = 1)		89.14	240.10*	ms
ATT2C15 (M3 = 0)		13.69	36.87*	ms
(M3 = 1)		109.52	294.99*	ms
ATT2C26 (M3 = 0)		19.03	51.25*	ms
(M3 = 1)		152.28	409.99*	ms
ATT2C40 (M3 = 0)		29.39	79.16*	ms
(M3 = 1)		235.12	633.31*	ms
Slave Serial and Synchronous Peripheral Modes				
Power-on Reset Delay	TPO	4.06	10.95	ms
CCLK Period	TCCLK	100.00	—	ns
Configuration Latency (noncompressed):	TCL			
ATT2C04		6.53	—	ms
ATT2C06		9.09	—	ms
ATT2C08		11.55	—	ms
ATT2C10		14.88	—	ms
ATT2C12		17.97	—	ms
ATT2C15		22.08	—	ms
ATT2C26		30.69	—	ms
ATT2C40		47.40	—	ms

* Not applicable to asynchronous peripheral mode.

Timing Characteristics (continued)

Table 34. General Configuration Mode Timing Characteristics (continued)

Commercial: V_{DD} = 5.0 V ± 5%, 0 °C ≤ T_A ≤ 70 °C; Industrial: V_{DD} = 5.0 V ± 10%, –40 °C ≤ T_A ≤ +85 °C; C_L = 50 pF.

Parameter	Symbol	Min	Max	Unit
Slave Parallel Mode				
Power-on Reset Delay	T_{PO}	4.06	10.95	ms
CCLK Period	T_{CCLK}	100.00	—	ns
Configuration Latency (noncompressed):	T_{CL}			
ATT2C04		0.82	—	ms
ATT2C06		1.14	—	ms
ATT2C08		1.44	—	ms
ATT2C10		1.86	—	ms
ATT2C12		2.25	—	ms
ATT2C15		2.76	—	ms
ATT2C26		3.84	—	ms
ATT2C40		5.93	—	ms
$\overline{\text{INIT}}$ Timing				
$\overline{\text{INIT}}$ High to CCLK Delay	T_{INIT_CCLK}			
Slave Parallel		1.00	—	µs
Slave Serial		1.00	—	µs
Synchronous Peripheral		1.00	—	µs
Master Serial				
M3 = 1		1.00	2.90	µs
M3 = 0		0.50	0.70	µs
Master Parallel				
M3 = 1		4.90	13.60	µs
M3 = 0		1.00	2.90	µs
Initialization Latency ($\overline{\text{PRGM}}$ high to $\overline{\text{INIT}}$ high)	T_{IL}			
ATT2C04		59.51	162.33	µs
ATT2C06		70.43	191.72	µs
ATT2C08		81.34	221.11	µs
ATT2C10		92.25	250.51	µs
ATT2C12		103.16	279.90	µs
ATT2C15		114.07	309.29	µs
ATT2C26		135.90	368.07	µs
ATT2C40		170.87	462.26	µs
$\overline{\text{INIT}}$ High to $\overline{\text{WR}}$, Asynchronous Peripheral	T_{INIT_WR}	1.50	—	µs

Note: T_{PO} is triggered when V_{DD} reaches between 3.0 V to 4.0 V.

Timing Characteristics (continued)

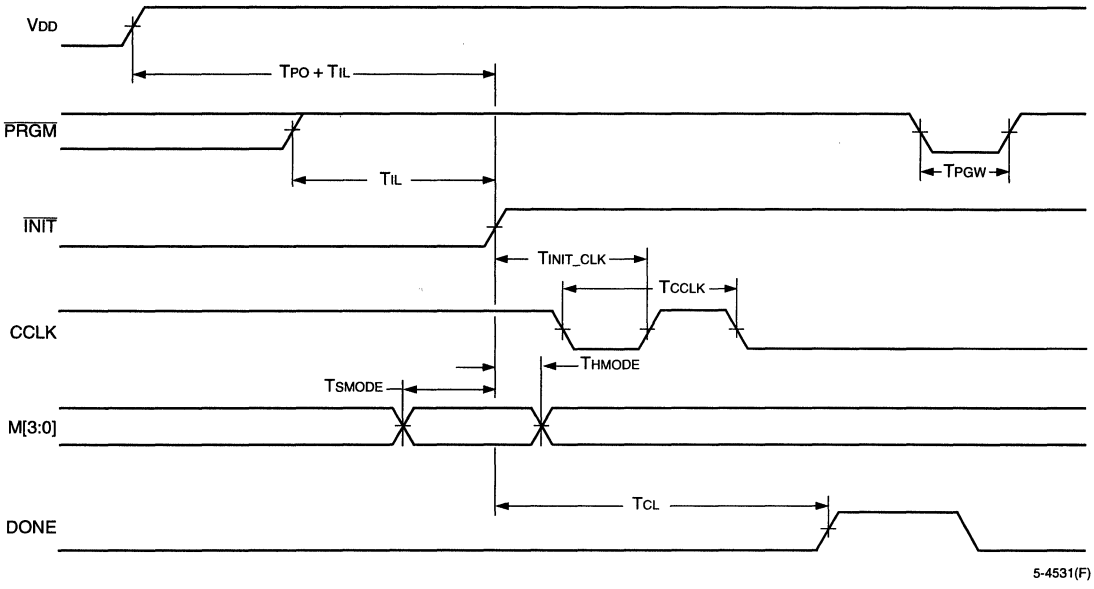


Figure 54. General Configuration Mode Timing Diagram

Timing Characteristics (continued)

Table 35. Master Serial Configuration Mode Timing Characteristics

Commercial: V_{DD} = 5.0 V ± 5%, 0 °C ≤ T_A ≤ 70 °C; Industrial: V_{DD} = 5.0 V ± 10%, –40 °C ≤ T_A ≤ +85 °C; C_L = 50 pF.

Parameter	Symbol	Min	Nom	Max	Unit
DIN Setup Time	T_S	60.0	—	—	ns
DIN Hold Time	T_H	0	—	—	ns
CCLK Frequency (M3 = 0)	F_C	6.0	10.0	16.0	MHz
CCLK Frequency (M3 = 1)	F_C	0.75	1.25	2.0	MHz
CCLK to DOUT Delay	T_D	—	—	30	ns

Note: Serial configuration data is transmitted out on DOUT on the falling edge of CCLK after it is input DIN.

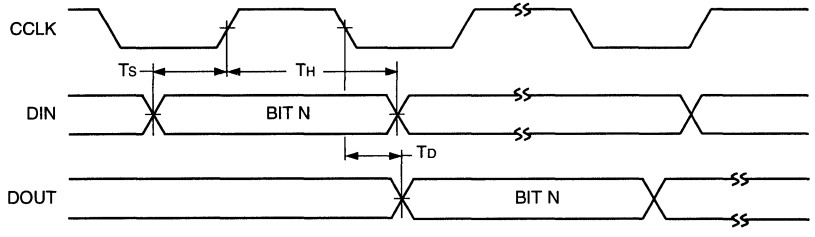


Figure 55. Master Serial Configuration Mode Timing Diagram

Timing Characteristics (continued)

Table 36. Master Parallel Configuration Mode Timing Characteristics

Commercial: V_{DD} = 5.0 V ± 5%, 0 °C ≤ T_A ≤ 70 °C; Industrial: V_{DD} = 5.0 V ± 10%, –40 °C ≤ T_A ≤ +85 °C; C_L = 50 pF.

Parameter	Symbol	Min	Max	Unit
RCLK to Address Valid	TAV	0	200	ns
D[7:0] Setup Time to RCLK High	TS	60	—	ns
D[7:0] Hold Time to RCLK High	TH	0	—	ns
RCLK Low Time (M3 = 0)	TCL	434	1169	ns
RCLK High Time (M3 = 0)	TCH	62	167	ns
RCLK Low Time (M3 = 1)	TCL	3472	9352	ns
RCLK High Time (M3 = 1)	TCH	496	1336	ns
CCLK to DOUT	TD	—	30	ns

Notes: The RCLK period consists of seven CCLKs for RCLK low and one CCLK for RCLK high.
Serial data is transmitted out on DOUT 1.5 CCLK cycles after the byte is input D[7:0].

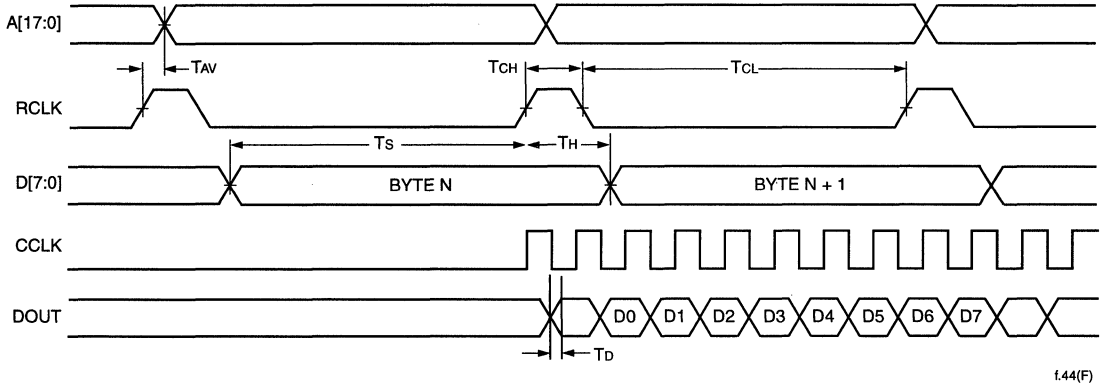


Figure 56. Master Parallel Configuration Mode Timing Diagram

Timing Characteristics (continued)

Table 37. Asynchronous Peripheral Configuration Mode Timing Characteristics

Commercial: V_{DD} = 5.0 V ± 5%, 0 °C ≤ T_A ≤ 70 °C; Industrial: V_{DD} = 5.0 V ± 10%, –40 °C ≤ T_A ≤ +85 °C; C_L = 50 pF.

Parameter	Symbol	Min	Max	Unit
$\overline{WR}$, $\overline{CS0}$, and CS1 Pulse Width	T_{WR}	100	—	ns
D[7:0] Setup Time	T_S	20	—	ns
D[7:0] Hold Time	T_H	0	—	ns
RDY/$\overline{BUSY}$ Delay	T_{RDY}	—	60	ns
RDY/$\overline{BUSY}$ Low	T_B	2	9	CCLK Periods
Earliest $\overline{WR}$ After End of $\overline{BUSY}$	T_{WR2}	0	—	ns
CCLK to DOUT	T_D	—	30	ns

Note: Serial data is transmitted out on DOUT on the falling edge of CCLK after the byte is input D[7:0].

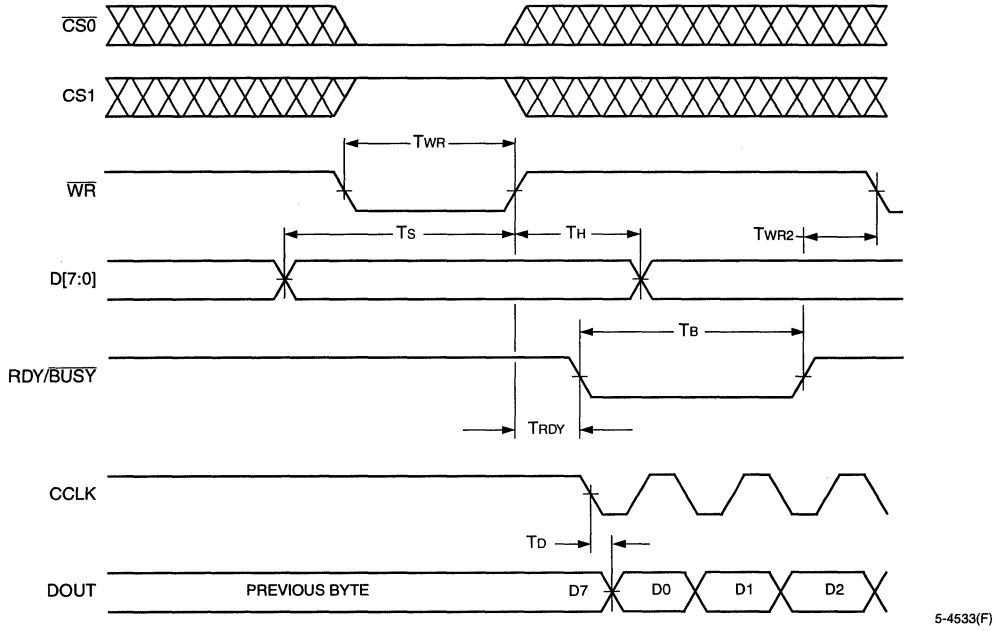


Figure 57. Asynchronous Peripheral Configuration Mode Timing Diagram

Timing Characteristics (continued)

Table 38. Synchronous Peripheral Configuration Mode Timing Characteristics

Commercial: V_{DD} = 5.0 V ± 5%, 0 °C ≤ T_A ≤ 70 °C; Industrial: V_{DD} = 5.0 V ± 10%, –40 °C ≤ T_A ≤ +85 °C; C_L = 50 pF.

Parameter	Symbol	Min	Max	Unit
D[7:0] Setup Time	T_S	20	—	ns
D[7:0] Hold Time	T_H	0	—	ns
CCLK High Time	T_{CH}	50	—	ns
CCLK Low Time	T_{CL}	50	—	ns
CCLK Frequency	F_C	—	10	MHz
CCLK to DOUT	T_D	—	30	ns

Note: Serial data is transmitted out on DOUT 1.5 clock cycles after the byte is input D[7:0].

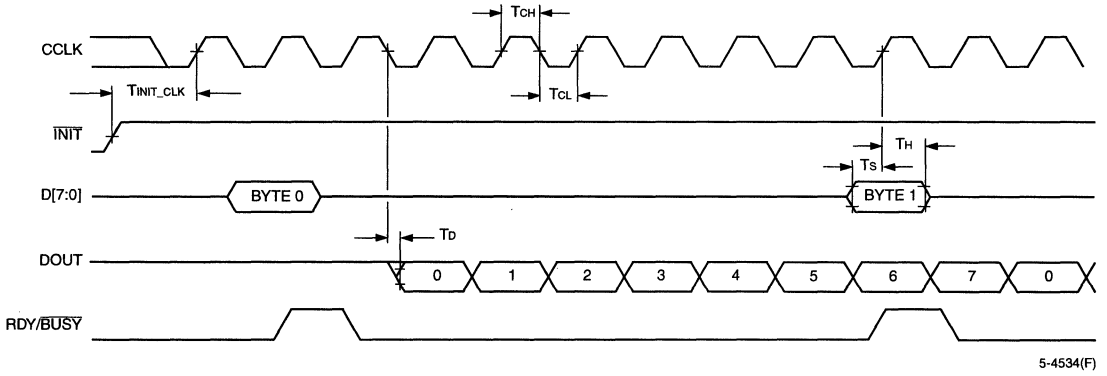


Figure 58. Synchronous Peripheral Configuration Mode Timing Diagram

Timing Characteristics (continued)

Table 39. Slave Serial Configuration Mode Timing Characteristics

Commercial: V_{DD} = 5.0 ± 5%, 0 °C ≤ T_A ≤ 70 °C; Industrial: V_{DD} = 5.0 V ± 10%, –40 °C ≤ T_A ≤ +85 °C; C_L = 50 pF.

Parameter	Symbol	Min	Max	Unit
DIN Setup Time	T_S	20	—	ns
DIN Hold Time	T_H	0	—	ns
CCLK High Time	T_{CH}	50	—	ns
CCLK Low Time	T_{CL}	50	—	ns
CCLK Frequency	F_C	—	10	MHz
CCLK to DOUT	T_D	—	30	ns

Note: Serial configuration data is transmitted out on DOUT on the falling edge of CCLK after it is input on DIN.

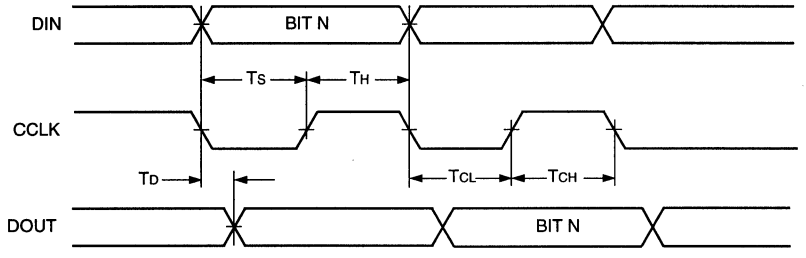


Figure 59. Slave Serial Configuration Mode Timing Diagram

Timing Characteristics (continued)

Table 40. Slave Parallel Configuration Mode Timing Characteristics

Commercial: V_{DD} = 5.0 V ± 5%, 0 °C ≤ T_A ≤ 70 °C; Industrial: V_{DD} = 5.0 V ± 10%, –40 °C ≤ T_A ≤ +85 °C.

Parameter	Symbol	Min	Max	Unit
$\overline{CS0}$, CS1, $\overline{WR}$ Setup Time	T_{S1}	60	—	ns
$\overline{CS0}$, CS1, $\overline{WR}$ Hold Time	T_{H1}	20	—	ns
D[7:0] Setup Time	T_{S2}	20	—	ns
D[7:0] Hold Time	T_{H2}	0	—	ns
CCLK High Time	T_{CH}	50	—	ns
CCLK Low Time	T_{CL}	50	—	ns
CCLK Frequency	F_C	—	10	MHz

Note: Daisy chaining of FPGAs is not supported in this mode.

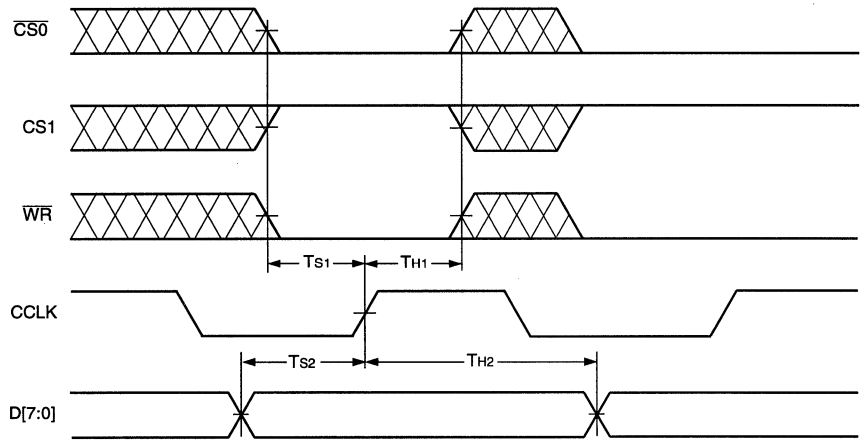


Figure 60. Slave Parallel Configuration Mode Timing Diagram

Timing Characteristics (continued)

Table 41. Readback Timing Characteristics

Commercial: V_{DD} = 5.0 V ± 5%, 0 °C ≤ T_A ≤ 70 °C; Industrial: V_{DD} = 5.0 V ± 10%, –40 °C ≤ T_A ≤ +85 °C; C_L = 50 pF.

Parameter	Symbol	Min	Max	Unit
$\overline{RD\_CFGN}$ to CCLK Setup Time	T_S	50	—	ns
$\overline{RD\_CFGN}$ High Width to Abort Readback	T_{RBA}	2	—	CCLK
CCLK Low Time	T_{CL}	50	—	ns
CCLK High Time	T_{CH}	50	—	ns
CCLK Frequency	F_C	—	10*	MHz
CCLK to RD_DATA Delay	T_D	—	50	ns

* The maximum readback CCLK frequency for the ATT2C40 is 8 MHz.

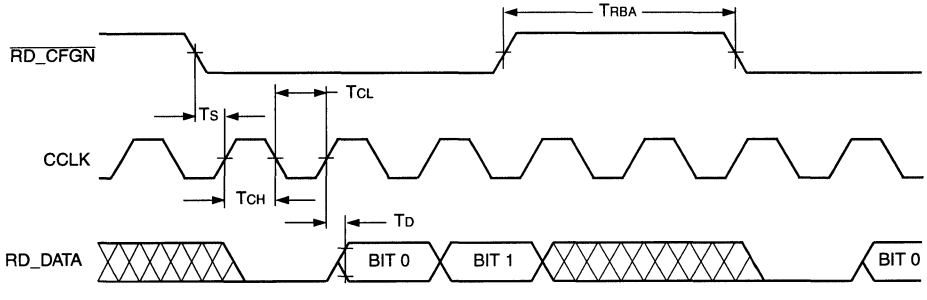


Figure 61. Readback Timing Diagram

Timing Characteristics (continued)

Table 42. Boundary-Scan Timing Characteristics

Commercial: V_{DD} = 5.0 V ± 5%, 0 °C ≤ T_A ≤ 70 °C; Industrial: V_{DD} = 5.0 V ± 10%, –40 °C ≤ T_A ≤ +85 °C; C_L = 50 pF.

Parameter	Symbol	Min	Max	Unit
TDI/TMS to TCK Setup Time	T_S	25	—	ns
TDI/TMS Hold Time from TCK	T_H	0	—	ns
TCK Low Time	T_{CL}	50	—	ns
TCK High Time	T_{CH}	50	—	ns
TCK to TDO Delay	T_D	—	20	ns
TCK Frequency	T_{TCK}	—	10	MHz

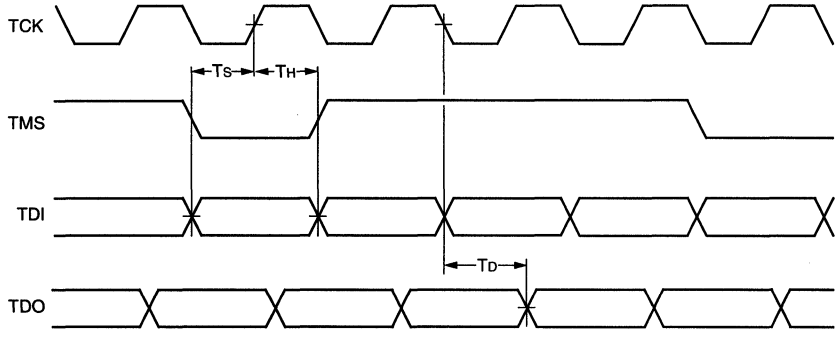
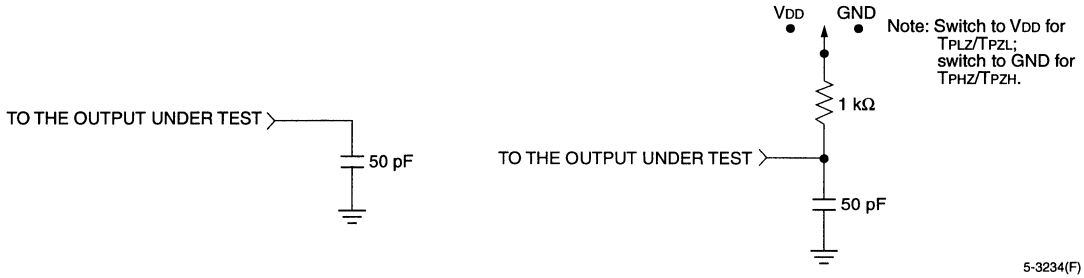


Figure 62. Boundary-Scan Timing Diagram

Measurement Conditions



A. Load Used to Measure Propagation Delay **B. Load Used to Measure Rising/Falling Edges**

Figure 63. ac Test Loads

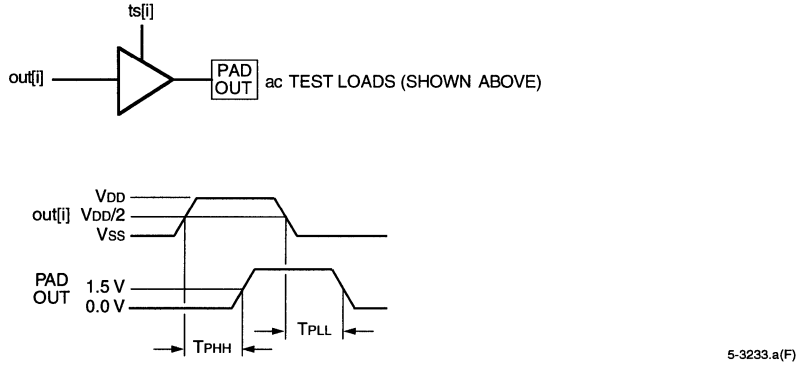


Figure 64. Output Buffer Delays

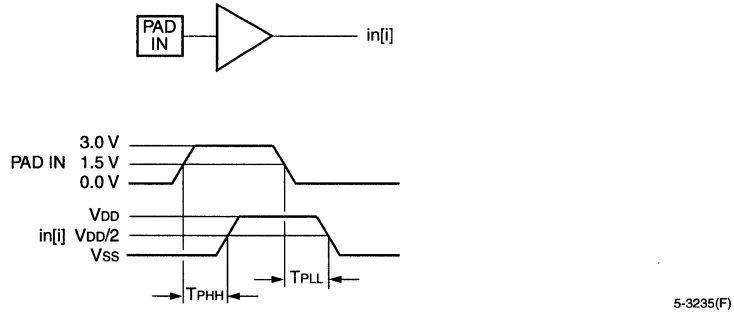
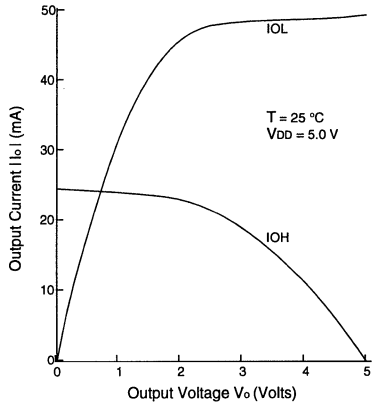
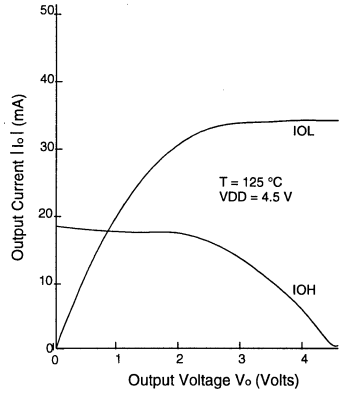


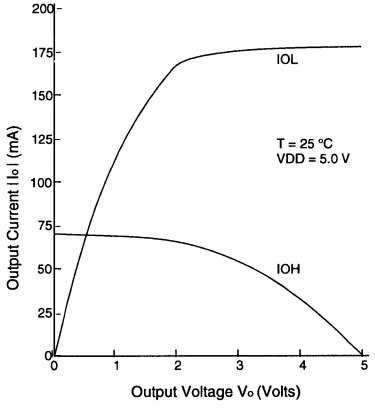
Figure 65. Input Buffer Delays

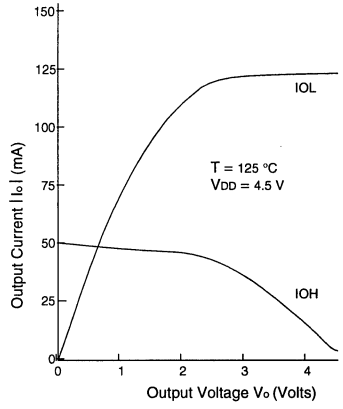
Output Buffer Characteristics



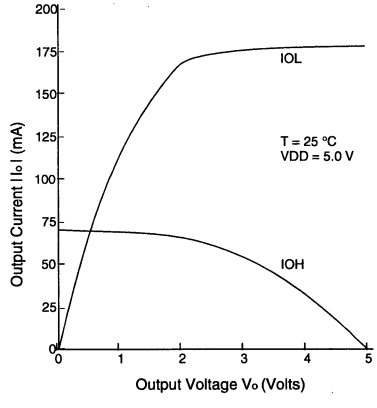


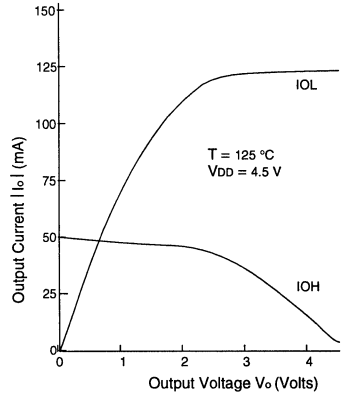
Sinklim





Slewlim

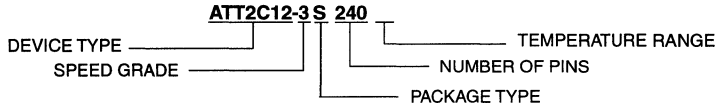




Fast

Ordering Information

Example:



ATT2C12, -3 Speed Grade, 240-pin Shrink Quad Flat Pack, Commercial Temperature.

Table 43. FPGA Temperature Options

Symbol	Description	Temperature
(Blank)	Commercial	0 °C to 70 °C
I	Industrial	–40 °C to +85 °C

Table 44. FPGA Package Options

Symbol	Description
B	Plastic Ball Grid Array (PBGA)
J	Quad Flat Package (QFP)
M	Plastic Leaded Chip Carrier (PLCC)
PS	Power Quad Shrink Flat Package (SQFP-PQ2)
R	Ceramic Pin Grid Array (CPGA)
S	Shrink Quad Flat Package (SQFP)
T	Thin Quad Flat Package (TQFP)

Table 45. *ORCA* 2C Series Package Matrix

Packages	84-Pin PLCC	100-Pin TQFP	144-Pin TQFP	160-Pin QFP	208-Pin EIAJ SQFP/ SQFP-PQ2	240-Pin EIAJ SQFP/ SQFP-PQ2	256-Pin PBGA	304-Pin EIAJ SQFP/ SQFP-PQ2	364-Pin CPGA	428-Pin CPGA
	M84	T100	T144	J160	S208/ PS208	S240/ PS240	B256	S304/ PS304	R364	R428
ATT2C04	CI	CI	CI	CI	CI	—	—	—	—	—
ATT2C06	CI	CI	CI	CI	CI	CI	—	—	—	—
ATT2C08	CI	—	—	CI	CI	CI	CI	CI	—	—
ATT2C10	CI	—	—	CI	CI	CI	CI	CI	—	—
ATT2C12	—	—	—	—	CI	CI	CI	CI	CI	—
ATT2C15	—	—	—	—	CI	CI	—	CI	CI	—
ATT2C26	—	—	—	—	CI	CI	—	CI	—	CI
ATT2C40	—	—	—	—	CI	CI	—	CI	—	CI

Key: C = commercial, I = industrial.

Note: The package options with the SQFP/SQFP-PQ2 designation in the table above use the SQFP package for all densities up to and including the ATT2C15, while the ATT2C26 uses the SQFP-PQ2 package (chip-up orientation), and the ATT2C40 uses the SQFP-PQ2 package (chip-down orientation).

Notes

Lucent Technologies
Bell Labs Innovations

ATT3000 Series Field-Programmable Gate Arrays

Features

- High performance:
  - — Up to 270 MHz toggle rates
  - — 4-input LUT delays < 3 ns
- User-programmable gate arrays
- Flexible array architecture:
  - — Compatible arrays, 2000 to 9000 gate logic complexity
  - — Extensive register and I/O capabilities
  - — Low-skew clock nets
  - — High fan-out signal distribution
  - — Internal 3-state bus capabilities
  - — TTL or CMOS input thresholds
  - — On-chip oscillator amplifier
- Standard product availability:
  - — Low-power 0.6 µm CMOS, static memory technology
  - — Pin-for-pin compatible with *Xilinx XC3000* and *XC3100* families
  - — Cost-effective, high-speed FPGAs
  - — 100% factory pretested
  - — Selectable configuration modes
- *ORCA* Foundry for ATT3000 Development System support
- All FPGAs processed on a QML-certified line
- Extensive packaging options

Description

The CMOS ATT3000 Series Field-Programmable Gate Array (FPGA) family provides a group of 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 an FPGA is shown in Figure 1.

The *ORCA* Foundry for ATT3000 development system provides automatic place and route of netlists. Logic and timing simulation 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 powerup. A serial configuration PROM can provide a very simple serial configuration program storage.

Table 1. ATT3000 Series FPGAs

FPGA	Logic Capacity (Available 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.

Table of Contents

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 powerup 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 *ORCA* Foundry for ATT3000 Development System generates the configuration program bit stream used to configure the FPGA. The memory loading process is independent of the user logic functions.

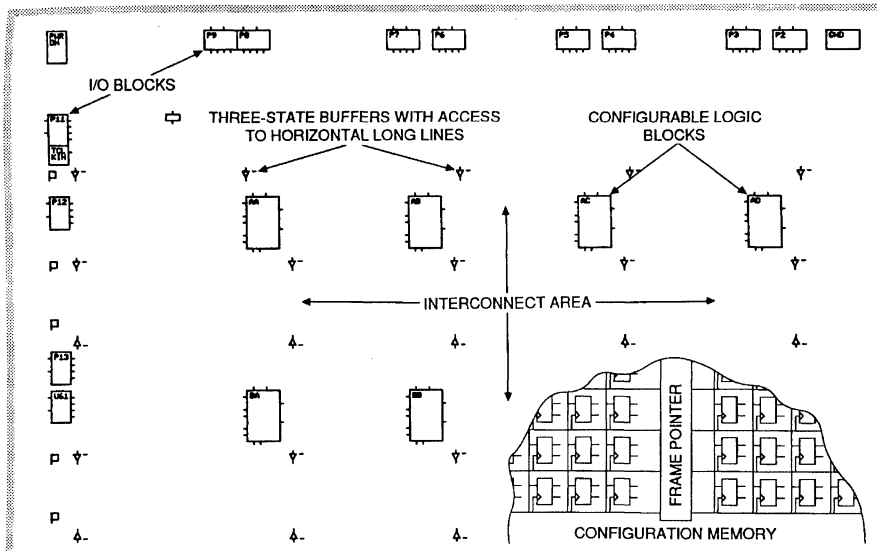


Figure 1. Field-Programmable Gate Array Structure

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 ensured 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 to during configuration and only read from during readback. During normal operation, the cell provides continuous control and the pass transistor is off and does not affect cell stability. This is quite different from the operation of conventional memory devices, in which the cells are frequently read and rewritten.

The memory cell outputs Q and $\overline{Q}$ use full Ground and V_{CC} levels and provide continuous, direct control. The additional capacitive load and the absence of address decoding and sense amplifiers provide high stability to the cell. Due to their structure, the configuration memory cells are not affected by extreme power supply excursions or very high levels of alpha particle radiation. Soft errors have not been observed in reliability testing.

Two methods of loading configuration data use serial data, while three use byte-wide data. The internal configuration logic utilizes framing information, embedded in the program data by the *ORCA* Foundry Development System, to direct memory cell loading. The serial data framing and length count preamble provide programming compatibility for mixes of various Lucent programmable gate arrays in a synchronous, serial, daisy-chain fashion.

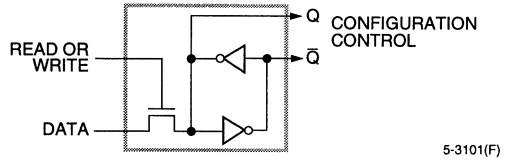


Figure 2. Static Configuration Memory Cell

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 IOB includes both registered and direct input paths and a programmable 3-state output buffer which may be driven by a registered or direct output signal. Configuration options allow each IOB an inversion, a controlled slew rate, and a high-impedance pull-up. Each input circuit also provides input clamping diodes to provide electrostatic protection and circuits to inhibit latch-up produced by input currents.

The input buffer portion of each IOB provides threshold detection to translate external signals applied to the package pin to internal logic levels. The global input-buffer threshold of the IOB 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 (IOB 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 IOB pin .q) signals are available for interconnect.

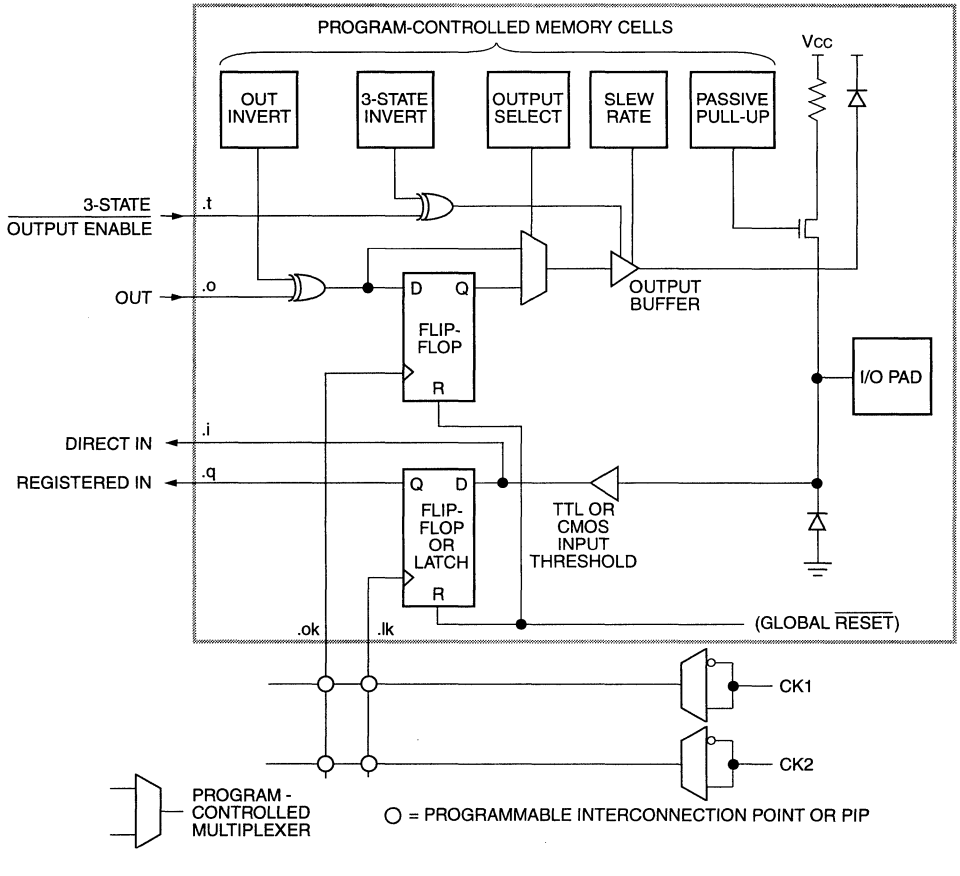


Figure 3. Input/Output Block

I/O Block (continued)

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 IOB includes a programmable high-impedance pull-up resistor which is selected by the program to provide a constant high for otherwise undriven package pins. Normal CMOS handling precautions should be observed.

Flip-flop loop delays for the IOB and logic block flip-flops are approximately 3 ns. 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 IOB flip-flops can be used to synchronize external signals applied to the device. When synchronized in the IOB, the signals can be used internally without further consideration of their clock relative timing, except as it applies to the internal logic and routing path delays.

Output buffers of the IOBs provide CMOS-compatible 4 mA source-or-sink drive for high fan-out CMOS or TTL compatible signal levels. The network driving IOB pin .o becomes the registered or direct data source for the output buffer. The 3-state control signal (IOB pin .t) can control output activity. An open-drain type output may be obtained by using the same signal for driving the output and 3-state signal nets so that the buffer output is enabled only for a LOW.

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

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

- Logical inversion of the output is controlled by one configuration program bit per IOB.
- Logical 3-state control of each IOB output buffer is determined by the states of configuration program bits which turn the buffer on or off or select the output buffer 3-state control interconnection (IOB pin .t). When this IOB output control signal is high, a logic 1, the buffer is disabled and the package pin is high impedance. When this IOB output control signal is low, a logic 0, the buffer is enabled and the package pin is active. Inversion of the buffer 3-state control logic sense (output enable) is controlled by an additional configuration program bit.
- Direct or registered output is selectable for each IOB. The register uses a positive-edge, clocked flip-flop. The clock source may be supplied (IOB pin .ok) by either of two metal lines available along each die edge. Each of these lines is driven by an invertible buffer.
- Increased output transition speed can be selected to improve critical timing. Slower transitions reduce capacitive load peak currents of noncritical outputs and minimize system noise.
- A high-impedance pull-up resistor may be used to prevent unused inputs from floating.

Summary of I/O Options

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

Configurable Logic Block

The array of configurable logic blocks (CLBs) provides the functional elements from which the user's logic is constructed. The logic blocks are arranged in a matrix within the perimeter of IOBs. The ATT3020 has 64 such blocks arranged in eight rows and eight columns. The *ORCA* Foundry Development System is used to compile the configuration data for loading into the internal configuration memory to define the operation and interconnection of each block. User definition of CLBs and their interconnecting networks may be done by automatic translation from a schematic capture logic diagram or optionally by installing library or user macros.

Each CLB has a combinatorial logic section, two flip-flops, and an internal control section; see Figure 4 below. There are five logic inputs (.a, .b, .c, .d, and .e); a common clock input (.k); an asynchronous direct reset input (.rd); and an enable clock (.ec). All may be driven from the interconnect resources adjacent to the blocks.

Each CLB also has two outputs (.x and .y) which may drive interconnect networks. Data input for either flip-flop within a CLB is supplied from the function F or G outputs of the combinatorial logic, or the block input, data-in (.di). Both flip-flops in each CLB share the asynchronous reset (.rd) which, when enabled and high, is dominant over clocked inputs. All flip-flops are reset by the active-low chip input, $\overline{\text{RESET}}$, or during the configuration process.

The flip-flops share the enable clock (.ec) which, when low, recirculates the flip-flops' present states and inhibits its 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 x 1 look-up table to implement Boolean functions. Variables selected from the five logic inputs and the 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.

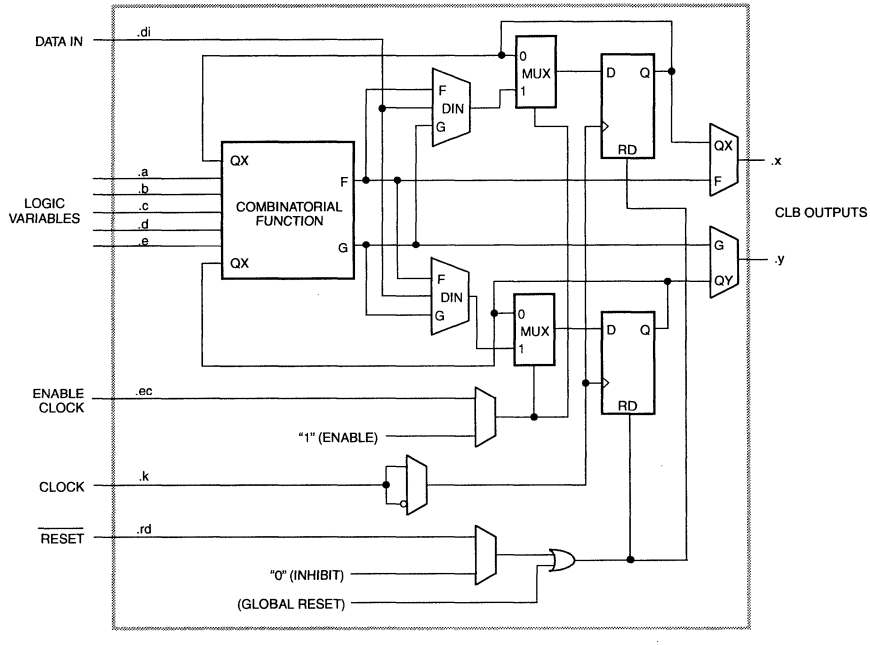
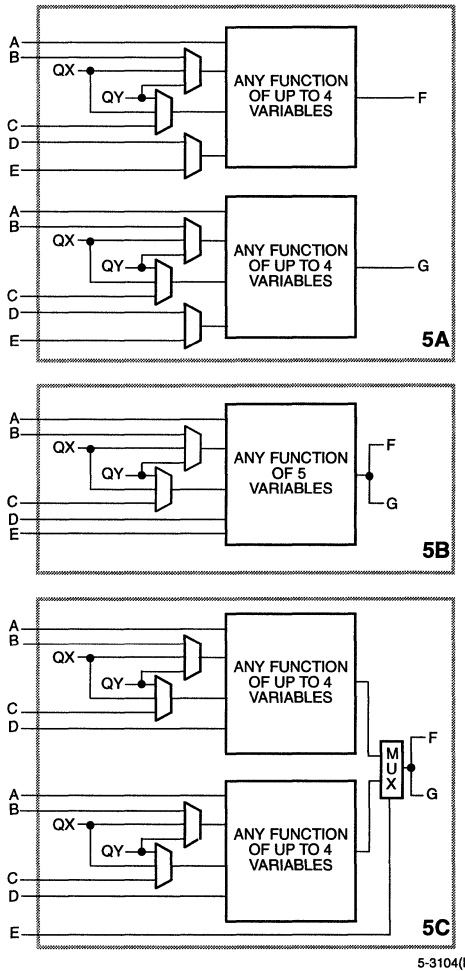


Figure 4. Configurable Logic Block

Configurable Logic Block (continued)



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 variables can be any choice among B, C, Qx, and Qy. The fourth variable can be either D or E.

5B. Combinatorial Logic Option 2 generates any function of five variables: A, D, E, and two choices among 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 among B, C, Qx, and Qy for the remaining two variables. Option 3 can then implement some functions of six or seven variables.

Figure 5. Combinatorial Logic Diagram

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 by 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 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 and IOBs.

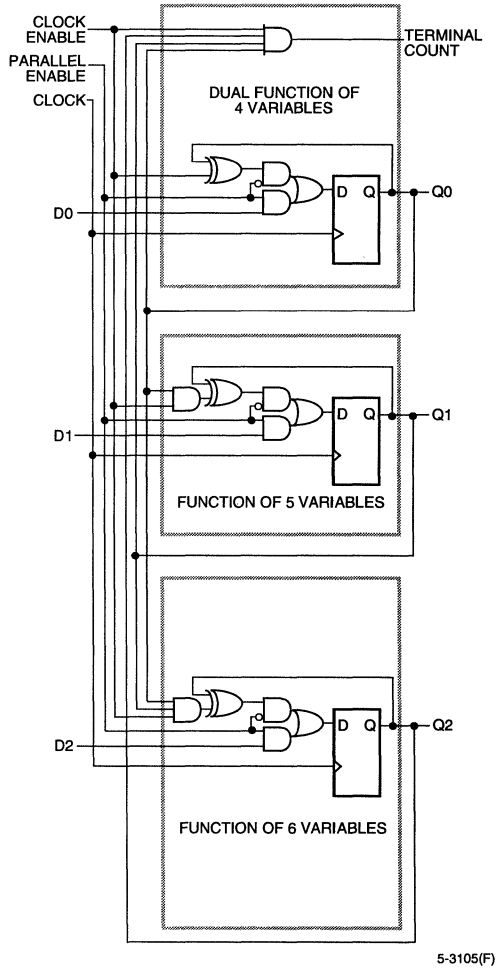


Figure 6. C8BCP Macro

Programmable Interconnect

Programmable interconnection resources in the FPGA provide routing paths to connect inputs and outputs of the IOBs 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 *ORCA* Foundry Development System provides automatic routing of these interconnections. Interactive routing is also available for design optimization. The inputs of the logic or IOBs are multiplexers which can be programmed to select an input network from the adjacent interconnect segments. Since 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 buses and wide-AND gates)

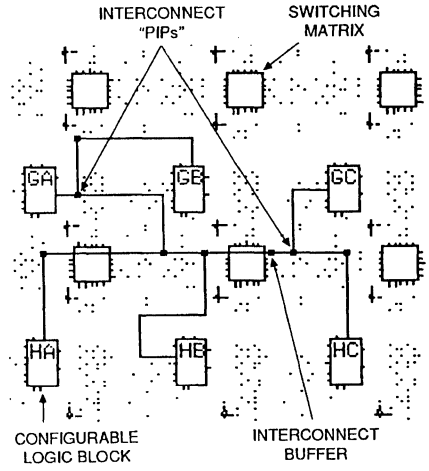


Figure 7. Example of Routing Resources

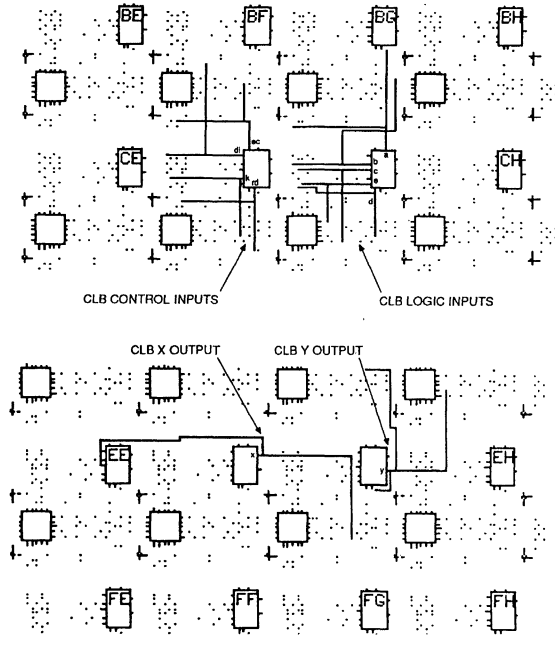


Figure 8. CLB Input and Output Routing

Programmable Interconnect (continued)

General-Purpose Interconnect

General-purpose interconnect, as shown in Figure 9, consists of a grid of five horizontal and five vertical metal segments located between the rows and columns of logic and IOBs. Each segment is the height or width of a logic block. Switching matrices join the ends of these segments and allow programmed interconnections between the metal grid segments of adjoining rows and columns. The switches of an unprogrammed device are all nonconducting. The connections through the switch matrix may be established by automatic or interactive routing by selecting the desired pairs of matrix pins to be connected or disconnected. The legitimate switching matrix combinations for each pin are indicated in Figure 10.

Special buffers within the general interconnect areas provide periodic signal isolation and restoration for improved performance of lengthy nets. The interconnect buffers are available to propagate signals in either direction on a given general interconnect segment. These bidirectional (bidi) buffers are found adjacent to the switching matrices, above and to the right. The other PIPs adjacent to the matrices are accessed 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 in the *ORCA* Foundry 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 also provided by the development system.

Some of the interconnect PIPs are directional, as indicated below:

- 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 nonconducting; P1 is "on."

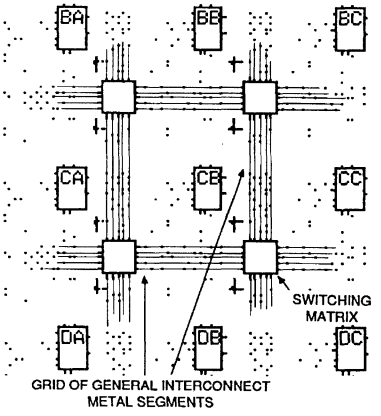


Figure 9. FPGA General-Purpose Interconnect

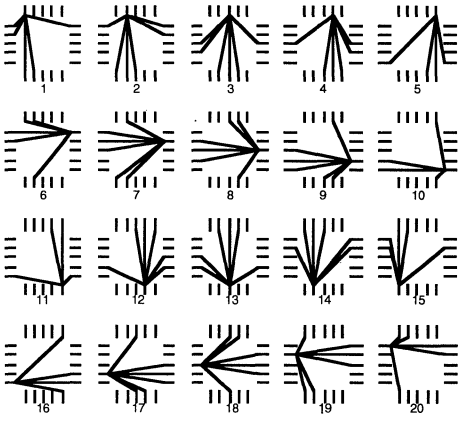


Figure 10. Switch Matrix Interconnection Options

Programmable Interconnect (continued)

Direct Interconnect

Direct interconnect (shown in Figure 11) provides the most efficient implementation of networks between adjacent logic or IOBs. Signals routed from block to block using the direct interconnect exhibit minimum interconnect propagation and use no general interconnect resources. For each CLB, the .x output may be connected directly to the .b input of the CLB immediately to its right and to the .c input of the CLB to its left. The .y output can use direct interconnect to drive the .d input of the block immediately above, and the .a input of the block below. Direct interconnect should be used to maximize the speed of high-performance portions of logic. Where logic blocks are adjacent to IOBs, direct connect is provided alternately to the IOB inputs (.i) and outputs (.o) on all four edges of the die. The right edge provides additional direct connects from CLB outputs to adjacent IOBs. Direct interconnections of IOBs with CLBs are shown in Figure 12.

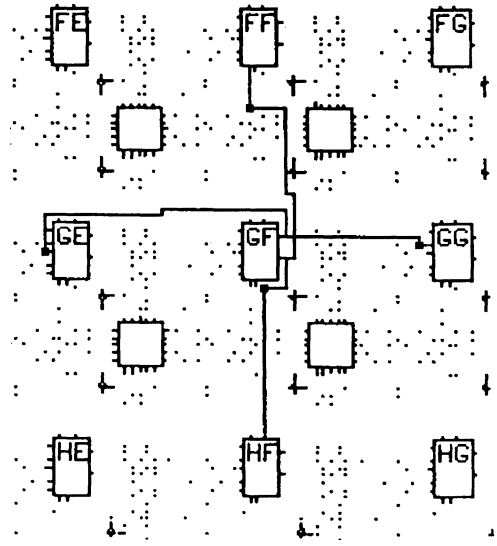


Figure 11. Direct Interconnect

Programmable Interconnect (continued)

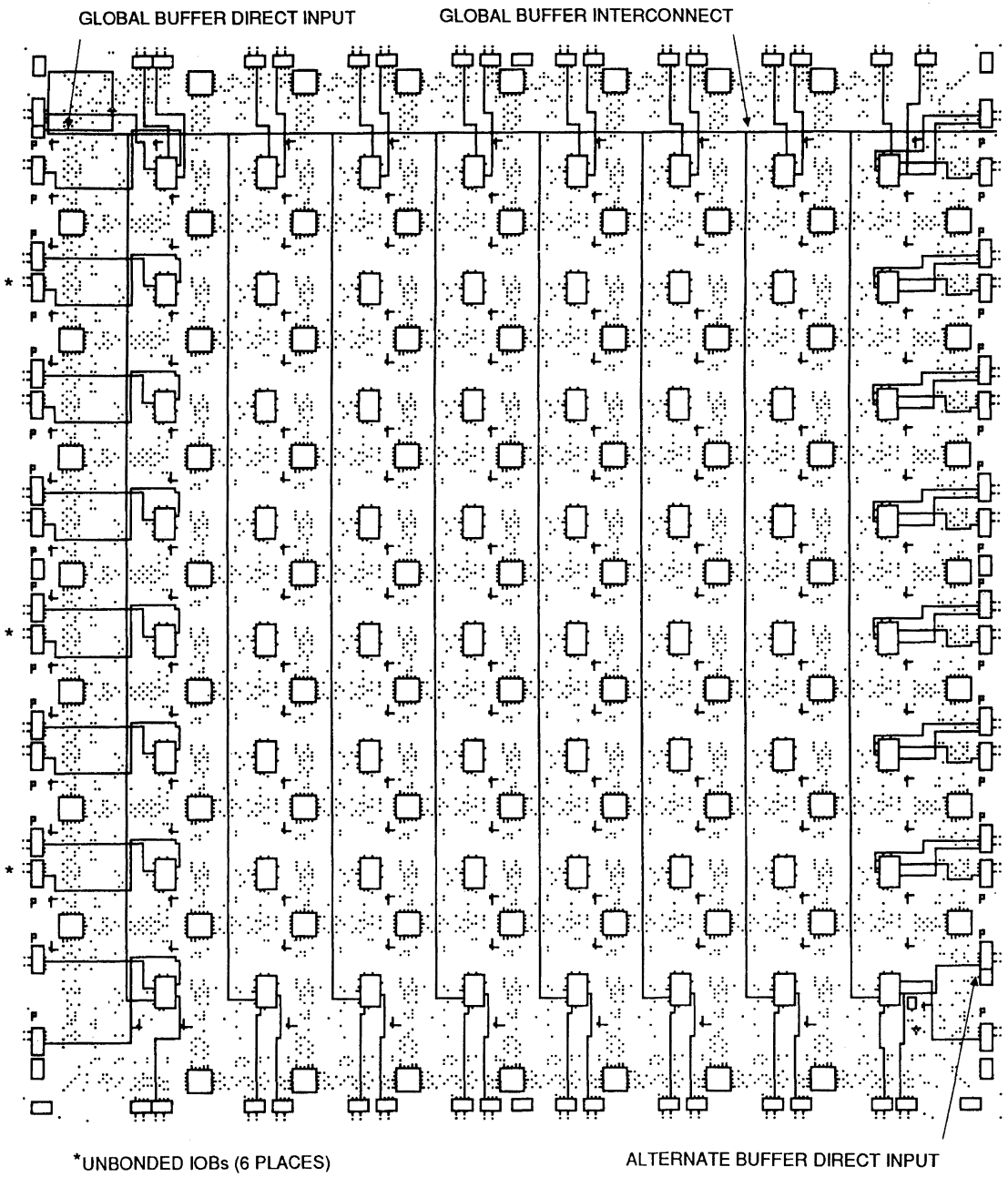


Figure 12. ATT3020 Die Edge I/O Blocks with Direct Access to Adjacent CLB

Programmable Interconnect (continued)

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. Additionally, two long lines are located adjacent to the outer sets of switching matrices. Two vertical long lines in each column are connectable 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 IOB output on a column-by-column basis. This capability provides a common low-skew control or clock line within each column of logic blocks. Interconnections of these long lines are shown in Figure 14. Isolation buffers are provided at each input to a long line and are enabled automatically by the development system when a connection is made.

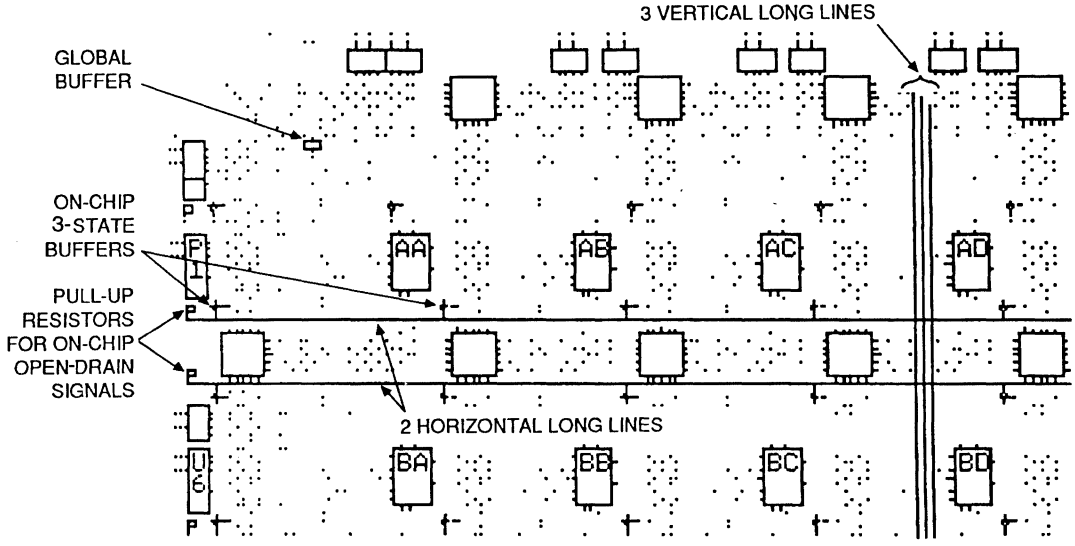


Figure 13. Horizontal and Vertical Long Lines in the FPGA

Programmable Interconnect (continued)

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 IOB use. An enhanced speed, CMOS threshold, offers direct access to this buffer and is available at the second pad from the top of the left die edge.

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

Internal Buses

A pair of 3-state buffers is located adjacent to each CLB. These buffers allow logic to drive the horizontal long lines. Logical operation of the 3-state buffer controls allows them to implement wide multiplexing functions. Any 3-state buffer input can be selected as drive for the horizontal long line bus by applying a low logic level on its 3-state control line (see Figure 15A). The user is required to avoid contention that can result from multiple drivers with opposing logic levels. Control of the 3-state input by the same signal that drives the buffer input creates an open-drain wired-AND function. A 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 nonconducting. This forms fast, wide gating functions. When data drives the inputs and separate signals drive the 3-state control lines, these buffers form multiplexers (3-state 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 3-state buffers, long lines, and pull-up resistors.

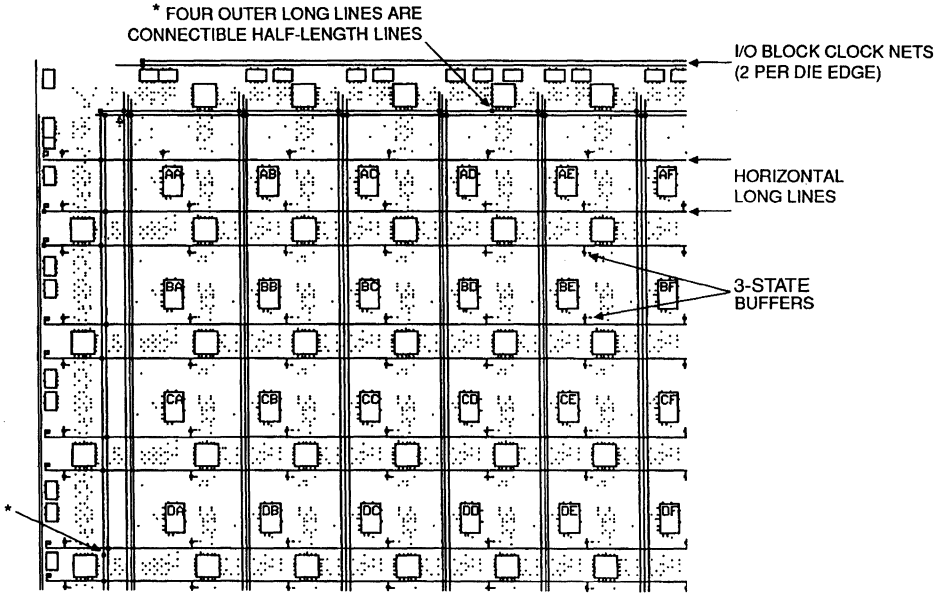


Figure 14. Programmable Interconnection of Long Lines

Programmable Interconnect (continued)

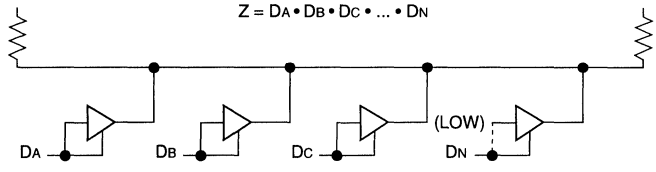


Figure 15A. 3-State Buffers Implement a Wired-AND Function

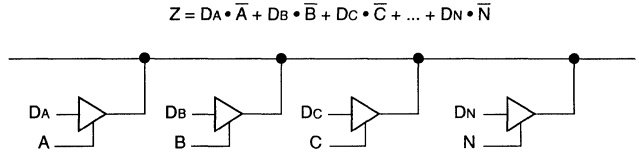


Figure 15B. 3-State Buffers Implement a Multiplexer

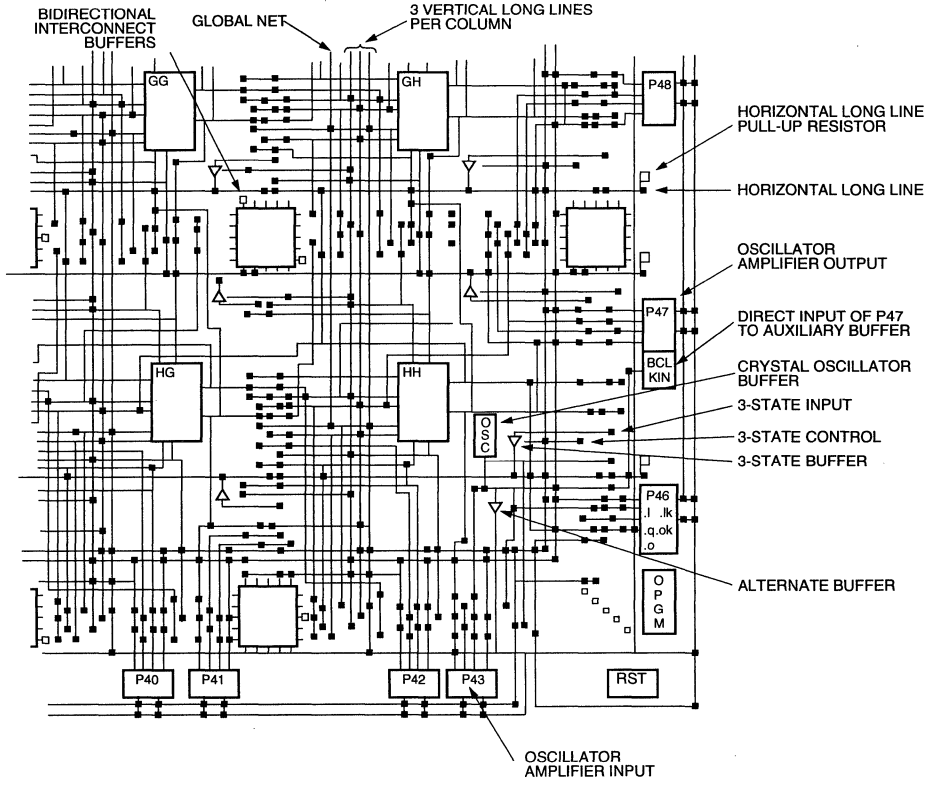
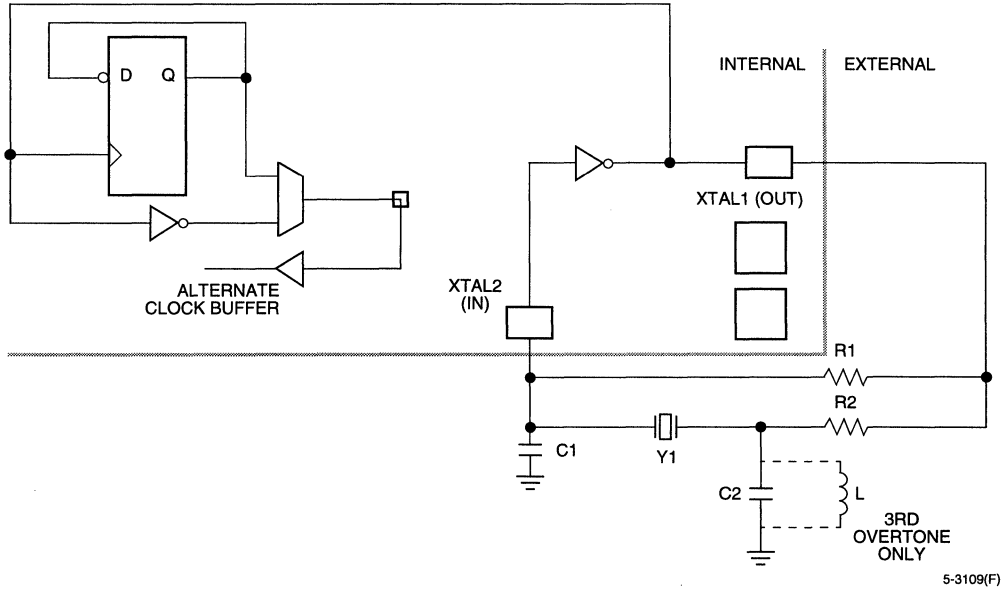


Figure 16. Lower-Right Corner of ATT3020

Programmable Interconnect (continued)

Crystal Oscillator

Figure 16 shows the location of an internal high-speed inverting amplifier which may be used to implement an on-chip crystal oscillator. It is associated with the auxiliary buffer in the lower right corner of the die. When the oscillator is configured by MAKEBITS and connected as a signal source, two special user IOBs are also configured to connect the oscillator amplifier with external crystal oscillator components as shown in Figure 17. A divide-by-two option is available to ensure 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 is 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, produces the 360° 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 or crystal resistance matching, or to limit the amplifier input swing to control clipping at large amplitudes. Excess feedback voltage may be corrected by the ratio of C2/C1. The amplifier is designed to be used from 1 MHz to one-half the specified CLB toggle frequency. Use at frequencies below 1 MHz may require individual characterization with respect to a series resistance. Crystal oscillators above 20 MHz generally require a crystal which operates in a third overtone mode, where the fundamental frequency must be suppressed by the R-C networks. When the oscillator inverter is not used, these IOBs and their package pins are available for general user I/O.



Suggested component values:

R1—1 μΩ to 4 μΩ
R2—0 kΩ to 1 kΩ (may be required for low frequency, phase shift, and/or compensation level for Crystal Q)
C1, C2—10 pF to 40 pF
Y1—1 MHz to 20 MHz AT cut series resonant

Pin	44-Pin PLCC	68-Pin PLCC	84-Pin	100-Pin		132-Pin PPGA	144-Pin TQFP	160-Pin MQFP	175-Pin PPGA	208-Pin SQFP
			PLCC	MQFP	TQFP					
XTAL1 (OUT)	30	47	57	82	79	P13	75	82	T14	110
XTAL2 (IN)	26	43	53	76	73	M13	69	76	P15	100

Figure 17. Crystal Oscillator Inverter

Programming

Initialization Phase

An internal power-on-reset circuit is triggered when power is applied. When V_{CC} reaches the voltage where portions of the FPGA begin to operate (2.5 V to 3 V), the programmable I/O output buffers are disabled and a high-impedance pull-up resistor is provided for the user I/O pins. A time-out delay is initiated to allow the power supply voltage to stabilize. During this time, the power-down mode is inhibited. The initialization state time-out (about 11 ms 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 MHz to 1.5 MHz. As shown in Table 2, five configuration mode choices are available, as determined by the input levels of three mode pins: M0, M1, and M2.

Table 2. Configuration Modes

M0	M1	M2	Clock	Mode	Data
0	0	0	Active	Master	Bit Serial
0	0	1	Active	Master	Byte Wide (Address = 0000 up)
0	1	0	—	Reserved	—
0	1	1	Active	Master	Byte Wide (Address = FFFF down)
1	0	0	—	Reserved	—
1	0	1	Passive	Peripheral	Byte Wide
1	1	0	—	Reserved	—
1	1	1	Passive	Slave	Bit Serial

In master-configuration modes, the FPGA becomes the source of configuration clock (CCLK). Beginning configuration of devices using peripheral or slave modes must be delayed long enough for their initialization to be completed. An FPGA with mode lines selecting a master configuration mode extends its initialization state using four times the delay (43 ms to 130 ms) to ensure that all daisy-chained slave devices it may be driving will be ready, even if the master is very fast and the slave(s) very slow (see Figure 18). At the end of initialization, the FPGA enters the clear state where it clears 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 reassertion 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 resample $\overline{\text{RESET}}$ and the mode lines before re-entering the configuration state. A reprogram 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 configuration memory is cleared and mode lines resampled, as for an aborted configuration. The complete configuration program is cleared and loaded during each configuration program cycle.

Programming (continued)

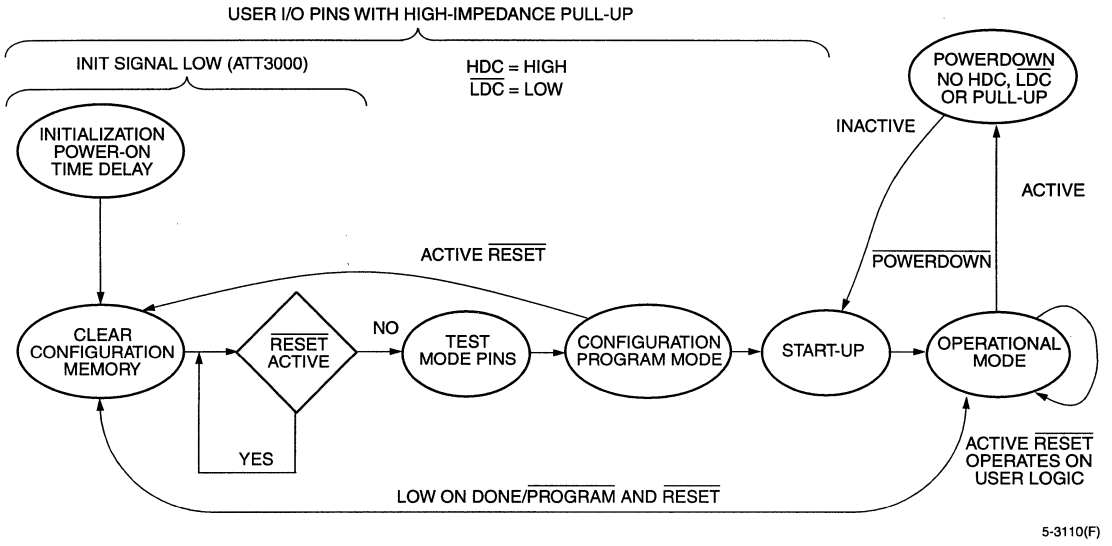


Figure 18. State Diagram of Configuration Process for Powerup and Reprogram

Length count control allows a system of multiple FPGAs in assorted sizes to begin operation in a synchronized fashion. The configuration program generated by the MakePROM program of the *ORCA* Foundry Development System begins with a preamble of 111111110010 (binary) 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) CCLK 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 on page 312). 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 ($\overline{\text{LDC}}$) are two user I/O pins which are driven active when an FPGA is in initialization, clear, or configure states. These signals and DONE/$\overline{\text{PROG}}$ provide for control of external logic signals such as reset, bus enable, or PROM enable during configuration.

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

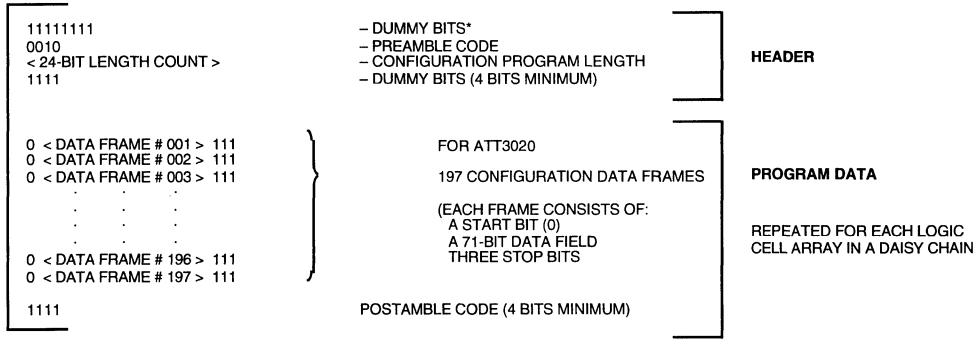
User I/O inputs can be programmed to be either TTL or CMOS compatible thresholds. At powerup, all inputs have TTL thresholds and can change to CMOS thresholds at the completion of configuration, if the user has selected CMOS thresholds. The threshold of $\overline{\text{PWRDWN}}$ and the direct clock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

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

Programming (continued)

Configuration Data

Configuration data to define the function and interconnection within an FPGA are loaded from an external storage at powerup and on a reprogram signal. Several methods of automatic and controlled loading of the required data are available. Logic levels applied to mode selection pins at the start of configuration time determine the method to be used (see Table 2). The data may be either bit-serial or byte-parallel, depending on the configuration mode. Various Lucent 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 FPGA devices require four dummy bits minimum.

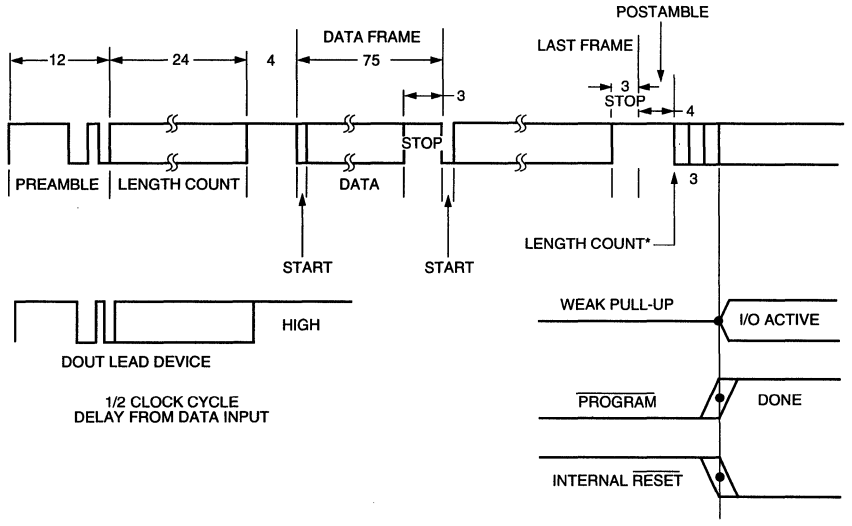
Figure 19. Internal Configuration Data Structure

Programming (continued)

Table 3. ATT3000 Device Configuration Data

Device	ATT3020	ATT3030	ATT3042	ATT3064	ATT3090
Gates	2000	3000	4200	6400	9000
CLBs (row x column)	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 (with 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

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



* 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 byte boundary. The length count is two less than the number of resulting bits. Timing of the assertion of DONE and termination of the internal RESET may each be programmed to occur one cycle before or after the I/O outputs become active.

Figure 20. FPGA Configuration and Start-Up

Programming (continued)

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 MAKEPROM command of the *ORCA* Foundry 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 indeterminate levels which might produce parasitic supply currents. TIE can be omitted for quick breadboard iterations where a few additional mA of I_{CC} are acceptable.

The configuration bit stream begins with high preamble bits, a 4-bit preamble code, and a 24-bit length count. When configuration is initiated, a counter in the FPGA is set to 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.

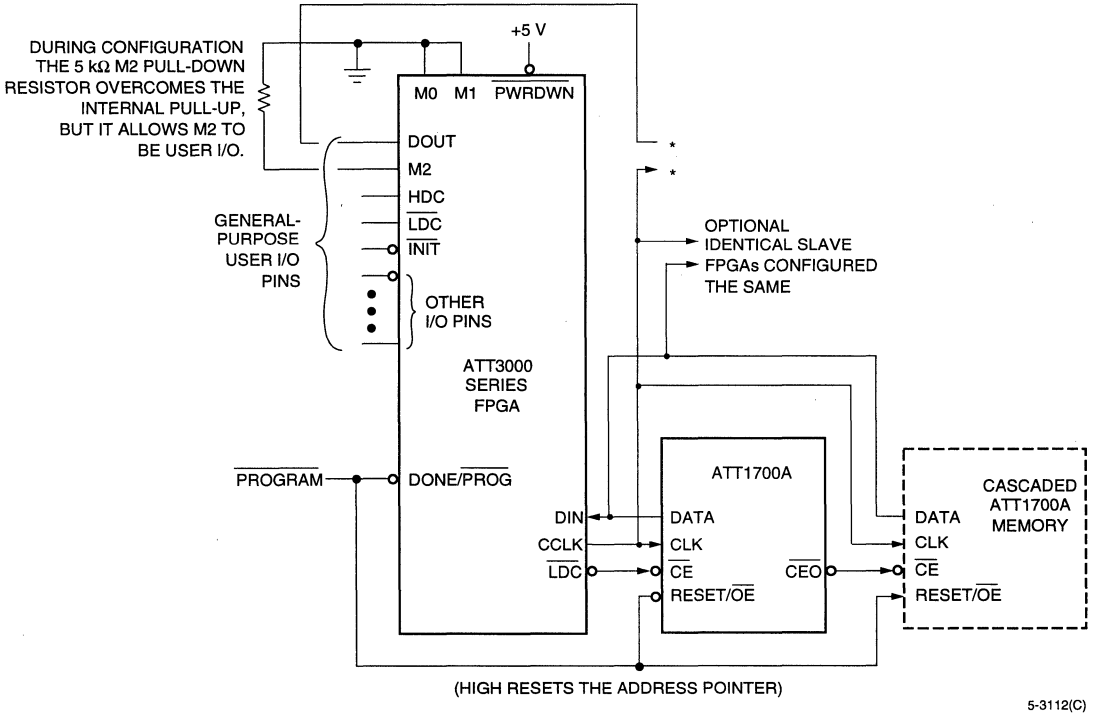
Two user-programmable pins are defined in the unconfigured FPGA: high during configuration (HDC) and low during configuration ($\overline{LDC}$), and DONE/$\overline{PROG}$ may be used as external control signals during configuration. In master mode configurations, it is convenient to use $\overline{LDC}$ as an active-low EPROM chip enable. After the last configuration data bit is loaded and the length count compares, the user I/O pins become active. Options in the MAKEBITS program allow timing choices of one clock earlier or later for the timing of the end of the internal logic reset and the assertion of the DONE signal. The open-drain DONE/$\overline{PROG}$ output can be AND-tied with multiple 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.

Programming (continued)

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 19. Parallel master low and master high modes automatically use parallel data supplied to the D[7:0] 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 read in parallel by each read clock ($\overline{RCLK}$) 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.



Note: The serial configuration PROM supports automatic loading of configuration programs up to 36/64/128 Kbits. Multiple devices can be cascaded to support additional FPGAs. An early DONE inhibits the data output one CCLK cycle before the FPGA I/O becomes active.

Figure 21. Master Serial Mode

Programming (continued)

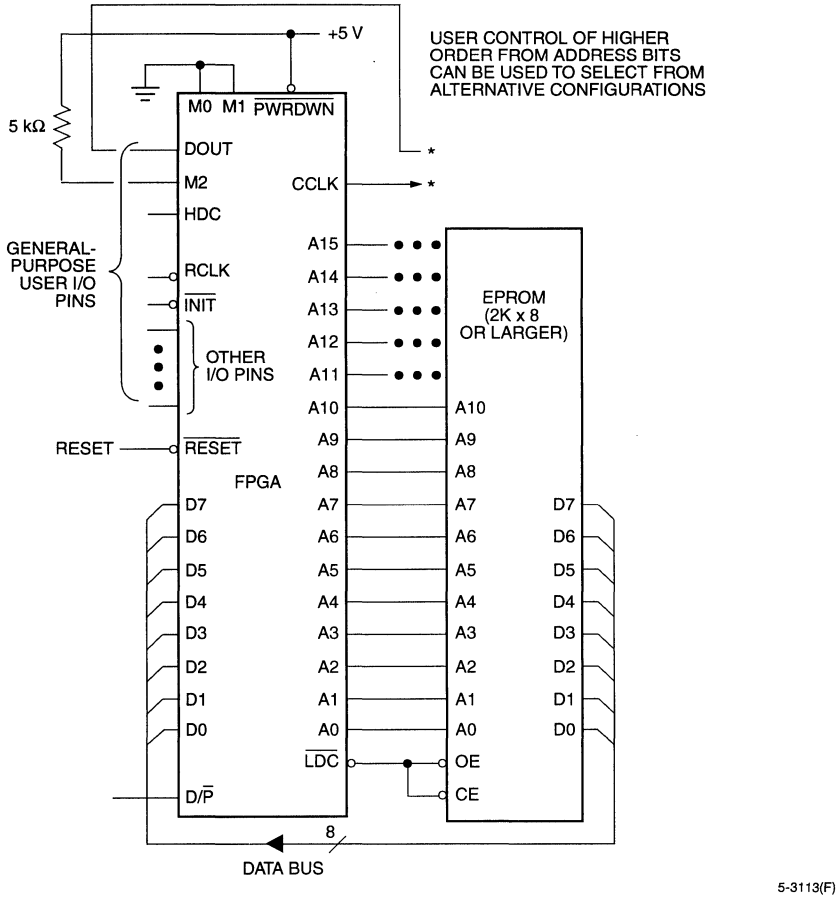


Figure 22. Master Parallel Mode

Programming (continued)

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 1 byte of configuration data on the D[7:0] inputs for each selected processor write cycle. Each byte of data is loaded into a buffer register. The FPGA generates a CCLK from the internal timing generator and serializes the parallel input data for internal framing or for succeeding slaves on data out (DOUT). An 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.

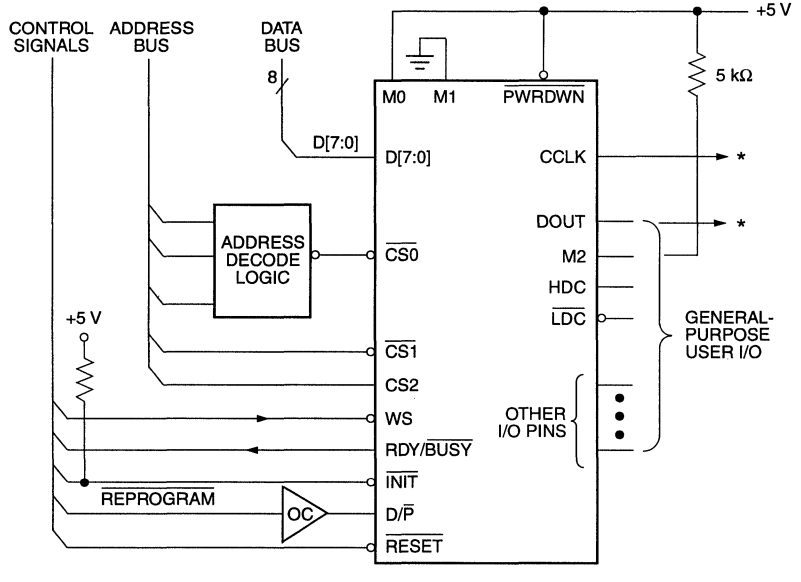


Figure 23. Peripheral Mode

Programming (continued)

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 is 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.

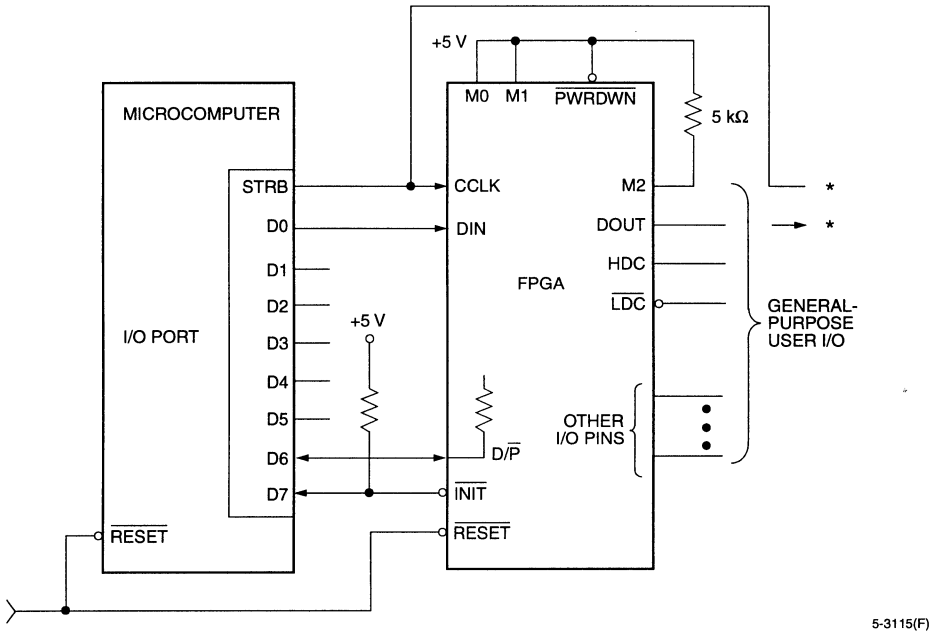


Figure 24. Slave Mode

Programming (continued)

Daisy Chain

The *ORCA* Foundry for ATT3000 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, a postamble, plus an additional fill bit per device in the serial chain. After loading and passing on the preamble and length count to a possible daisy chain, a lead device will load its configuration data frames while providing a high DOUT to possible downstream 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. 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 CCLK to synchronize the serial output data and data in of downstream FPGAs. Data is read in on DIN of slave devices by the positive edge of CCLK and shifted out the DOUT on the negative edge of CCLK. A parallel master mode device uses its internal timing generator to produce an internal CCLK of eight times its EPROM address rate, while a peripheral mode device produces a burst of eight 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.

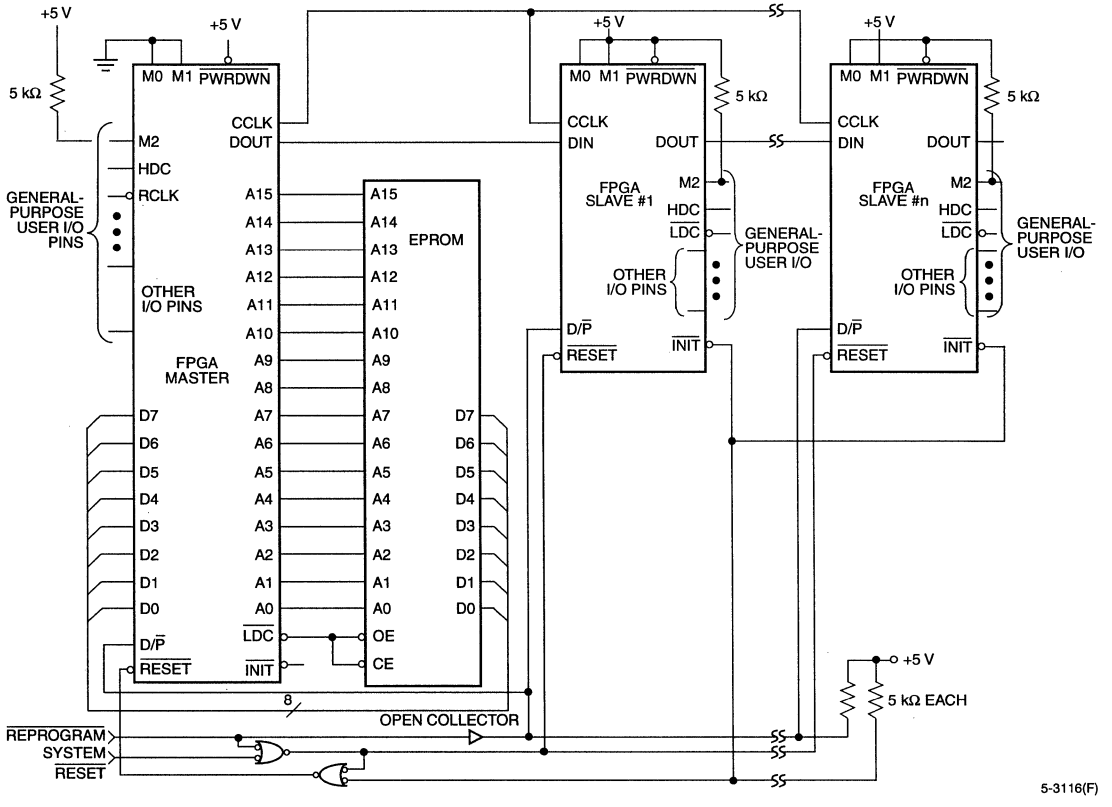


Figure 25. Master Mode with Daisy-Chained Slave Mode Devices

Special Configuration Functions

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

- 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 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 $\overline{\text{PWRDWN}}$ input and direct clocks which always have a CMOS input. Prior to the completion of configuration, the user I/O pins each have a high-impedance pull-up. The configuration program can be used to enable the IOB pull-up resistors in the operational mode to act either as an input load or to avoid a floating input on an otherwise unused pin.

Readback

The contents of an 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 for determining the state of internal logic nodes. 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/RDATA (read data) pin. All data frames must be read back to complete the process and return the mode select and CCLK pins to their normal functions.

The readback data includes the current state of each internal logic block storage element, and the state of the (.i and .ri) connection pins on each IOB. The data is 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 read back a uniform time sample of all storage elements, it may be necessary to inhibit the system clock.

Reprogram

The FPGA configuration memory can be rewritten while the device is operating in the user's system. To initiate a reprogramming 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 two cycles of the FPGA's internal timing generator. When reprogram begins, the user-programmable I/O output buffers are disabled and high-impedance pull-ups are provided for the package pins. The device returns to the clear state and clears the configuration memory before it prompts **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, wire-AND the slave $\overline{\text{INIT}}$ pins and use them to force a $\overline{\text{RESET}}$ on the master (see Figure 25). Reprogram control is often implemented by 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 reprogram request is externally held low beyond the configuration period, the FPGA will begin operation upon completion of configuration.

Special Configuration Functions
(continued)

DONE Pull-Up

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

DONE Timing

The timing of the DONE status signal can be controlled by a selection 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 $\overline{\text{RESET}}$ 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 Q to form the toggle flip-flop.

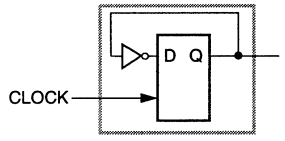


Figure 26. Toggle Flip-Flop

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 *ORCA* Foundry Development System timing calculator or *ORCA* Foundry-generated simulation models should be used to calculate worst-case paths by using actual impedance and loading information.

Figure 27 shows a variety of elements which are involved in determining system performance. Table 20 gives the parameter values for the different speed grades. 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 the 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 setup 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.

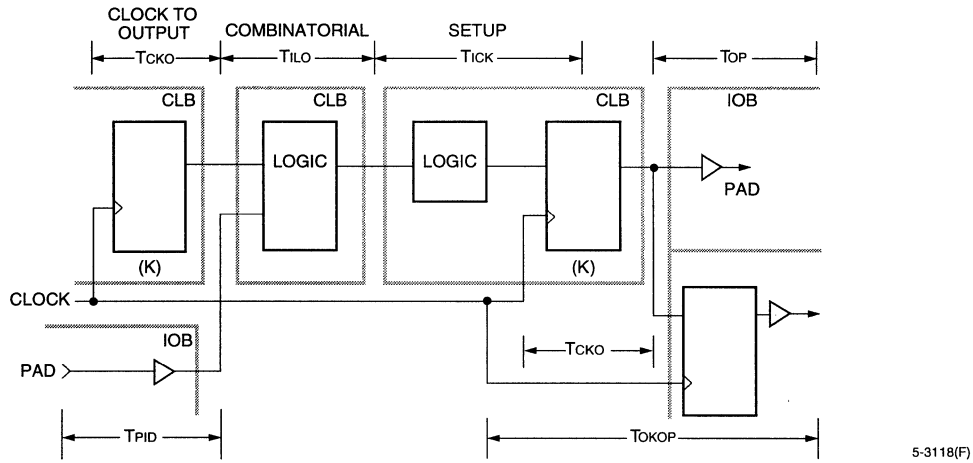


Figure 27. Examples of Primary Block Speed Factors

Performance (continued)

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 setup 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 bidirectional repowering buffers, and the overall loading on the signal path at all points along the path. In calculating the worst-case timing for a general interconnect path, the timing calculator portion of the *ORCA* Foundry 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 repowering buffer. The capacitance of the actual block inputs is not significant; the capacitance is in the interconnect metal and switches. Figure 30 illustrates this.

Performance (continued)

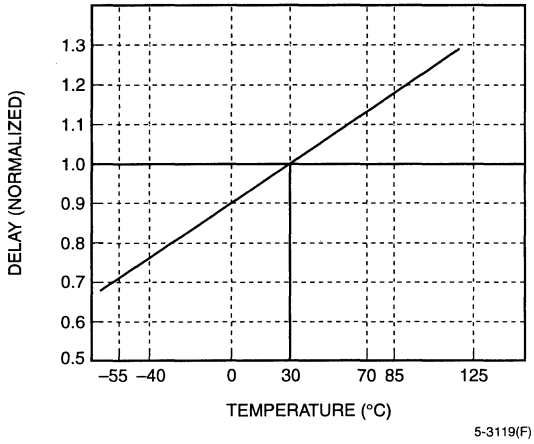


Figure 28. Change in Speed Performance

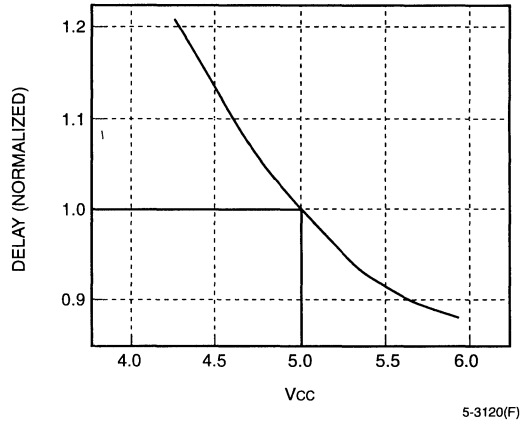


Figure 29. Speed Performance of a CMOS Device

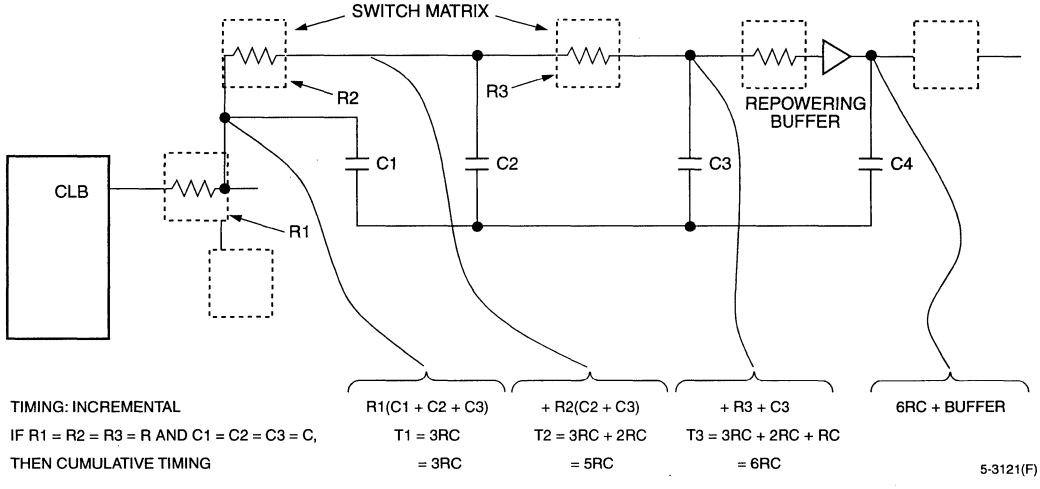


Figure 30. Interconnection Timing Example

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 below). 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 which drive the specified 4 mA loads under worst-case conditions may drive 25 to 30 times this amount under best-case process 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 IOB 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 simultaneously.

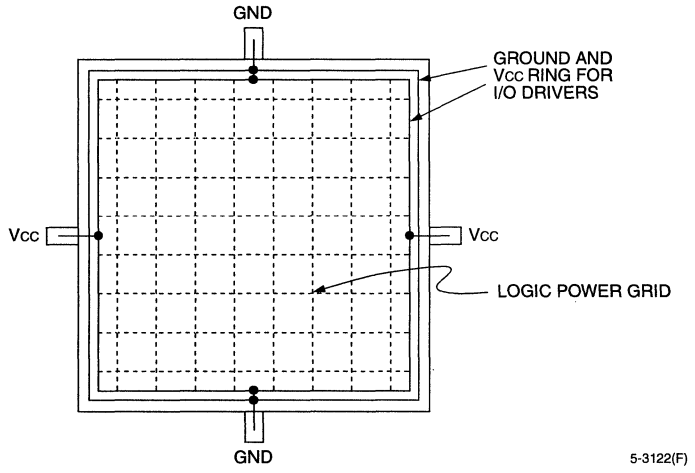


Figure 31. FPGA Power Distribution

Power (continued)

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 powerdown 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.

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% to 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:

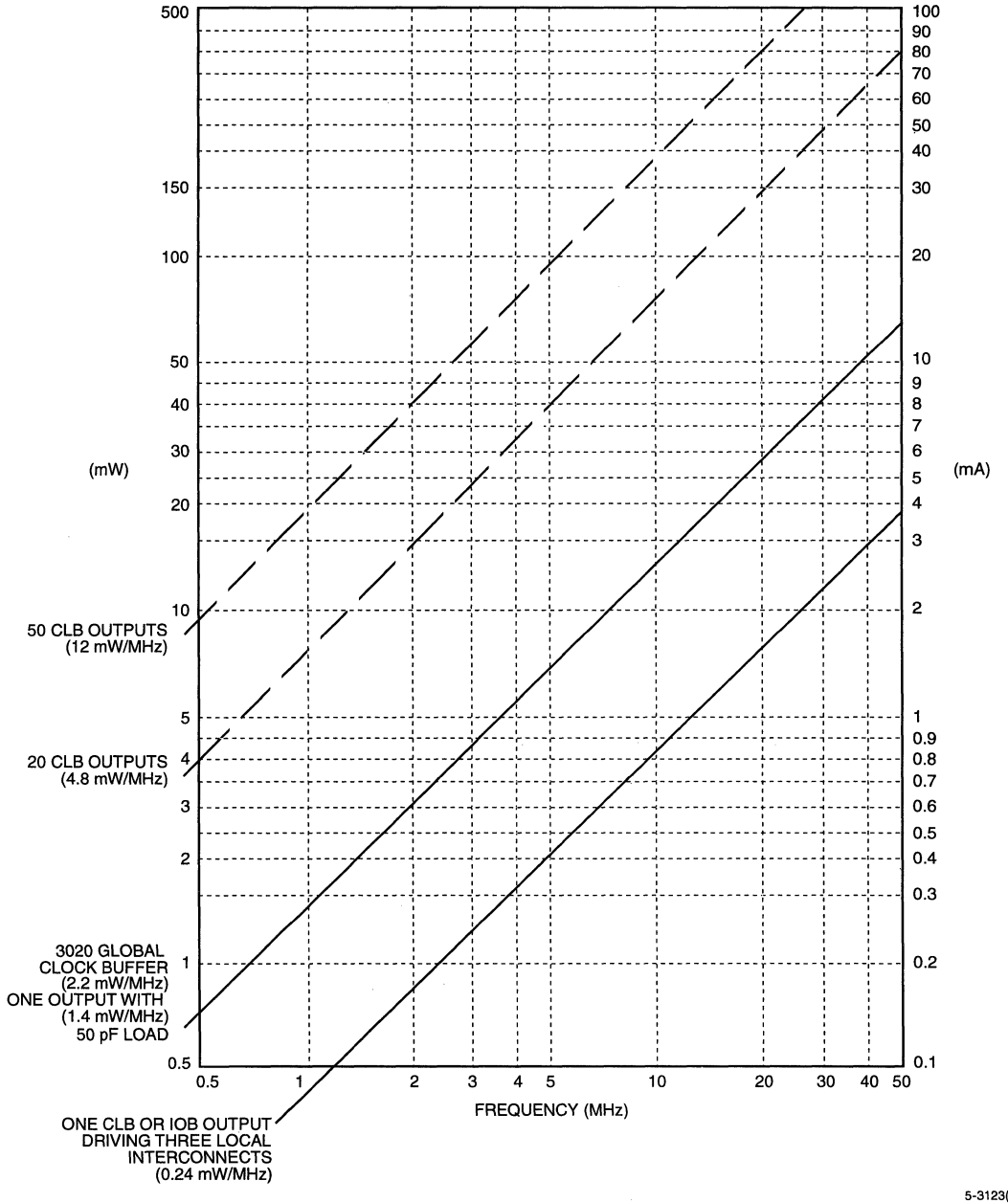
Total Power = V_{CC} + I_{CCO} + External (dc + Capacitive) + Internal (CLB + IOB + Long Line + Pull-up)

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 powerdown 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. Powerdown data retention is possible with a simple battery backup circuit, because the power requirement is extremely low. For retention at 2.4 V, the required current is typically on the order of 50 nA.

To force the FPGA into the powerdown state, the user must pull the $\overline{\text{PWRDWN}}$ pin low and continue to supply a retention voltage to the V_{CC} pins of the package. When normal power is restored, V_{CC} 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 electrostatic input protection is implemented with diodes to the supply and ground. A positive voltage applied to an I/O 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 bipolar buffer may be used to isolate the input signal.

Power (continued)



5-3123(F)

Note: Total chip power is the sum of VCC x ICCO plus effective internal and external values of frequency-dependent capacitive charging currents and duty-factor-dependent resistive loads.

Figure 32. FPGA Power Consumption by Element

Pin Information

Table 4. Permanently Dedicated Pins

Symbol	Name/Description
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; configuration is retained. When the $\overline{\text{PWRDWN}}$ pin returns high, the device returns to operation with the same sequence of buffer enable and DONE/$\overline{\text{PROG}}$ 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 reinitialized 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	**Configuration Clock**. During configuration, this 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 filtered out.
DONE/ $\overline{\text{PROG}}$	**DONE Output**. 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 occurs. Once configuration is done, a high-to-low transition of this pin will cause an initialization of the FPGA and start a reconfiguration.
M0	**Mode 0**. This input, M1, and M2 are sampled before the start of configuration to establish the configuration mode to be used.

Pin Information (continued)

Table 5. I/O Pins with Special Functions

Symbol	Name/Description
M2	**Mode 2.** This input has a passive pull-up 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.** HDC 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.** This active-low signal 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 four generates a write to the internal data buffer. The removal of any assertion clocks in the D[7:0] data present. In the master parallel mode, $\overline{\text{WS}}$ and CS2 are the A0 and A1 outputs. After configuration, the pins are user-programmable I/O pins.

Pin Information (continued)

Table 5. I/O Pins with Special Functions (continued)

Symbol	Name/Description
$\overline{\text{RCLK}}$	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.
D[7:0]	This set of eight pins represents the parallel configuration byte for the parallel master and peripheral modes. After configuration is complete, they are user-programmed I/O pins.
A[15:0]	This set of 16 pins presents an address output for a configuration EPROM during master parallel mode. After configuration is complete, they are user-programmed I/O pins.
DIN	This user I/O pin is used as serial data input during slave or master serial configuration. This pin is data zero 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.
I/O	**Input/Output (Unrestricted).** 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).

Pin Information (continued)

Table 6A. ATT3000 Family Configuration (44-, 68-, and 84-PLCC; 100-MQFP; and 100-TQFP)

Configuration Mode (M2:M1:M0)					44 PLCC*	68 PLCC	84 PLCC†	100 MQFP	100 TQFP	User Operation
Slave (1:1:1)	Master-Serial (0:0:0)	Peripheral (1:0:1)	Master-High (1:1:0)	Master-Low (1:0:0)						
PWRDWN	PWRDWN	PWRDWN	PWRDWN	PWRDWN	7	10	12	29	26	PWRDWN
VCC	VCC	VCC	VCC	VCC	12	18	22	41	38	VCC
M1 (High)	M1 (Low)	M1 (Low)	M1 (High)	M1 (Low)	16	25	31	52	49	RDATA
M0 (High)	M0 (Low)	M0 (Low)	M0 (High)	M0 (Low)	17	26	32	54	51	RTRIG
M2 (High)	M2 (Low)	M2 (High)	M2 (High)	M2 (Low)	18	27	33	56	53	I/O
HDC (High)	HDC (High)	HDC (High)	HDC (High)	HDC (High)	19	28	34	57	54	I/O
LDC (Low)	LDC (Low)	LDC (Low)	LDC (Low)	LDC (Low)	20	30	36	59	56	I/O
INIT ‡	INIT ‡	INIT ‡	INIT ‡	INIT ‡	22	34	42	65	62	I/O
GND	GND	GND	GND	GND	23	35	43	66	63	GND
					26	43	53	76	73	XTL2–I/O
RESET	RESET	RESET	RESET	RESET	27	44	54	78	75	RESET
DONE	DONE	DONE	DONE	DONE	28	45	55	80	77	PROG
		D7	D7	D7	—	46	56	81	78	I/O
					30	47	57	82	79	XTL1–I/O
		D6	D6	D6	—	48	58	83	80	I/O
		D5	D5	D5	—	49	60	87	84	I/O
		CS0	—	—	—	50	61	88	85	I/O
		D4	D4	D4	—	51	62	89	86	I/O
VCC	VCC	VCC	VCC	VCC	34	52	64	91	88	VCC
		D3	D3	D3	—	53	65	92	89	I/O
		CS1	—	—	—	54	66	93	90	I/O
		D2	D2	D2	—	55	67	94	91	I/O
		D1	D1	D1	—	56	70	98	95	I/O
		RDY/BUSY	RCLK	RCLK	—	57	71	99	96	I/O
DIN	DIN	D0	D0	D0	38	58	72	100	97	I/O
DOUT	DOUT	DOUT	DOUT	DOUT	39	59	73	1	98	I/O
CCLK	CCLK	CCLK	CCLK	CCLK	40	60	74	2	99	CCLK
		WS	A0	A0	—	61	75	5	2	I/O
		CS2	A1	A1	—	62	76	6	3	I/O
			A2	A2	—	63	77	8	5	I/O
			A3	A3	—	64	78	9	6	I/O
			A15	A15	—	65	81	12	9	I/O
			A4	A4	—	66	82	13	10	I/O
			A14	A14	—	67	83	14	11	I/O
			A5	A5	—	68	84	15	12	I/O
GND	GND	GND	GND	GND	1	1	1	16	13	GND
			A13	A13	—	2	2	17	14	I/O
			A6	A6	—	3	3	18	15	I/O
			A12	A12	—	4	4	19	16	I/O
			A7	A7	—	5	5	20	17	I/O
			A11	A11	—	6	8	23	20	I/O
			A8	A8	—	7	9	24	21	I/O
			A10	A10	—	8	10	25	22	I/O
			A9	A9	—	9	11	26	23	I/O

Represents a 50 kΩ to 100 kΩ pull-up.

* Peripheral mode and master parallel mode are not supported in the 44-PLCC package; see Table 7.

† Pin assignments for the ATT3064/ATT3090 differ from those shown; see page 2-261.

‡ INIT is an open-drain output during configuration.

Pin Information (continued)

Table 6B. ATT3000 Family Configuration (132-PPGA, 144-TQFP, 160-MQFP, 175-PPGA, 208-SQFP)

Configuration Mode (M2:M1:M0)					132 PPGA	144 TQFP	160 MQFP	175 PPGA	208 SQFP	User Operation
Slave (1:1:1)	Master-Serial (0:0:0)	Peripheral (1:0:1)	Master-High (1:1:0)	Master-Low (1:0:0)						
PWRDWN	PWRDWN	PWRDWN	PWRDWN	PWRDWN	A1	1	159	B2	3	PWRDWN
VCC	VCC	VCC	VCC	VCC	C8	19	20	D9	26	VCC
M1 (High)	M1 (Low)	M1 (Low)	M1 (High)	M1 (Low)	B13	36	40	B14	48	RDATA
M0 (High)	M0 (Low)	M0 (Low)	M0 (High)	M0 (Low)	A14	38	42	B15	50	RTRIG
M2 (High)	M2 (Low)	M2 (High)	M2 (High)	M2 (Low)	C13	40	44	C15	56	I/O
HDC (High)	HDC (High)	HDC (High)	HDC (High)	HDC (High)	B14	41	45	E14	57	I/O
LDC (Low)	LDC (Low)	LDC (Low)	LDC (Low)	LDC (Low)	D14	45	49	D16	61	I/O
INIT *	INIT *	INIT *	INIT *	INIT *	G14	53	59	H15	77	I/O
GND	GND	GND	GND	GND	H12	55	19	J14	25	GND
					M13	69	76	P15	100	XTL2–I/O
RESET	RESET	RESET	RESET	RESET	P14	71	78	R15	102	RESET
DONE	DONE	DONE	DONE	DONE	N13	73	80	R14	107	PROG
		D7	D7	D7	M12	74	81	N13	109	I/O
					P13	75	82	T14	110	XTL1–I/O
		D6	D6	D6	N11	78	86	P12	115	I/O
		D5	D5	D5	M9	84	92	T11	122	I/O
		CS0	—	—	N9	85	93	R10	123	I/O
		D4	D4	D4	N8	88	98	R9	128	I/O
VCC	VCC	VCC	VCC	VCC	M8	90	100	N9	130	VCC
		D3	D3	D3	N7	92	102	P8	132	I/O
		CS1	—	—	P6	93	103	R8	133	I/O
		D2	D2	D2	M6	96	108	R7	138	I/O
		D1	D1	D1	M5	102	114	R5	145	I/O
		RDY/BUSY	RCLK	RCLK	N4	103	115	P5	146	I/O
DIN	DIN	D0	D0	D0	N2	106	119	R3	151	I/O
DOUT	DOUT	DOUT	DOUT	DOUT	M3	107	120	N4	152	I/O
CCLK	CCLK	CCLK	CCLK	CCLK	P1	108	121	R2	153	CCLK
		WS	A0	A0	M2	111	124	P2	161	I/O
		CS2	A1	A1	N1	112	125	M3	162	I/O
			A2	A2	L2	115	128	P1	165	I/O
			A3	A3	L1	116	129	N1	166	I/O
			A15	A15	K1	119	132	M1	172	I/O
			A4	A4	J2	120	133	L2	173	I/O
			A14	A14	H1	123	136	K2	178	I/O
			A5	A5	H2	124	137	K1	179	I/O
GND	GND	GND	GND	GND	H3	126	139	J3	182	GND
			A13	A13	G2	128	141	H2	184	I/O
			A6	A6	G1	129	142	H1	185	I/O
			A12	A12	F2	133	147	F2	192	I/O
			A7	A7	E1	134	148	E1	193	I/O
			A11	A11	D1	137	151	D1	199	I/O
			A8	A8	D2	138	152	C1	200	I/O
			A10	A10	B1	141	155	E3	203	I/O
			A9	A9	C2	142	156	C2	204	I/O

Represents a 50 kΩ to 100 kΩ pull-up.

* INIT is an open-drain output during configuration.

Pin Assignments

Table 7. ATT3030 44-Pin PLCC Pinout

Pin No.	Function	Pin No.	Function
1	GND	23	GND
2	I/O	24	I/O
3	I/O	25	I/O
4	I/O	26	XTL2–I/O
5	I/O	27	$\overline{\text{RESET}}$
6	I/O	28	DONE–$\overline{\text{PROG}}$
7	$\overline{\text{PWRDWN}}$	29	I/O
8	TCLKIN–I/O	30	XTL1–BCLKIN–I/O
9	I/O	31	I/O
10	I/O	32	I/O
11	I/O	33	I/O
12	VCC	34	VCC
13	I/O	35	I/O
14	I/O	36	I/O
15	I/O	37	I/O
16	M1–$\overline{\text{RDATA}}$	38	DIN–I/O
17	M0–RTRIG	39	DOUT–I/O
18	M2–I/O	40	CCLK
19	HDC–I/O	41	I/O
20	$\overline{\text{LDC}}$–I/O	42	I/O
21	I/O	43	I/O
22	$\overline{\text{INIT}}$–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.

Pin Assignments (continued)

Table 8. ATT3020, ATT3030, and ATT3042; 68-PLCC and 84-PLCC Pinout*

Pin Numbers		Function	Pin Numbers		Function
68 PLCC	84 PLCC		68 PLCC	84 PLCC	
10	12	$\overline{\text{PWRDWN}}$	38	46	I/O
11	13	TCLKIN–I/O	39	47	I/O
—	14	I/O†	40	48	I/O
12	15	I/O	41	49	I/O
13	16	I/O	—	50	I/O†
—	17	I/O	—	51	I/O†
14	18	I/O	42	52	I/O
15	19	I/O	43	53	XTL2–I/O
16	20	I/O	44	54	$\overline{\text{RESET}}$
17	21	I/O	45	55	DONE–$\overline{\text{PROG}}$
18	22	Vcc	46	56	D7–I/O
19	23	I/O	47	57	XTL1–BCLKIN–I/O
—	24	I/O	48	58	D6–I/O
20	25	I/O	—	59	I/O
21	26	I/O	49	60	D5–I/O
22	27	I/O	50	61	$\overline{\text{CS0}}$–I/O
—	28	I/O	51	62	D4–I/O
23	29	I/O	—	63	I/O
24	30	I/O	52	64	Vcc
25	31	M1–$\overline{\text{RDATA}}$	53	65	D3–I/O
26	32	M0–RTRIG	54	66	$\overline{\text{CS1}}$–I/O
27	33	M2–I/O	55	67	D2–I/O
28	34	HDC–I/O	—	68	I/O
29	35	I/O	—	69	I/O†
30	36	$\overline{\text{LDC}}$–I/O	56	70	D1–I/O
31	37	I/O	57	71	RDY/$\overline{\text{BUSY}}$–$\overline{\text{RCLK}}$–I/O
—	38	I/O†	58	72	D0–DIN–I/O
32	39	I/O	59	73	DOUT–I/O
33	40	I/O	60	74	CCLK
—	41	I/O†	61	75	A0–$\overline{\text{WS}}$–I/O
34	42	$\overline{\text{INIT}}$–I/O	62	76	A1–CS2–I/O
35	43	GND	63	77	A2–I/O
36	44	I/O	64	78	A3–I/O
37	45	I/O	—	79	I/O†

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

† Indicates unconnected package pins for the ATT3020.

Pin Assignments (continued)

Table 8. ATT3020, ATT3030, and ATT3042; 68-PLCC and 84-PLCC Pinout* (continued)

Pin Numbers		Function	Pin Numbers		Function
68 PLCC	84 PLCC		68 PLCC	84 PLCC	
—	80	I/O†	4	4	A12–I/O
65	81	A15–I/O	5	5	A7–I/O
66	82	A4–I/O	—	6	I/O†
67	83	A14–I/O	—	7	I/O†
68	84	A5–I/O	6	8	A11–I/O
1	1	GND	7	9	A8–I/O
2	2	A13–I/O	8	10	A10–I/O
3	3	A6–I/O	9	11	A9–I/O

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

† Indicates unconnected package pins for the ATT3020.

Note: Table 8 describes the pin assignments for three different chips in two different packages. The function column lists 84 of the 118 pads on the ATT3042 and 84 of the 98 pads on the ATT3030. Ten pads (indicated by an asterisk) do not exist on the ATT3020, which has 74 pads; therefore, the corresponding pins on the 84-pin packages have no connections to an ATT3020.

Pin Assignments (continued)

Table 9. ATT3064 and ATT3090 84-PLCC Pinout

84 PLCC	Function	84 PLCC	Function	84 PLCC	Function
12	$\overline{\text{PWRDWN}}$	40	I/O	68	D2–I/O*
13	TCLKIN–I/O	41	$\overline{\text{INIT}}$–I/O*	69	I/O
14	I/O	42	VCC*	70	D1–I/O
15	I/O	43	GND	71	RDY/$\overline{\text{BUSY}}$–$\overline{\text{RCLK}}$–I/O
16	I/O	44	I/O	72	D0–DIN–I/O
17	I/O	45	I/O	73	DOUT–I/O
18	I/O	46	I/O	74	CCLK
19	I/O	47	I/O	75	A0–$\overline{\text{WS}}$–I/O
20	I/O	48	I/O	76	A1–CS2–I/O
21	GND*	49	I/O	77	A2–I/O
22	VCC	50	I/O	78	A3–I/O
23	I/O	51	I/O	79	I/O*
24	I/O	52	I/O	80	I/O*
25	I/O	53	XTL2–I/O	81	A15–I/O
26	I/O	54	$\overline{\text{RESET}}$	82	A4–I/O
27	I/O	55	DONE–$\overline{\text{PROG}}$	83	A14–I/O
28	I/O	56	D7–I/O	84	A5–I/O
29	I/O	57	XTL1–BCLKIN–I/O	1	GND
30	I/O	58	D6–I/O	2	VCC*
31	M1–$\overline{\text{RDATA}}$	59	I/O	3	A13–I/O*
32	M0–RTRIG	60	D5–I/O	4	A6–I/O*
33	M2–I/O	61	$\overline{\text{CS0}}$–I/O	5	A12–I/O*
34	HDC–I/O	62	D4–I/O	6	A7–I/O*
35	I/O	63	I/O	7	I/O
36	$\overline{\text{LDC}}$–I/O	64	VCC	8	A11–I/O
37	I/O	65	GND*	9	A8–I/O
38	I/O	66	D3–I/O*	10	A10–I/O
39	I/O	67	$\overline{\text{CS1}}$–I/O*	11	A9–I/O

* Different pin definition than ATT3020/ATT3030/ATT3042 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.

Pin Assignments (continued)

Table 10. ATT3020, ATT3030, and ATT3042 100-MQFP Pinout

100 MQFP	Function	100 MQFP	Function	100 MQFP	Function
16	GND	50	I/O*	84	I/O*
17	A13–I/O	51	I/O*	85	I/O*
18	A6–I/O	52	M1–$\overline{\text{RDATA}}$	86	I/O
19	A12–I/O	53	GND*	87	D5–I/O
20	A7–I/O	54	M0–RTRIG	88	$\overline{\text{CS0}}$–I/O
21	I/O*	55	Vcc*	89	D4–I/O
22	I/O*	56	M2–I/O	90	I/O
23	A11–I/O	57	HDC–I/O	91	Vcc
24	A8–I/O	58	I/O	92	D3–I/O
25	A10–I/O	59	$\overline{\text{LDC}}$–I/O	93	$\overline{\text{CS1}}$–I/O
26	A9–I/O	60	I/O*	94	D2–I/O
27*	Vcc	61	I/O*	95	I/O
28*	GND	62	I/O	96	I/O*
29	$\overline{\text{PWRDWN}}$	63	I/O	97	I/O*
30	TCLKIN–I/O	64	I/O	98	D1–I/O
31	I/O**	65	$\overline{\text{INIT}}$–I/O	99	$\overline{\text{RCLK}}$–$\overline{\text{RDY/BUSY}}$–I/O
32	I/O*	66	GND	100	D0–DIN–I/O
33	I/O*	67	I/O	1	DOUT–I/O
34	I/O	68	I/O	2	CCLK
35	I/O	69	I/O	3	Vcc*
36	I/O	70	I/O	4	GND*
37	I/O	71	I/O	5	A0–$\overline{\text{WS}}$–I/O
38	I/O	72	I/O	6	A1–CS2–I/O
39	I/O	73	I/O	7	I/O**
40	I/O	74	I/O*	8	A2–I/O
41	Vcc	75	I/O*	9	A3–I/O
42	I/O	76	XTL2–I/O	10	I/O*
43	I/O	77*	GND	11	I/O*
44	I/O	78	$\overline{\text{RESET}}$	12	A15–I/O
45	I/O	79	Vcc*	13	A4–I/O
46	I/O	80	DONE–$\overline{\text{PROG}}$	14	A14–I/O
47	I/O	81	D7–I/O	15	A5–I/O
48	I/O	82	XTL1–BCLKIN–I/O	—	—
49	I/O	83	D6–I/O	—	—

* Only 100 of the 118 pads on the ATT3042 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.

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 11. ATT3030, ATT3042, and ATT3064 100-TQFP Pinout

100 TQFP	Function	100 TQFP	Function	100 TQFP	Function
13	GND	47	I/O	81	I/O
14	A13–I/O	48	I/O	82	I/O
15	A6–I/O	49	M1–$\overline{\text{RDATA}}$	83	I/O
16	A12–I/O	50	GND	84	D5–I/O
17	A7–I/O	51	M0–RTRIG	85	$\overline{\text{CS0}}$–I/O
18	I/O	52	Vcc	86	D4–I/O
19	I/O	53	M2–I/O	87	I/O
20	A11–I/O	54	HDC–I/O	88	Vcc
21	A8–I/O	55	I/O	89	D3–I/O
22	A10–I/O	56	$\overline{\text{LDC}}$–I/O	90	$\overline{\text{CS1}}$–I/O
23	A9–I/O	57	I/O	91	D2–I/O
24	Vcc	58	I/O	92	I/O
25	GND	59	I/O	93	I/O
26	$\overline{\text{PWRDWN}}$	60	I/O	94	I/O
27	TCLKIN–I/O	61	I/O	95	D1–I/O
28	I/O*	62	$\overline{\text{INIT}}$–I/O	96	$\overline{\text{RCLK}}$–RDY/$\overline{\text{BUSY}}$–I/O
29	I/O	63	GND	97	D0–DIN–I/O
30	I/O	64	I/O	98	DOUT–I/O
31	I/O	65	I/O	99	CCLK
32	I/O	66	I/O	100	Vcc
33	I/O	67	I/O	1	GND
34	I/O	68	I/O	2	A0–$\overline{\text{WS}}$–I/O
35	I/O	69	I/O	3	A1–CS2–I/O
36	I/O	70	I/O	4	I/O*
37	I/O	71	I/O	5	A2–I/O
38	Vcc	72	I/O	6	A3–I/O
39	I/O	73	XTL2–I/O	7	I/O
40	I/O	74	GND	8	I/O
41	I/O	75	$\overline{\text{RESET}}$	9	A15–I/O
42	I/O	76	Vcc	10	A4–I/O
43	I/O	77	DONE–$\overline{\text{PROG}}$	11	A14–I/O
44	I/O	78	D7–I/O	12	A5–I/O
45	I/O	79	XTL1–BCLKIN–I/O	—	—
46	I/O	80	D6–I/O	—	—

* Indicates unconnected package pins for the ATT3030.

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 12. ATT3042 and ATT3064 132-PPGA Pinout

132 PPGA	Function	132 PPGA	Function	132 PPGA	Function
C4	GND	F12	I/O	N6	I/O*
A1	$\overline{\text{PWRDWN}}$	E14	I/O	P5	I/O*
C3	TCLKIN–I/O	F13	I/O	M6	D2–I/O
B2	I/O	F14	I/O	N5	I/O
B3	I/O	G13	I/O	P4	I/O
A2	I/O*	G14	$\overline{\text{INIT}}$–I/O	P3	I/O
B4	I/O	G12	VCC	M5	D1–I/O
C5	I/O	H12	GND	N4	$\overline{\text{RCLK}}$–RDY/$\overline{\text{BUSY}}$–I/O
A3	I/O*	H14	I/O	P2	I/O
A4	I/O	H13	I/O	N3	I/O
B5	I/O	J14	I/O	N2	D0–DIN–I/O
C6	I/O	J13	I/O	M3	DOUT–I/O
A5	I/O	K14	I/O	P1	CCLK
B6	I/O	J12	I/O	M4	VCC
A6	I/O	K13	I/O	L3	GND
B7	I/O	L14	I/O*	M2	A0–$\overline{\text{WS}}$–I/O
C7	GND	L13	I/O	N1	A1–CS2–I/O
C8	VCC	K12	I/O	M1	I/O
A7	I/O	M14	I/O	K3	I/O
B8	I/O	N14	I/O	L2	A2–I/O
A8	I/O	M13	XTL2–I/O	L1	A3–I/O
A9	I/O	L12	GND	K2	I/O
B9	I/O	P14	$\overline{\text{RESET}}$	J3	I/O
C9	I/O	M11	VCC	K1	A15–I/O
A10	I/O	N13	DONE–$\overline{\text{PROG}}$	J2	A4–I/O
B10	I/O	M12	D7–I/O	J1	I/O*
A11	I/O*	P13	XTL1–BCLKIN–I/O	H1	A14–I/O
C10	I/O	N12	I/O	H2	A5–I/O
B11	I/O	P12	I/O	H3	GND
A12	I/O*	N11	D6–I/O	G3	VCC
B12	I/O	M10	I/O	G2	A13–I/O
A13	I/O*	P11	I/O*	G1	A6–I/O
C12	I/O	N10	I/O	F1	I/O*
B13	M1–$\overline{\text{RDATA}}$	P10	I/O	F2	A12–I/O
C11	GND	M9	D5–I/O	E1	A7–I/O
A14	M0–RTRIG	N9	$\overline{\text{CS0}}$–I/O	F3	I/O
D12	VCC	P9	I/O*	E2	I/O
C13	M2–I/O	P8	I/O*	D1	A11–I/O
B14	HDC–I/O	N8	D4–I/O	D2	A8–I/O
C14	I/O	P7	I/O	E3	I/O
E12	I/O	M8	VCC	C1	I/O
D13	I/O	M7	GND	B1	A10–I/O
D14	$\overline{\text{LDC}}$–I/O	N7	D3–I/O	C2	A9–I/O
E13	I/O*	P6	$\overline{\text{CS1}}$–I/O	D3	VCC

* Indicates unconnected package pins for the ATT3030.

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 13. ATT3042 and ATT3064 144-TQFP Pinout

144 TQFP	Function	144 TQFP	Function	144 TQFP	Function	144 TQFP	Function
1	$\overline{\text{PWRDWN}}$	37	GND	73	DONE—$\overline{\text{PROG}}$	109	Vcc
2	TCLKIN—I/O	38	M0–RTRIG	74	D7–I/O	110	GND
3	I/O*	39	Vcc	75	XTL1–BCLKIN–I/O	111	A0–$\overline{\text{WS}}$–I/O
4	I/O	40	M2–I/O	76	I/O	112	A1–CS2–I/O
5	I/O	41	HDC–I/O	77	I/O	113	I/O
6	I/O*	42	I/O	78	D6–I/O	114	I/O
7	I/O	43	I/O	79	I/O	115	A2–I/O
8	I/O	44	I/O	80	I/O*	116	A3–I/O
9	I/O*	45	$\overline{\text{LDC}}$—I/O	81	I/O	117	I/O
10	I/O	46	I/O*	82	I/O	118	I/O
11	I/O	47	I/O	83	I/O*	119	A15–I/O
12	I/O	48	I/O	84	D5–I/O	120	A4–I/O
13	I/O	49	I/O	85	$\overline{\text{CS0}}$–I/O	121	I/O*
14	I/O	50	I/O*	86	I/O*	122	I/O*
15	I/O*	51	I/O	87	I/O*	123	A14–I/O
16	I/O	52	I/O	88	D4–I/O	124	A5–I/O
17	I/O	53	$\overline{\text{INIT}}$–I/O	89	I/O	125	—
18	GND	54	Vcc	90	Vcc	126	GND
19	Vcc	55	GND	91	GND	127	Vcc
20	I/O	56	I/O	92	D3—I/O	128	A13–I/O
21	I/O	57	I/O	93	$\overline{\text{CS1}}$—I/O	129	A6–I/O
22	I/O	58	I/O	94	I/O*	130	I/O*
23	I/O	59	I/O	95	I/O*	131	—
24	I/O	60	I/O	96	D2—I/O	132	I/O*
25	I/O	61	I/O	97	I/O	133	A12–I/O
26	I/O	62	I/O	98	I/O	134	A7–I/O
27	I/O	63	I/O*	99	I/O*	135	I/O
28	I/O*	64	I/O*	100	I/O	136	I/O
29	I/O	65	I/O	101	I/O*	137	A11–I/O
30	I/O	66	I/O	102	D1—I/O	138	A8–I/O
31	I/O*	67	I/O	103	$\overline{\text{RCLK}}$—$\overline{\text{BUSY}}$/RDY—I/O	139	I/O
32	I/O*	68	I/O	104	I/O	140	I/O
33	I/O	69	XTL2—I/O	105	I/O	141	A10–I/O
34	I/O*	70	GND	106	D0—DIN—I/O	142	A9–I/O
35	I/O	71	$\overline{\text{RESET}}$	107	DOUT–I/O	143	Vcc
36	M1–$\overline{\text{RDATA}}$	72	Vcc	108	CCLK	144	GND

* 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.

Pin Assignments (continued)

Table 14. ATT3064 and ATT3090 160-MQFP Pinout

160 MQFP	Function	160 MQFP	Function	160 MQFP	Function	160 MQFP	Function
1	I/O*	41	GND	81	D7–I/O	121	CCLK
2	I/O*	42	M0–RTRIG	82	XTL1–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	$\overline{RCLK}$–RDY/$\overline{BUSY}$–I/O	155	A10–I/O
36	I/O	76	XTL2–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	DONE–$\overline{PROG}$	120	DOUT–I/O	160	TCLKIN–I/O

* Indicates unconnected package pins for the 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.

Pin Assignments (continued)

Table 15. ATT3000 Family 175-PPGA Pinout

175 PPGA	Function	175 PPGA	Function	175 PPGA	Function	175 PPGA	Function
B2	$\overline{\text{PWRDWN}}$	D13	I/O	R14	DONE–$\overline{\text{PROG}}$	R3	D0–DIN–I/O
D4	TCLKIN–I/O	B14	M1–$\overline{\text{RDATA}}$	N13	D7–I/O	N4	DOUT–I/O
B3	I/O	C14	GND	T14	XTL1–BCLKIN–I/O	R2	CCLK
C4	I/O	B15	M0–RTRIG	P13	I/O	P3	VCC
B4	I/O	D14	VCC	R13	I/O	N3	GND
A4	I/O	C15	M2–I/O	T13	I/O	P2	A0–$\overline{\text{WS}}$–I/O
D5	I/O	E14	HDC–I/O	N12	I/O	M3	A1–CS2–I/O
C5	I/O	B16	I/O	P12	D6–I/O	R1	I/O
B5	I/O	D15	I/O	R12	I/O	N2	I/O
A5	I/O	C16	I/O	T12	I/O	P1	A2–I/O
C6	I/O	D16	$\overline{\text{LDC}}$–I/O	P11	I/O	N1	A3–I/O
D6	I/O	F14	I/O	N11	I/O	L3	I/O
B6	I/O	E15	I/O	R11	I/O	M2	I/O
A6	I/O	E16	I/O	T11	D5–I/O	M1	A15–I/O
B7	I/O	F15	I/O	R10	$\overline{\text{CS0}}$–I/O	L2	A4–I/O
C7	I/O	F16	I/O	P10	I/O	L1	I/O
D7	I/O	G14	I/O	N10	I/O	K3	I/O
A7	I/O	G15	I/O	T10	I/O	K2	A14–I/O
A8	I/O	G16	I/O	T9	I/O	K1	A5–I/O
B8	I/O	H16	I/O	R9	D4–I/O	J1	I/O
C8	I/O	H15	$\overline{\text{INIT}}$–I/O	P9	I/O	J2	I/O
D8	GND	H14	VCC	N9	VCC	J3	GND
D9	VCC	J14	GND	N8	GND	H3	VCC
C9	I/O	J15	I/O	P8	D3–I/O	H2	A13–I/O
B9	I/O	J16	I/O	R8	$\overline{\text{CS1}}$–I/O	H1	A6–I/O
A9	I/O	K16	I/O	T8	I/O	G1	I/O
A10	I/O	K15	I/O	T7	I/O	G2	I/O
D10	I/O	K14	I/O	N7	I/O	G3	I/O
C10	I/O	L16	I/O	P7	I/O	F1	I/O
B10	I/O	L15	I/O	R7	D2–I/O	F2	A12–I/O
A11	I/O	M16	I/O	T6	I/O	E1	A7–I/O
B11	I/O	M15	I/O	R6	I/O	E2	I/O
D11	I/O	L14	I/O	N6	I/O	F3	I/O
C11	I/O	N16	I/O	P6	I/O	D1	A11–I/O
A12	I/O	P16	I/O	T5	I/O	C1	A8–I/O
B12	I/O	N15	I/O	R5	D1–I/O	D2	I/O
C12	I/O	R16	I/O	P5	RDY/$\overline{\text{BUSY}}$–$\overline{\text{RCLK}}$–I/O	B1	I/O
D12	I/O	M14	I/O	N5	I/O	E3	A10–I/O
A13	I/O	P15	XTL2–I/O	T4	I/O	C2	A9–I/O
B13	I/O	N14	GND	R4	I/O	D3	VCC
C13	I/O	R15	$\overline{\text{RESET}}$	P4	I/O	C3	GND
A14	I/O	P14	VCC	—	—	—	—

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

Pin Assignments (continued)

Table 16. ATT3000 Family 208-SQFP Pinout

208 SQFP	Function	208 SQFP	Function	208 SQFP	Function	208 SQFP	Function
1	—	53	—	105	—	157	—
2	GND	54	—	106	VCC	158	—
3	PWRDWN	55	VCC	107	DONE–PROG	159	—
4	TCLKIN–I/O	56	M2–I/O	108	—	160	GND
5	I/O	57	HDC–I/O	109	D7–I/O	161	A0–WS–I/O
6	I/O	58	I/O	110	XTL1–BCLKIN–I/O	162	A1–CS2–I/O
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	A2–I/O
10	I/O	62	I/O	114	I/O	166	A3–I/O
11	I/O	63	I/O	115	D6–I/O	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	I/O	67	—	119	—	171	—
16	I/O	68	I/O	120	I/O	172	A15–I/O
17	I/O	69	I/O	121	I/O	173	A4–I/O
18	I/O	70	I/O	122	D5–I/O	174	I/O
19	I/O	71	I/O	123	CS0–I/O	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	A14–I/O
23	I/O	75	I/O	127	I/O	179	A5–I/O
24	I/O	76	I/O	128	D4–I/O	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	D3–I/O	184	A13–I/O
29	I/O	81	I/O	133	CS1–I/O	185	A6–I/O
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	D2–I/O	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	A12–I/O
37	—	89	I/O	141	I/O	193	A7–I/O
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	D1–I/O	197	I/O
42	I/O	94	I/O	146	RDY/BUSY–RCLK–I/O	198	I/O
43	I/O	95	I/O	147	I/O	199	A11–I/O
44	I/O	96	I/O	148	I/O	200	A8–I/O
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	D0–DIN–I/O	203	A10–I/O
48	M1–RDATA	100	XTL2–I/O	152	DOUT–I/O	204	A9–I/O
49	GND	101	GND	153	CCLK	205	VCC
50	M0–RTRIG	102	RESET	154	VCC	206	—
51	—	103	—	155	—	207	—
52	—	104	—	156	—	208	—

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.

Package Thermal Characteristics

When silicon die junction temperature is below the recommended junction temperature of 125 °C, the temperature-activated failure mechanisms are minimized. There are four major factors that affect the thermal resistance value: silicon device size/paddle size, board mounting configuration (board density, multilayer nature of board), package type and size, and system airflow over the package. The values in the table below reflect the capability of the various package types to dissipate heat at given airflow rates. The numbers represent the delta °C/W between the ambient temperature and the device junction temperature.

To test package thermal characteristics, a single package containing a 0.269 in. sq. test IC of each configuration is mounted at the center of a printed-circuit board (PCB) measuring 8 in. x 13 in. x 0.062 in. The assembled PCB is mounted vertically in the center of the rectangular test section of a wind tunnel. The walls of the wind tunnel simulate adjacent boards in the electronic rack and can be adjusted to study the effects of PCB spacing. Forced air at room temperature is supplied by a pair of push-pull blowers which can be regulated to supply the desired air velocities. The air velocity is measured with a hot-wire anemometer at the center of the channel, 3 in. upstream from the package.

A typical test consists of regulating the wind tunnel blowers to obtain the desired air velocity and applying power to the test IC. The power to the IC is adjusted until the maximum junction temperature (as measured by its diodes) reaches 115 °C to 120 °C. The thermal resistance Θ_{JA} (°C/W) is computed by using the power supplied to the IC, junction temperature, ambient temperature, and air velocity:

$$\Theta_{JA} = \frac{T_J - T_A}{Q_C}$$

where:

T_J = peak temperature on the active surface of the IC

T_A = ambient air temperature

Q_C = IC power

The tests are repeated at several velocities from 0 fpm (feet per minute) to 1000 fpm.

The definition of the junction to case thermal resistance Θ_{JC} is:

$$\Theta_{JC} = \frac{T_J - T_C}{Q_C}$$

where:

T_C = temperature measured to the thermocouple at the top dead center of the package

The actual Θ_{JC} measurement performed at Lucent, Θ_{J-TDC}, uses a different package mounting arrangement than the one defined for Θ_{JC} in MIL-STD-883D and SEMI standards. Please contact Lucent for a diagram.

The maximum power dissipation for a package is calculated from the maximum junction temperature, maximum operating temperature, and the junction to ambient characteristic Θ_{JA}. The maximum power dissipation for commercial grade ICs is calculated as follows: max power (watts) = (125 °C – 70 °C) x ($1/\Theta_{JA}$), where 125 °C is the maximum junction temperature. Table 17 lists the ATT3000 plastic package thermal characteristics.

Package Thermal Characteristics (continued)

Table 17. ATT3000 Plastic Package Thermal Characteristics

Package	Θ_{JA} (°C/W)			Θ_{JC} (°C/W)	Max Power (70 °C—0 fpm)
	0 fpm	200 fpm	400 fpm		
44-Pin PLCC	49	41	40	—	1.12 W
68-Pin PLCC	43	38	35	11	1.28 W
84-Pin PLCC	40	35	32	9	1.38 W
100-Pin MQFP	81	67	64	11	0.68 W
100-Pin TQFP	61	49	46	6	0.90 W
132-Pin PPGA	22	18	16	—	2.50 W
144-Pin TQFP	52	39	36	4	1.06 W
160-Pin MQFP	40	36	32	8	1.38 W
175-Pin PPGA	23	20	17	—	2.39 W
208-Pin SQFP	37	33	29	8	1.49 W

Package Coplanarity

The coplanarity of Lucent Technologies postmolded packages is 4 mils. The coplanarity of selected packages is scheduled to be reduced to 3.1 mils. All Lucent ATT3000 Series FPGA ceramic packages are through-hole mount.

Package Parasitics

The electrical performance of an IC package, such as signal quality and noise sensitivity, is directly affected by the package parasitics. Table 18 lists eight parasitics associated with the ATT3000 packages. These parasitics represent the contributions of all components of a package, which include the bond wires, all internal package routing, and the external leads.

Four inductances in nH are listed: L_W and L_L, the self-inductance of the lead; and L_{MW} and L_{ML}, the mutual inductance to the nearest neighbor lead. These parameters are important in determining ground bounce noise and inductive crosstalk noise. Three capacitances in pF are listed: C_M, the mutual capacitance of the lead to the nearest neighbor lead; and C_1 and C_2, the total capacitance of the lead to all other leads (all other leads are assumed to be grounded). These parameters are important in determining capacitive crosstalk and the capacitive loading effect of the lead.

The parasitic values in Table 18 are for the circuit model of bond wire and package lead parasitics. If the mutual capacitance value is not used in the designer's model, then the value listed as mutual capacitance should be added to each of the C1 and C2 capacitors. The PGAs contain power and ground planes that will make the inductance value for power and ground leads the minimum value listed. The PGAs also have a significant range of parasitic values. This is due to the large variation in internal trace lengths and is also due to two signal metal layers that are separated from the ground plane by different distances. The upper signal layer is more inductive but less capacitive than the closer, lower signal layer.

Package Parasitics (continued)

Table 18. Package Parasitics

Package Type	L_W	M_W	R_W	C_1	C_2	C_M	L_L	M_L
44-Pin PLCC	3	1	140	0.5	0.5	0.3	5—6	2—2.5
68-Pin PLCC	3	1	140	0.5	0.5	0.4	6—9	3—4
84-Pin PLCC	3	1	140	1	1	0.5	7—11	3—6
100-Pin MQFP	3	1	160	1	1	0.5	7—9	4—5
100-Pin TQFP	3	1	150	0.5	0.5	0.4	4—6	2—3
132-Pin PPGA	3	1	150	1	1	0.25	4—10	0.5—1
144-Pin TQFP	3	1	140	1	1	0.6	4—6	2—2.5
160-Pin MQFP	4	1.5	180	1.5	1.5	1	10—13	6—8
175-Pin PPGA	3	1	150	1	1	0.3	5—11	1—1.5
208-Pin SQFP	4	2	200	1	1	1	7—10	4—6

* Leads designated as ground (power) can be connected to the ground plane, reducing the trace inductance to the minimum value listed.

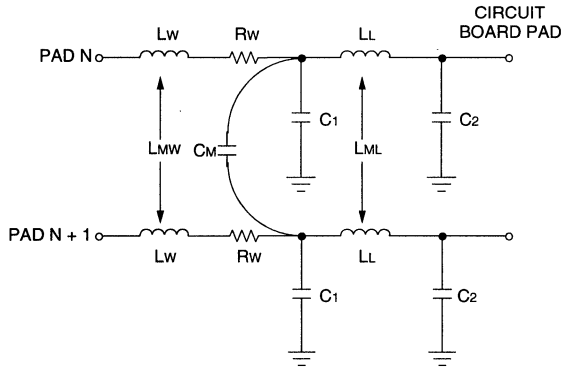


Figure 33. Package Parasitics

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 Relative to GND	V_{IN}	–0.5	0.5	V
Voltage Applied to 3-state Output	V_{TS}	–0.5	0.5	V
Storage Temperature (ambient)	T_{stg}	–65	150	°C
Maximum Soldering Temperature (10 seconds at 1/16 in.)	T_{SOL}	—	260	°C
Junction Temperature	T_J	—	125	°C

Electrical Characteristics

Table 19. dc Electrical Characteristics Over Operating Conditions

Commercial: V_{CC} = 5.0 V ± 5%; 0 °C ≤ T_A ≤ 70 °C; Industrial: V_{CC} = 5.0 ± 10%, –40 °C ≤ T_A ≤ +85 °C.

Parameter/Conditions	Symbol	-50, -70, -100, -125, and -150 MHz		-3, -4, and -5		Unit
		Min	Max	Min	Max	
High-level Input Voltage						
CMOS Level	V_{IHC}	70%	100%	70%	100%	V
TTL Level	V_{IHT}	2.0	V_{CC}	2.0	V_{CC}	V
Low-level Input Voltage						
CMOS Level	V_{ILC}	0	20%	0	20%	V
TTL Level	V_{ILT}	0	0.8	0	0.8	V
Output Voltage						
High						
(I_{OH} = –4 mA)	V_{OH}	3.86	—	—	—	V
(I_{OH} = –8 mA)	V_{OH}	—	—	3.86	—	V
Low						
(I_{OL} = 4 mA)	V_{OL}	—	0.40	—	—	V
(I_{OL} = 8 mA)	V_{OL}	—	—	—	0.40	V
Input Signal Transition Time	T_{IN}	—	250	—	250	ns
Powerdown Supply Current	I_{CCPD}					
ATT3020		—	50	—	50	µA
ATT3030		—	80	—	80	µA
ATT3042		—	120	—	120	µA
ATT3064		—	170	—	170	µA
ATT3090		—	250	—	250	µA
Quiescent FPGA Supply Current (in addition to ICCPD)	I_{CCO}					
CMOS Inputs				—	10	mA
ATT3020		—	500			µA
ATT3030		—	500			µA
ATT3042		—	500			µA
ATT3064		—	500			µA
ATT3090		—	500			µA
TTL Inputs		—	10	—	20	mA
Leakage Current	I_{IL}	–10	10	–10	10	µA
Input Capacitance*	C_{IN}					
All Packages Except 175-PGA:						
All Pins Except XTL1/XTL2		—	10	—	10	pF
XTL1 and XTL2		—	15	—	15	pF
175-PGA Package:						
All Pins Except XTL1/XTL2		—	15	—	15	pF
XTL1 and XTL2		—	20	—	20	pF
Pad Pull-up* (when selected) (at V_{IN} = 0 V)	I_{RIN}	0.02	0.17	0.02	0.17	mA
Horizontal Long Line Pull-up (when selected) at Logic LOW	I_{RLL}	0.2	2.5	0.2	2.8	mA

* Sample tested.

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

Electrical Characteristics (continued)

Table 20. CLB Switching Characteristics (-50, -70, -100, -125, and -150)

Commercial: V_{CC} = 5.0 V ± 5%; 0 °C ≤ T_A ≤ 70 °C; Industrial: V_{CC} = 5.0 ± 10%, –40 °C ≤ T_A ≤ +85 °C.

Description	Symbol		-50		-70		-100		-125		-150		Units
			Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
Combinatorial Delay	1	TILO	—	14.0	—	9.0	—	7.0	—	5.5	—	4.6	ns
Sequential Delay													
Clock K to Outputs x or y	8	TCKO	—	12.0	—	6.0	—	5.0	—	4.5	—	4.0	ns
Clock K to Outputs x or y when Q Returned Through Function Generators F or G to Drives x or y	—	TQLO	—	23.0	—	13.0	—	10.0	—	8.0	—	6.7	ns
Setup Time													
Logic Variables	2	TICK	12.0	—	8.0	—	7.0	—	5.5	—	4.6	—	ns
Data In	4	TDICK	8.0	—	5.0	—	4.0	—	3.0	—	2.0	—	ns
Enable Clock	6	TECCK	10.0	—	7.0	—	5.0	—	4.5	—	4.0	—	ns
Reset Direct Active	—	TRDCK	1.0	—	1.0	—	1.0	—	1.0	—	1.0	—	ns
Hold Time													
Logic Variables	3	TCKI	1.0	—	0	—	0	—	0	—	0	—	ns
Data In	5	TCKDI	6.0	—	4.0	—	2.0	—	1.5	—	1.2	—	ns
Enable Clock	7	TCKEC	0	—	0	—	0	—	0	—	0	—	ns
Clock													
High Time*	11	TCH	9.0	—	5.0	—	4.0	—	3.0	—	2.5	—	ns
Low Time*	12	TCL	9.0	—	5.0	—	4.0	—	3.0	—	2.5	—	ns
Flip-flop Toggle Rate*	—	FCLK	50	—	70	—	100	—	125	—	150	—	MHz
Reset Direct (rd)													
rd Width	13	TRPW	12.0	—	8.0	—	7.0	—	6.0	—	5.0	—	ns
Delay from rd to Outputs x, y	9	TRIO	—	12.0	—	8.0	—	7.0	—	6.0	—	5.0	ns
Master Reset (MR)													
MR Width	—	TMRW	30	—	25	—	21	—	20	—	19	—	ns
Delay from MR to Outputs x, y	—	TMRQ	—	27	—	23	—	19	—	17	—	17	ns

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

Note: The CLKB 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.

Electrical Characteristics (continued)

Table 21. CLB Switching Characteristics (-3, -4, and -5)

Commercial: V_{CC} = 5.0 V ± 5%; 0 °C ≤ T_A ≤ 70 °C; Industrial: V_{CC} = 5.0 ± 10%, –40 °C ≤ T_A ≤ +85 °C.

Description	Symbol		-5		-4		-3		Units
			Min	Max	Min	Max	Min	Max	
Combinatorial Delay	1	TILO	—	4.1	—	3.3	—	2.7	ns
Sequential Delay									
Clock K to Outputs x or y	8	TCKO	—	3.1	—	2.5	—	2.1	ns
Clock K to Outputs x or y when Q Returned Through Function Generators F or G to Drives x or y	—	TQLO	—	6.3	—	5.2	—	4.3	ns
Setup Time									
Logic Variables	2	TICK	3.1	—	2.5	—	2.1	—	ns
Data In	4	TDICK	2.0	—	1.6	—	1.4	—	ns
Enable Clock	6	TECCK	3.8	—	3.2	—	2.7	—	ns
Reset Direct Active	—	TRDCK	1.0	—	1.0	—	1.0	—	ns
Hold Time									
Logic Variables	3	TCKI	0	—	0	—	0	—	ns
Data In	5	TCKDI	1.2	—	1.0	—	0.9	—	ns
Enable Clock	7	TCKEC	1.0	—	0.8	—	0.7	—	ns
Clock									
High Time*	11	TCH	2.4	—	2.0	—	1.6	—	ns
Low Time*	12	TCL	2.4	—	2.0	—	1.6	—	ns
Flip-flop Toggle Rate*	—	FCLK	190	—	230	—	270	—	MHz
Reset Direct (rd)									
rd Width	13	TRPW	3.8	—	3.2	—	2.7	—	ns
Delay from rd to Outputs x, y	9	TRIO	—	4.4	—	3.7	—	3.1	ns
Master Reset (MR)									
MR Width	—	TMRW	18.0	—	15.0	—	13.0	—	ns
Delay from MR to Outputs x, y	—	TMRQ	—	17.0	—	14.0	—	12.0	ns

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

Note: The CLKB 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.

Electrical Characteristics (continued)

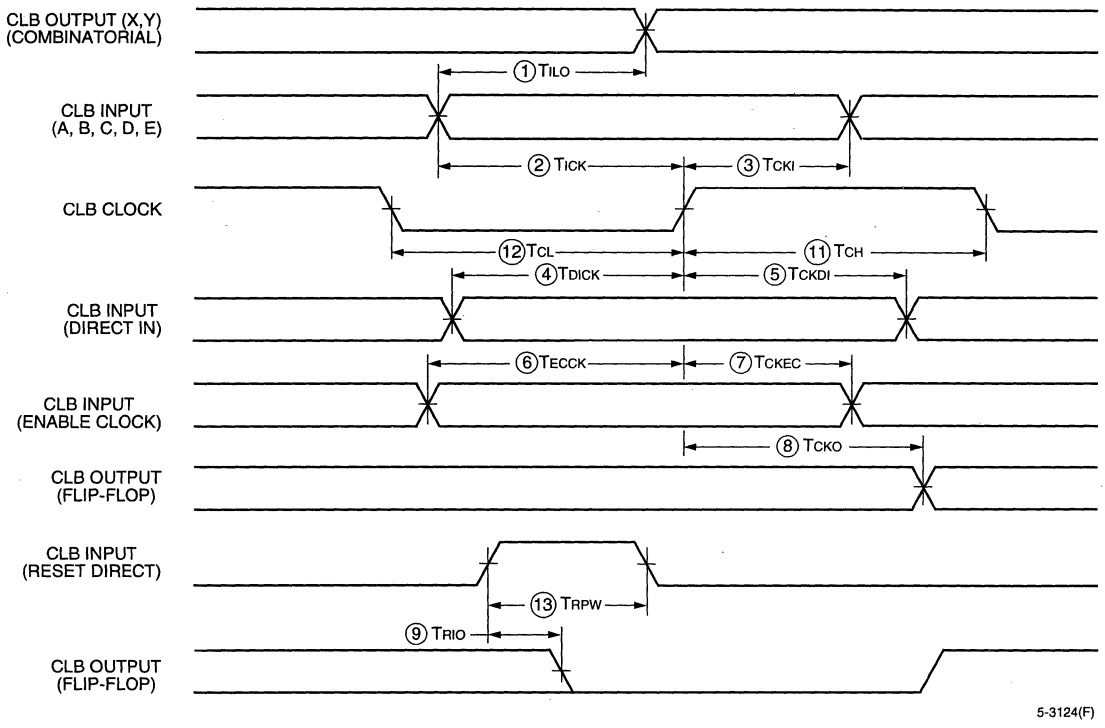


Figure 34. CLB Switching Characteristics

Electrical Characteristics (continued)

Table 22. IOB Switching Characteristics (-50, -70, -100, -125, and -150)

Commercial: V_{CC} = 5.0 V ± 5%; 0 °C ≤ T_A ≤ 70 °C; Industrial: V_{CC} = 5.0 ± 10%, –40 °C ≤ T_A ≤ +85 °C.

Description	Symbol		-50		-70		-100		-125		-150		Units
			Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
Input Delays													
Pad to Direct In	3	TPID	—	9.0	—	6.0	—	4.0	—	3.0	—	2.8	ns
Pad to Registered In	—	TPTG	—	34.0	—	21.0	—	17.0	—	16.0	—	15.0	ns
Clock to Registered In	4	TIKRI	—	11.0	—	5.5	—	4.0	—	3.0	—	2.8	ns
Setup Time (Input):													
Clock Setup Time	1	TPICK	30.0	—	20.0	—	17.0	—	16.0	—	14.5	—	ns
Output Delays													
Clock to Pad													
Fast	7	TOKPO	—	18.0	—	13.0	—	10.0	—	9.0	—	7.0	ns
Slew-rate Limited	7	TOKPO	—	43.0	—	33.0	—	27.0	—	24.0	—	22.0	ns
Output to Pad													
Fast	10	TOPF	—	15.0	—	9.0	—	6.0	—	5.0	—	4.5	ns
Slew-rate Limited	10	TOPS	—	40.0	—	29.0	—	23.0	—	20.0	—	15.0	ns
3-state to Pad Hi-Z													
Fast	9	TTSHZ	—	10.0	—	8.0	—	8.0	—	7.0	—	7.0	ns
Slew-rate Limited	9	TTSHZ	—	37.0	—	28.0	—	25.0	—	24.0	—	22.0	ns
3-state to Pad Valid													
Fast	8	TTSON	—	20.0	—	14.0	—	12.0	—	11.0	—	11.0	ns
Slew-rate Limited	8	TTSON	—	45.0	—	34.0	—	29.0	—	27.0	—	26.0	ns
Setup and Hold Times (output)													
Clock Setup Time	5	TOCK	15.0	—	10.0	—	9.0	—	8.0	—	7.0	—	ns
Clock Hold Time	6	TOKO	0	—	0	—	0	—	0	—	0	—	ns
Clock													
High Time*	11	TCH	9.0	—	5.0	—	4.0	—	3.0	—	2.5	—	ns
Low Time*	12	TCL	9.0	—	5.0	—	4.0	—	3.0	—	2.5	—	ns
Max. Flip-flop Toggle*	—	FCLK	—	50	—	70	—	100	—	125	—	150	MHz
Master Reset Delays													
$\overline{\text{RESET}}$ to:													
Registered In	13	TRRI	—	35	—	25	—	24	—	23	—	20	ns
Output Pad (fast)	15	TRPO	—	50	—	35	—	33	—	29	—	25	ns
Output Pad (slew-rate limited)	15	TRPO	—	68	—	53	—	45	—	42	—	40	ns

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

Notes:

Timing is measured at pin threshold with 50 pF external capacitive loads (including 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. Exceeding this maximum capacitive load can result in ground bounce of >1.5 V amplitude and <5 ns duration, which may cause problems when the $\overline{\text{LCA}}$ drives clocks and other asynchronous signals.

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 setup time is specified with respect to the internal clock (ik).

To calculate system setup time, subtract clock delay (pad to ik) from the input pad setup time value. Input pad hold time with respect to the internal clock (ik) is negative. This means that pad levels changed immediately before the internal clock edge (ik) will not be recognized.

Electrical Characteristics (continued)

Table 23. IOB Switching Characteristics (-3, -4, and -5)

Commercial: V_{CC} = 5.0 V ± 5%; 0 °C ≤ T_A ≤ 70 °C; Industrial: V_{CC} = 5.0 ± 10%, –40 °C ≤ T_A ≤ +85 °C.

Description	Symbol		-5		-4		-3		Units
			Min	Max	Min	Max	Min	Max	
Input Delays									
Pad to Direct In	3	TPID	—	2.8	—	2.5	—	2.2	ns
Pad to Registered In	—	TPTG	—	16.0	—	15.0	—	13.0	ns
Clock to Registered In	4	TIKRI	—	2.8	—	2.5	—	2.2	ns
Setup Time (Input):									
Clock Setup Time	1	TPICK	15.0	—	14.0	—	12.0	—	ns
Output Delays									
Clock to Pad									
Fast	7	TOKPO	—	5.5	—	5.0	—	4.4	ns
Slew-rate Limited	7	TOKPO	—	14.0	—	12.0	—	10.0	ns
Output to Pad									
Fast	10	TOPF	—	4.1	—	3.7	—	3.3	ns
Slew-rate Limited	10	TOPS	—	13.0	—	11.0	—	9.0	ns
3-state to Pad Hi-Z									
Fast	9	TTSHZ	—	6.9	—	6.2	—	5.5	ns
Slew-rate Limited	9	TTSHZ	—	21.0	—	19.0	—	17.0	ns
3-state to Pad Valid									
Fast	8	TTSON	—	12.0	—	10.0	—	9.0	ns
Slew-rate Limited	8	TTSON	—	20.0	—	17.0	—	15.0	ns
Setup and Hold Times (output)									
Clock Setup Time	5	TOCK	6.2	—	5.6	—	5.0	—	ns
Clock Hold Time	6	TOKO	0	—	0	—	0	—	ns
Clock									
High Time*	11	TIOH	2.4	—	2.0	—	1.6	—	ns
Low Time*	12	TCL	2.4	—	2.0	—	1.6	—	ns
Max. Flip-flop Toggle*	—	FCLK	190	—	230	—	270	—	MHz
Master Reset Delays									
$\overline{\text{RESET}}$ to:									
Registered In	13	TRRI	—	18	—	15	—	13	ns
Output Pad (fast)	15	TRPO	—	24	—	20	—	17	ns
Output Pad (slew-rate limited)	15	TRPO	—	32	—	27	—	23	ns

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

Notes:

Timing is measured at pin threshold with 50 pF external capacitive loads (including 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. Exceeding this maximum capacitive load can result in ground bounce of >1.5 V amplitude and <5 ns duration, which may cause problems when the LCA drives clocks and other asynchronous signals.

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 setup time is specified with respect to the internal clock (ik).

To calculate system setup time, subtract clock delay (pad to ik) from the input pad setup time value. Input pad hold time with respect to the internal clock (ik) is negative. This means that pad levels changed immediately before the internal clock edge (ik) will not be recognized.

Electrical Characteristics (continued)

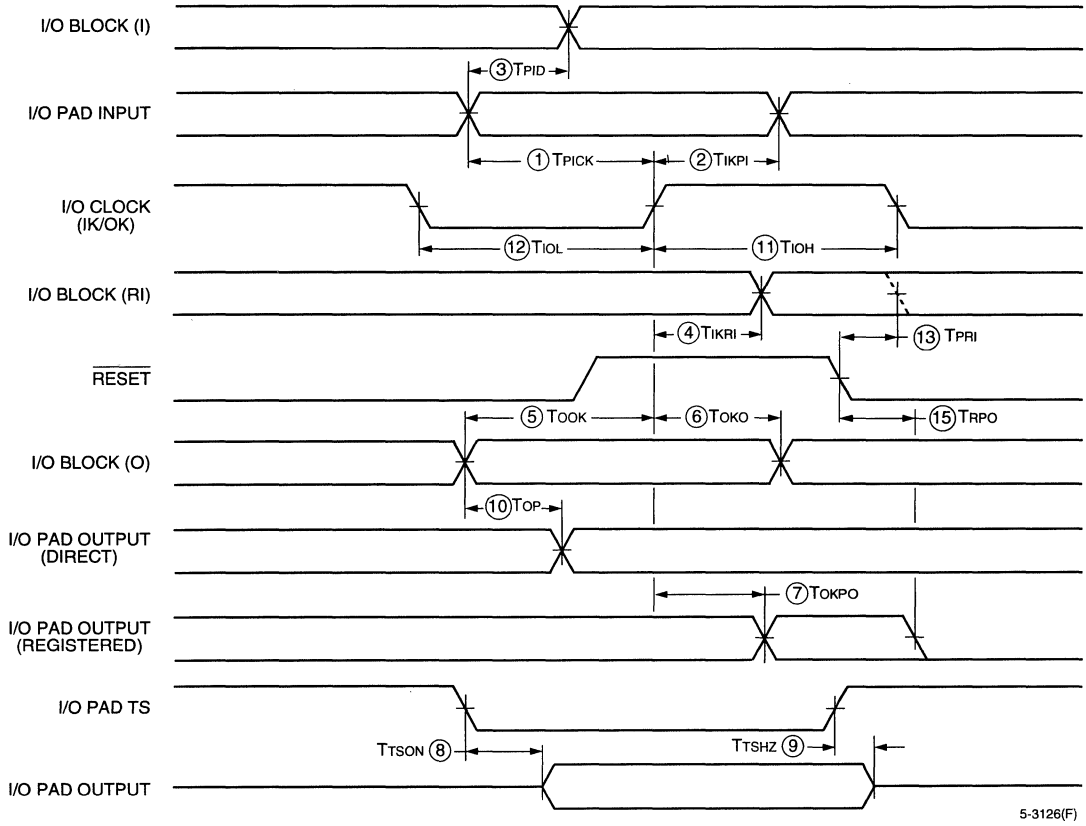


Figure 35. IOB Switching Characteristics

Electrical Characteristics (continued)

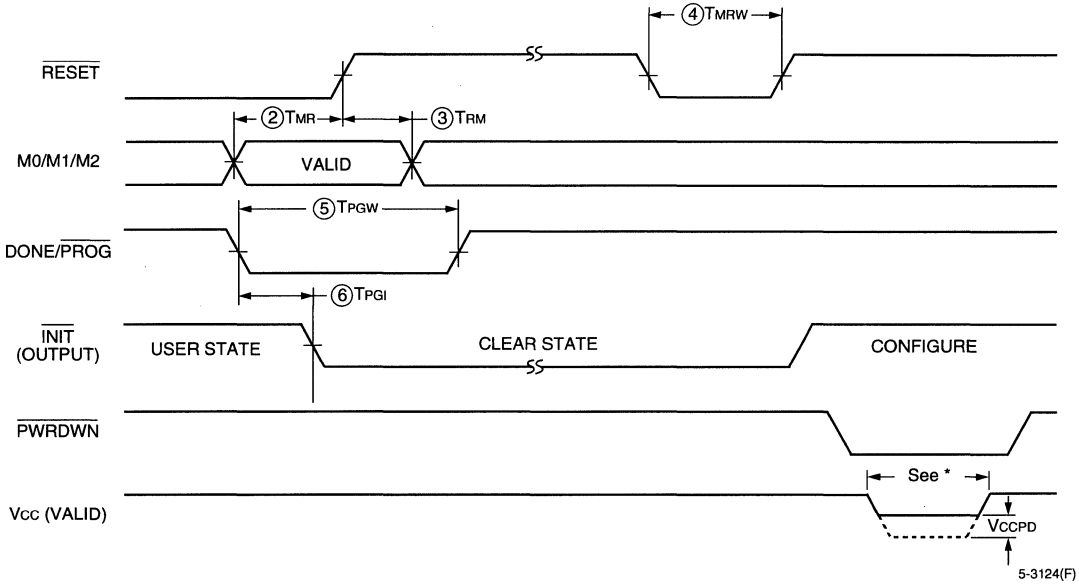
Table 24. Buffer (Internal) Switching Characteristics

Commercial: V_{CC} = 5.0 V ± 5%; 0 °C ≤ T_A ≤ 70 °C; Industrial: V_{CC} = 5.0 ± 10%, –40 °C ≤ T_A ≤ +85 °C.

Description	Symbol	-50	-70	-100	-125	-150	-5	-4	-3	Unit
		Max	Max	Max	Max	Max	Max	Max	Max	
Global and Alternate Clock Distribution*:										
Either Normal IOB Input Pad to Clock Buffer Input	T_{PID}	10.0	8.0	7.5	7.0	6.8	6.8	6.5	5.6	ns
Or Fast (CMOS only) Input Pad to Clock Buffer Input	T_{PIDC}	8.0	6.5	6.0	5.7	5.5	5.4	5.1	4.3	ns
TBUF Driving a Horizontal Long Line (LL)*:										
I to LL While T Is Low (buffer active)	T_{IO}	8.0	5.0	4.7	4.5	4.1	4.1	3.7	3.1	ns
T↓ to LL Active and Valid with Single Pull-up Resistor	T_{ON}	12.0	11.0	10.0	9.0	5.6	5.6	5.0	4.2	ns
T↓ to LL Active and Valid with Pair of Pull-up Resistors	T_{ON}	14.0	12.0	11.0	10.0	7.1	7.1	6.5	5.7	ns
T↑ to LL High with Single Pull-up Resistor	T_{PUS}	42.0	24.0	22.0	17.0	15.6	15.6	13.5	11.4	ns
T↑ to LL High with Pair of Pull-up Resistors	T_{PUF}	22.0	17.0	15.0	12.0	12.0	12.0	10.5	8.8	ns
Bidirectional Buffer Delay	T_{BIDI}	6.0	2.0	1.8	1.7	1.4	1.4	1.2	1.0	ns

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

Electrical Characteristics (continued)



* At powerup, VCC must rise from 2 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 V. A very long VCC rise time of >100 ms or a nonmonotonically rising VCC may require a >1 µs high level on $\overline{\text{RESET}}$, followed by a >6 µs low level on $\overline{\text{RESET}}$ and DONE/$\overline{\text{PROG}}$ after VCC has reached 4 V.

Figure 36. General FPGA Switching Characteristics

Testing of the switching characteristics is modeled after testing specified by MIL-M-38510/605. Devices are 100% functionally tested. Actual worst-case timing is provided by the timing calculator or simulation.

Table 25. General FPGA Switching Characteristics

Signal	Description	Symbol	Min	Max	Unit
$\overline{\text{RESET}}$*	M0, M1, and M2 Setup Time	TMR (2)	1	—	µs
	M0, M1, and M2 Hold Time	TRM (3)	1	—	µs
	$\overline{\text{RESET}}$ Width (LOW) Required for Abort	TMRW (4)	6	—	µs
DONE/$\overline{\text{PROG}}$	Width Low Required for Reconfiguration	TPGW (5)	6	—	µs
	$\overline{\text{INIT}}$ Response after DONE/$\overline{\text{PROG}}$ Is Pulled Low	TPGI (6)	—	7	µs
VCC†	Powerdown VCC (commercial/industrial)	VCCPD	2.3	—	V

* $\overline{\text{RESET}}$ timing relative to valid mode lines (M0, M1, M2) is relevant when $\overline{\text{RESET}}$ is used to delay configuration.

† $\overline{\text{PWRDWN}}$ transitions must occur while VCC > 4 V.

Electrical Characteristics (continued)

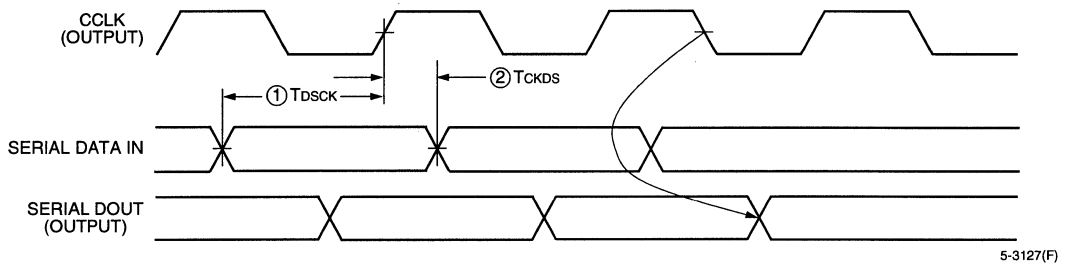


Figure 37. Master Serial Mode Switching Characteristics

Table 26. Master Serial Mode Switching Characteristics

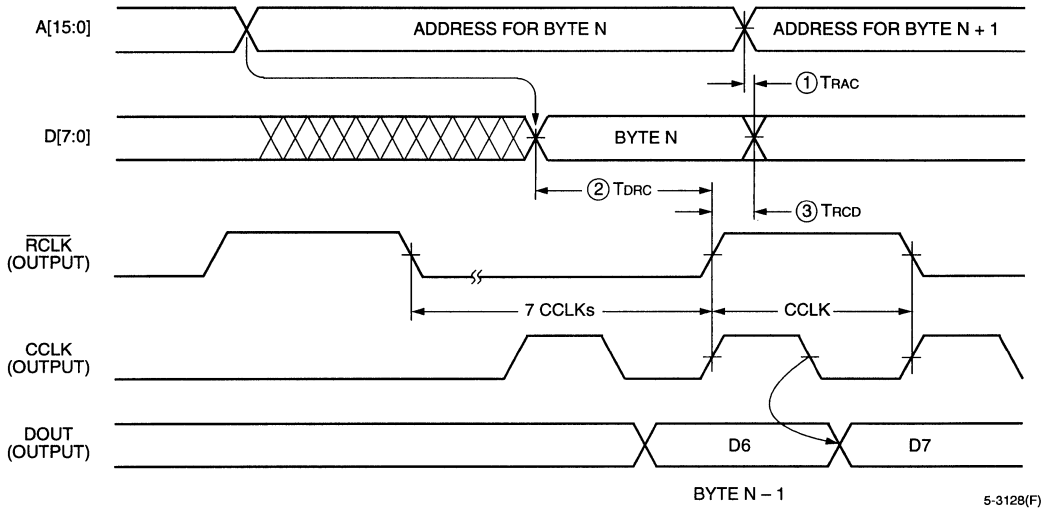
Signal	Description	Symbol		Min	Max	Unit
CCLK	Data-In Setup	1	TDSCK	60	—	ns
	Data-In Hold	2	TCKDS	0	—	ns

Notes:
At powerup, 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 nonmonotonically rising VCC may require a >1 µs high level on $\overline{\text{RESET}}$, followed by >6 µs low level on $\overline{\text{RESET}}$ and D/$\overline{\text{P}}$ after VCC has reached 4.0 V.

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

Master serial mode timing is based on slave mode testing.

Electrical Characteristics (continued)



Note: The EPROM requirements in this timing diagram are extremely relaxed; EPROM access time can be longer than 4000 ns. EPROM data output has no hold time requirements.

Figure 38. Master Parallel Mode Switching Characteristics

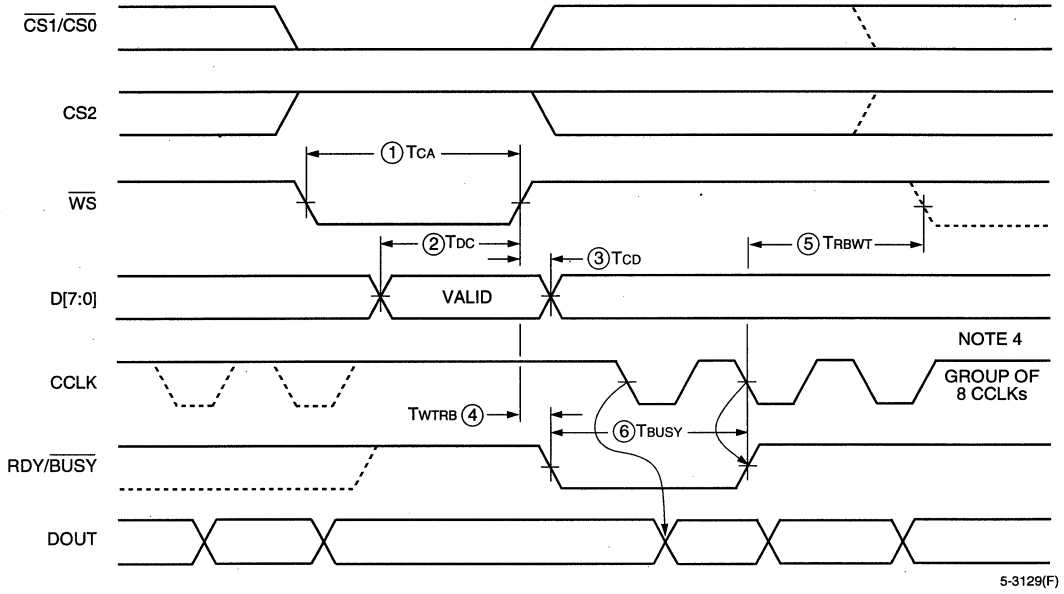
Table 27. Master Parallel Mode Switching Characteristics

Signal	Description	Symbol		Min	Max	Unit
RCLK	To Address Valid	1	TRAC	0	200	ns
	To Data Setup	2	TDRC	60	—	ns
	To Data Hold	3	TRCD	0	—	ns
	RCLK High	—	TRCH	600	—	ns
	RCLK Low	—	TRCL	4.0	—	µs

Notes:
At powerup, 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 nonmonotonically rising VCC may require a >1 µs high level on $\overline{\text{RESET}}$, followed by >6 µs low level on $\overline{\text{RESET}}$ and D/$\overline{\text{P}}$ after VCC has reached 4.0 V.

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

Electrical Characteristics (continued)



Note: The requirements in this timing diagram are extremely relaxed; 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 $\overline{BUSY}$.

Figure 39. Peripheral Mode Switching Characteristics

Table 28. Peripheral Mode Programming Switching Characteristics

Signal	Description	Symbol		Min	Max	Unit
Write Signal	Effective Write Time Required ($\overline{CS0}$ x $\overline{CS1}$ x CS2 x $\overline{WS}$)	1	TCA	100	—	ns
D[7:0]	DIN Setup Time Required	2	TDC	60	—	ns
	DIN Hold TIme Required	3	TCD	0	—	ns
RDY/$\overline{BUSY}$	RDY/$\overline{BUSY}$ Delay after End of $\overline{WS}$	4	TWTRB	—	—	ns
	Earliest Next $\overline{WS}$ after End of $\overline{BUSY}$	5	TRBWT	0	60	ns
	$\overline{BUSY}$ Low Time Generated	6	TBUSY	2	9	CCLK Periods

Notes:

At powerup, 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 nonmonotonically rising VCC may require a >1 µs high level on $\overline{RESET}$, followed by >6 µs low level on $\overline{RESET}$ and D/$\overline{P}$ after VCC has reached 4.0 V.

Configuration must be delayed until the $\overline{INIT}$ of all LCAs is high.

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.

CCLK and DOUT timing is tested in slave mode.

Electrical Characteristics (continued)

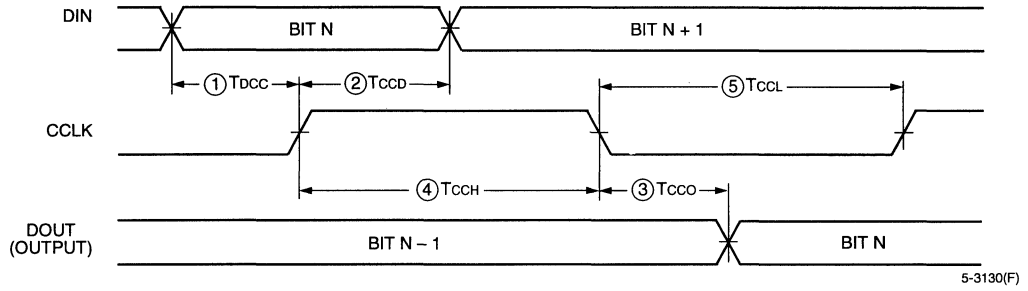


Figure 40. Slave Mode Switching Characteristics

Table 29. Slave Mode Switching Characteristics

Commercial: V_{CC} = 5.0 V ± 5%; 0 °C ≤ T_A ≤ 70 °C; Industrial: V_{CC} = 5.0 ± 10%, –40 °C ≤ T_A ≤ +85 °C.

Signal	Description	Symbol		Min	Max	Unit
CCLK	To DOUT	3	T_{CCO}	—	100	ns
	DIN Setup	1	T_{DCC}	60	—	ns
	DIN Hold	2	T_{CCD}	0	—	ns
	HIGH Time	4	T_{CCH}	0.05	—	µs
	LOW Time	5	T_{CCL}	0.05	5.0	µs
	Frequency	—	F_{CC}	—	10.0	MHz

Notes:
The maximum limit of CCLK LOW time is caused by dynamic circuitry inside the LCA device.

Configuration must be delayed until the $\overline{INIT}$ of all LCAs is high.

At powerup, V_{CC} must rise from 2.0 V to V_{CC} minimum in less than 25 ms. If this is not possible, configuration can be delayed by holding $\overline{RESET}$ low until V_{CC} has reached 4.0 V. A very long V_{CC} rise time of >100 ms, or a nonmonotonically rising V_{CC}, may require a >1 µs high level on $\overline{RESET}$, followed by >6 µs low level on $\overline{RESET}$ and D/$\overline{P}$ after V_{CC} has reached 4.0 V.

Electrical Characteristics (continued)

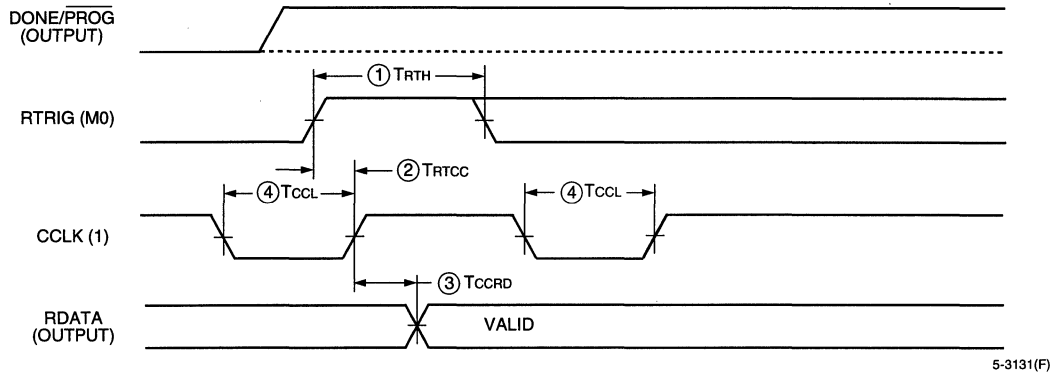


Figure 41. Program Readback Switching Characteristics

Table 30. Program Readback Switching Characteristics

Commercial: V_{CC} = 5.0 V ± 5%; 0 °C ≤ T_A ≤ 70 °C; Industrial: V_{CC} = 5.0 ± 10%, –40 °C ≤ T_A ≤ +85 °C.

Signal	Description	Symbol		Min	Max	Unit
RTRIG	RTRIG HIGH	1	T_{RTH}	250	—	ns
CCLK	RTRIG Setup	2	T_{RTCC}	200	—	ns
	$\overline{\text{RDATA}}$ Delay	3	T_{CCRD}	—	100	ns
	HIGH Time	5	T_{CCH}	0.05	—	µs
	LOW Time	4	T_{CCL}	0.05	5.0	µs

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

RTRIG (M0 positive transition) must not be done until after one clock following active I/O pins.

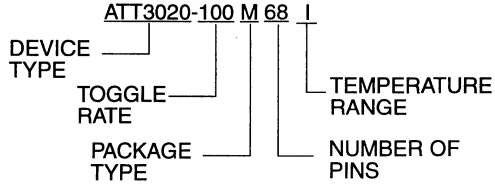
Readback should not be initiated until after configuration is complete.

Ordering Information

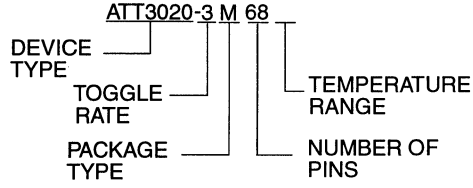
The ATT3000 Series includes standard and high-performance FPGAs. The part nomenclature uses two different suffixes for speed designation. The lower-speed ATT3000 Series devices use a flip-flop toggle rate (-50, -70, -100, -125, -150), which corresponds to *XC3000* Series nomenclature. The ATT3000 Series High-Performance FPGAs use a suffix which is an approximation of the look-up table delay (-5, -4, and -3), which corresponds to *XC3100* nomenclature.

For burn-in diagrams and/or package assembly information call 1-800-EASY-FPG(A) or 1-800-327-9374.

Example: ATT3020, 100 MHz, 68-Lead PLCC, Industrial Temperature



Example: ATT3020, 270 MHz, 68-Lead PLCC, Commercial Temperature



Note: For availability of device types or packaging options, please contact your Lucent Technologies Sales Representative or an authorized distributor.

Table 31. FPGA Temperature Options

Symbol	Description	Temperature
(Blank)	Commercial	0 °C to 70 °C
I	Industrial	–40 °C to +85 °C

Table 32. FPGA Package Options

Symbol	Description
H	Plastic Pin Grid Array
J	Metric Quad Flat Pack
M	Plastic Leaded Chip Carrier
S	Shrink Quad Flat Pack
T	Thin Quad Flat Pack

Ordering Information (continued)

Table 33. ATT3000 Package Matrix

Device	Speed	44-Pin PLCC M44	68-Pin PLCC M68	84-Pin PLCC M84	100-Pin MQFP J100	100-Pin TQFP T100	132-Pin PPGA H132	144-Pin TQFP T144	160-Pin MQFP J160	175-Pin PPGA H175	208-Pin SQFP S208
ATT3020	-70	—	CI	CI	CI	—	—	—	—	—	—
	-100	—	CI	CI	CI	—	—	—	—	—	—
	-125	—	CI	CI	CI	—	—	—	—	—	—
	-5	—	CI	CI	CI	—	—	—	—	—	—
	-4	—	C	C	C	—	—	—	—	—	—
	-3	—	C	C	C	—	—	—	—	—	—
ATT3030	-70	CI	CI	CI	CI	CI	—	—	—	—	—
	-100	CI	CI	CI	CI	CI	—	—	—	—	—
	-125	CI	CI	CI	CI	CI	—	—	—	—	—
	-5	CI	CI	CI	CI	CI	—	—	—	—	—
	-4	C	C	C	C	C	—	—	—	—	—
	-3	C	C	C	C	C	—	—	—	—	—
ATT3042	-70	—	—	CI	CI	CI	CI	CI	—	—	—
	-100	—	—	CI	CI	CI	CI	CI	—	—	—
	-125	—	—	CI	CI	CI	CI	CI	—	—	—
	-5	—	—	CI	CI	CI	CI	CI	—	—	—
	-4	—	—	C	C	C	C	C	—	—	—
	-3	—	—	C	C	C	C	C	—	—	—
ATT3064	-70	—	—	CI	—	CI	CI	CI	CI	—	—
	-100	—	—	CI	—	CI	CI	CI	CI	—	—
	-125	—	—	CI	—	CI	CI	CI	CI	—	—
	-5	—	—	CI	—	CI	CI	CI	CI	—	—
	-4	—	—	C	—	C	C	C	C	—	—
	-3	—	—	C	—	C	C	C	C	—	—
ATT3090	-70	—	—	CI	—	—	—	—	CI	CI	CI
	-100	—	—	CI	—	—	—	—	CI	CI	CI
	-125	—	—	CI	—	—	—	—	CI	CI	CI
	-5	—	—	CI	—	—	—	—	CI	CI	CI
	-4	—	—	C	—	—	—	—	C	C	C
	-3	—	—	C	—	—	—	—	C	C	C

Key: C = commercial, I = industrial.

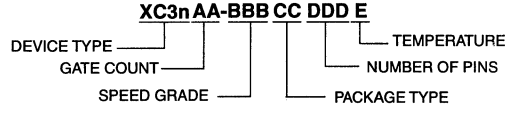
ATT3000 Series Cross-Reference Guide

Cross-Referencing ATT3000 Series FPGAs with *Xilinx XC3000, XC3000A, XC3100,* and *XC3100A* FPGAs

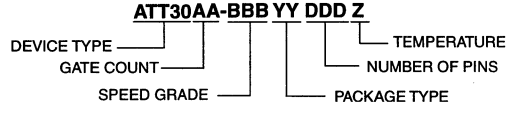
Xilinx XC3000 and *XC3100*

The Lucent Technologies ATT3000 family is a direct second-source replacement for the *Xilinx XC3000* family. The devices are 100% I/O and bitstream compatible. In general, a *Xilinx* part number corresponds with an associated ATT3000 part number. However, certain identifiers in each part number vary in meaning. For example, identifiers AA, BBB, and DDD in Figure 1 represent the same values in both the *Xilinx* and Lucent parts. However, identifier CC (package type) in the *Xilinx* part number is related to identifier YY in the Lucent part number. Also, identifier E in the *Xilinx* part number relates to identifier Z in the Lucent part number (see Figure 1 and Table 1).

In January 1993, *Xilinx* changed the nomenclature for their *XC3000* series, moving away from the -70, -125, etc. speed grade ratings and creating the *XC3100* series with a -3, and -4 type rating. As with the *XC3000*, the *XC3100* series and ATT3000 series of FPGAs are identical in I/O and are bitstream compatible. The family identifiers in each part number vary in the same manner described for the *XC3000* series above and as illustrated in Figure 1.



***Xilinx XC3000* and *XC3100* Series Part Numbers**



Lucent ATT3000 Series Part Numbers

Figure 1. *Xilinx* and Lucent Part Numbers Compared

Table 1. Lucent/*Xilinx* Package Options

Lucent	*Xilinx*	Description
M	PC	Plastic Leaded Chip Carrier (PLCC)
J	PQ	Quad Flat Package (QFP)
H	PP	Plastic Pin Grid Array (PPGA)
S	PQ	Shrink Quad Flat Package (SQFP)
T	TQ	Thin Quad Flat Package (TQFP)

Table 2. Lucent/*Xilinx* Temperature Options

Lucent	*Xilinx*	Description
(Blank)	C	Commercial
I	I	Industrial

XC3000A and *XC3100A*

The "A" revision to *Xilinx*'s 3000 and 3100 families involved the addition of minimal routing resources. These devices are not directly crossable to ATT3000 devices, but can be cross-referenced under two conditions:

1. The design may have been created as a standard *XC3000* or *XC3100* device. The bill of materials may call out an "A" part due to lower pricing or availability issues. "A" devices are backward compatible with earlier bitstreams. If a device was originally created as a "non-A" device, the comparable ATT3000 device can be dropped-in, just as with the other *XC3000* devices.
2. If the bitstream originally targeted the "A" family, it is necessary to retarget the original design to the ATT3000 family and change the bitstream on the PWB. This can usually be accomplished with minimal effort using the *ORCA* Foundry suite of tools. Because the *ORCA* Foundry tools have superior routing algorithms, an ATT3000 design can usually succeed, regardless of results obtained with *Xilinx*'s *XACT* software.

ATT3000 Series Cross-Reference to *Xilinx XC3000* Series

Xilinx Part Number	Lucent Part Number	Description*	Package
XC3020-70PC68C	ATT3020-70M68	2000 Gate 70 MHz	68-Lead PLCC
XC3020-70PC68I	ATT3020-70M68I	2000 Gate 70 MHz	68-Lead PLCC
XC3020-100PC68C	ATT3020-100M68	2000 Gate 100 MHz	68-Lead PLCC
XC3020-100PC68I	ATT3020-100M68I	2000 Gate 100 MHz	68-Lead PLCC
XC3020-125PC68C	ATT3020-125M68	2000 Gate 125 MHz	68-Lead PLCC
XC3020-70PC84C	ATT3020-70M84	2000 Gate 70 MHz	84-Lead PLCC
XC3020-70PC84I	ATT3020-70M84I	2000 Gate 70 MHz	84-Lead PLCC
XC3020-100PC84C	ATT3020-100M84	2000 Gate 100 MHz	84-Lead PLCC
XC3020-100PC84I	ATT3020-100M84I	2000 Gate 100 MHz	84-Lead PLCC
XC3020-125PC84C	ATT3020-125M84	2000 Gate 125 MHz	84-Lead PLCC
XC3020-70PQ100C	ATT3020-70J100	2000 Gate 70 MHz	100-Lead PQFP
XC3020-70PQ100I	ATT3020-70J100I	2000 Gate 70 MHz	100-Lead PQFP
XC3020-100PQ100C	ATT3020-100J100	2000 Gate 100 MHz	100-Lead PQFP
XC3020-100PQ100I	ATT3020-100J100I	2000 Gate 100 MHz	100-Lead PQFP
XC3020-125PQ100I	ATT3020-125J100	2000 Gate 125 MHz	100-Lead PQFP
XC3030-70PC44C	ATT3030-70M44	3000 Gate 70 MHz	44-Lead PLCC
XC3030-70PC44I	ATT3030-70M44I	3000 Gate 70 MHz	44-Lead PLCC
XC3030-100PC44C	ATT3030-100M44	3000 Gate 100 MHz	44-Lead PLCC
XC3030-100PC44I	ATT3030-100M44I	3000 Gate 100 MHz	44-Lead PLCC
XC3030-125PC44C	ATT3030-125M44	3000 Gate 125 MHz	44-Lead PLCC
XC3030-70PC68C	ATT3030-70M68	3000 Gate 70 MHz	68-Lead PLCC
XC3030-70PC68I	ATT3030-70M68I	3000 Gate 70 MHz	68-Lead PLCC
XC3030-100PC68C	ATT3030-100M68	3000 Gate 100 MHz	68-Lead PLCC
XC3030-100PC68I	ATT3030-100M68I	3000 Gate 100 MHz	68-Lead PLCC
XC3030-125PC68C	ATT3030-125M68	3000 Gate 125 MHz	68-Lead PLCC
XC3030-70PC84C	ATT3030-70M84	3000 Gate 70 MHz	84-Lead PLCC
XC3030-70PC84I	ATT3030-70M84I	3000 Gate 70 MHz	84-Lead PLCC
XC3030-100PC84C	ATT3030-100M84	3000 Gate 100 MHz	84-Lead PLCC
XC3030-100PC84I	ATT3030-100M84I	3000 Gate 100 MHz	84-Lead PLCC
XC3030-125PC84C	ATT3030-125M84	3000 Gate 125 MHz	84-Lead PLCC
XC3030-70PQ100C	ATT3030-70J100	3000 Gate 70 MHz	100-Lead PQFP
XC3030-70PQ100I	ATT3030-70J100I	3000 Gate 70 MHz	100-Lead PQFP
XC3030-100PQ100C	ATT3030-100J100	3000 Gate 100 MHz	100-Lead PQFP
XC3030-100PQ100I	ATT3030-100J100I	3000 Gate 100 MHz	100-Lead PQFP
XC3030-100PQ125C	ATT3030-125J100	3000 Gate 125 MHz	100-Lead PQFP
XC3042-70PC84C	ATT3042-70M84	4200 Gate 70 MHz	84-Lead PLCC
XC3042-70PC84I	ATT3042-70M84I	4200 Gate 70 MHz	84-Lead PLCC
XC3042-100PC84C	ATT3042-100M84	4200 Gate 100 MHz	84-Lead PLCC
XC3042-100PC84I	ATT3042-100M84I	4200 Gate 100 MHz	84-Lead PLCC
XC3042-125PC84C	ATT3042-125M84	4200 Gate 125 MHz	84-Lead PLCC
XC3042-70PQ100C	ATT3042-70J100	4200 Gate 70 MHz	100-Lead PQFP
XC3042-70PQ100I	ATT3042-70J100I	4200 Gate 70 MHz	100-Lead PQFP
XC3042-100PQ100C	ATT3042-100J100	4200 Gate 100 MHz	100-Lead PQFP
XC3042-100PQ100I	ATT3042-100J100I	4200 Gate 100 MHz	100-Lead PQFP
XC3042-125PQ100C	ATT3042-125J100	4200 Gate 125 MHz	100-Lead PQFP
XC3042-70PP132C	ATT3042-70H132	4200 Gate 70 MHz	132-Lead PPGA
XC3042-70PP132I	ATT3042-70H132I	4200 Gate 70 MHz	132-Lead PPGA
XC3042-100PP132C	ATT3042-100H132	4200 Gate 100 MHz	132-Lead PPGA
XC3042-100PP132I	ATT3042-100H132I	4200 Gate 100 MHz	132-Lead PPGA

* The speed designated is the maximum flip-flop toggle rate.

ATT3000 Series Cross-Reference to *Xilinx XC3000* Series (continued)

Xilinx Part Number	Lucent Part Number	Description*	Package
XC3042-125PP132C	ATT3042-125H132	4200 Gate 125 MHz	132-Lead PPGA
XC3064-70PC84C	ATT3064-70M84	6400 Gate 70 MHz	84-Lead PLCC
—	ATT3064-70M84I	6400 Gate 70 MHz	84-Lead PLCC
XC3064-100PC84C	ATT3064-100M84	6400 Gate 100 MHz	84-Lead PLCC
—	ATT3064-100M84I	6400 Gate 100 MHz	84-Lead PLCC
XC3064-125PC84C	ATT3064-125M84	6400 Gate 125 MHz	84-Lead PLCC
XC3064-70PP132C	ATT3064-70H132	6400 Gate 70 MHz	132-Lead PPGA
XC3064-70PP132I	ATT3064-70H132I	6400 Gate 70 MHz	132-Lead PPGA
XC3064-100PP132C	ATT3064-100H132	6400 Gate 100 MHz	132-Lead PPGA
XC3064-100PP132I	ATT3064-100H132I	6400 Gate 100 MHz	132-Lead PPGA
XC3064-125PP132C	ATT3064-125H132	6400 Gate 125 MHz	132-Lead PPGA
XC3064-70PQ160C	ATT3064-70J160	6400 Gate 70 MHz	160-Lead PQFP
XC3064-70PQ160I	ATT3064-70J160I	6400 Gate 70 MHz	160-Lead PQFP
XC3064-100PQ160C	ATT3064-100J160	6400 Gate 100 MHz	160-Lead PQFP
XC3064-100PQ160I	ATT3064-100J160I	6400 Gate 100 MHz	160-Lead PQFP
XC3064-125PQ160C	ATT3064-125J160	6400 Gate 125 MHz	160-Lead PQFP
XC3090-70PC84C*	ATT3090-70M84	9000 Gate 70 MHz	84-Lead PLCC
—	ATT3090-70M84I	9000 Gate 70 MHz	84-Lead PLCC
XC3090-100PC84C	ATT3090-100M84	9000 Gate 100 MHz	84-Lead PLCC
—	ATT3090-100M84I	9000 Gate 100 MHz	84-Lead PLCC
XC3090-125PC84C	ATT3090-125M84	9000 Gate 125 MHz	84-Lead PLCC
XC3090-70PQ160C	ATT3090-70J160	9000 Gate 70 MHz	160-Lead PQFP
XC3090-70PQ160I	ATT3090-70J160I	9000 Gate 70 MHz	160-Lead PQFP
XC3090-100PQ160C	ATT3090-100J160	9000 Gate 100 MHz	160-Lead PQFP
XC3090-100PQ160I	ATT3090-100J160I	9000 Gate 100 MHz	160-Lead PQFP
XC3090-125PQ160C	ATT3090-125J160	9000 Gate 125 MHz	160-Lead PQFP
XC3090-70PP175C	ATT3090-70H175	9000 Gate 70 MHz	175-Lead PPGA
XC3090-70PP175I	ATT3090-70H175I	6400 Gate 70 MHz	175-Lead PPGA
XC3090-100PP175C	ATT3090-100H175	9000 Gate 100 MHz	175-Lead PPGA
XC3090-100PP175I	ATT3090-100H175I	6400 Gate 100 MHz	175-Lead PPGA
XC3090-125PP175C	ATT3090-125H175	9000 Gate 125 MHz	175-Lead PPGA
XC3090-70PQ208C	ATT3090-70Q208	9000 Gate 70 MHz	208-Lead PQFP
XC3090-70PQ208I	ATT3090-70Q208I	9000 Gate 70 MHz	208-Lead PQFP
XC3090-100PQ208C	ATT3090-100Q208	9000 Gate 100 MHz	208-Lead PQFP
XC3090-100PQ208I	ATT3090-100Q208I	9000 Gate 100 MHz	208-Lead PQFP
XC3090-125PQ208C	ATT3090-125Q208	9000 Gate 125 MHz	208-Lead PQFP

* The speed designated is the maximum flip-flop toggle rate.

High-Speed ATT3000 Series Cross-Reference to *Xilinx XC3100* Series

Xilinx Part No.	Lucent Part No.	Description*	Package
XC3120-5PC68C	ATT3020-5M68	2000 Gate 4.1 ns	68-Lead PLCC
XC3120-5PC68I	ATT3020-5M68I	2000 Gate 4.1 ns	68-Lead PLCC
XC3120-4PC68C	ATT3020-4M68	2000 Gate 3.3 ns	68-Lead PLCC
XC3120-4PC68I	ATT3020-4M68I	2000 Gate 3.3 ns	68-Lead PLCC
XC3120-3PC68C	ATT3020-3M68	2000 Gate 2.7 ns	68-Lead PLCC
XC3120-5PC84C	ATT3020-5M84	2000 Gate 4.1 ns	84-Lead PLCC
XC3120-5PC84I	ATT3020-5M84I	2000 Gate 4.1 ns	84-Lead PLCC
XC3120-4PC84C	ATT3020-4M84	2000 Gate 3.3 ns	84-Lead PLCC
XC3120-4PC84I	ATT3020-4M84I	2000 Gate 3.3 ns	84-Lead PLCC
XC3120-3PC84C	ATT3020-3M84	2000 Gate 2.7 ns	84-Lead PLCC
XC3120-5PQ100C	ATT3020-5J100	2000 Gate 4.1 ns	100-Lead PQFP
XC3120-5PQ100I	ATT3020-5J100I	2000 Gate 4.1 ns	100-Lead PQFP
XC3120-4PQ100C	ATT3020-4J100	2000 Gate 3.3 ns	100-Lead PQFP
XC3120-4PQ100I	ATT3020-4J100I	2000 Gate 3.3 ns	100-Lead PQFP
XC3120-3PQ100C	ATT3020-3J100	2000 Gate 2.7 ns	100-Lead PQFP
XC3130-5PC44C	ATT3030-5M44	3000 Gate 4.1 ns	44-Lead PLCC
XC3130-5PC44I	ATT3030-5M44I	3000 Gate 4.1 ns	44-Lead PLCC
XC3130-4PC44C	ATT3030-4M44	3000 Gate 3.3 ns	44-Lead PLCC
XC3130-4PC44I	ATT3030-4M44I	3000 Gate 3.3 ns	44-Lead PLCC
XC3130-3PC44C	ATT3030-3M44	3000 Gate 2.7 ns	44-Lead PLCC
XC3130-5PC68C	ATT3030-5M68	3000 Gate 4.1 ns	68-Lead PLCC
XC3130-5PC68I	ATT3030-5M68I	3000 Gate 4.1 ns	68-Lead PLCC
XC3130-4PC68C	ATT3030-4M68	3000 Gate 3.3 ns	68-Lead PLCC
XC3130-4PC68I	ATT3030-4M68I	3000 Gate 3.3 ns	68-Lead PLCC
XC3130-3PC68C	ATT3030-3M68	3000 Gate 2.7 ns	68-Lead PLCC
XC3130-5PC84C	ATT3030-5M84	3000 Gate 4.1 ns	84-Lead PLCC
XC3130-5PC84I	ATT3030-5M84I	3000 Gate 4.1 ns	84-Lead PLCC
XC3130-4PC84C	ATT3030-4M84	3000 Gate 3.3 ns	84-Lead PLCC
XC3130-4PC84I	ATT3030-4M84I	3000 Gate 3.3 ns	84-Lead PLCC
XC3130-3PC84C	ATT3030-3M84	3000 Gate 2.7 ns	84-Lead PLCC
XC3130-5PQ100C	ATT3030-5J100	3000 Gate 4.1 ns	100-Lead PQFP
XC3130-5PQ100I	ATT3030-5J100I	3000 Gate 4.1 ns	100-Lead PQFP
XC3130-4PQ100C	ATT3030-4J100	3000 Gate 3.3 ns	100-Lead PQFP
XC3130-4PQ100I	ATT3030-4J100I	3000 Gate 3.3 ns	100-Lead PQFP
XC3130-3PQ100C	ATT3030-3J100	3000 Gate 2.7 ns	100-Lead PQFP
XC3130-5TQ100C	ATT3030-5T100	3000 Gate 4.1 ns	100-Lead TQFP
XC3130-4TQ100C	ATT3030-4T100	3000 Gate 3.3 ns	100-Lead TQFP
XC3130-3TQ100C	ATT3030-3T100	3000 Gate 2.7 ns	100-Lead TQFP
XC3142-5PC84C	ATT3042-5M84	4200 Gate 4.1 ns	84-Lead PLCC
XC3142-5PC84I	ATT3042-5M84I	4200 Gate 4.1 ns	84-Lead PLCC
XC3142-4PC84C	ATT3042-4M84	4200 Gate 3.3 ns	84-Lead PLCC
XC3142-4PC84I	ATT3042-4M84I	4200 Gate 3.3 ns	84-Lead PLCC
XC3142-3PC84C	ATT3042-3M84	4200 Gate 2.7 ns	84-Lead PLCC
XC3142-5PQ100C	ATT3042-5J100	4200 Gate 4.1 ns	100-Lead QFP
XC3142-5PQ100I	ATT3042-5J100I	4200 Gate 4.1 ns	100-Lead QFP
XC3142-4PQ100C	ATT3042-4J100	4200 Gate 3.3 ns	100-Lead PQFP
XC3142-4PQ100I	ATT3042-4J100I	4200 Gate 3.3 ns	100-Lead PQFP
XC3142-3PQ100C	ATT3042-3J100	4200 Gate 2.7 ns	100-Lead PQFP

* The timing specified is the minimum CLB look-up table propagation delay.

High-Speed ATT3000 Series Cross-Reference to *Xilinx XC3100* Series (continued)

Xilinx Part No.	Lucent Part No.	Description*	Package
XC3142-5TQ100C	ATT3042-5T100	4200 Gate 4.1 ns	100-Lead TQFP
XC3142-4TQ100C	ATT3042-4T100	4200 Gate 3.3 ns	100-Lead TQFP
XC3142-3TQ100C	ATT3042-3T100	4200 Gate 2.7 ns	100-Lead TQFP
XC3142-5PP132C	ATT3042-5H132	4200 Gate 4.1 ns	132-Lead PPGA
—	ATT3042-5H132I	4200 Gate 4.1 ns	132-Lead PPGA
XC3142-4PP132C	ATT3042-4H132	4200 Gate 3.3 ns	132-Lead PPGA
—	ATT3042-4H132I	4200 Gate 3.3 ns	132-Lead PPGA
XC3142-3PP132C	ATT3042-3H132	4200 Gate 2.7 ns	132-Lead PPGA
XC3142-5TQ144C	ATT3042-5T144	4200 Gate 4.1 ns	144-Lead TQFP
XC3142-4TQ144I	ATT3042-4T144I	4200 Gate 3.3 ns	144-Lead TQFP
XC3164-5PC84C	ATT3064-5M84	6400 Gate 4.1 ns	84-Lead PLCC
XC3164-5PC84I	ATT3064-5M84I	6400 Gate 4.1 ns	84-Lead PLCC
XC3164-4PC84C	ATT3064-4M84	6400 Gate 3.3 ns	84-Lead PLCC
XC3164-4PC84I	ATT3064-4M84I	6400 Gate 3.3 ns	84-Lead PLCC
XC3164-3PC84C	ATT3064-3M84	6400 Gate 2.7 ns	84-Lead PLCC
XC3164-5PP132C	ATT3064-5H132	6400 Gate 4.1 ns	132-Lead PPGA
XC3164-5PP132I	ATT3064-5H132I	6400 Gate 4.1 ns	132-Lead PPGA
XC3164-4PP132C	ATT3064-4H132	6400 Gate 3.3 ns	132-Lead PPGA
XC3164-4PP132I	ATT3064-4H132I	6400 Gate 3.3 ns	132-Lead PPGA
XC3164-3PP132C	ATT3064-3H132	6400 Gate 2.7 ns	132-Lead PPGA
XC3164-5PQ160C	ATT3064-5J160	6400 Gate 4.1 ns	160-Lead PQFP
XC3164-5PQ160I	ATT3064-5J160I	6400 Gate 4.1 ns	160-Lead PQFP
XC3164-4PQ160C	ATT3064-4J160	6400 Gate 3.3 ns	160-Lead PQFP
XC3164-4PQ160I	ATT3064-4J160I	6400 Gate 3.3 ns	160-Lead PQFP
XC3164-3PQ160C	ATT3064-3J160	6400 Gate 2.7 ns	160-Lead PQFP
XC3190-5PC84C	ATT3090-5M84	9000 Gate 4.1 ns	84-Lead PLCC
XC3190-5PC84I	ATT3090-5M84I	9000 Gate 4.1 ns	84-Lead PLCC
XC3190-4PC84C	ATT3090-4M84	9000 Gate 3.3 ns	84-Lead PLCC
XC3190-4PC84I	ATT3090-4M84I	9000 Gate 3.3 ns	84-Lead PLCC
XC3190-3PC84C	ATT3090-3M84	9000 Gate 2.7 ns	84-Lead PLCC
XC3190-5PQ160C	ATT3090-5J160	9000 Gate 4.1 ns	160-Lead PQFP
XC3190-4PQ160C	ATT3090-4J160	9000 Gate 3.3 ns	160-Lead PQFP
XC3190-3PQ160C	ATT3090-3J160	9000 Gate 2.7 ns	160-Lead PQFP
XC3190-5PP175C	ATT3090-5H175	9000 Gate 4.1 ns	175-Lead PPGA
XC3190-5PP175I	ATT3090-5H175I	9000 Gate 4.1 ns	175-Lead PPGA
XC3190-4PP175C	ATT3090-4H175	9000 Gate 3.3 ns	175-Lead PPGA
XC3190-4PP175I	ATT3090-4H175I	9000 Gate 3.3 ns	175-Lead PPGA
XC3190-3PP175C	ATT3090-3H175	9000 Gate 2.7 ns	175-Lead PPGA
XC3190-5PQ208C	ATT3090-5S208	9000 Gate 4.1 ns	208-Lead SQFP
XC3190-5PQ208I	ATT3090-5S208I	9000 Gate 4.1 ns	208-Lead SQFP
XC3190-4PQ208C	ATT3090-4S208	9000 Gate 3.3 ns	208-Lead SQFP
XC3190-4PQ208I	ATT3090-4S208I	9000 Gate 3.3 ns	208-Lead SQFP
XC3190-3PQ208C	ATT3090-3S208	9000 Gate 2.7 ns	208-Lead SQFP

* The timing specified is the minimum CLB look-up table propagation delay.

High-Speed ATT3000 Series Cross-Reference to *Xilinx XC3xxxA* Series

Xilinx Part No.	Lucent Part No.	Description*	Package
XC3130A-5PC44C	ATT3030-5M44	3000 Gate 4.1 ns	44-Lead PLCC
XC3130A-5PC44I	ATT3030-5M44I	3000 Gate 4.1 ns	44-Lead PLCC
XC3130A-4PC44C	ATT3030-4M44	3000 Gate 3.3 ns	44-Lead PLCC
XC3130A-4PC44I	ATT3030-4M44I	3000 Gate 3.3 ns	44-Lead PLCC
XC3130A-3PC44C	ATT3030-3M44	3000 Gate 2.7 ns	44-Lead PLCC
XC3130A-5PC68C	ATT3030-5M68	3000 Gate 4.1 ns	68-Lead PLCC
XC3130A-5PC68I	ATT3030-5M68I	3000 Gate 4.1 ns	68-Lead PLCC
XC3130A-4PC68C	ATT3030-4M68	3000 Gate 3.3 ns	68-Lead PLCC
XC3130A-4PC68I	ATT3030-4M68I	3000 Gate 3.3 ns	68-Lead PLCC
XC3130A-3PC68C	ATT3030-3M68	3000 Gate 2.7 ns	68-Lead PLCC
XC3130A-5PC84C	ATT3030-5M84	3000 Gate 4.1 ns	84-Lead PLCC
XC3130A-5PC84I	ATT3030-5M84I	3000 Gate 4.1 ns	84-Lead PLCC
XC3130A-4PC84C	ATT3030-4M84	3000 Gate 3.3 ns	84-Lead PLCC
XC3130A-4PC84I	ATT3030-4M84I	3000 Gate 3.3 ns	84-Lead PLCC
XC3130A-3PC84C	ATT3030-3M84	3000 Gate 2.7 ns	84-Lead PLCC
XC3130A-5PQ100C	ATT3030-5J100	3000 Gate 4.1 ns	100-Lead PQFP
XC3130A-5PQ100I	ATT3030-5J100I	3000 Gate 4.1 ns	100-Lead PQFP
XC3130A-4PQ100C	ATT3030-4J100	3000 Gate 3.3 ns	100-Lead PQFP
XC3130A-4PQ100I	ATT3030-4J100I	3000 Gate 3.3 ns	100-Lead PQFP
XC3130A-3PQ100C	ATT3030-3J100	3000 Gate 2.7 ns	100-Lead PQFP
XC3142A-5PC84C	ATT3042-5M84	4200 Gate 4.1 ns	84-Lead PLCC
XC3142A-5PC84I	ATT3042-5M84I	4200 Gate 4.1 ns	84-Lead PLCC
XC3142A-4PC84C	ATT3042-4M84	4200 Gate 3.3 ns	84-Lead PLCC
XC3142A-4PC84I	ATT3042-4M84I	4200 Gate 3.3 ns	84-Lead PLCC
XC3142A-3PC84C	ATT3042-3M84	4200 Gate 2.7 ns	84-Lead PLCC
XC3142A-5PQ100C	ATT3042-5J100	4200 Gate 4.1 ns	100-Lead PQFP
XC3142A-5PQ100I	ATT3042-5J100I	4200 Gate 4.1 ns	100-Lead PQFP
XC3142A-4PQ100C	ATT3042-4J100	4200 Gate 3.3 ns	100-Lead PQFP
XC3142A-4PQ100I	ATT3042-4J100I	4200 Gate 3.3 ns	100-Lead PQFP
XC3142A-3PQ100C	ATT3042-3J100	4200 Gate 2.7 ns	100-Lead PQFP
XC3142A-5PP132C	ATT3042-5H132	4200 Gate 4.1 ns	132-Lead PPGA
XC3142A-4PP132C	ATT3042-4H132	4200 Gate 3.3 ns	132-Lead PPGA
XC3142A-3PP132C	ATT3042-3H132	4200 Gate 2.7 ns	132-Lead PPGA
XC3142A-5TQ144C	ATT3042-5T144	4200 Gate 4.1 ns	144-Lead TQFP
XC3142A-5TQ144I	ATT3042-5T144I	4200 Gate 4.1 ns	144-Lead TQFP
XC3142A-4TQ144C	ATT3042-4T144	4200 Gate 3.3 ns	144-Lead TQFP
XC3142A-4TQ144I	ATT3042-4T144I	4200 Gate 3.3 ns	144-Lead TQFP
XC3142A-3TQ144C	ATT3042-3T144	4200 Gate 2.7 ns	144-Lead TQFP
XC3164A-5PC84C	ATT3064-5M84	6400 Gate 4.1 ns	84-Lead PLCC
XC3164A-5PC84I	ATT3064-5M84I	6400 Gate 4.1 ns	84-Lead PLCC
XC3164A-4PC84C	ATT3064-4M84	6400 Gate 3.3 ns	84-Lead PLCC
XC3164A-4PC84I	ATT3064-4M84I	6400 Gate 3.3 ns	84-Lead PLCC
XC3164A-3PC84C	ATT3064-3M84	6400 Gate 2.7 ns	84-Lead PLCC
XC3164A-5PP132C	ATT3064-5H132	6400 Gate 4.1 ns	132-Lead PPGA
XC3164A-4PP132C	ATT3064-4H132	6400 Gate 3.3 ns	132-Lead PPGA
XC3164A-3PP132C	ATT3064-3H132	6400 Gate 2.7 ns	132-Lead PPGA
XC3164A-5PQ160C	ATT3064-5J160	6400 Gate 4.1 ns	160-Lead PQFP
XC3164A-4PQ160C	ATT3064-4J160	6400 Gate 3.3 ns	160-Lead PQFP
XC3164A-3PQ160C	ATT3064-3J160	6400 Gate 2.7 ns	160-Lead PQFP

* The timing specified is the minimum CLB look-up table propagation delay.

High-Speed ATT3000 Series Cross-Reference to *Xilinx XC3xxxA* Series (continued)

Xilinx Part No.	Lucent Part No.	Description*	Package
XC3190A-5PC84C	ATT3090-5M84	9000 Gate 4.1 ns	84-Lead PLCC
XC3190A-5PC84I	ATT3090-5M84I	9000 Gate 4.1 ns	84-Lead PLCC
XC3190A-4PC84C	ATT3090-4M84	9000 Gate 3.3 ns	84-Lead PLCC
XC3190A-4PC84I	ATT3090-4M84I	9000 Gate 3.3 ns	84-Lead PLCC
XC3190A-3PC84C	ATT3090-3M84	9000 Gate 2.7 ns	84-Lead PLCC
XC3190A-5PQ160C	ATT3090-5J160	9000 Gate 4.1 ns	160-Lead PQFP
XC3190A-5PQ160I	ATT3090-5J160I	9000 Gate 4.1 ns	160-Lead PQFP
XC3190A-4PQ160C	ATT3090-4J160	9000 Gate 3.3 ns	160-Lead PQFP
XC3190A-4PQ160I	ATT3090-4J160I	9000 Gate 3.3 ns	160-Lead PQFP
XC3190A-3PQ160C	ATT3090-3J160	9000 Gate 2.7 ns	160-Lead PQFP
XC3190A-5PP175C	ATT3090-5H175	9000 Gate 4.1 ns	175-Lead PPGA
XC3190A-5PP175I	ATT3090-5H175I	9000 Gate 4.1 ns	175-Lead PPGA
XC3190A-4PP175C	ATT3090-4H175	9000 Gate 3.3 ns	175-Lead PPGA
XC3190A-4PP175I	ATT3090-4H175I	9000 Gate 3.3 ns	175-Lead PPGA
XC3190A-3PP175C	ATT3090-3H175	9000 Gate 2.7 ns	175-Lead PPGA
XC3190A-5PQ208C	ATT3090-5S208	9000 Gate 4.1 ns	208-Lead SQFP
XC3190A-5PQ208I	ATT3090-5S208I	9000 Gate 4.1 ns	208-Lead SQFP
XC3190A-4PQ208C	ATT3090-4S208	9000 Gate 3.3 ns	208-Lead SQFP
XC3190A-4PQ208I	ATT3090-4S208I	9000 Gate 3.3 ns	208-Lead SQFP
XC3190A-3PQ208C	ATT3090-3S208	9000 Gate 2.7 ns	208-Lead SQFP

* The timing specified is the minimum CLB look-up table propagation delay.

Note: Designs originally done utilizing a *Xilinx XC3xxxA* series part cannot be directly crossed into a corresponding Lucent device. However, if a design was originally done using the *XC3xxx* part, rather than the *XC3xxxA* part, the above crosses do apply.

Notes

Lucent Technologies
Bell Labs Innovations

ATT1700A Series Serial ROMs

Features

- 32K, 64K, and 128K x 1 Serial ROMs for configuration of ATT3000 and *ORCA* Series FPGAs
- Pinout and functional replacement of *Xilinx XC1700* series
- Simple 4-wire interface
- Cascadable to support large FPGAs, multiple configurations, and multiple FPGAs
- 8-pin, plastic DIP; 8-pin SOIC; and 20-pin PLCC packages
- Programming support from leading programmer manufacturers
- Programmable polarity on RESET/OE pin
- Full static operation
- Standby current—100 µA typical
- Operating current—10 mA maximum
- 10 MHz maximum clock rate
- Electrostatic discharge protection: >4000 V
- Temperature ranges:
  Commercial: 0 °C to 70 °C
  Industrial: –40 °C to +85 °C

Description

The ATT1700A Series Serial ROM family provides easy-to-use, cost-effective, nonvolatile memory for configuring ATT3000 and *ORCA* Series FPGAs. The ATT1700A Series consists of one-time programmable (OTP) devices. The ATT1700A devices are available in 8-pin, plastic DIP, 8-pin SOIC, and 20-pin PLCC packages.

The ATT1700A Series is a pinout and functional replacement for the ATT1700 and *Xilinx XC1700* families (see Figure 1) and can be programmed by most commercially available programmers. FPGA development tools, such as *ORCA* Foundry, generate configuration files in *Intel, Motorola*, and *Tektronix* formats for use in programmers.

The ATT1700A Series is most often used when the ATT3000 Series and *ORCA* Series FPGAs are configured in the master serial mode. The primary advantage of this configuration mode is that it provides a simple, four-wire interface between the FPGA and serial ROM (see Figure 2).

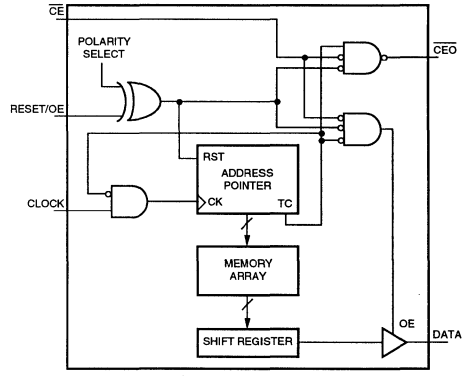


Figure 1. Block Diagram

Pin Information

Table 1. Pin Descriptions

Symbol	Pin Numbers		I/O	Function
	8-Pin	20-Pin		
DATA	1	2	O	DATA output from the serial ROM to FPGA synchronous with the CLOCK input. DATA is 3-stated when either $\overline{CE}$ or OE is inactive.
CLOCK	2	4	I	CLOCK is an input used to increment the address pointer which strobes data out of the DATA pin.
RESET/OE	3	6	I	RESET/OUTPUT ENABLE is a dual-function pin used to reset and enable the ATT1700A Series device. An active level on both $\overline{CE}$ and OE inputs enables data out of the DATA pin. An active level on RESET resets the address pointer. When the serial ROM is programmed, the polarity of RESET/OE is set either with RESET active-high and $\overline{OE}$ active-low or with $\overline{RESET}$ active-low and OE active-high.
$\overline{CE}$	4	8	I	$\overline{CHIP\ ENABLE}$ is an input used to select the device. An active level on both $\overline{CE}$ and OE enables data out of the device. A high on $\overline{CE}$ disables the address pointer and forces the serial ROM into a low-power mode.
VSS	5	10	I	Ground.
$\overline{CEO}$	6	14	O	$\overline{CHIP\ ENABLE\ OUT}$ is asserted low on the clock cycle following the last bit read from the device. $\overline{CEO}$ remains low as long as $\overline{CE}$ and OE are both active.
VPP	7	17	I	VPP is an input used by programmers when programming the serial ROM. The programming operations, voltages, and timing are defined later in this data sheet. For read operations, VPP must be tied directly to VDD.
VDD	8	20	I	Power supply.

FPGA Configuration

The functionality of Lucent Technologies Microelectronics Groups' FPGAs is determined by the contents of the FPGA's configuration memory. The configuration memory is loaded either automatically at powerup or with a configuration command by pulsing the $\overline{PRGM}$ pin low. The FPGAs can be programmed in a variety of modes, and the mode used is determined by the inputs into the FPGA's M[2:0] pins. The configuration modes allow the FPGA to act as a master or a slave and also allow configuration data to be transmitted either serially or in parallel. The ATT1700A Series ROMS are targeted for use when the FPGA is configured serially, primarily in the master serial mode. Table 2 provides the configuration memory requirements for Lucent's FPGAs.

FPGA Master Serial Configuration Mode

The master serial mode provides a simple interface between the FPGA and the serial ROM. Four interface lines, DATA, CLOCK, $\overline{CE}$, and RESET/OE, are required to configure the FPGA. Upon powerup or a configure command ($\overline{PRGM}$ in *ORCA*, $\overline{PROG}$ in ATT3000), when the FPGA's M[2:0] pins are low, the FPGA configures using the master serial mode. The configuration data is transmitted serially into the FPGA's DIN pin from the serial ROM's DATA pin. To synchronize the data, the FPGA's CCLK output is routed into the serial ROM's CLOCK input.

Because the FPGA DIN signal may be unused after FPGA configuration, it is necessary to avoid an unresolved state once the serial ROM has finished sending configuration data. If this pin is used only for the configuration process, it should be configured so that it does not float. This can be accomplished by programming it as an output during normal operation or by programming it as an input with an internal pull-up resistor enabled. CCLK must also be pulled up following configuration.

Signal contention on the DIN pin must be avoided if it is to be used for a user I/O signal after configuration. To avoid contention, the FPGA DONE signal may be programmed (selected in *ORCA* Foundry) to go high prior to the FPGA I/O signals being enabled. An alternative is to use the FPGA's $\overline{LDC}$ to drive the serial ROM's $\overline{CE}$ pin, rather than DONE, and configure $\overline{LDC}$ to output a constant logic 1 high-voltage level after configuration. Control of the serial ROM's $\overline{CE}$ and RESET/OE pins varies, depending upon the FPGA series being used, and is described in subsequent sections.

Table 2. Configuration Requirements

Lucent FPGA	Memory Requirements
ATT3020	14,819
ATT3030	22,216
ATT3042	30,824
ATT3064	46,104
ATT3090	64,200
ATT1C03	57,144
ATT1C05	76,376
ATT1C07	98,296
ATT1C09	122,904
ATT2C04/OR2C04A/OR2T04A	65,424
ATT2C06/OR2C06A/OR2T06A	91,024
ATT2C08/OR2C08A/OR2T08A	115,600
ATT2C10/OR2C10A/OR2T10A	148,944
ATT2C12/OR2C12A/OR2T12A	179,856
ATT2C15/OR2C15A/OR2T15A	220,944
ATT2C26/OR2C26A/OR2T26A	307,024
ATT2C40/OR2C40A/OR2T40A	474,176

The FPGA serial ROM interface used also depends upon the system and configuration requirements. The following are some typical system requirements:

- Configuring an FPGA at powerup
- Configuring an FPGA in response to a configure command
- One serial ROM configures an FPGA with multiple configuration programs
- Cascaded serial ROMs configure daisy-chained FPGAs

In addition to the clock and data lines, the FPGA pins used in configuration/startup are $\overline{RESET}$, DONE, $\overline{PRGM}$, $\overline{LDC}$, HDC, and $\overline{INIT}$. Normally, only a small subset of these pins is used to control the serial ROM's $\overline{CE}$ and RESET/OE pins. In some applications, the RESET/OE signal is generated by the system host, not the FPGA. For example, the host may generate a system reset, allowing the FPGA and the serial ROM to be reset synchronously.

FPGA Configuration (continued)

ATT3000 Series/*ORCA* Series Differences

While both the ATT3000 and *ORCA* Series have $\overline{\text{RESET}}$, $\overline{\text{LDC}}$, HDC, $\overline{\text{INIT}}$, DIN, CCLK, and DOUT pins, there are some configuration differences in the FPGAs. The ATT3000 Series DONE/$\overline{\text{PROG}}$ pin is a shared open-drain I/O, while the *ORCA* Series has discrete DONE and $\overline{\text{PRGM}}$ pins. When the system generates a configure command to the ATT3000, the DONE/$\overline{\text{PROG}}$ pin is held low throughout the configuration cycle. For the *ORCA* Series, the $\overline{\text{PRGM}}$ pin is pulsed low and returned high to initiate configuration. A second difference is the internal pull-ups on the mode select pins. For the ATT3000 Series, only M2 has an internal pull-up during configuration, but for the *ORCA* Series, M[3:0] have pull-ups.

Configuring the FPGA at Powerup

The ATT1700A series can configure FPGAs at powerup. There is level-sensitive, power-on-reset circuitry included in the device that resets the address pointer during powerup. The ATT3000 and *ORCA* FPGAs enable the serial ROM using either the DONE (or $\overline{\text{LDC}}$) and $\overline{\text{INIT}}$ pins. If these signals are low at powerup and they are connected to the $\overline{\text{CE}}$ and RESET/$\overline{\text{OE}}$ pins on the serial ROM, the FPGA is programmed from the serial ROM (see Figures 2 and 3). When these FPGA signals go high at the end of configuration, the serial ROM is disabled.

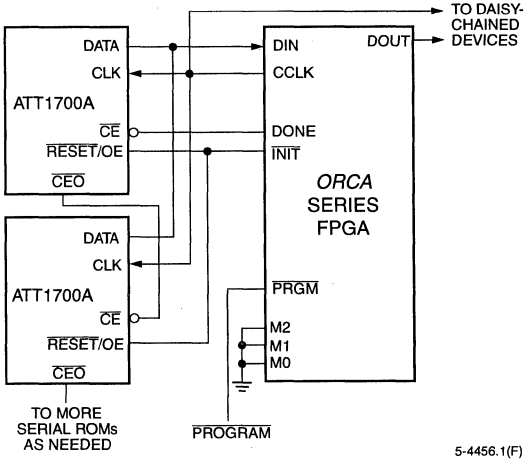


Figure 2. *ORCA* Master Serial Configuration

Configuring the *ORCA* Series FPGA with a Configure Command

The FPGA needs to enable the serial ROM's RESET/OE and $\overline{\text{CE}}$ inputs. The polarity of the RESET/OE input is programmable in the ATT1700A series. In the method shown in Figure 2, the system generates an active-low configure pulse to the FPGA's $\overline{\text{PRGM}}$ pin. This configuration pulse causes the FPGA to drive its $\overline{\text{INIT}}$ pin low, which forces a RESET on the serial ROM RESET/OE pin (with the RESET/OE pin programmed for active-low RESET and active-high OE). The FPGA's DONE pin is routed to the serial ROM's $\overline{\text{CE}}$ pin. At the end of configuration, DONE returns high, disabling the serial ROM. Alternatively, the $\overline{\text{LDC}}$ pin can be used instead of the DONE pin to enable the serial ROM.

Configuring the ATT3000 Series FPGA with a Configure Command

In the method illustrated in Figure 3, the system generates an active-low configure pulse on the FPGA's DONE/$\overline{\text{PROG}}$ pin. The system then releases the open-drain DONE/$\overline{\text{PROG}}$ pin, allowing the FPGA to control it and drive it low during configuration. DONE/$\overline{\text{PROG}}$ is generally connected to both the $\overline{\text{CE}}$ and RESET/OE pins of the serial ROM, which has been programmed so that RESET is active-high and OE is active-low. At the end of configuration, the DONE/$\overline{\text{PROG}}$ pin returns high, disabling and resetting the serial ROM. The $\overline{\text{LDC}}$ pin may be used instead of the DONE/$\overline{\text{PROG}}$ pin to enable the serial ROM, as shown.

FPGA Configuration (continued)

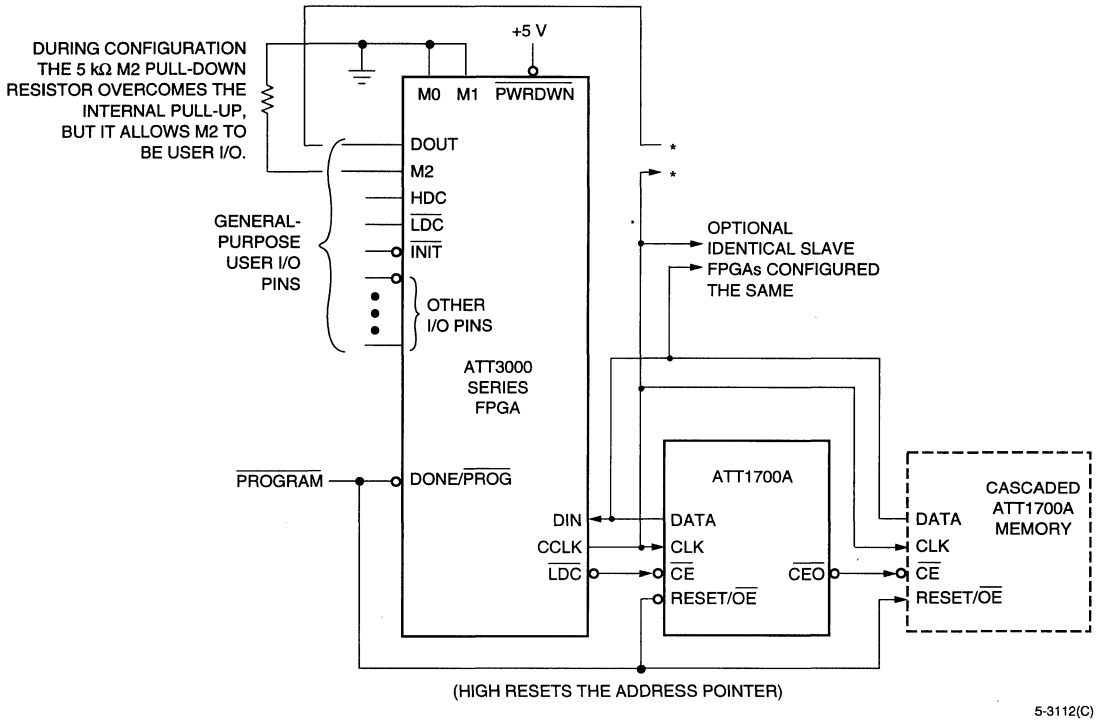


Figure 3. ATT3000 Master Serial Configuration

Programming the FPGA with the Address Pointer Unchanged Upon Completion

In the two interfaces previously discussed, the serial ROM is reset at the completion of configuration. This is typically the case when one or more serial ROMs are used to configure one or more FPGAs with one configuration program. In applications in which a serial ROM is used to configure an FPGA with multiple configuration programs, the address pointer should not be reset. This allows the next configuration program to be loaded at the next internal ROM address.

When multiple FPGA configurations are stored in a serial ROM, the OE pin of the serial ROM should be tied low. Upon powerup, the internal address pointer is reset and configuration begins with the first set of configuration data stored in memory. Since the OE pin is held low, the address pointer is left unchanged after configuration is complete. To reprogram the FPGA with another program, the DONE/$\overline{\text{PROG}}$ or $\overline{\text{PRGM}}$ pin is pulled low, and configuration begins at the last value of the address pointer.

FPGA Configuration (continued)

Cascading Serial ROMs

Figure 2 and Figure 3 also illustrate the cascading of serial ROMs. This is done to provide additional memory for large FPGAs and/or for configuring multiple FPGAs in a daisy chain. The serial ROMs are cascaded with the next ROM's $\overline{CE}$ input connected to the $\overline{CEO}$ output of the previous serial ROM. All of the cascaded serial ROM's DATA lines are routed to the FPGA's DIN input, and the FPGA's CCLK output is routed in parallel to all of the serial ROMs' CLOCK inputs.

After the last bit from the first serial ROM is read, the first serial ROM asserts $\overline{CEO}$ low and disables its DATA output. The next serial ROM recognizes the low on its $\overline{CE}$ input and enables its DATA output. The inactive $\overline{CE}$ into all serial ROMs causes the inactive DATA pins to be 3-stated after configuration is finished.

The ATT3000 DONE/$\overline{PROG}$ signal and the *ORCA* DONE signal are open-drain outputs with optional internal pull-ups and can be used to control the output enable of multiple serial ROMs. Extremely large, cascaded serial memories may require additional logic if the DONE/$\overline{PROG}$ or DONE signals are too slow to activate many serial ROMs.

Standby Mode

The ATT1700A Series enters a low-power standby mode when $\overline{CE}$ is high. In standby mode, the serial ROM consumes less than 100 µA of current. The DATA pin remains in the high-impedance state regardless of the state of the RESET/OE input.

RESET/OE Polarity

The ATT1700A Series allows the user to select the polarity of the dual-function RESET/OE pin. The PROM programmer software is used to program the desired polarity. The method used to select a polarity depends on the prom programmer user interface.

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_{DD}	–0.6	6.6	V
Programming Voltage Relative to GND	V_{PP}	–0.6	14.0	V
Input Voltage with Respect to GND	V_{IN}	–0.6	V_{DD} + 0.6	V
Voltage Applied to 3-state Output	V_{TS}	–0.6	V_{DD} + 0.6	V
Ambient Storage Temperature	T_{stg}	–65	150	°C
Maximum Soldering Temperature	T_{SOL}	—	300	°C
Maximum Junction Temperature	T_J	—	125	°C

Electrical Characteristics

Table 3. dc Electrical Characteristics

Commercial: 0 °C ≤ T_A ≤ 70 °C, V_{DD} = 5.0 V ± 5%; Industrial: –40 °C ≤ T_A ≤ +85 °C, V_{DD} = 5.0 V ± 10%.

Parameter	Symbol	Conditions	Min	Max	Unit
High-level Input Voltage	V_{IH}	—	2.0	V_{DD}	V
Low-level Input Voltage	V_{IL}	—	–0.3	0.8	V
High-level Output Voltage	V_{OH} V_{OH}	V_{DD} = 3.0 V, I_{OH} = –4.0 mA V_{DD} = 4.5 V, I_{OH} = –4.0 mA	2.40 3.86	— —	V V
Low-level Output Voltage	V_{OL}	V_{DD} = 5.5 V, I_{OL} = 4.0 mA	—	0.32	V
Supply Voltage Relative to V_{SS}: Commercial Industrial	 — —	 — —	 4.75 4.50	 5.25 5.50	 V V
Standby Supply Current	I_{DDSB}	V_{IN} = V_{DD} = 5.5 V V_{IN} = V_{DD} = 3.6 V	— —	100 50	µA µA
Operating Supply Current	I_{DD}	V_{DD} = 5.5 V, Clock = 10 MHz V_{DD} = 3.6 V, Clock = 2.5 MHz	— —	10 2	mA mA
Input Leakage Current	I_{IL}	V_{DD} = 5.5 V, V_{IN} = V_{DD} and 0 V	–10	10	µA
Output Leakage Current	I_{IL}	V_{DD} = 5.5 V, V_{IN} = V_{DD} and 0 V	–10	10	µA
Pin Capacitance	C_{IN}	V_{CC} = 5 V, T_A = 25 °C, F_{CLK} = 1 MHz	—	10	pF

Electrical Characteristics (continued)

Table 4. ac Characteristics During Read

Commercial: 0 °C ≤ TA ≤ 70 °C, VDD = 5.0 V ± 5%; Industrial: –40 °C ≤ TA ≤ +85 °C, VDD = 5.0 V ± 10%.

Parameter	Symbol	Test Conditions	Limits 3.0 V ≤ VDD ≤ 6.0 V		Limits 4.5 V ≤ VDD ≤ 6.0 V		Unit
			Min	Max	Min	Max	
OE to Data Delay	TOE	—	—	45	—	45	ns
$\overline{CE}$ to Data Delay	TCE	—	—	60	—	50	ns
CLOCK to DATA Delay	TCAC	—	—	200	—	60	ns
DATA Hold from $\overline{CE}$, OE, or CLOCK	TOH	—	0	—	0	—	ns
$\overline{CE}$ or OE to DATA Float Delay	TDF	—	—	50	—	50	ns
CLOCK Frequency	TCLK	—	—	2.5	—	10	MHz
CLOCK Low Time	TCL	—	100	—	25	—	ns
CLOCK High Time	TCH	—	100	—	25	—	ns
$\overline{CE}$ Setup Time to CLOCK (Guarantees correct counting.)	TSCE	—	40	—	25	—	ns
$\overline{CE}$ Hold Time from CLOCK (Guarantees correct counting.)	THCE	—	0	—	0	—	ns
OE High Time (Guarantees counters are reset.)	THOE	$\overline{CE}$ high or low	100	—	20	—	ns

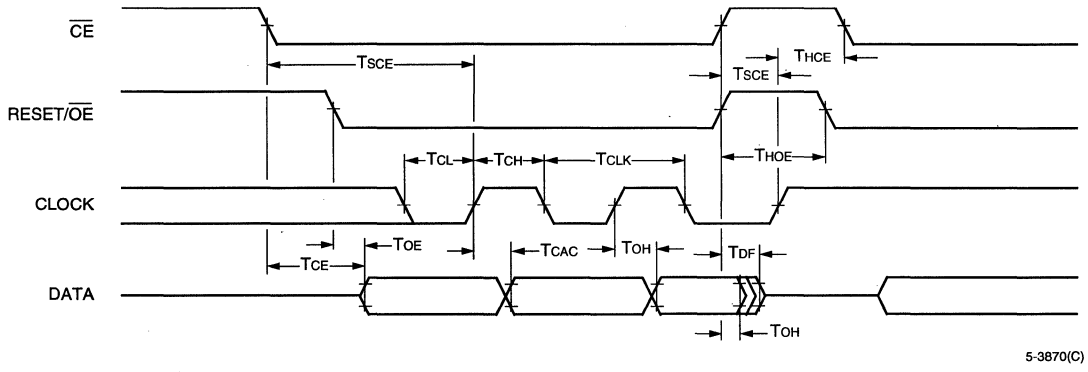


Figure 4. Read Characteristics

Electrical Characteristics (continued)

Table 5. ac Characteristics at End of Read

Commercial: 0 °C ≤ TA ≤ 70 °C, VDD = 5.0 V ± 5%; Industrial: –40 °C ≤ TA ≤ +85 °C, VDD = 5.0 V ± 10%.

Parameter	Symbol	Limits 3.0 V ≤ VDD ≤ 6.0 V		Limits 4.5 V ≤ VDD ≤ 6.0 V		Unit
		Min	Max	Min	Max	
CLOCK to DATA Disable Delay	TCDF	—	50	—	50	ns
CLOCK to $\overline{CEO}$ Delay	TOCK	—	65	—	40	ns
$\overline{CE}$ to $\overline{CEO}$ Delay	TOCE	—	45	—	40	ns
$\overline{OE}$ to $\overline{CEO}$ Delay	TOOE	—	40	—	40	ns

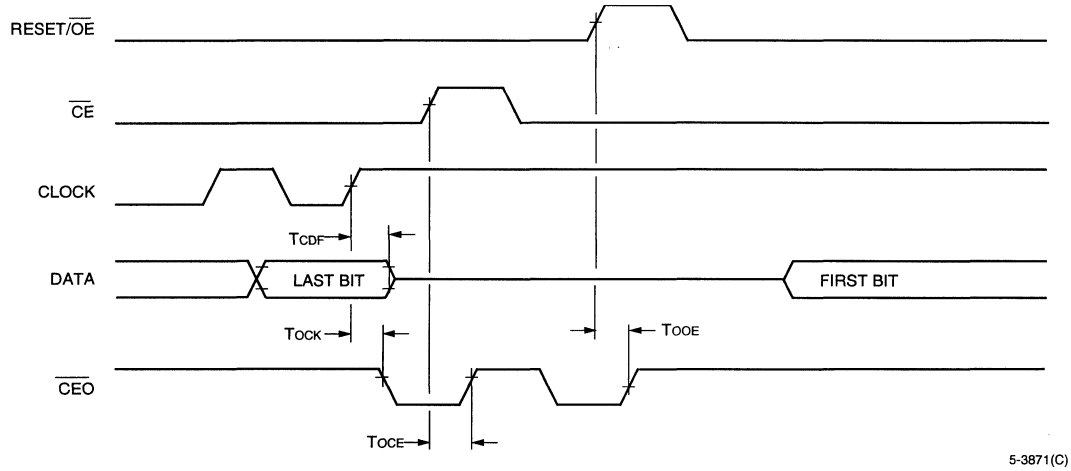


Figure 5. Read Characteristics at End of Array

Electrical Characteristics (continued)

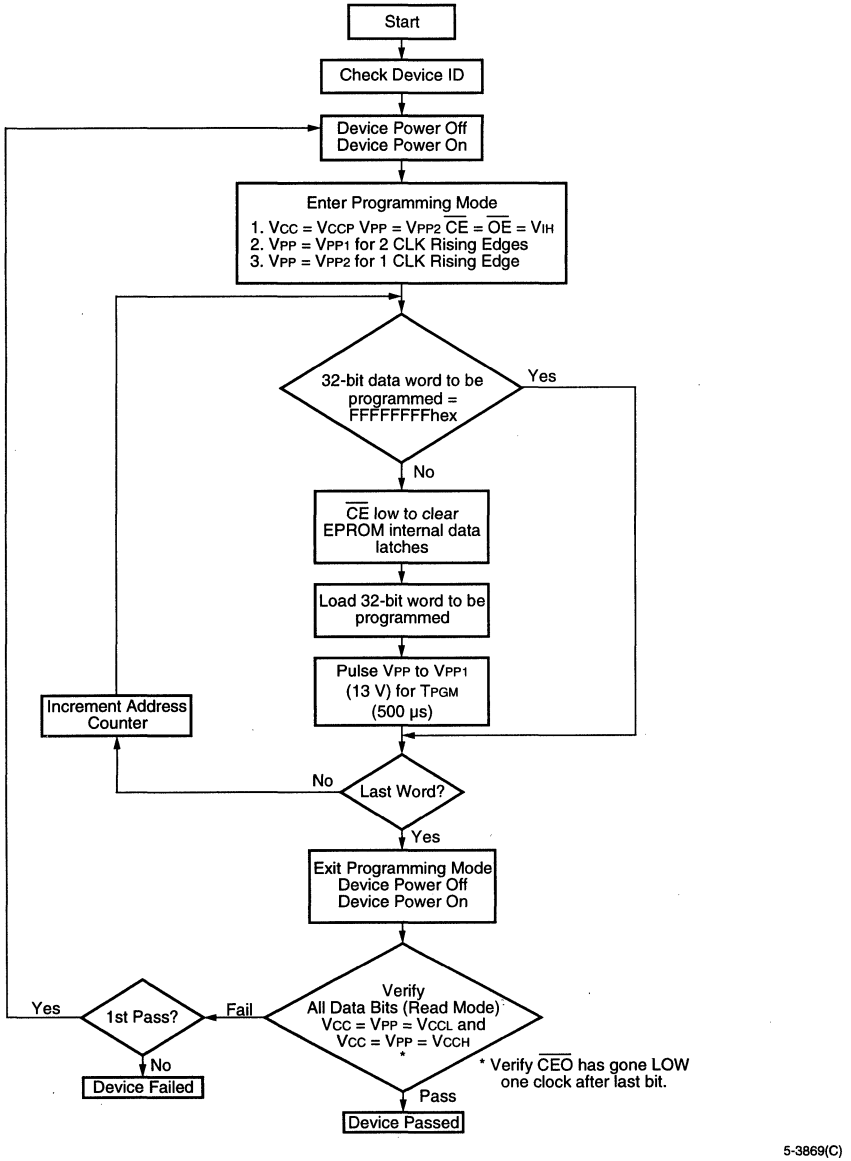


Figure 6. ATT1700A Programming

Electrical Characteristics (continued)

Table 6. dc Programming Specifications

Commercial: 0 °C ≤ T_A ≤ 70 °C, V_{DD} = 5.0 V ± 5%; Industrial: –40 °C ≤ T_A ≤ +85 °C, V_{DD} = 5.0 V ± 10%.

Parameter	Symbol	Min	Max	Unit
Supply Voltage During Programming	V_{CCP}	5.0	6.0	V
Low-level Input Voltage	V_{IL}	0.0	0.5	V
High-level Input Voltage	V_{IH}	2.4	V_{CC}	V
Low-level Output Voltage	V_{OL}	—	0.4	V
High-level Output Voltage	V_{OH}	3.7	—	V
Programming Voltage*	V_{PP1}	12.5	13.5	V
Programming Mode Access Voltage	V_{PP2}	V_{CCP}	V_{CCP} + 1	V
Supply Current in Programming Mode	I_{PPP}	—	100	mA
Input or Output Leakage Current	I_L	–10	10	µA
First-pass, Low-level Supply Voltage for Final Verification	V_{DDL}	2.8	3.0	V
Second-pass, High-level Supply Voltage for Final Verification	V_{DDH}	6.0	8.2	V

* No overshoot is permitted on this signal. V_{PP} must not be allowed to exceed 14 V.

Electrical Characteristics (continued)

Table 7. ac Programming Specifications

Commercial: 0 °C ≤ T_A ≤ 70 °C, V_{DD} = 5.0 V ± 5%; Industrial: –40 °C ≤ T_A ≤ +85 °C, V_{DD} = 5.0 V ± 10%.

Parameter	Test Conditions	Symbol	Min	Max	Unit
10% to 90% Rise Time of V_{PP}	*	T_{RPP}	1	—	µs
90% to 10% Fall Time of V_{PP}	*	T_{FPP}	1	—	µs
V_{PP} Programming Pulse Width	—	T_{PGM}	0.5	1.05	ms
V_{PP} Setup to Clock for Entering Programming Mode	*	T_{SVC}	100	—	ns
$\overline{CE}$ Setup to Clock for Entering Programming Mode	*	T_{SVCE}	100	—	ns
$\overline{OE}$ Setup to Clock for Entering Programming Mode	*	T_{SVOE}	100	—	ns
V_{PP} Hold from Clock for Entering Programming Mode	*	T_{HVC}	300	—	ns
Data Setup to Clock for Programming	—	T_{SDP}	50	—	ns
Data Hold from Clock for Programming	—	T_{HDP}	0	—	ns
$\overline{CE}$ Low Time to Clear Data Latches	—	T_{LCE}	100	—	ns
$\overline{CE}$ Setup to Clock for Programming/Verifying	—	T_{SCC}	100	—	ns
$\overline{OE}$ Setup to Clock for Incrementing Address Counter	—	T_{SIC}	100	—	ns
$\overline{OE}$ Hold from Clock for Incrementing Address Counter	—	T_{HIC}	0	—	ns
$\overline{OE}$ Hold from V_{PP}	*	T_{HOV}	200	—	ns
Clock to Data Valid	—	T_{PCAC}	—	400	ns
Data Hold from Clock	—	T_{POH}	0	—	ns
$\overline{CE}$ Low to Data Valid	—	T_{PCE}	—	250	ns

* This parameter is periodically sampled and is not 100% tested.

Note: While in programming mode, $\overline{CE}$ should only be changed while $\overline{OE}$ is HIGH and has been HIGH for 200 ns, and $\overline{OE}$ should only be changed while $\overline{CE}$ is HIGH and has been HIGH for 200 ns.

Electrical Characteristics (continued)

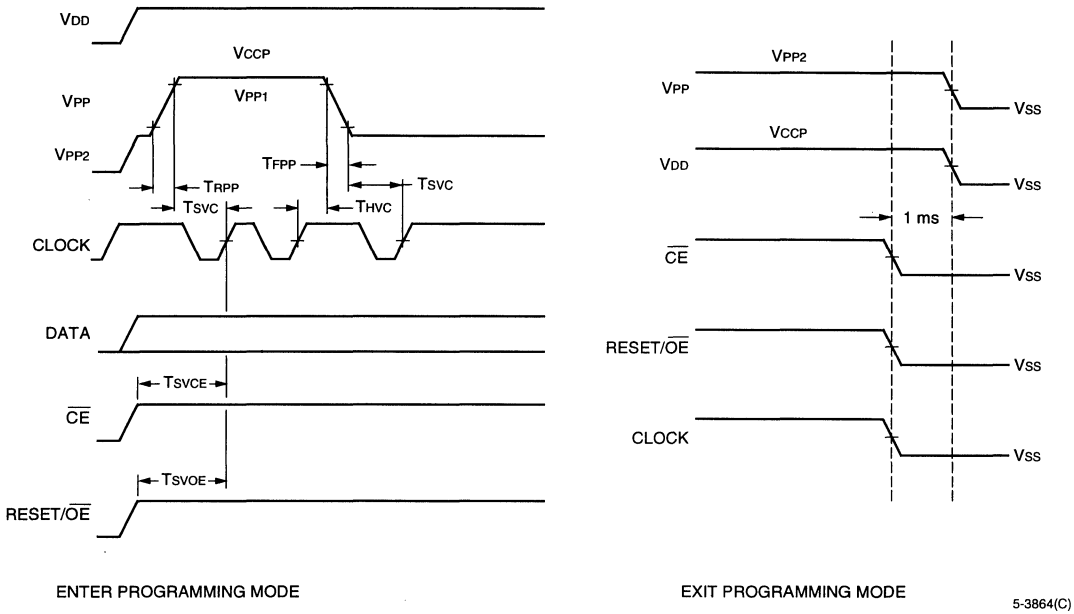
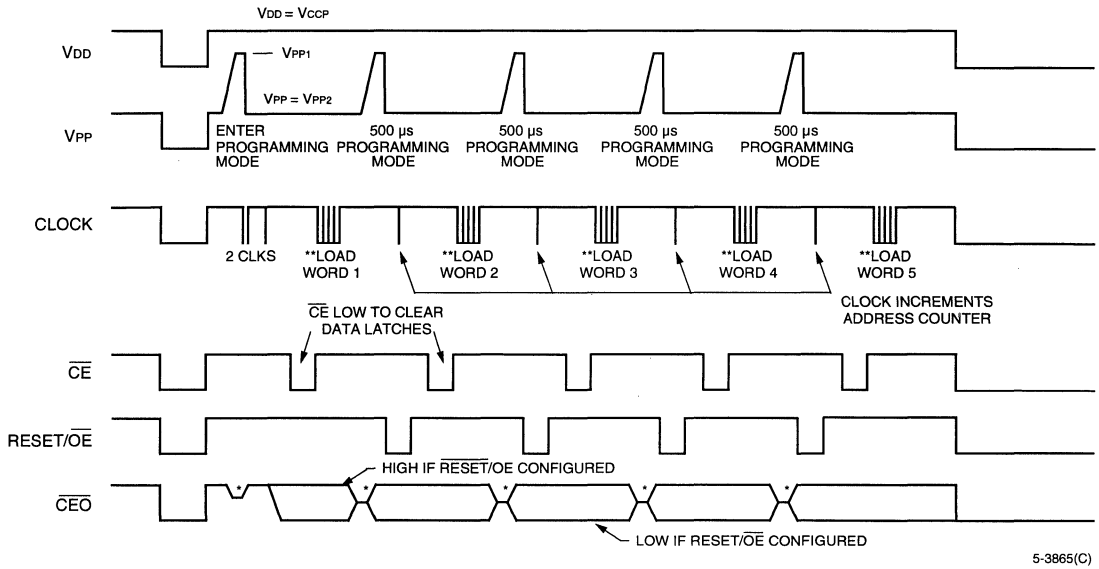


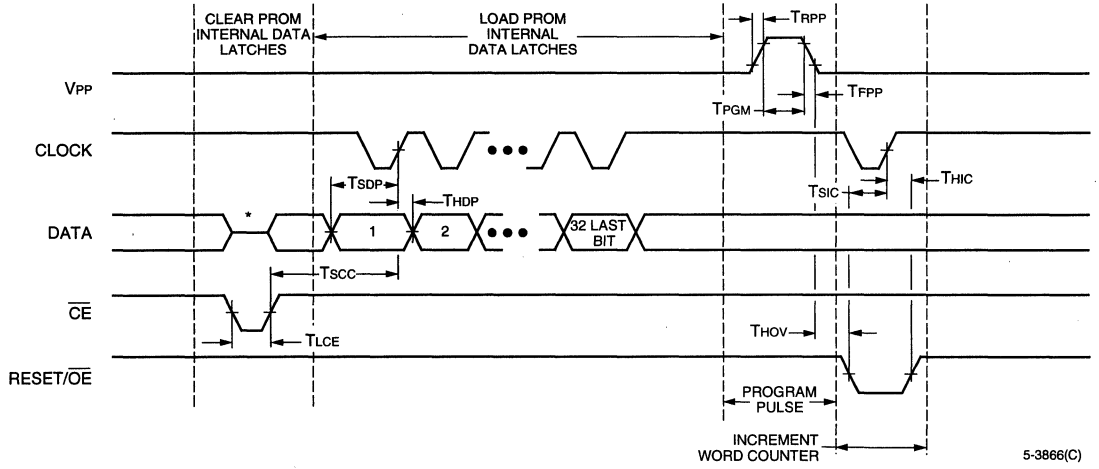
Figure 7. Entering and Exiting Programming Mode



* The $\overline{CEO}$ pin is high impedance when VPP = VPP1.
** 32 clocks.

Figure 8. Programming Cycle Overview

Electrical Characteristics (continued)



* The programmer must float the data pin while $\overline{CE}$ is low to avoid bus contention.

Figure 9. Details of Programming Cycle

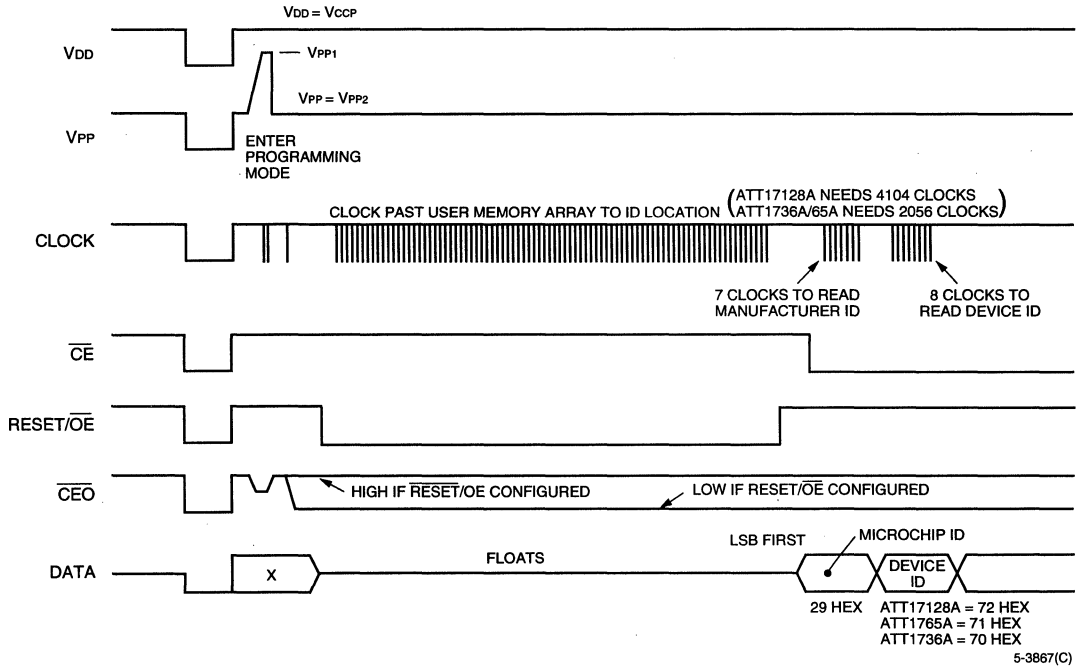


Figure 10. Read Manufacturer and Device ID Overview

Electrical Characteristics (continued)

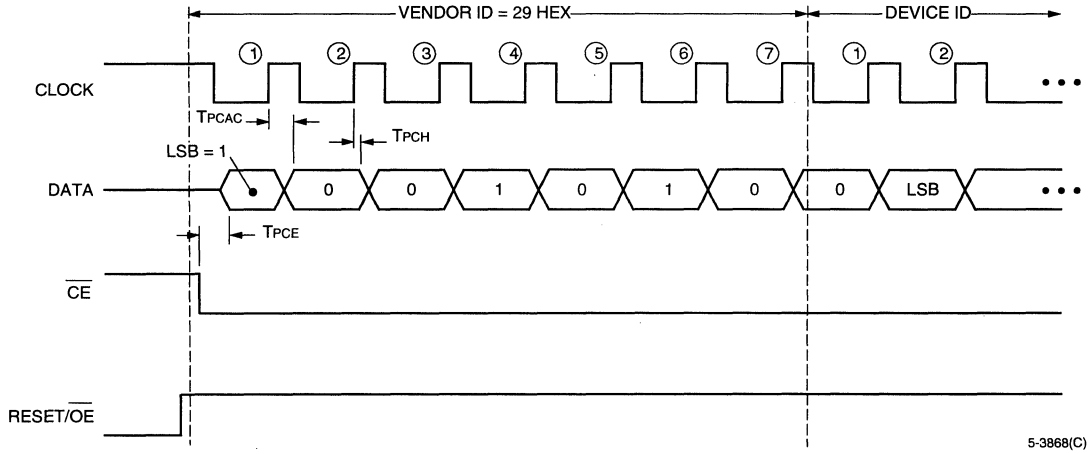
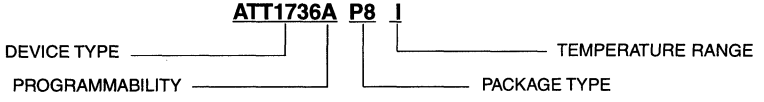


Figure 11. Details of Read Manufacturer and Device ID

Ordering Information

Example: **ATT1736A P8 I**



ATT1736A; One-Time Programmable; 8-pin, Plastic DIP; Industrial Temperature

Table 8. Device Type

Device	Size
ATT1736A	36,288
ATT1765A	65,536
ATT17128A	131,072

Table 9. Programmability

Designation	Programmability
Blank or A	One-Time Programmable

Table 10. Package Type

Designation	Package
P8	8-pin, plastic DIP
SO8	8-pin SOIC
M20	20-pin PLCC

Table 11. Temperature Range

Designation	Type	Operating Range
Blank	Commercial	0 °C to 70 °C
I	Industrial	–40 °C to +85 °C

Lucent Technologies
Bell Labs Innovations

Development Systems

Overview

Developing high-density, high-performance field-programmable gate arrays in today's competitive market requires a unique yet robust design environment. The design flow must be easy to use, compatible with the user's existing environment, and powerful enough to meet the time-to-market schedules.

Lucent Technologies supports a number of different development system products optimized for the *ORCA* and ATT3000 FPGA architectures to meet the user's design requirements. Products include interfaces for popular CAE tools, as well as a versatile map/place/route development system. Supported platforms include personal computers and *Sun* and *HP* workstations.

Development systems are only as effective as the methodology with which they are applied. Lucent has taken its vast ASIC design experience, applied that knowledge and expertise, and developed an FPGA design methodology that meets today's market demands.

Design Methodology

The design methodology that Lucent recommends consists of three steps: design entry, design implementation, and design verification (see Figure 1 below). Lucent recommends that the user follow a straightforward methodology, but recognizes that the process is wrought with change. The user will often need to move among the three design steps throughout the design cycle to correct or change a circuit. This is one of the many advantages of using an FPGA.

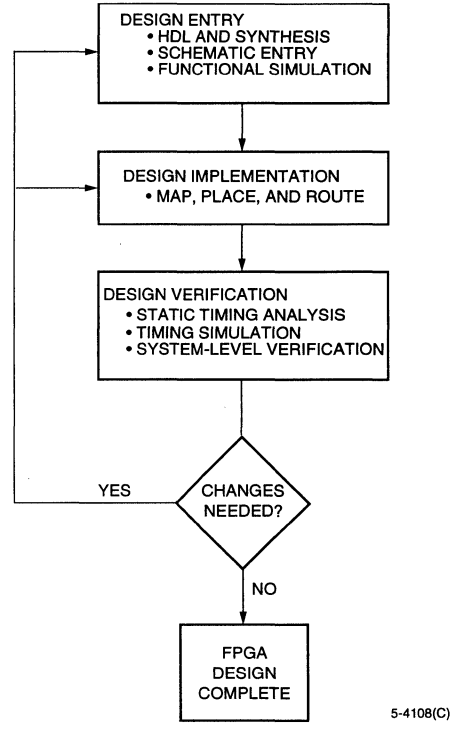


Figure 1. FPGA Design Methodology

Design Methodology (continued)

Design Entry

A design can be captured in a number of different ways. Traditional schematic capture tools remain a popular design entry method. Lucent's *ORCA* Foundry Development System supports schematic editors from *Viewlogic* and *Mentor Graphics*. The libraries for these tools contain macros and primitives that allow efficient access to the *ORCA* or ATT3000 architectures.

As designs increase in complexity, the use of hardware description languages (*Verilog* HDL or VHDL) and synthesis tools can increase productivity. *Synopsys, Exemplar Logic, Synplicity, Viewlogic,* and *Mentor Graphics* are just some of the supported CAE synthesis vendors. Libraries, design flows, and algorithms, such as Lucent's proprietary *SCUBA* (Synthesis Compiler for User-Programmable Arrays), have been developed to synthesize the Lucent FPGA architectures efficiently.

More than one of these forms of design entry can be used to capture the user's design for a single device. If more than one type is used, the top level of a hierarchical design is created using the first of any one of these design methods. This top level then calls other circuit descriptions in each succeedingly lower level of hierarchy (which have been entered by using whichever design entry method is required). In this way, the designer can use any type of design entry desired for each portion of the circuit.

Lucent recommends that chip and/or board functional simulation be performed before proceeding to the Design Implementation stage. This step will identify many problems that may be difficult to solve during the Verification step. Simulators from *Cadence, Mentor Graphics,* Synopsys, *Viewlogic,* and *Model Technology* are supported by *ORCA* Foundry. In addition, the Logic Modeling Group of Synopsys, Inc. supports board-level simulation models for *ORCA* FPGAs from Lucent.

Design Implementation

After design entry is complete, the design must be implemented into a selected architecture. This generally requires that the three major steps of mapping, placing, and routing the design be complete. The first step in design implementation is to map the circuit created during design entry into the distinct logic blocks of the target FPGA architecture. Once this has been accomplished, the next step is to place these logic blocks into specific positions in the target device which has a repeating array of these logic blocks. The third step is to route the signals that connect the logic blocks together.

The *ORCA* Foundry Development System is the tool that is used to perform these three functions, as well as static timing analysis and generating the bit stream used to configure the devices. This tool has a menu-driven interface and allows the process to be automated. With *ORCA* Foundry, the designer can specify many parameters, such as clock frequency, I/O delays, signal skews, and so on, in what is called a preference file. The software then uses this information and attempts to create a circuit that meets the required specifications automatically. Although this generally meets the designer's needs, the ability to manually change the mapping, placement, or routing through a graphical interface is also included in the software.

Once the design has been implemented, a bit stream file that contains the programming information for the RAM cells in the FPGA can be created. This bit stream can then be downloaded into the FPGA using any of the configuration modes explained in the ATT3000 and *ORCA* data sheets. Once downloaded, it causes the FPGA to perform the desired function.

Design Methodology (continued)

Design Verification

Verification of the FPGA design can be performed using one or more of the following three methods:

- Postlayout timing simulation
- Postlayout static timing analysis
- Using the device in the target system

Postlayout timing simulation can be done using any of the simulators that were used for functional simulation. An interface is provided from the *ORCA* Foundry Development System to these third-party simulators to allow timing to be transferred to the simulator after the map, place, and route steps have been completed. This timing simulation generally uses the same input stimulus as the functional simulation and is used to find such problems as implementations that are too slow, have race conditions, or have setup/hold time violations.

Postlayout static timing analysis is supported in *ORCA* Foundry using the *TRACE* tool, as well as other third-party tools, such as *MOTIVE* or *Veritime*. These tools do not require input stimulus but allow the designer to evaluate the timing characteristics of the implemented device, including such things as maximum clock frequency.

The third method for design verification is to test the FPGA in the target systems; however, Lucent recommends that the designer use at least one of the other methods noted above to fully verify the design. This is due to the fact that the device being used will probably be faster than the worst-case specification guaranteed by Lucent due to process variation. This process variation occurs in any semiconductor manufacturing process, but it is guaranteed by Lucent to be within a certain range. This range defines the timing that is used for both timing simulation and static timing analysis.

Individual Design Tools Overview

The remainder of this development systems section contains overviews, a features list, platforms supported, and ordering information for each of the following design tools:

- *ORCA* Foundry Development System
- *ORCA* FPGA Library for *Synopsys*
- *ORCA* FPGA Library for *Viewlogic*
- *ORCA* FPGA Library for *Mentor Graphics*
- *ORCA* FPGA Library for Bell Labs Design Automation
- *ORCA* FPGA Library for *Verilog* HDL Simulation
- *ORCA* FPGA Library for VHDL Simulation
- *Exemplar Logic's Galileo* and *Model Technology's V-System/VHDL*: HDL-Based Synthesis and Simulation for *ORCA* FPGAs
- *WorkView Office* from Viewlogic Systems, Inc.: Integrated Schematics, Synthesis, and Simulation for *ORCA* FPGAs

ORCA Foundry Development System

Features

- Complete, fully integrated FPGA design tool set
- Supports *ORCA* and ATT3000 Series FPGAs
- Integrates into existing CAE environments
- Supports timing- and frequency-driven design
- Performs device-specific optimization and technology mapping
- Performs both automatic and manual place & route
- *SCUBA* macro compiler generates *ORCA*-specific implementations of macrocells
- Performs detailed static timing analysis
- Allows for back-annotated timing simulation and system-level verification

ORCA Foundry Benefits

- Automatic completion of difficult designs
- Maximum device utilization
- Faster clock speeds
- Ease of use means fast time-to-market benefits

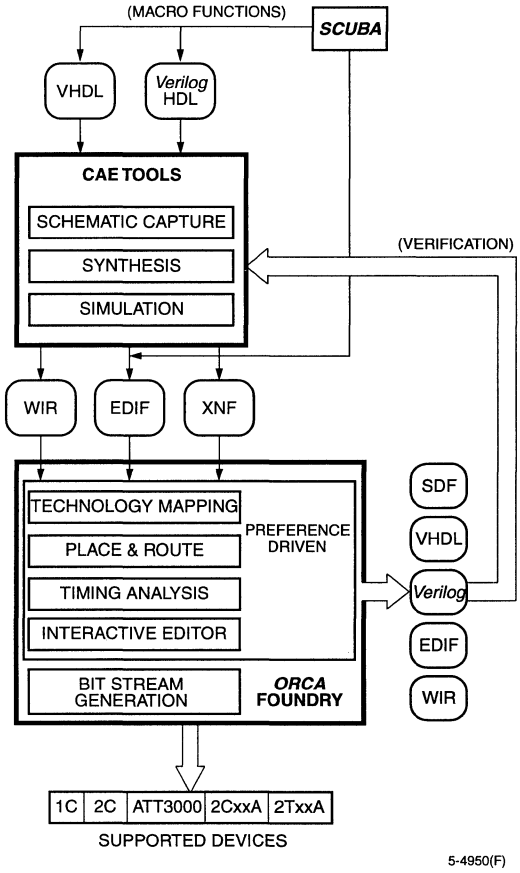


Figure 1. *ORCA* Foundry Environment

Smarter, Faster Tools

FPGA devices are growing in size and complexity—straining the capabilities of both designers and early-generation tool sets. High-performance tools are critical to realizing the full potential of today's larger, more complicated devices. Such tools not only significantly shorten your design cycles, but also produce chip designs with higher device utilization and faster operating frequencies. *ORCA* Foundry is such a tool set.

Capture, Mapping, and Optimization

ORCA Foundry allows designs to be captured using device-specific libraries, vendor-independent libraries, or a combination of both. No other design tool set lets you designate the specific design capture method that best supports your requirements. As a result, vendor-independent libraries and industry-standard netlists can be easily implemented in either *ORCA* or ATT3000 Series devices.

ORCA Foundry's device- and architecture-specific optimization, combined with superior place and route capabilities, produces consistently high device utilization. Of course, *ORCA* Foundry fully supports device-specific features, such as hard macros, RAM, and automatic routing of clocks.

With complete back-annotation and the ability to preserve hierarchy throughout the design process, *ORCA* Foundry gives you as much help in updating and debugging your design as it does in implementing it.

Advanced Place & Route Capabilities (PAR)

Using the most powerful combination of algorithms available, *ORCA* Foundry's place & route (PAR) program consistently completes designs with the fewest iterations and with no manual intervention. PAR's fast execution time and built-in incremental change capability result in the shortest possible design cycle.

PAR is a timing-driven place & route tool, which means users can specify critical timing and PAR will work towards desired goals, resulting in higher performance. Lucent's Preference Language is a robust language that allows designers to specify timing preferences like clock frequency, net and path delays, multicycle paths, setup requirements, and clock-to-output constraints.

Powerful Interactive Layout Editor (Lucent *EPIC*)

The Lucent Technologies Editor for Programmable ICs (Lucent *EPIC*) is a powerful, interactive layout editor found in *ORCA* Foundry that streamlines the debugging and tuning of FPGA designs. Lucent *EPIC*'s easy-to-use graphical interface provides a choice of push button, menu-driven, or command-line editing capabilities that can be customized to suit any set of requirements. In addition, Lucent *EPIC* has been tuned to guarantee the fastest graphics response, eliminating the unproductive waiting while a large design is panning, zooming, or simply highlighting a net.

Many advanced features have been designed into Lucent *EPIC* to make working with complex devices easier. Among these are manual placement and routing, auto placement, auto routing, and integration of *ORCA* Foundry's powerful timing analyzer. Lucent *EPIC*'s on-line design rule checks (DRC) can be used in logical mode (allowing changes to placement and routing, but preventing any changes to the logic during the editing session) or in physical mode (allowing logic and signals to be added and deleted while guaranteeing that changes are valid within the physical constraints of the specified FPGA).

FPGA-Specific Timing Analyzer (Lucent *TRACE*)

Lucent *TRACE* provides complete analysis of a circuit's timing characteristics. Using actual component and interconnect delays, Lucent *TRACE* exhaustively examines every signal path and automatically evaluates the circuit for setup and hold violations and adherence to specified timing preferences.

Lucent *TRACE* runs its analysis using user-specified timing preferences (such as desired operating frequency) and feeds back detailed results that identify specifically where the design fails to meet those requirements, thereby eliminating the need to read through reams of paper to pinpoint potential timing problems.

SCUBA

SCUBA (Synthesis Compiler for User-programmable Arrays) is a design tool for creating high-density, high-performance macrocells optimized for *ORCA* FPGAs. *SCUBA* will generated VHDL and *Verilog* HDL for support of synthesis and simulation tools or EDIF for input to *ORCA* Foundry.

SCUBA will generate the following macrocells in parameterizable sizes:

- Memory (asynchronous, synchronous, and dual-port)
- Adders
- Subtractors
- Adder/subtractors
- Comparators
- Multipliers

System Requirements

PC-Based:

Personal Computer with *Intel* 486, *Pentium*, or *Pentium-Pro*

Microsoft Windows 3.1 (or higher), *Windows* '95, or *Windows* NT

RAM: 32 Mbytes minimum

Hard Disk: 70 Mbytes spare capacity

Installation Media: 2x CD-ROM

Workstation-Based:

SPARC-based *Sun Workstation* running *SunOS* 4.1.3 or running *Solaris* 5.4 (or higher)

HP 9000 Series 400/700 running *HP*-UX 9.0.x (or higher)

X-Windows version X11R4 (or higher) and *OSF/MOTIF* 1.1

RAM: 32 Mbytes

Hard Disk: 70 Mbytes spare capacity

Swap: 100 Mbytes minimum

Ordering Information

PC Solutions:

- Low-density starter system for Lucent FPGAs. Supports ATT3000, ATT2Cxx, OR2CxxA, and OR2TxxA series devices to 10,000 gates. (OR-ORCAVISTA-PC).
- Complete support package for **all** ATT3000 and **all** *ORCA* Series FPGAs. (OR-ORCAAPEX-PC).
- Evaluation version of complete software suite. No bit stream generation capability or download cable (OR-ORCAEVAL-PC).
- Upgrade Vista package to Apex package (OR-ORCAVISAP-PC).

Workstation Solutions:

- Low-density starter system for Lucent FPGAs. Supports ATT3000, ATT2Cxx, OR2CxxA, and OR2TxxA series devices to 10,000 gates (OR-ORCAVISTA-WS).
- Complete support package for **all** ATT3000 and **all** *ORCA* Series FPGAs (OR-ORCAAPEX-WS).
- Evaluation version of complete software suite. No bit stream generation capability or download cable (OR-ORCAEVAL-WS).
- Upgrade Vista package to Apex package (OR-ORCAVISAP-WS).

Notes

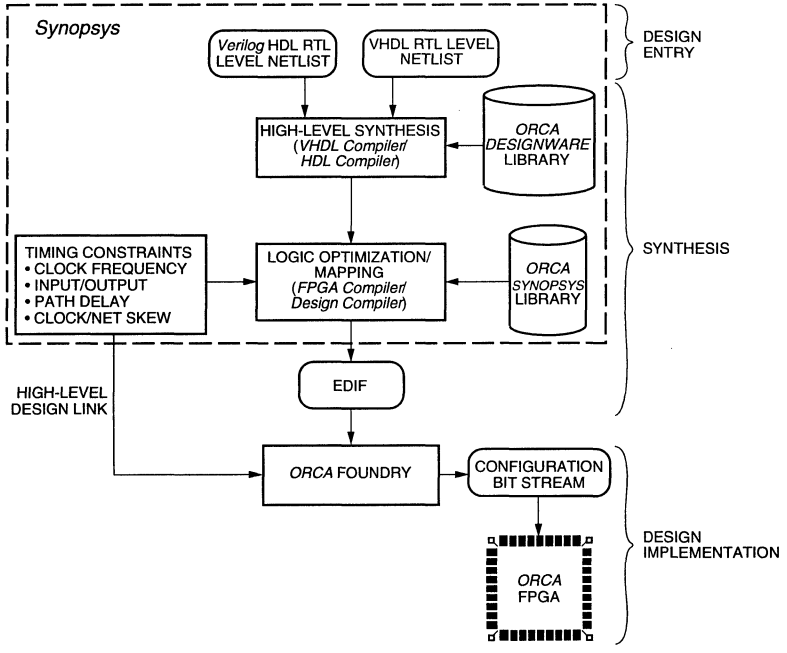
Lucent Technologies
Bell Labs Innovations

ORCA FPGA Library for *Synopsys*

Features

- Performs area/timing synthesis to take advantage of the ATT2C, OR2CxxA, and OR2TxxA series of Optimized Reconfigurable Cell Array (*ORCA*) FPGAs from Lucent
- Allows top-down design, with the input being either VHDL or *Verilog* HDL
- Support for both *FPGA Compiler* and *Design Compiler*
- Interfaces to the *ORCA* Foundry Development System, including a high-level design link to transfer timing information
- Supports *Synopsys* FPGA Compiler LUT library to allow synthesis directly to *ORCA* look-up tables (LUTs) to increase area efficiency and performance estimation accuracy
- Supports *DesignWare* data path synthesis for adders/subtracters and comparators

Design Flow



Synopsys Interface Kit

The *ORCA* FPGA Library for *Synopsys* supports customers performing high-level synthesis, logic optimization, and timing optimization using the *Synopsys* FPGA Compiler and *Design Compiler* synthesis products. The input into *Synopsys* is either a VHDL or a *Verilog* HDL device development netlist, and the output is a netlist optimized to be used with *ORCA* Series FPGAs. When combined with the *ORCA* Foundry Development System to map, place, and route *ORCA* FPGAs, the result is a fully integrated system for implementing *ORCA* designs from concept to fully programmed devices.

The library contains elements that support both the 1C and 2C Series of *ORCA* FPGAs and are synthesized by *Synopsys*. With *Synopsys'* FPGA Compiler, users can synthesize directly to *ORCA* LUTs to improve area efficiency and timing accuracy. In addition, *Synopsys' DesignWare* library is used for inferred data path synthesis, with the output being a netlist with elements of the needed bit-length included.

Synopsys to *ORCA* Foundry High-Level Design Link

Synopsys can perform area/timing-optimized synthesis targeting *ORCA*. To support this, all *ORCA* elements from the gate library, LUT library, and *DesignWare* library now include area and delay estimation values. The designer can input timing constraints such as clock frequency, input/output delays, path delays, and skew values into *Synopsys*, which *Synopsys* then uses to perform timing-optimized synthesis. *ORCA* Foundry then accepts timing constraints from *Synopsys* and produces a timing- and frequency-driven layout.

Ordering Information

The *ORCA* FPGA Library for *Synopsys* is included with the *ORCA* Foundry Development System:

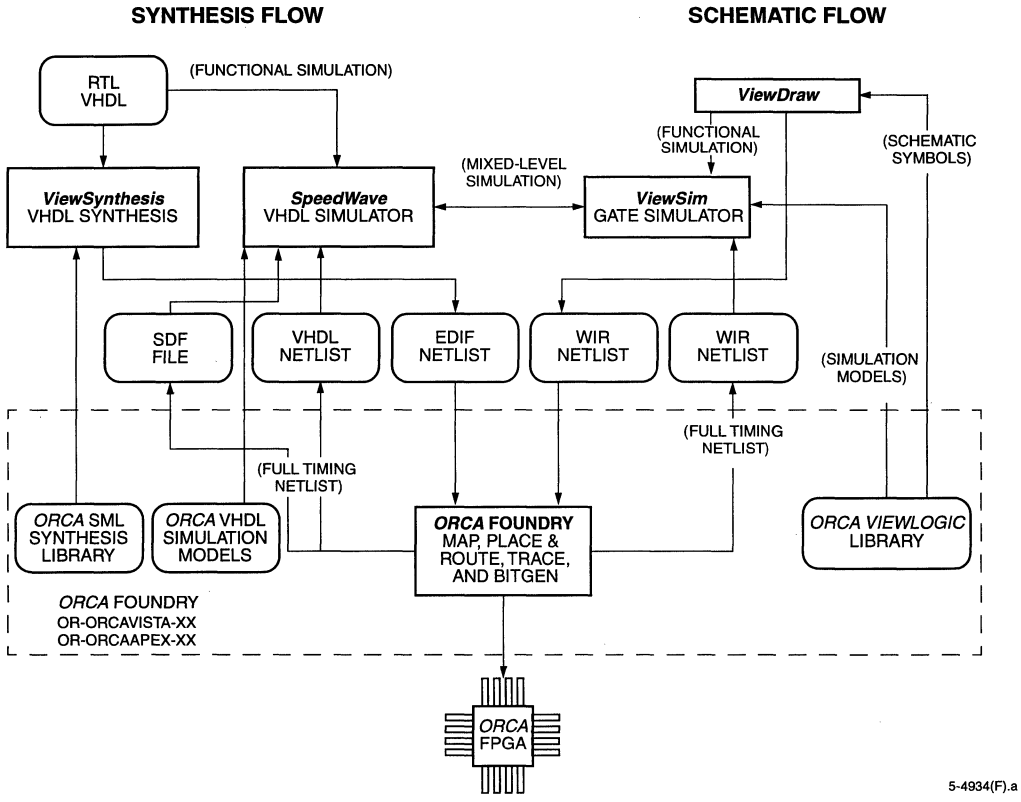
- The VISTA configuration is a complete design system supporting the ATT3000 and *ORCA* 2Cxx, OR2CxxA, and OR2TxxA devices up through 10,000 gates (OR-ORCAVISTA-WS).
- The APEX configuration supports **all** Lucent FPGAs (OR-ORCAAPEX-WS).

Lucent Technologies
Bell Labs Innovations

ORCA FPGA Library for *Viewlogic*

Features

- Supports *WorkView Office* (PC) and *PowerView* (*Sun*, *HP*) schematic capture tools from *Viewlogic*
- Supports *ViewSynthesis* synthesis tool from *Viewlogic*
- Includes support for *ViewSim* (PC, *Sun*, *HP*) and *Speedwave* (PC, *HP*, *Sun*) simulation environments from *Viewlogic*
- Seamless interface to the ORCA Foundry Development System for both schematic entry synthesis
- Library contains over 275 elements, including elements optimized for combinatorial logic, sequential logic, I/O, internal RAM, and data path circuits



Key: XX = PC (Personal Computer) or WS (Workstation). See *ORCA* Foundry Ordering Information.

Figure 1. *WorkView Office* Flow

Viewlogic Interface Kit

The Lucent Technologies Microelectronics Group *ORCA* FPGA Library for *Viewlogic* supports customers who perform schematic capture and simulation of *ORCA* Series FPGAs using *Powerview* and *WorkView Office*. When combined with the *ORCA* Foundry Development System used to map and place and route *ORCA* FPGAs, the result is a fully integrated system for implementing Lucent Technologies *ORCA* designs from concept to fully programmed devices.

The library contains over 275 elements supporting the *ORCA* 1C, 2C, 2CxxA, and 2TxxA Series of devices. This library contains many elements that take advantage of the architectural features of the *ORCA* Series, including the following:

- Sequential cells taking advantage of the many FF/latch options such as cken, front-end select, and synchronous/asynchronous reset/clear
- Combinatorial elements including optimized *ORCA*-unique PFU gates
- I/O elements
- Data path elements including counters, adders, etc.
- RAM/ROM elements for use with internal RAM capabilities

Libraries

Lucent ships several libraries for use with VL products. A symbol library is available for use with *ViewDraw*. An SML library is used with *ViewSynthesis* to synthesize *ORCA* FPGA designs from a VHDL circuit description. A simulation library is available for unit-gate or full-timing simulation with *ViewSim*. Finally, the *ORCA* Library for VHDL Simulation is available to allow designers functional and full timing simulation capability with *Speedwave*. For more information, please see the ORCA *Library for VHDL Simulation* Product Brief (PN96-058FPGA).

EDIF and WIR Netlists

The *Viewlogic* interface supports both EDIF and WIR netlist formats from *Viewlogic*'s front-end tools. *ViewSynthesis* generates EDIF for input to *ORCA* Foundry using **edif2ngd**, and *ViewDraw* generates WIR for input to *ORCA* Foundry using **wir2ngd**.

Timing Simulation Netlists

After a design has been placed and routed, users can perform full timing simulation with *Viewlogic*'s *ViewSim* and/or *Speedwave* simulators. *ORCA* Foundry's back-annotation interfaces produce both WIR and VHDL netlists for simulation with a *ViewSim* or *Speedwave*.

Ordering Information

The *ORCA* FPGA Library for *Viewlogic* is included with the *ORCA* Foundry Development System:

- The VISTA configuration is a complete design system supporting the ATT3000, *ORCA* 2Cxx, *ORCA* 2CxxA, and *ORCA* OR2TxxA devices up through 10,000 gates (OR-ORCAVISTA-WS).
- The APEX configuration supports **all** Lucent FPGAs (OR-ORCAAPEX-WS).

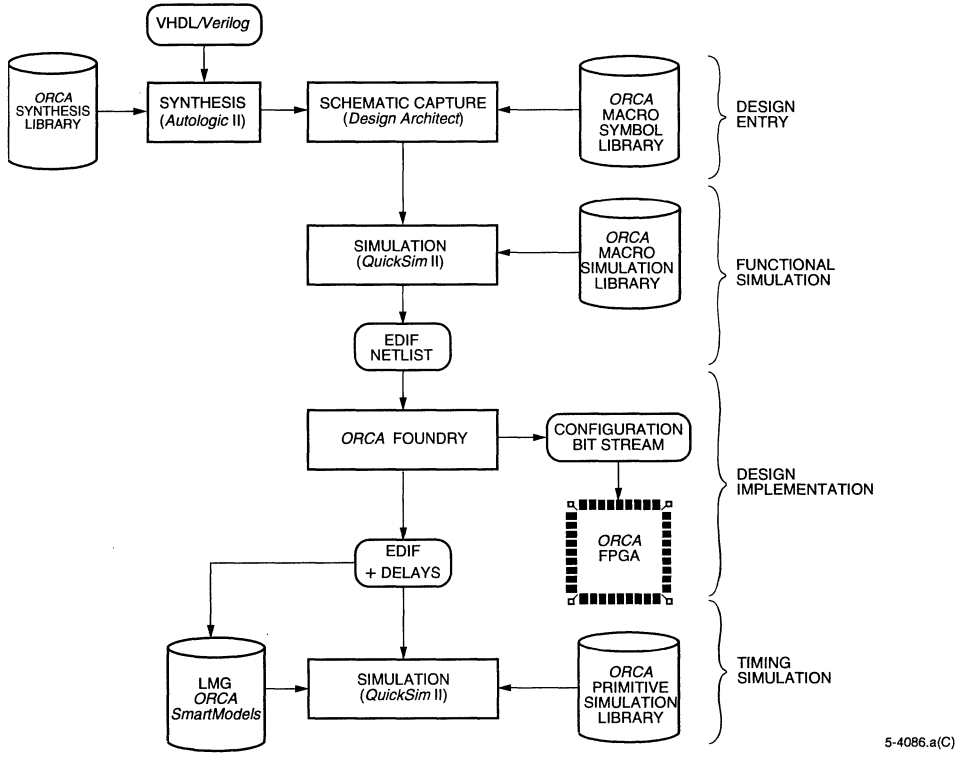
Lucent Technologies
Bell Labs Innovations

ORCA FPGA Library for *Mentor Graphics*

Features

- Library optimized to take advantage of all the Optimized Reconfigurable Cell Array (*ORCA*) FPGA families from Lucent Technologies
- Includes support for *Design Architect* schematic entry tools and the *QuickSim* II Simulation environment
- Lucent customized cell library menus for *Design Architect* to increase ease of use
- Interfaces to the *ORCA* Foundry Development System for both schematic entry and timing simulation
- Library contains over 275 elements, including elements optimized for combinatorial logic, sequential logic, I/O, internal RAM, and data path circuits
- On-line documentation of library elements
- Supports *Autologic* II synthesis

Design Flow



Mentor Graphics Interface Kit

The Lucent Technologies Microelectronics Group *ORCA* FPGA Library for *Mentor Graphics* supports customers who perform schematic capture, synthesis, and simulation of *ORCA* Series FPGAs using *Design Architect*, *Autologic* II, and *QuickSim* II. When combined with the *ORCA* Foundry Development System used to map, place, and route *ORCA* FPGAs, the result is a fully integrated system for implementing Lucent *ORCA* designs from concept to fully programmed devices.

The *ORCA* Foundry library contains over 275 elements supporting the *ORCA* 1C, 2C, 2CxxA, and 2TxxA Series of devices. This library contains many elements that take advantage of the architectural features of the *ORCA* Series including the following:

- Sequential cells taking advantage of the many FF/latch options such as cken, front-end select, and synchronous/asynchronous reset/clear
- Combinational elements including optimized *ORCA*-unique pfugate elements
- I/O elements
- Data path elements including counters, adders, etc.
- RAM/ROM elements for use with internal RAM capabilities

Mentor Graphics Schematic Capture and Simulation

Mentor Graphics' Design Architect provides a single graphical point of entry for any design targeted to the *ORCA* series. These schematic entry tools can be used with the *QuickSim* II simulator both to verify the design before proceeding to physical layout and to verify the timing after physical layout. An *ORCA SmartModel* from the Logic Modeling Group of Synopsys, Inc. can also have timing information back-annotated to it to further enhance *QuickSim* II timing simulation, if desired, especially for board-level simulation.

Ordering Information

The *ORCA* FPGA Library for *Mentor Graphics* is included with the *ORCA* Foundry Development System:

- The VISTA configuration is a complete design system supporting the ATT3000 and *ORCA* 2Cxx, OR2CxxA, and OR2TxxA devices up through 10,000 gates (OR-ORCAVISTA-WS).
- The APEX configuration supports **all** Lucent FPGAs (OR-ORCAAPEX-WS).

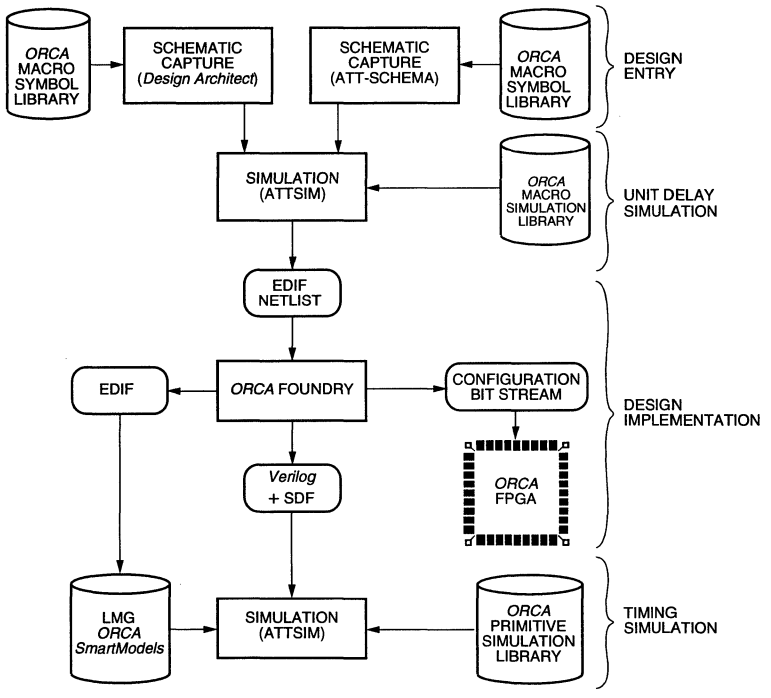
Lucent Technologies
Bell Labs Innovations

ORCA FPGA Library for Bell Labs Design Automation

Features

- Library optimized to take advantage of all the Optimized Reconfigurable Cell Array (*ORCA*) FPGA families from Lucent Technologies.
- Includes support for the ATTSIM simulation environment.
- *Mentor Graphics' Design Architect* schematic entry tools to ATTSIM interface also supported (requires *ORCA* FPGA Library for *Mentor Graphics*).
- Supports ATT-Schema schematic entry tools.
- Interfaces to the *ORCA* Foundry Development System.
- Library contains over 275 elements, including elements optimized for combinatorial logic, sequential logic, I/O, internal RAM, and data path circuits.

Design Flow



5-4089.a-R3 (C)

Bell Labs Design Automation Interface Kit

The *ORCA* FPGA Library for Bell Labs Design Automation supports customers who perform schematic capture and simulation of *ORCA* Series FPGAs using ATT-Schema and ATTSIM. Design entry using *Mentor Graphics* tools while using ATTSIM for simulation is accomplished by using both this interface kit and the *ORCA* FPGA Library for *Mentor Graphics* kit together. When combined with the *ORCA* Foundry Development System to map/place/route *ORCA* FPGAs, the result is a fully integrated system for implementing Lucent *ORCA* designs from concept to fully programmed devices.

3

The library contains over 275 elements supporting the *ORCA* 1C, 2C, 2CxxA, and 2TxxA Series of devices. This library contains many elements that take advantage of the architectural features of the *ORCA* Series, including the following:

- Sequential cells taking advantage of the many FF/latch options, such as cken, front-end select, and synchronous/asynchronous reset/clear
- Combinatorial elements, including optimized *ORCA*-unique pfugate elements
- I/O elements
- Data path elements, including counters, adders, etc.
- RAM/ROM elements for use with internal RAM capabilities

ATT-Schema Schematic Capture and ATTSIM Simulation

ATT-Schema provides a single graphical point of entry for any design targeted to the *ORCA* series. These schematic entry tools can be used with the ATTSIM simulator both to verify the design before proceeding to physical layout and to verify the timing after physical layout. An *ORCA SmartModel* from the Logic Modeling Group of Synopsys, Inc. can also have timing information back-annotated to it to further enhance ATTSIM timing simulation, if desired, especially for board-level simulation.

Ordering Information

The *ORCA* FPGA Library for Bell Labs Design Automation is included with the *ORCA* Foundry Development System:

- The VISTA configuration is a complete design system supporting the ATT3000 and *ORCA* 2Cxx, OR2CxxA, and OR2TxxA devices up through 10,000 gates (OR-ORCAVISTA-WS).
- The APEX configuration supports **all** Lucent FPGAs (OR-ORCAAPEX-WS).

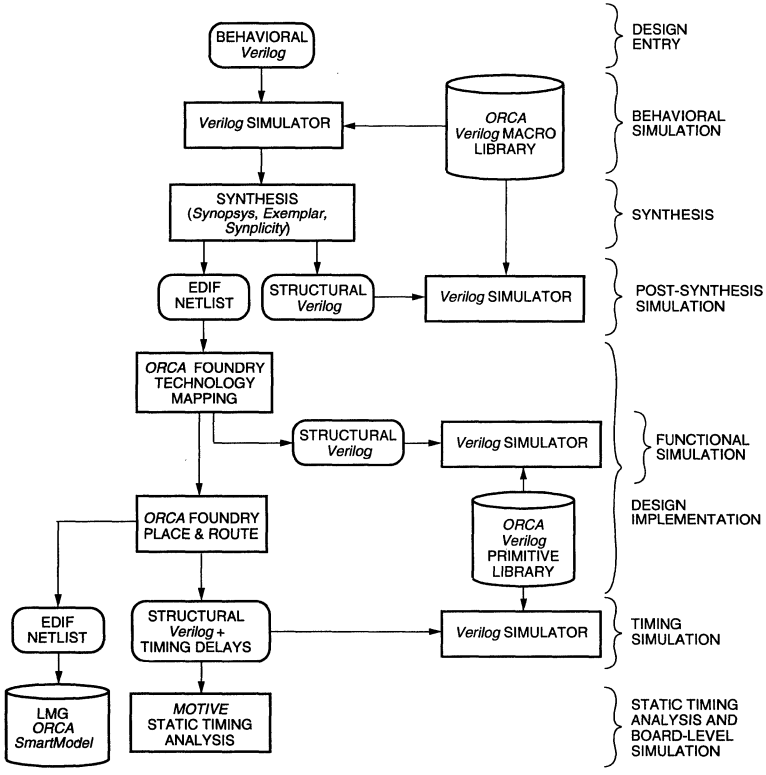
Lucent Technologies
Bell Labs Innovations

ORCA FPGA Library for *Verilog* Simulation

Features

- Allows simulation of *Verilog* HDL with Lucent Technologies' *ORCA* Series FPGAs pre- and postsynthesis, as well as after map/place/route.
- Interfaces industry-standard synthesis and simulation products to the *ORCA* Foundry Development System.
- Fully compatible with Cadence Design System's current version of *Verilog-XL*.
- Static timing analysis supported using *MOTIVE* also available when the libraries are purchased from the Quad Design Technologies Group of Viewlogic Systems, Inc.
- System verification supported using the *ORCA SmartModel* available from Synopsys, Inc.

Design Flow



5-4092.b(C)

Verilog Interface Kit

The Lucent Technologies *ORCA* FPGA Library for *Verilog* supports customers who are designing with the *Verilog* hardware description language. The combination of industry-standard synthesis tools and the *ORCA* Foundry Development System provides the designer with a fully integrated high-level design system for implementing Lucent Technologies *ORCA* FPGAs.

The *ORCA* Library for *Verilog* is compatible with the Lucent Technologies *ORCA* design kits for both *Synopsys* and *Exemplar Logic*, enabling complementary and powerful high-level design and synthesis techniques to be used. The library is designed to be used with the *Verilog-XL* simulator for design verification.

3

Simulation Capabilities

The *ORCA* Library for *Verilog* includes simulation models for the *ORCA* macro library which contains over 300 high-level macros, such as basic logic gates, ALUs, counters, and RAM. These models can be used to verify designs pre- and postsynthesis. An additional set of simulation models is supplied for full timing verification after map/place/route. This set of models supports the *Verilog* Integration Kit, which is part of the *ORCA* Foundry Development System. There are approximately 30 simulation primitives specifically designed to allow timing annotation of logic and routing delays using the standard delay format (SDF).

Postlayout analysis is supported for the *MOTIVE* static timing analysis tool from Viewlogic, Inc. An *ORCA SmartModel* from the Logic Modeling Group of Synopsys, Inc. can also have timing information back-annotated to it to further enhance timing simulation, if desired, especially for board-level simulation.

Ordering Information

The *ORCA* FPGA Library for *Mentor Graphics* is included with the *ORCA* Foundry Development System:

- The VISTA configuration is a complete design system supporting the ATT3000 and *ORCA* 2Cxx, OR2CxxA, and OR2TxxA devices up through 10,000 gates (OR-ORCAVISTA-WS for workstations, OR-ORCAVISTA-PC for personal computers).
- The APEX configuration supports **all** Lucent FPGAs (OR-ORCAAPEX-WS for workstations, OR-ORCAVISTA-PC for personal computers).

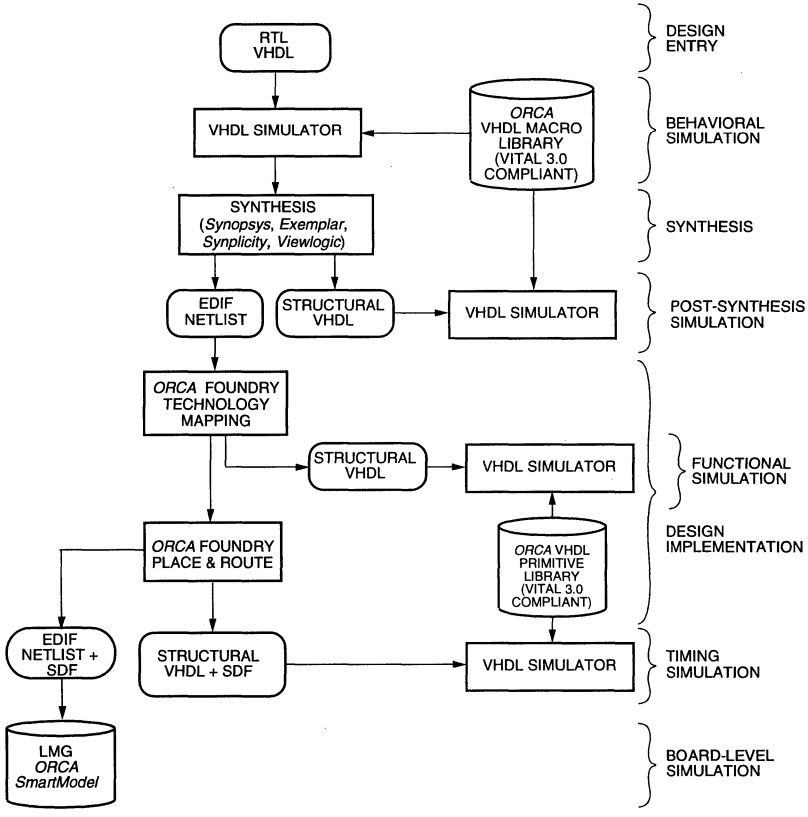
Lucent Technologies
Bell Labs Innovations

ORCA FPGA Library for VHDL Simulation

Features

- Allows simulation of VHDL with Lucent's *ORCA* Series FPGAs pre- and postsynthesis, as well as after map/place/route.
- Interfaces industry-standard simulation products to the *ORCA* Foundry Development System.
- Fully compatible with any VITAL 3.0-compliant simulator, such as *Model Technology's V-System/VHDL* and *Synopsys'* VSS simulators.
- System-level verification supported using the *SmartModel* for *ORCA* FPGAs available from the Logic Modeling Group of Synopsys, Inc.

Design Flow



VHDL Simulation Kit

The *ORCA* Foundry VHDL Simulation Kit supports customers who are designing with VHDL. The combination of industry-standard synthesis and simulation tools and the *ORCA* Foundry Development System provides the designer with a fully integrated high-level design system for implementing Lucent's *ORCA* FPGAs.

The VHDL Simulation interface is compatible with VITAL 3.0-compliant simulators, such as *Synopsys*' VSS and *Model Technology*'s V-System simulators.

Simulation Capabilities

The *ORCA* Foundry VHDL Simulation interface includes VITAL 3.0-compliant simulation models for the *ORCA* macrocell library, which contains over 300 high-level logic functions, such as basic logic gates, ALUs, counters, and RAM. These models can be used to verify designs before and after synthesis prior to map/place/route.

An additional set of VITAL 3.0-compliant simulation models is supplied for full timing simulation after map/place/route. There are approximately 30 simulation primitives to describe the logic functions of a completed design. Combined with a standard delay format (SDF) file containing logic and routing delays, these models can be used for timing verification.

An *ORCA*-designed *SmartModel* can also have timing information back-annotated to it to further enhance timing simulation and system-level verification.

Ordering Information

The *ORCA* FPGA Library for VHDL Simulation is included with the *ORCA* Foundry Development System:

- The VISTA configuration is a complete design system supporting the ATT3000 and *ORCA* 2Cxx, OR2CxxA, and OR2TxxA devices up through 10,000 gates (OR-ORCAVISTA-WS for workstations, OR-ORCAVISTA-PC for personal computers).
- The APEX configuration supports **all** Lucent FPGAs (OR-ORCAAPEX-WS for workstations, OR-ORCAVISTA-PC for personal computers).

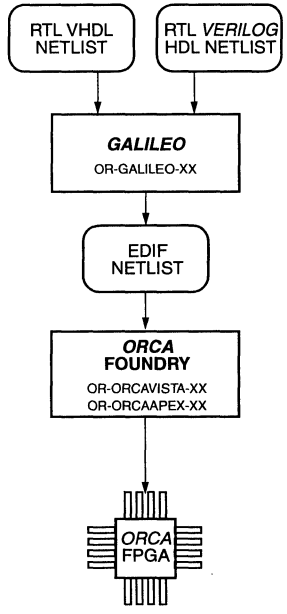
Lucent Technologies
Bell Labs Innovations

Exemplar Logic's Galileo and *Model Technology's V-System/VHDL*: HDL-Based Synthesis and Simulation for *ORCA* FPGAs

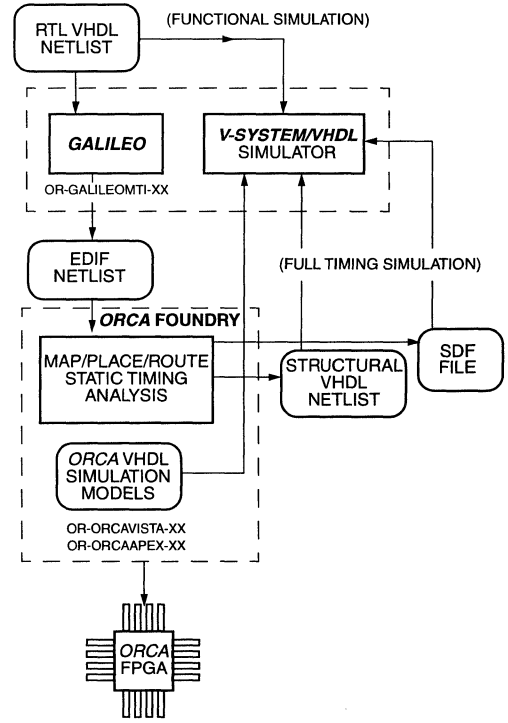
Features

- Synthesizes VHDL or *Verilog* HDL register transfer level (RTL) descriptions into *ORCA* devices
- Optimization for area and/or speed
- Look-up table synthesis produces results comparable to hand-drawn schematics
- User-defined timing constraints and synthesis direct optimization
- Fully compatible with *IEEE* 1076 and 1164 VHDL, and OVI 2.0 *Verilog* HDL
- PC and workstation support
- Optional simulation support with *Model Technology's V-System/VHDL* simulator

SYNTHESIS FLOW ONLY



VHDL SYNTHESIS AND SIMULATION FLOW



5-4934(F)

Key: XX = PC (Personal Computer) or WS (Workstation). See *ORCA* Foundry Ordering Information.

Figure 1. Synthesis and Simulation Flows

Description

Galileo from *Exemplar Logic* is a logic optimization and synthesis tool for designing Lucent Technologies' *ORCA* series FPGAs. *Galileo* takes a register transfer level VHDL or *Verilog* HDL file as input and produces an optimized netlist for *ORCA* FPGAs. When combined with the *ORCA* Foundry Development System that is used to map, place, and route the netlist, the result is a fully integrated system for implementing an *ORCA* FPGA from a high-level description.

A key feature of *Galileo* is that the synthesis algorithms used to produce the output netlist are tuned for the *ORCA* architecture. Furthermore, for structured logic and data paths, a module generation capability is included to generate optimized adders, subtractors, comparators, and incrementers. The result is a design with an area and speed that matches the quality of hand-drawn schematics.

While the *Galileo* tool is used strictly for synthesis designers, it can also perform functional and full-timing VHDL simulation with a *Galileo* plus MTI *V-System/VHDL* package. The *V-System/VHDL* simulator from *Model Technology* features fast-direct compile technology, interactive source-level debugging, and accelerated VITAL support. It also fully conforms to the *IEEE* standard 1076 for VHDL language.

Solutions

Galileo may be purchased as a synthesis-only option. This solution comes with a single language reader: either VHDL or *Verilog* HDL, but a second language reader may be added. Customers who wish to add *Model Technology's* VHDL simulator to a previously purchased *Galileo* solution must purchase it directly from *Model Technology.*

A full VHDL synthesis and simulation flow is available with the combined *Galileo* and *V-System/VHDL* solution. This solution allows designers to perform high-level synthesis, pre- and postsynthesis functional verification, and full timing verification after place and route.

Both the synthesis-only and synthesis-plus simulation solutions come with one year of product updates and hot-line support. Annual maintenance agreements are available for both solutions to extend support beyond the initial maintenance period.

System Requirements

Personal Computers

- Processor: 80486, *Pentium*, or *Pentium-Pro* based personal computers
- Operating environment: *MS-DOS* 6.22 or *Windows* 3.1x (enhanced mode) and *Windows* 95 or *Windows* NT 3.5 or later
- Memory: 16 Mbytes minimum
- Storage: Hard disk drive with at least 100 Mbytes spare capacity
- Installation media: CD-ROM drive for *Galileo*, 3.5" disk drive for MTI's *V-System/VHDL*

Workstations

- Processor: *SPARC*-based *Sun Microsystems* workstation, or *Hewlett-Packard* 700 Series workstation
- Operating environment: *SunOS* 4.1 (or higher) or *Solaris* 2.4 (or higher) and *OpenWindows* 3.0 or *OSF/Motif* 1.2
- Memory: 16 Mbytes minimum
- Storage: Hard disk drive with at least 100 Mbytes spare capacity
- Installation media: 1/4" tape QIC 24 format and CD-ROM for *Sun Workstations* or 4 mm DAT for *HP* workstations

Ordering Information

PC Solutions

Part	Description	Part Number	Comcode
Synthesis Only	*Galileo* (specify VHDL or *Verilog* HDL)	OR-GALILEO-PC	107740888
VHDL Synthesis and Simulation	*Galileo* and MTI's *V-System/VHDL*	OR-GALILEOMTI-PC	107740862
Additional Language Reader for *Galileo*	Specify VHDL or *Verilog* HDL	OR-GALILEO2-PC	107740839
1-Year Maintenance Contract	Includes hot line support and product updates.	OR-GALILEOMAIN-PC	107740896

Workstation Solutions

Part	Description	Part Number	Comcode
Synthesis Only	*Galileo* (specify VHDL or *Verilog* HDL)	OR-GALILEO-WS	107740821
VHDL Synthesis and Simulation	*Galileo* and MTI's *V-System/VHDL*	OR-GALILEOMTI-WS	107740870
Additional Language Reader for *Galileo*	Specify VHDL or *Verilog* HDL	OR-GALILEO2-WS	107740847
1-Year Maintenance Contract	Includes hot line support and product updates.	OR-GALILEOMAIN-WS	107740854

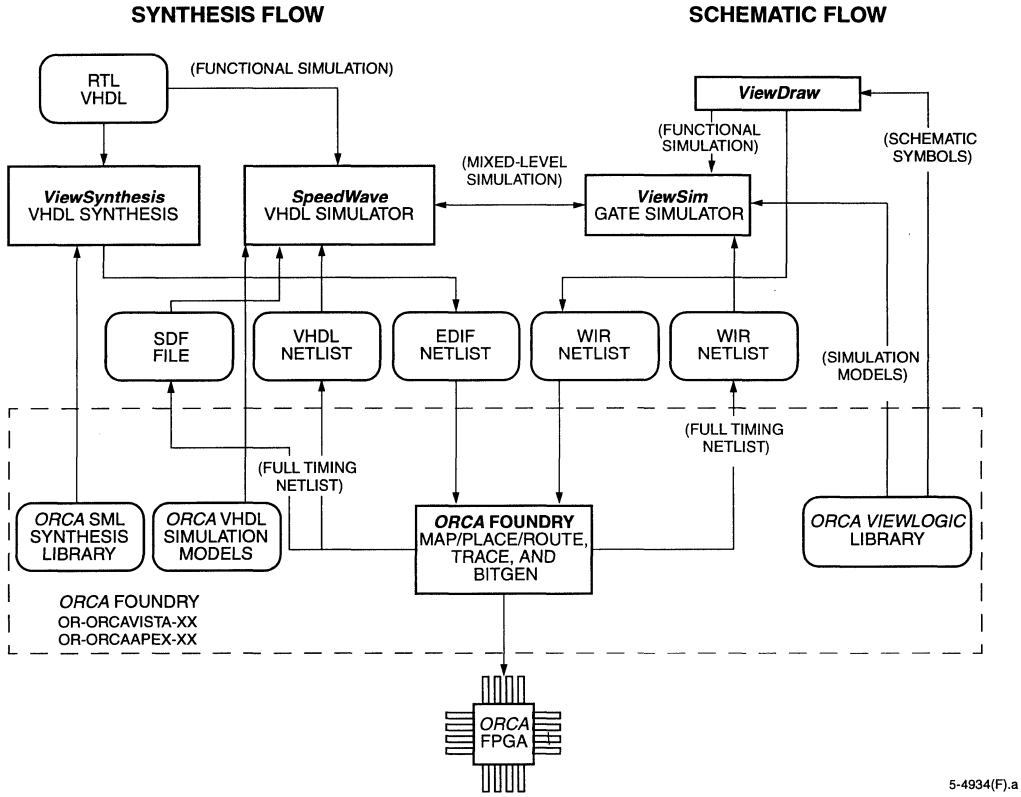
Notes

Lucent Technologies
Bell Labs Innovations

WorkView Office from Viewlogic Systems, Inc.: Integrated Schematics, Synthesis, and Simulation for *ORCA* FPGAs

Features

- Supports the *Windows* NT and *Windows* 95 environments
- Provides schematics, simulation, and synthesis—all in an integrated design environment
- Supports simulation of an unlimited number of gates
- Graphical Design Manager controls design threads
- Optional mixed gate-level and VHDL simulation
- Supports netlist to/from WIR and EDIF



Key: XX = PC (Personal Computer) or WS (Workstation). See *ORCA* Foundry Ordering Information.

Figure 1. *WorkView Office* Flow

Overview

WorkView Office is a complete suite of electronic design automation tools for *ORCA* FPGAs on the *Windows* 95 and *Windows* NT platforms. It offers designers the kind of design capability and high performance previously reserved for the *UNIX* platform. *WorkView Office* offers dynamic tools for schematic capture, gate-level simulation, VHDL synthesis and debugging, waveform analysis, and VHDL simulation—all in an integrated design environment. When combined with the *ORCA* Foundry Development System, a powerful design environment is created that enables designers to speed their FPGA designs to market.

ViewDraw

ViewDraw is a design entry tool for graphics and text. It combines schematics, block diagrams, state diagrams, and VHDL entry methods. Hierarchical design is also possible for the modular representation of any design. Dynamic links with *SpeedWave* for *Windows* permits graphical design debugging. Support for object linking and embedding (OLE) automation and visual editing enables seamless integration with popular packages like *Microsoft Word* and *Excel*.

ViewSim

ViewSim is *Viewlogic*'s easy-to-use, high-performance, interactive, digital simulation system. With a powerful built-in graphical navigator, it allows the user to view a design hierarchically, navigating through schematic blocks with a click of a button. *ViewSim* is integrated into *ViewTrace*, which is a fully featured waveform display method for digital simulation. *ViewTrace* enables the designer to review and measure the results of simulation and also provides the ability to edit waveforms and to back-annotate to digital simulations.

ViewSim also offers an optional cosimulation environment with *SpeedWave* for the simulation of both gates and *IEEE* 1076 VHDL.

ViewSynthesis

Viewlogic's *ViewSynthesis* improves productivity by synthesizing RTL-level specified designs into optimized implementations of *ORCA* FPGAs. Using VHDL, a designer can quickly create specifications at the RTL level without concern for the low-level logic components available in the Lucent *ORCA* macro library.

ViewSynthesis supports the following VHDL features:

- Signals and variables of type integer, bit, and bit vector
- Concurrent conditional signal assignments
- Multiple processes
- Clocked processes
- Combinational processes
- Built-in arithmetic
- Overloading for bit vector operators +, –, <, >, and =
- Description styles suitable for finite state machines permitting one-hot, sequential, and user-specified state encoding
- Generate statements
- Loop statements
- Asynchronous set/reset for registers, functions, and procedures
- Structural instantiation
- Optimal implementation of case settlement
- Support of std_logic and std_ulogic for *IEEE* 1164

SpeedWave

The optional *SpeedWave* simulator is a complete, easy-to-use, high-performance, interactive *IEEE* 1076 VHDL simulation system. With its native VHDL source code debugger, this powerful tool allows for quick analysis and debugging of complex designs.

Like *ViewSim*, *SpeedWave* is integrated with *ViewTrace*, a fully featured waveform display method for displaying digital simulation. The *SpeedWave* option also supports cosimulation with *Viewlogic*'s *ViewSim* for both gates and *IEEE* 1076 VHDL. This allows designs to mix gates and VHDL descriptions.

SpeedWave includes *Viewlogic*'s VHDL debugger, a true source-level debugger that can display individual sections of each source-level statement, such as a signal update or process execution. An expandable watchpoint window is also available to allow the user to view anything in a design, whether it be a signal, VHDL variable, or event name.

Solutions

Lucent Technologies offers two *WorkView Office* solutions: the Developer and the Expert. The *WorkView Office* Developer solution offers the following features: *ViewDraw, ViewTrace, ViewSim* (with unlimited gate simulations and full timing), and *ViewSynthesis.*

Lucent's *WorkView Office* Expert solution offers all of the features of the *WorkView Office* Developer solution plus the following: *SpeedWave* VHDL simulator, VHDL debugger, and mixed-level simulation. Mixed-level simulation allows the designer to use both *ViewSim* and *SpeedWave* to simulate both gates and VHDL.

For purchasers of the Developer solution, an upgrade option is available that adds the additional features of the Expert solution.

Both the Developer and Expert solutions include one year of product updates and hot-line support—available directly from Lucent Technologies. Annual maintenance agreements are also available that will extend product support beyond the initial one-year maintenance period.

System Requirements

- Processor: 80486, *Pentium*, or *Pentium-Pro* based personal computers
- Operating environment: *Windows* 95 or *Windows* NT version 3.5 or later
- Memory: 16 Mbytes minimum
- Storage: Hard disk drive with at least 240 Mbytes spare capacity
- Installation media: 2x CD-ROM drive

Ordering Information

Part	Description	Part Number	Comcode
WorkView Office Developer Solution	Features include *ViewDraw, ViewSim, ViewSynthesis,* and *ViewTrace.*	OR-WVODEVELOPER-PC	10780808
WorkView Office Expert Solution	Includes Developer solution features (listed above), plus *SpeedWave*, VHDL debugger, and mixed-level simulation.	OR-WVOEXPERT-PC	10780809
Upgrade Option	Upgrades Developer solution to Expert solution.	OR-WVOUPGRDD2E-PC	10783383
1-Year Maintenance Contract	Provides hot line support and product updates.	OR-WVOMAINT-PC	10783384

Notes

Lucent Technologies
Bell Labs Innovations

ORCA Series FPGAs Customer Solution Cores

Introduction

Customer Solution Cores (CSCs) are HDL descriptions of popular logic functions that enable FPGA designers to quickly and easily introduce prototype and production systems. These CSCs are designed to reduce development time and support design reuse among design teams so that designers can spend less time on low-level details and concentrate instead on higher-level, value-added features.

These *Verilog* HDL and VHDL macrocells are linked to high-level synthesis and have been optimized to target our *ORCA* series of FPGAs. Tight coupling to third-party synthesis vendors through the *ORCA* Alliance for Synthesis guarantees compact, high-speed results.

The following three CSCs are currently available:

- PCI Bus Target Interface
- PCI Bus Master Interface
- Parameterizable DSP FIR Filters

Other popular CSCs are also in the implementation stage, including a USB (Universal Serial Bus) interface, other DSP functions, and popular networking applications, such as ATM.

Notes

ORCA Series FPGAs in PCI Bus Target Applications

Introduction

This application note discusses a VHDL implementation of a peripheral component interface (PCI) bus target controller. This target controller interfaces an add-on application to the PCI bus. Because it is implemented in VHDL, the code can be synthesized to the Lucent Technologies *ORCA* 2C Series high performance FPGAs. These FPGAs range in density from 3,500—40,000 gates and are now produced in 0.35 µm and 0.5 µm, triple-layer metal CMOS processes. A feature analysis is also provided here that explains the rationale for using an *ORCA* device rather than another complex PLD or FPGA for a PCI bus design. The PCI bus target VHDL and a sample ATT2C15/OR2C15A/OR2T15A implementation are available from Lucent Technologies.

Technical Challenges of a PCI Bus Design

It generally requires 6—12 man months' development to implement a PCI bus interface in an add-on card design. This is in addition to the design work for the add-on's main functions. One reason for this is the critical electrical specifications that must be met by the design. Because the PCI bus is a high-speed, unterminated CMOS bus, the standard describes in detail the required I/O performance. The result is a set of critical parameters that almost no FPGA technology can currently meet:

- ac output drive characteristics defined as I/V curves (equations given for minimum and maximum drive current).
- Stringent input specifications (10 pF, <70 nA leakage current).
- Very high-speed performance:
  — Clock rate ≤33 MHz
  — 7 ns setup and 0 ns hold to system CLK
  — System clock to output valid delay: 11 ns
- Density and routability to handle 36-bit parity generation and checking, configuration registers, 36-bit input and output pipelines, and PCI bus control.
- Sufficient I/O count to handle 48 connections to the PCI bus and 70+ connections to the back end.

This application note shows that the *ORCA* family of FPGAs meets these requirements and has other features essential to the success of a project of this scope. Additionally, the *ORCA* series is the FPGA family with a working VHDL model for a PCI bus target.

ORCA FPGAs Are PCI Compliant

The *ORCA* ATT2Cxx/OR2CxxA/OR2TxxA Series allows designers to meet the PCI bus specifications with the following features:

- It meets the stringent I/O performance requirements of the PCI bus.
- It meets the timing requirements of the PCI bus (33 MHz).
- It meets the input loading requirements of the PCI bus.
- It has sufficient density to allow this generic PCI bus target to be further enhanced by the end user.
- It is SRAM-based for ease of use in development.
- It has nibble-based PLCs that fit bus-oriented logic very well.
- Its PLCs can be used as 16 x 4 SRAM cells for internal memory requirements.
- It has JTAG boundary-scan capability.
- It can be designed using an HDL which will allow the design to be retargeted to a volume technology where appropriate.
- The timing of routing passes is repeatable.

Table of Contents

PCI Bus Target Controller Features

This implementation of the bus target controller supports the following features:

- VHDL design, compliant with *IEEE* 1076 and 1076/1164 Extensions on VHDL interoperability.
- 32-bit PCI target interface—designed to PCI bus standard, Rev 2.1.
- Full 32-bit I/O and memory spaces supported.
- Full-speed burst support (132 Mbytes/s) in memory space.
- Address and data parity generation and checking for I/O, memory, and configuration spaces.
- PCI interrupt support.
- Simple interface to back-end application.
- Full 1149.1 JTAG Boundary Scan (optional).

- PCI configuration space registers:
  - — Device ID, vendor ID
  - — Status, command
  - — Class code, revision ID
  - — Memory base address
  - — I/O base address
  - — Interrupt line
  - — Interrupt pin

The following features are not supported by this implementation, but can be added by the user:

- 64-bit operation.
- Interrupt acknowledge or special cycles.
- Cache support.
- Exclusive access (lock).
- Master operation.

PCI Bus Design Criteria

This section provides background information on the PCI bus. It also contains detailed information relating to critical design parameters that influence the design, especially in the area of FPGA device selection.

Description of the PCI Bus

The PCI bus is an interconnect for personal computer (PC) and add-on boards that provides substantial performance gains over the usual ISA or EISA expansion slots. Among these are the following:

- High bus bandwidth—132 Mbytes/s on a 33 MHz bus at full-burst speed.
- PCI bus-to-system bridges give add-on PCI bus masters a high bandwidth path to main memory.
- Cache and exclusive access support.
- Transparent upgrade paths to 64 bits (data and address) and 3.3 V operation.
- Full autoconfiguration of PCI add-ons through uniformly defined configuration registers. Jumpers are not required to configure a system.
- Bus electrical specifications designed for direct drive by the FPGA or ASIC, eliminating the need for external bus drivers. This allows significant cost reduction because PCI interfacing can be designed into an add-on ASIC without MSI glue.
- The PCI bus is processor independent. It is not an extension of a processor's bus control scheme. This will extend the life of add-on designs.

Typical PCI applications are add-on boards that require high-speed memory or I/O access, including LAN adapters, video adapters, hard drive controllers, and SCSI cards. Using the PCI bus allows system designers to implement critical system components on a high-bandwidth bus using low-cost ASIC components, enhancing system price/performance. For example, the PCI bus is the bus used in full-performance *Pentium* systems.

There are three main types of devices that operate on the PCI bus:

- PCI Bus/System Bridge. Interfaces the PCI bus to the system processor, main memory, etc. This device can act as a PCI bus master. This includes arbitration for systems that allow multiple bus masters.
- PCI Bus Add-on Masters. Add-on devices that can operate the bus and may need access to other PCI add-ons or main memory on the system.
- PCI Bus Target-only Add-ons. Add-on devices that can only operate as targets. These devices respond to but do not initiate bus cycles.

This implementation is a target-only device.

PCI Bus Design Criteria (continued)

PCI Bus Signaling Method

For ease of use, this section briefly describes the PCI bus transfer methodology. Refer to the PCI Bus Specification, Revision 2.0 for more detailed information. (Note that the PCI bus standard uses "#" to indicate low-true signals.) Table 1 gives a description of all I/O signals used in this implementation.

Basics

The PCI bus signals consist of a 32-bit multiplexed address/data bus (AD[31:00]) and control signals. The bus is synchronous (all bus devices assert and sample data using the bus clock). Most signals are tristate and bidirectional and will be driven only when a device is selected. The only purely input pins are CLK, IDSEL (slot select for configuration cycles), RST# (bus reset), and the JTAG interface pins (optional) TDI, TCK, and TMS. The only purely output pin is INTA#, which is defined as an open-drain signal.

Signals Sent and Received By Targets

In relation to a target-only device, some signals are input-only, since only masters will drive those signals. These are FRAME# (sampled by all PCI bus devices to detect start and end of a transaction), IRDY# (initiator ready, used to assert master wait-states), and C/BE[3:0]# (command/byte enables, used to classify transactions and identify active byte lanes for a transaction).

When it is the selected device (determined by address/command decoded with base addresses and enable bits in the configuration registers), a target device will drive the signals DEVSEL# (indicates it has decoded and accepted the transaction), TRDY# (used to assert target wait-states), and STOP# (used to assert target-initiated transaction terminations). These signals are also driven during configuration cycles (master accesses of configuration register space) and interrupt acknowledge cycles (not supported in this implementation). When both TRDY# and IRDY# are true, a data word is clocked from sending to receiving device. In Figure 1, the signal directions are drawn to illustrate the bus from a target's point of view.

Special Tristate Considerations

The FRAME# signal is sampled by bus targets to detect the start of a transaction. A master that starts a cycle samples DEVSEL# for a response, even though DEVSEL# is not driven until a device actually accepts the cycle. Therefore, some signals that are constantly sampled may also, at times, not be driven. The PCI bus defines these signals (FRAME# and DEVSEL#) in an "off" state with pull-ups.

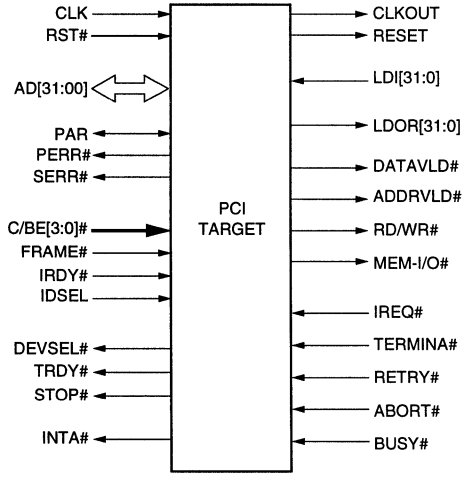


Figure 1. PCI Pin Diagram

PCI Bus Design Criteria (continued)

Table 1. PCI I/O Signals

Name	Type	Description
PCI Interface		
CLK	IN	CLK provides the reference signal for all other PCI interface signals, except RST# and INTA#. The frequency of CLK ranges from dc to 33 MHz.
RST#	IN	RST# is an input that initializes the FPGA's PCI interface circuitry to a known state. When reset, the PCI output signals are 3-stated, and the open-drain signals, such as SERR#, are floated.
AD[31:0]	TS	AD[31:0] are time-multiplexed address/data signals, with each bus transaction consisting of an address phase followed by one or more data phases. The FRAME# input signal identifies the start of an address phase. The data phases occur when IRDY# and TRDY# are both asserted.
C/BE[3:0]	TS (in)	Command and byte enable inputs are multiplexed on CBE[3:0]. The bus command (Table 3) is indicated during the address phase of a bus cycle. The byte enables are active during the data phase of a bus transaction.
PAR	TS	PAR is a 3-stated output of even parity calculated on the concatenation of the AD[31:0] and C/BE[3:0] fields.
FRAME#	S/T/S (in)	FRAME# is an output from the current bus master that indicates the beginning and duration of a bus operation. When FRAME# is first asserted, the address and command signals are present on AD[31:0] and C/BE[3:0]. FRAME# remains asserted during the data operation and is deasserted to identify the end of a data operation.
IRDY#	S/T/S (in)	Initiate or ready is output by a bus master to a target to indicate that the bus master can complete a data operation. In write operations, IRDY# indicates that data is on AD[31:0].
DEVSEL#	S/T/S (out)	Target asserts DEVSEL# as a decode acknowledge that address and bus commands are valid.
TRDY#	S/T/S (out)	Target ready is a target output that indicates that the current data operation can occur. In a read operation, TRDY# indicates that the target is providing data on AD[31:0].
STOP#	S/T/S (out)	STOP# is a target output that requests that the bus master stop the current transaction.
IDSEL	IN	Initialization device select is a chip select for configuration read or write transactions.
PERR#	S/T/S (out)	Data parity error indicates parity error on a data operation.
SERR#	O/D	SERR# indicates system error and address parity error.
INTA#	O/D	Interrupt A is an active-low interrupt to the host. INTA# must be used for any single-function device requiring an interrupt capability.
JTAG Signals		
TCK	I	Test clock input used to clock test commands into TMS and test data into TDI.
TMS	I	Test mode select is used to specify JTAG boundary-scan instruction or Lucent-defined instruction to execute.
TDI	I	Test data input into boundary-scan register, instruction register, or programmable scan ring.
TDO	O	Test data output from bypass register, boundary-scan register, instruction register, or programmable scan ring.

PCI Bus Design Criteria (continued)

Table 1. PCI I/O Signals (continued)

Name	Type	Description
Back-End Signals		
LDI[31:0]	I	LDI[31:0] is a 32-bit input data bus from the back-end application.
LDOR[31:0]	O	LDOR[31:0] is a multiplexed address/data bus to the back-end application.
DATAVLD#	O	DATAVLD# is an active-low strobe. For write operations, DATAVLD# is used to indicate valid data on LDOR[31:0]. For read operations, the back-end application must provide valid data into LDI[31:0] on the cycle after it detects DATAVLD# low.
ADDRVLD#	O	ADDRVLD# is an active-low strobe used to indicate a valid address on LDOR[31:0].
CLKOUT	O	CLKOUT is the buffered output of the PCI CLK input, and as such, ranges from dc to 33 MHz.
RESET#	O	RESET# is an active-low output that is the buffered output of the RST# input signal. It is not synchronized to CLK.
RD/WR#	O	RD/WR# is an output strobe used to specify a read operation when high, and a write operation when low.
MEM-I/O#	O	MEM-I/O# is an output strobe used to specify a memory operation when high, and an I/O operation when low.
IREQ#	I	IREQ# is an active-low signal from the back-end application used to request an interrupt.
TERMINA#	I	TERMINA# is an active-low input from the back-end application used to terminate the data flow.
RETRY#	I	RETRY# is an active-low input from the back-end application used to request that the master reattempt the bus transaction later.
ABORT#	I	ABORT# is an active-low input from the back-end application used to request that the master abort the bus transaction.
BUSY#	I	BUSY# is used by the add-on application to request wait-state(s).

For this approach to work at speed (pull-ups mean slow rise times on open-drain or 3-state outputs), these signals are defined as sustained tristate (s/t/s). This means that the PCI bus standard requires using one clock cycle to assert the signal false (high) before being 3-stated. The standard also requires that any signal that is being released, such as the master releasing AD after asserting the address on a read operation, is given a full cycle to 3-state before another device can start driving it. This is a turnaround cycle and prevents contention on the bus.

Two output signals, SERR# and INTA#, are defined as open-drain in the PCI standard. This is done by tying the signal to both the input of the tristate output buffer and the output enable, with the output enable active-low.

Parity

Each cycle asserted on the bus includes parity. Every device that transmits on AD[31:0] must also drive the PAR signal, including masters outputting the address. Since parity on the PCI bus is even, the sum of AD[31:0], C/BE[3:0]#, and PAR must be even. The PAR bit lags the AD bus by one clock.

Targets are not absolutely required to support parity checking, but if they do, a master must configure them to do so through the target's command register.

Table 2. PCI I/O Types

Signal Type	Function
I	Standard input.
O	Standard output.
TS	3-stated output or 3-stated bidirectional I/O.
S/T/S	Sustained 3-state is an active-low signal that must be driven high for a minimum of one clock cycle before it is floated. This signal cannot be driven prior to one clock cycle after it has been released. A pull-up resistor is required to sustain the inactive state.
O/D	Open-drain output signals allow multiple outputs to function as a wired-OR.

PCI Bus Design Criteria (continued)

Address parity errors are signaled on the SERR# pin, and data parity errors are signaled on the PERR# pin. Parity errors lag the PAR bit by one clock, and thus the address or data by two clocks.

Bus Commands

The PCI bus has no read/write signals. Control signals are embedded in the command transmitted during the address phase of a cycle. However, a latched C/BE[0]# signal can be used as an RD#/WR indicator. Table 3 indicates which cycles are acknowledged by this implementation and which are not. Note that Memory Read Line and Memory Read Multiple are mapped to (i.e., treated the same as) the standard Memory Read command, and Memory Write and Invalidate is mapped to the standard Memory Write. This is in accordance with the standard for implementations that do not provide cache support.

Interrupt acknowledge cycles are ignored by this implementation because such cycles are acknowledged only by the device that contains the interrupt controller. Since this is an add-on card implementation, this cycle is not needed. Special cycles are ignored because the specification requires all devices to ignore this type of cycle as far as overt acknowledgement in the form of asserting DEVSEL# is concerned. Special cycles are for broadcast messages from master to bus, and any receiving devices will capture the cycle as it goes by. This is easily added to any design, if needed. Dual address cycles are used for transferring 64-bit addresses, which are beyond the scope of this implementation.

The AD[1:0] signals affect how a target responds to a cycle. For example, if the I/O space implemented by the target doesn't supply all 4 byte lanes, then the target issues a target abort when AD[1:0] and the following C/BE signals do not agree. A memory address asserted with AD[1:0] = 00b will use a linear increment burst, but if AD[1:0] = 1xb, it must disconnect after the first data phase. On configuration cycles, AD[1:0] = 00b indicates a regular configuration cycle and AD[1:0] = 01b indicates a cycle intended for a bridge to another PCI bus. Multifunction devices with several sets of configuration registers would also decode AD[10:8] on configuration cycles, but this implementation will ignore those bits since it does not implement multiple functions requiring multiple sets of configuration registers.

Exclusive Access

This implementation does not implement exclusive access, which allows a master to lock out other masters from accessing a target.

Table 3. PCI Bus Commands

C/BE[3:0]#	Command	Support
0000	Interrupt Acknowledge	Ignore
0001	Special Cycle	Ignore
0010	I/O Read	Yes
0011	I/O Write	Yes
0100	Reserved	Ignore
0101	Reserved	Ignore
0110	Memory Read	Yes
0111	Memory Write	Yes
1000	Reserved	Ignore
1001	Reserved	Ignore
1010	Configuration Read	Yes
1011	Configuration Write	Yes
1100	Memory Read Multiple	Yes
1101	Dual Address Cycle	Ignore
1110	Memory Read Line	Yes
1111	Memory Write/Invalidate	Yes

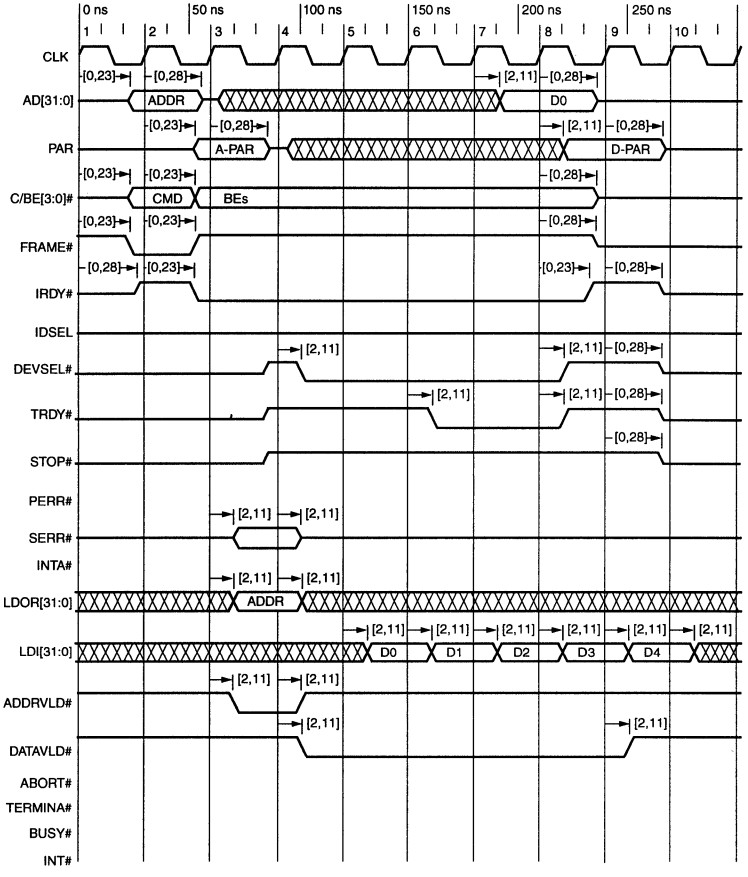
This feature is required if you implement system memory. However, since this implementation is intended for add-ons, exclusive access is not needed, so it was not included.

PCI Bus Cycle

Figure 2 shows the timing of a basic single data phase PCI bus read cycle. FRAME# low indicates the start of a cycle. Targets latch the address and command on the clock edge where FRAME# is low and then start the decode. In the example given, FRAME# is deasserted and IRDY# is asserted immediately after the address phase, indicating that the master is ready for data. If the master needs more time, IRDY# is delayed and FRAME# is not deasserted until the clock before IRDY# goes true. If the following cycles were multiple data phases, FRAME# would remain asserted until the last data phase.

A target that asserts DEVSEL# has accepted the transaction. In Figure 2, DEVSEL# is asserted at clock 4, and is sampled by the master on clock 5. This is "slow decode" timing. There are fast (DEVSEL# asserted on clock 2), medium (DEVSEL# asserted on clock 3), and slow (DEVSEL# asserted on clock 4) decode targets. If no DEVSEL# is detected by clock 5, a subtractive decode device may respond. Only one device on the bus can use subtractive decode, in which the decoding device accepts a transaction after detecting that no other device has asserted DEVSEL# by clock 5. This implementation is a slow decode implementation.

PCI Bus Design Criteria (continued)



5-4689(F)

Figure 2. Basic PCI Target Bus Read Cycle

PCI Bus Design Criteria (continued)

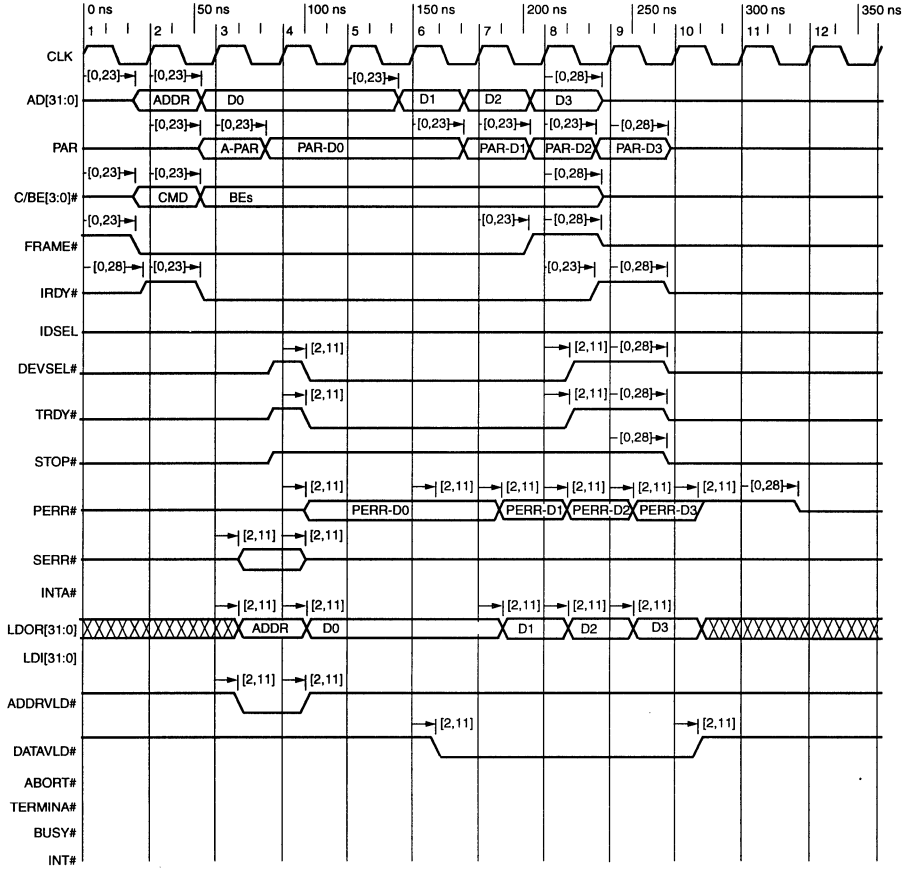


Figure 3. Burst PCI Target Bus Write Cycle

PCI Bus Design Criteria (continued)

Turnaround cycles are shown in the timing diagram. Because this is a read cycle, the master 3-states AD[31:0] on clock 2, when address is sampled. The target drives AD[31:0] in clock 3. At the end of the cycle, the data is transferred on clock 8, when both TRDY# and IRDY# are true. On this edge, the target releases the AD bus (the data has been sampled by the master), and the master releases C/BE#. PAR is released by the target one clock later, as is DEVSEL#, TRDY#, and STOP#. They are driven high on clock 8, so the s/t/s requirement of driving the signal high for one clock before releasing it is met. Though shown in 3-state, IRDY#, TRDY#, DEVSEL#, and STOP# are actually high. As mentioned, pull-up resistors are used to keep s/t/s signals high. They are shown 3-stated to illustrate when devices start and stop driving the signals.

PCI Bus Bursts

Figure 3 illustrates a write operation burst. If both the master and the target support burst operations, the PCI bus can burst data at the clock speed, as shown. Note the timing of PERR#, two clocks after the data. Since PERR# can be either 0 or 1, the last data assertion on clock 9 cannot be the "last clock asserted high" cycle for sustained tristate. One more cycle must be applied high before PERR# can be released. This affects when devices on the next cycle are allowed to drive PERR#, shown here at clock 4.

Target Termination

The PCI bus provides a mechanism for targets to disconnect cycles that it cannot support. The three types of termination are disconnect, retry request, and abort. An example of a use of termination is to halt a master's attempt to burst when the target cannot support one. Targets disconnect by asserting STOP# synchronous with TRDY#. A retry request is initiated when STOP# is asserted with TRDY# deasserted. The master ends the current data phase without a data transfer occurring and may reattempt the transfer later. A target abort request is initiated when STOP# is asserted and DEVSEL# deasserted on the same clock edge. In a target abort, the target indicates to the master that the transaction should not be attempted again.

Table 4. Target-Initiated Terminations

Termination	DEVSEL#	STOP#	TRDY#	Action
Disconnect	assert	assert	assert	End after current transaction.
Retry	assert	assert	deassert	Data is not transferred. Retry transaction later.
Abort	deassert	assert	don't care	Fatal error. Data is not transferred.

PCI Bus Configuration Registers

Uniformly defined configuration registers allow the PCI bus add-on cards to be configured by the system without requiring the end user to set jumpers. Configuration registers are accessed via normal PCI bus transaction cycles, but there is no throughput to the back-end application. Configuration cycles have their own command code defined in Table 3. The target must detect IDSEL in order for it to accept the cycle.

These configuration registers, shown in Figures 4 to 7, have the following functions:

Address 00h—03h, 08h: **Device, Revision, and Vendor ID Registers**: Read-only registers with ID values that uniquely identify the device to the system and application software. The vendor ID identifies the manufacturer and is assigned by the PCI SIG. The device ID and revision ID are set by the manufacturer.

Address 04h—05h: **Command Registers**: The command register controls the device, enabling functions such as I/O space, memory space, parity error generation, etc.

Address 06h—7h: **Status Register**: The status register reports on basic capabilities, events, and errors.

Address 09h—0Bh: **Class Code Register**: Identifies the type of device to system and application software, with codes that are defined by the standard.

PCI Bus Design Criteria (continued)

Address 10h: **Memory Base Address Register**: Allows system software to control the location of memory and on read tells the system about the capabilities and requirements of the memory space. MBAR[0] is a read-only 0, indicating a memory BAR. MBAR[2:1] are read-only and set by the user (in VHDL) to indicate memory type. MBAR[3] is read-only and set by the user to indicate whether memory is prefetchable. MBAR[31:20] are read/write for the memory base address.

Address 14h: **I/O Base Address Register**: Allows system software to control where the I/O space is located. IOBAR[1:0] are a read-only 01b to indicate an I/O BAR. IOBAR[31:15] are read-only zeros. IOBAR[15:4] are read/write for the I/O base address.

Address 3Ch—3Dh: **Interrupt Line and Pin Registers**: Interrupt line register is used by system to define to which interrupt controller line the device is connected. This will probably be slot dependent.

Interrupt_Line[7:0] is an 8-bit read/write register. The interrupt pin register is read-only and indicates which PCI bus interrupt pin (INTA#, INTB#, INTC#, or INTD#) the device is using. This application uses INTA#, but could easily be changed if desired.

Note that memory and I/O BARs return zeros in the unused address bits (bits that the address comparator doesn't look at) so that the system can determine the size of the memory or I/O space by writing all 1s to that register, reading it back and seeing how many zeros were forced by the add-on.

Besides the registers, this block also contains two comparators that will produce MEMHIT and IOHIT by comparing the ADIR bus during the address phase with the outputs of the memory and I/O BAR registers. These values are part of the decode that causes DEVSEL# to go true and makes the state machine start the cycles that pass data across the AD bus. The memory space and I/O spaces are limited by the BAR registers and the size of the address comparators. Currently, the memory BAR allows writes and does an address compare on the upper 12 bits of the address input to it from the PCI bus. This corresponds to a 1 Mbyte memory size. The I/O BAR forces the upper 16 bits [31:16] to 0 and allows writes only to bits [15:4]. This produces an I/O space size of 16 bytes and allows the back end to ignore the upper 16 bits of address for I/Os. These characteristics can easily be changed by the end designer, if required.

PCI Bus I/O Drive Characteristics

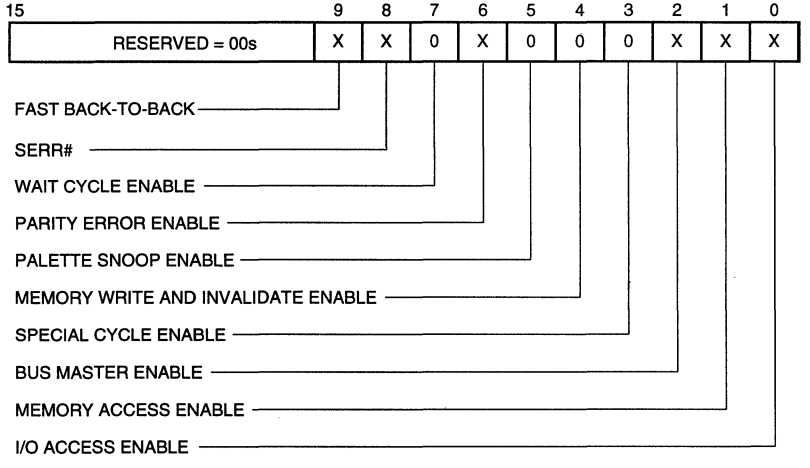
The PCI bus is defined as an unterminated CMOS bus. This means that steady-state current is very small, with almost all due to the pull-ups on control signals, and most current is transient current. The drive characteristics for output drivers on the PCI bus are thoroughly defined for ac as well as dc conditions. A study of Section 4.2 of the PCI bus standard is required to select an FPGA for a PCI bus interface.

PCI Bus Design Criteria (continued)

31 24	23 16	15 8	7 00	
DEVICE ID		VENDOR ID		00
STATUS		COMMAND		04
CLASS CODE			REV ID	08
BIST	HEADER TYPE = 0	LATENCY TIMER	CACHE LINE SIZE	0C
BASE ADDRESS REGISTER #1				10
BASE ADDRESS REGISTER #2				14
BASE ADDRESS REGISTER #3				18
BASE ADDRESS REGISTER #4				1C
BASE ADDRESS REGISTER #5				20
BASE ADDRESS REGISTER #6				24
RESERVED = 0s				28
RESERVED = 0s				2C
EXPANSION ROM BASE ADDRESS				30
RESERVED = 0s				34
RESERVED = 0s				38
MAX_LAT	MIN_GNT	INTERRUPT PIN	INTERRUPT LINE	3C

5-3854(F)

Figure 4. Configuration Registers



5-3855(F)

Figure 5. Command Register

PCI Bus Design Criteria (continued)

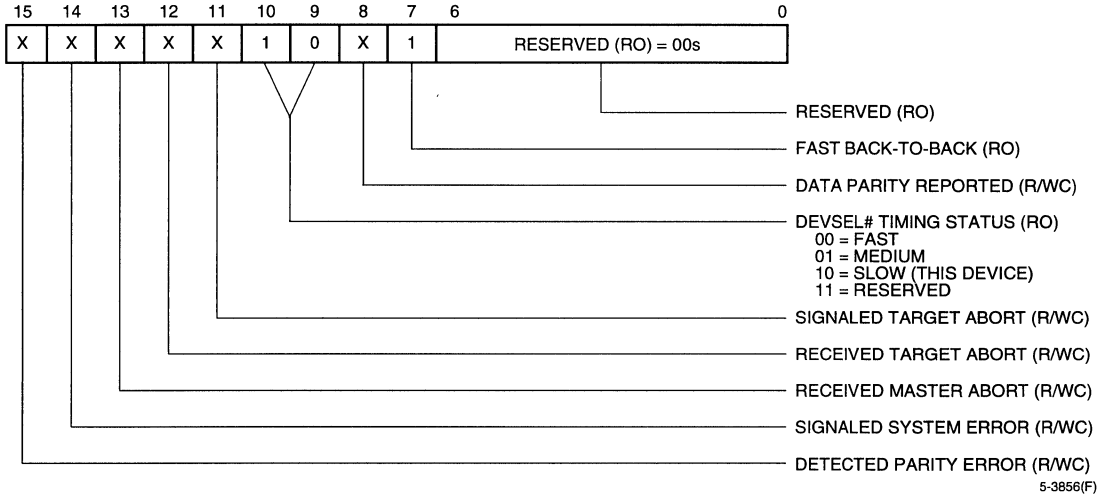


Figure 6. Status Register

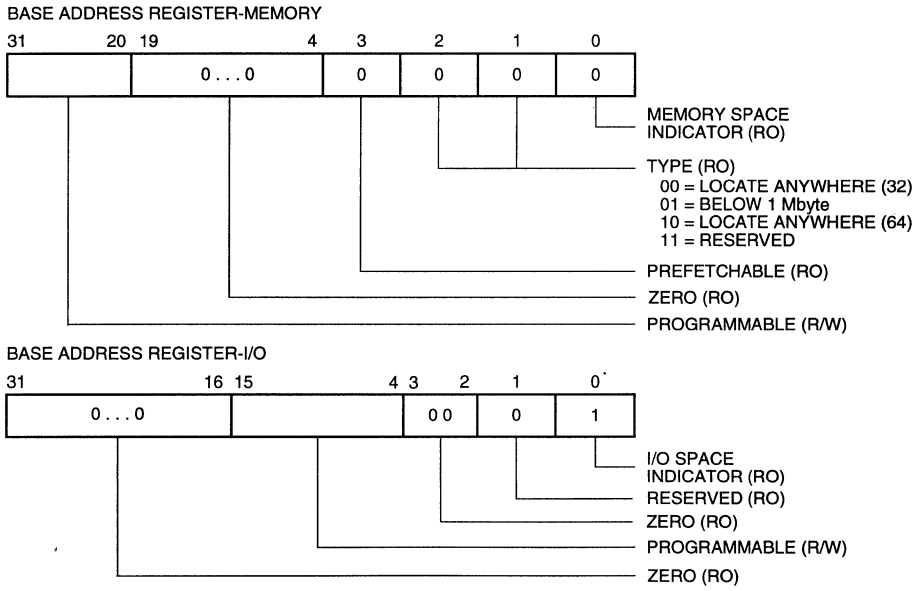


Figure 7. Base Address Registers

PCI Bus Design Criteria (continued)

The specification of I/O drive characteristics for the PCI bus includes definition of the voltage/current relationship through the driver's active switching range. In Figure 8, the *ORCA* output driver V/I curve is superimposed over the V/I specification for the PCI bus. Many vendors do not provide this data for their drivers, or the devices do not meet the specification.

The specification also includes device protection requirements. These relate directly to the unterminated environment and require inputs to handle transients from signal reflections. The specification requires the inputs to withstand an 11 V, 11 ns transient pulse (5.5 V overshoot) and a –5.5 V, 11 ns input undershoot.

While the PCI specification supports driving the bus directly with ASICs and FPGAs, this requires attention to details by the designer. Some FPGA technologies meet the specification, but most do not. For reference, PCI bus 5 V signaling dc and ac requirements as well as the *ORCA* 2C family associated values are indicated in Tables 5 and 6.

4

PCI Bus Timing Requirements

The PCI clock cycle is defined in Figure 3, with parameter values given in Table 7. To operate at 33 MHz, the period cannot exceed 30 ns. In the *ORCA* series, registers are located in programmable logic cells (PLCs). To meet input timing requirements, the *ORCA* 2C Series provides direct inputs to registers, located in the PLCs, from I/O pads. To meet output timing requirements, direct outputs to pads are available from registers in the PLCs to I/O pads.

An effective placement of logic in critical timing paths is essential to meeting PCI timing requirements. The *ORCA* layout tools allow timing to be specified in what is called a preference file. Excerpts of the preference file, pci.prf, define the critical setup, hold time, and propagation delay requirements.

```
/* CLK FREQUENCY */
FREQUENCY NET CLK 33.0000 MHz;
/* PROP DELAY */
OFFSET OUT COMP "AD<0>" 11.0 NS AFTER COMP
"CLK";
/* SETUP TIME */
.OFFSET IN COMP AD<0> 7.000 NS BEFORE COMP
"CLK";
```

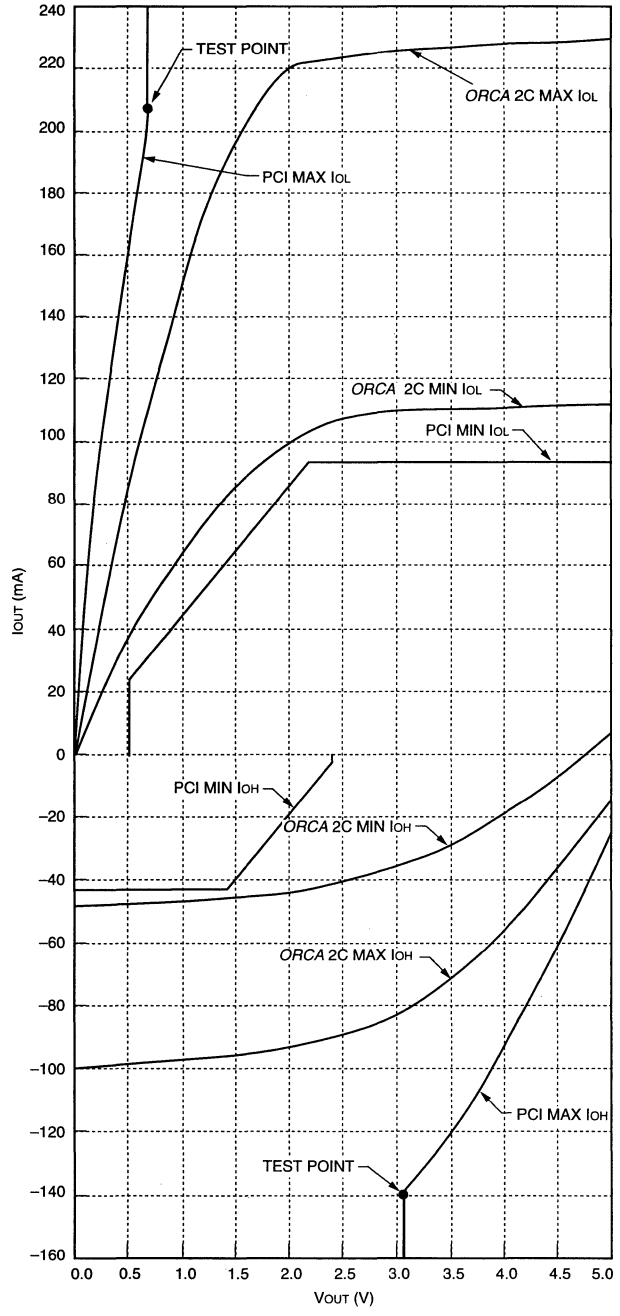
The two methods used to verify that the VHDL code meets PCI timing requirements are static timing analysis and timing simulation.

The preference file is also used as input into the *ORCA* Foundry *TRACE* static timing analysis tool to identify critical paths to analyze. An example of the timing analysis report file pci.twr is given below.

```
--------------------------------------------------------------------------
Design file:   pci.ncd
Preference file: pci.prf
Device, speed:   att2c08,3
Report level:   error report, limited to 3 items per
preference

--------------------------------------------------------------------------
Preference: FREQUENCY NET "manual_CLK"
33.000000 MHz;8546 items scored, 0 timing errors
detected.

--------------------------------------------------------------------------

Report:  37.594 MHz is the maximum frequency for
this preference.

--------------------------------------------------------------------------
Preference: OFFSET IN COMP "CBEN<0>"
7.000000 ns BEFORE COMP "CLK";17 items scored,
0 timing errors detected.

--------------------------------------------------------------------------
```

In the *ORCA* series, speed grades are designated using single-digit prefixes, with faster devices designated by higher numbers. A -3 speed grade is needed to meet the PCI requirements.

PCI Bus Design Criteria (continued)



5-3990(C)

Figure 8. *ORCA* vs. PCI V/I Curves

PCI Bus Design Criteria (continued)

Table 5. PCI Bus I/O dc Specification

Symbol	Parameter	PCI Specification		*ORCA* 2C		Unit
		Min	Max	Min	Max	
VDD	Supply Voltage	4.75	5.25	4.75	5.25	V
VIH	Input High Voltage	2.0	VDD + 0.5	2.0	VDD + 0.5	V
VIL	Input Low Voltage	–0.5	0.8	–0.5	0.8	V
IIH	Input High Leakage Current	—	70	—	10	µA
IIL	Input Low Leakage Current	—	–70	—	–10	µA
VOH	Output High Voltage	2.4	—	2.4	—	V
VOL	Output Low Voltage	—	0.55	—	0.4	V
CIN*	Input Pin Capacitance	—	10	—	7	pF
CCLK*	Clock Pin Capacitance	5	12	—	7	pF
CIDSEL*	IDSEL Pin Capacitance	—	8	—	7	pF
LPIN	Pin Inductance	—	20	—	†	nH

* Parameter met by *ORCA* but not most other FPGA families.
† This value depends on the package used. Please see the FPGA Data Book (MN95-001FPGA).

4

Table 6. PCI Bus I/O ac Specification

Symbol	Parameter	Condition	PCI Specification		*ORCA* 2C		Unit
			Min	Max	Min	Max	
IOH	ac Switching Current High*	0 < VOUT < 1.4	–44	—	–47	—	mA
		1.4 < VOUT < 2.4	–44 + (VOUT – 1.4)/0.024	Eq. A†	See Fig. 8.	See Fig. 8.	mA
		VOUT = 3.1	—	–142	—	–84	mA
IOL	ac Switching Current Low*	VOUT > 2.2	95	—	104	—	mA
		2.2 > VOUT > 0.55	VOUT/0.023	Eq. B†	See Fig. 8.	See Fig. 8.	mA
		VOUT = 0.71	—	206	—	116	mA
ICL	Low Clamp Current	–5 < VIN < –1	–25 + (VIN + 1)/0.015	—	Complies	—	mA
TR	Output Rise Time	0.4 V – 2.4 V	1	5‡	3.1	6.4	V/ns
TF	Output Fall Time	2.4 V – 0.4 V	1	5‡	4.2	9.7	V/ns

* Parameter met by *ORCA* but not most other FPGA families.
† Eq. A: IOH (max) = 11.0 x (VOUT – 5.25) x (VOUT + 2.45), Eq. B: IOL (max) = 78.5 x VOUT x (4.4 – VOUT).
‡ A guideline only. See PCI Standard, Revision 2.0 p., 89—91.

Table 7. PCI Bus I/O Timing Specification

Symbol	Parameter	PCI Spec		*ORCA* 2C		Unit
		Min	Max	Min	Max	
TVAL	Clock to Data Valid*	2	11	—	<10.8	ns
TON	Float to Active Delay	2	—	—	—	ns
TOFF	Active to Float Delay	—	28	—	<28	ns
TSU	Input Setup Time* (ATT2C26)	7	—	<5	—	ns
TH	Input Hold Time*	0	—	0	—	ns
TCUC	Clock Cycle Time*	30	—	<30	—	ns
THIGH	Clock High Time	12	—	<12	—	ns
TLOW	Clock Low Time	12	—	<12	—	V/ns

* Parameter met by *ORCA* but not most other FPGA families.

PCI Bus Design Criteria (continued)

The pci.prf specifies a setup time of 7 ns and a hold time of 0 ns for all PCI input signals. In order to guarantee the 0 ns hold time, the input buffers are placed into a delayed input mode. This still allows the setup time of 7 ns to be met for all *ORCA* 2C devices (the ATT2C15-3 has a system setup time of less than 4.7 ns, for example).

A clock to output valid delay (generally 11 ns) is specified in the pci.prf file for all PCI outputs as well. This is a specification from the system CLK input pin to FFs to output pins that have a 50 pF load. All *ORCA* devices meet this 11 ns specification.

Other PCI Bus Requirements

The standard specifies a maximum trace length of 1.5 in. from card edge connector to the PCI device pins for all 32-bit signals. If the optional 64-bit bus is used, the trace length maximum is 2.0 in. The clock must have a total trace length of 2.5 ± 0.1 in. Since there are 47 to 49 signals interfacing to the PCI connector, the number of devices should be the minimum. While it might be possible to use two devices co-located adjacent to the PCI connector, this is generally not an effective use of PCB space. Also, the standard requires that only one device load be placed on any signal on the PCI bus. For example, a separate address and data path cannot be used. Most applications should use a single, high pin count device, such as the 208-pin SQFP.

Though not in the scope of this application note, the PCI bus has mechanical and software requirements in addition to the electrical specifications addressed here.

Technology Selection

This section discusses the criteria for the selection of a device for the PCI bus target implementation. In this analysis, gate array, or standard-cell, technologies are not evaluated, although the ability to retarget the design to a low-cost ASIC is a consideration. Only complex programmable logic devices and FPGAs are evaluated in this analysis.

General Considerations

Besides PCI bus compliance, this analysis reviews general issues that affect the suitability of a part.

Density

This design provides the interface between the PCI bus and the back-end application. The density of the 2C Series FPGAs allows large application-specific functionality to be added. Although the 2C08 has been used to implement the design, there are currently *ORCA* devices available with >26,000 gates, allowing greater than two-thirds of the logic to be application-specific.

Tools

The availability of VHDL source code allows synthesis using several CAE vendors' VHDL synthesis tools.

Functional and Repeatable Routing

This is a qualitative evaluation of a programmable ICs suitability for timing critical design. If the routing's effect on timing delays is not repeatable, then timing requirements may not be met consistently.

PCI Bus Parameters to Support

Critical PCI bus parameters that the device must support are as follows:

- Complies with output V/I curve
- Complies with input buffer requirements: 10 pF, <70 µA leakage current
- Input setup time: 7 ns
- Input hold time: 0 ns
- Clock to output valid delay: 2 ns—11 ns
- FMAX: 33 MHz

Structurally, the device needs input buffers with one load on any signal. For example, AD[31:0] is input to both an address latch and to parity generation and check circuitry. This must appear as one load to the bus. Also, there needs to be the ability to control the output enables of output buffers with relatively complex combinatorial circuitry.

Technology Selection (continued)

FPGA Competitive Analysis

Several FPGA families were analyzed for suitability for this design. Some were omitted because their I/Os didn't meet the PCI bus specifications; others, because they didn't meet the density requirements. The *ORCA* Series FPGAs meet the specifications, are available, and have the highest gate count of any FPGA on the market. Other *ORCA* features are the following:

- Same pinout for all devices from the ATT2C04 to the ATT2C40 allows package selection based on design requirements. Upgrading to a larger *ORCA* series FPGA has no impact on pinout. This allows the designer to layout the PCB sooner. An added bonus is that a working pinout for the PCI bus is provided with this design, allowing the designer to start board layout even sooner.
- Nibble orientation of PLCs lends itself well to processing 32-bit buses.
- The look-up tables (LUTs) in the PLCs can be used as static RAM. This is useful for configuration registers, internal FIFOs, and/or RAM. Each PLC can be a 16 x 4 RAM.
- Boundary-scan (JTAG/*IEEE* 1149.1) capability can be connected to PCI bus JTAG pins.
- The reconfigurability of an SRAM-based FPGA lends itself to product development cost control because money can be used up in other technologies by throwing away experiments that don't work.

VHDL Implementation

In this section, the important characteristics of the VHDL implementation as they relate to PCI bus features and performance are discussed. In other words, the design is emphasized, not the VHDL. This information is fundamentally no different than if the design is implemented using *Verilog* (also available from Lucent Technologies), equations, or a schematic editor. For details on how to use the VHDL, see the *PCI Bus Target VHDL Source Code* Manual (MN96-018FPGA). Figure 9 is a detailed block diagram of this implementation and should be referred to when reading this section.

Back-End Signals and Data Path Flow

The purpose of this design is to interface the PCI bus to the user's back-end application. The signals provided to the back-end application are given in Table 1.

Write data and address is transmitted out LDOR[31:0], and read data is input LDI[31:0]. In this application, a bidirectional data bus for the back end is not provided. If bidirectional I/Os are needed, they can be added.

The timing of the control signals TERMINA#, RETRY#, ABORT#, and BUSY# will be shown later. TERMINA# is used by the back end to signal when to end a burst transaction. One use of this signal is to stop a burst that is about to cross an end-of-memory boundary. Another use is to enforce single data phases if the back-end application cannot support bursts. Note that the controller implementation enforces single data phases on configuration cycles, without the back end asserting TERMINA#. TERMINA# allows the application to do the same.

If the target terminates the burst attempt correctly, the master, when appropriate, advances the address and tries again. For the scenario when the burst steps across an end-of-memory boundary, the new attempt results in either another PCI bus device accepting the access, or the master generating a master abort because of a time-out on a DEVSEL# response.

RETRY# is similar to TERMINA#, except that instead of asserting STOP# on the last data phase, the controller asserts it afterwards. This type of termination will result in the master attempting to read the address again. The timing requirements of these two signals are somewhat different.

ABORT# is used by the back-end application to signal a catastrophic failure on the part of the back-end application. This type of termination indicates to the master that the target has concluded that it is incapable of performing the transaction it originally accepted. This response is required if the target detects an error in the signals asserted by the master. In most add-on applications, this signal probably will not be implemented by a back-end application, but it is provided in case there is a need.

BUSY# is used by the back end to request that the controller assert TRDY# wait-states in the transaction. The target must respond within eight clocks, or the master will generate a master abort. If the application needs more time, a retry terminate is the correct way to handle the pause.

VHDL Implementation (continued)

The VHDL code includes a burst address counter for use with SRAM. The application may require different functionality than what is provided, so the user can remove or modify this address counter if desired. If the address increments beyond the end of memory during burst mode, the target will disconnect.

The timing for read and write operations is relatively simple. Figure 10 shows the back-end signals for a single data phase read. Figure 11 shows a multiple data phase write operation. Both figures include the required back-end signals.

Write Operations

A register (INPUT REG) receives data from the AD and C/BE# buses and latches in the signals on each clock edge, resulting in a registered input bus for AD (ADIR[31:0]) and C/BE (CBER[3:0]). The ADIR bus and CBER bus are used for address decode and parity generation. ADIR is also one step in the pipeline that writes to the back end. This is implemented as a two-register pipeline:

INPUT REG (ADIR[31:0] → WRITE DATA OUTPUT REGISTER (LDOR[31:0].

The first phase in this pipe will be the address. Observe in Figure 11 on the multiphase write that LDOR data comes out of the controller two clocks after the data is sampled. For example, the D1 phase is clocked into the controller on clock 6 and shows up on LDOR on clock 7 and is clocked by the back end on clock 8. If the master imposes wait-states (IRDY# asserted), then DATAVLD# will reflect this and will be deasserted in accordance with the pipeline.

This is shown in the timing diagram in Figure 9, where IRDY# is deasserted on clock 6 and DATAVLD# is forced deasserted on clock 7.

Read Operations

For a read operation, master-imposed wait-states are possible. This requires the addition of a holding register for read operations.

The controller's output to AD[31:0] is from the OUTPUT REG, ADOR[31:0], which is fed by the readback multiplexer.

The possible outputs from the controller, selected with the readback multiplexer, are as follows:

- Local data input register LDIR[31:0]
- Local data delayed input register LDIRDLY[31:0]
- Device/vendor ID register
- Status/command register
- Class code/revision ID register
- Interrupt line and pin register
- I/O base address register (IOBAR)
- Memory base address register (MBAR)

The read data path from the back end is either a two- or three-register pipeline:

LDIR[31:0] → LDRDLY[31:0] → ADOR[31:0]

or

LDIR[31:0] → ADOR[31:0}

The holding register is normally not needed, unless there are master-imposed wait-states. As shown in Figure 10, the state of the LDIR[31:0], LDIRDLY[31:0], and ADOR[31:0] registered buses illustrate the need for a holding register when there are IRDY#-imposed wait-states. In this example, the pipeline is filled with data at the beginning of the cycle. Initially, only two stages of the pipeline (both LDIR[31:0] and LDIRDLY[31:0] provide data to the readback multiplexer) are used, so D1 arrives at LDI on clock 7 and is output by the controller on clock 9. IRDY# is then sampled as deasserted on clock 9. The back-end logic is allowed one clock cycle to respond to DATAVLD#. (If DATAVLD# is asserted on clock N, data is clocked in at N + 2, giving the back-end clock N + 1 to sample DATAVLD# true.) DATAVLD# is then deasserted on clock 10. The back end doesn't start holding the data stream until clock 11. The result is that D2 would be overwritten by D3 without a holding register; thus one has been provided (note that ADOR and the AD bus are the same for the data phases of this cycle).

More than one holding register is required if the memory in use in not prefetchable (like a FIFO). This is needed to hold prefetched contents for the next read cycle. Some changes in the control state machines will be required to support this. It must be able to detect the condition when data in the pipeline must be used as valid data. Besides LDOR[31:0], ADIR[31:0] is routed to the parity checking mechanism, the configuration register inputs, and the address comparators for memory and I/O decode.

VHDL Implementation (continued)

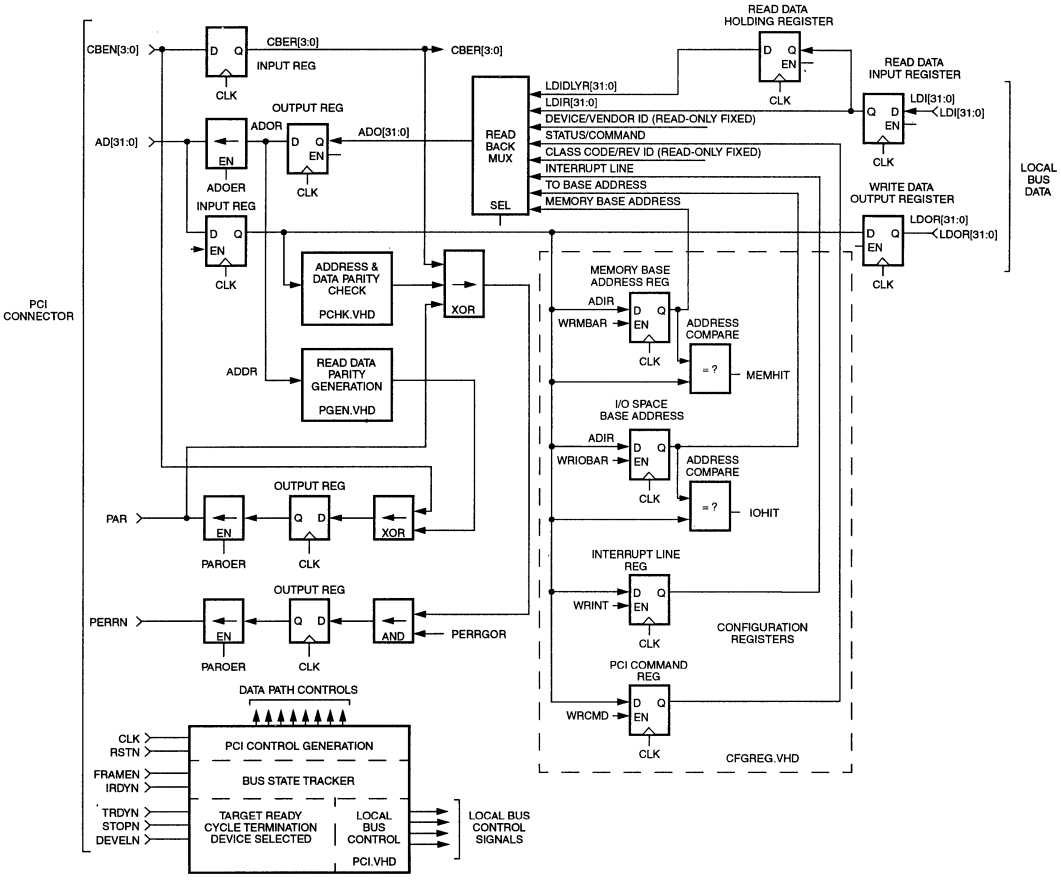


Figure 9. Detailed VHDL Target Block Diagram

VHDL Implementation (continued)

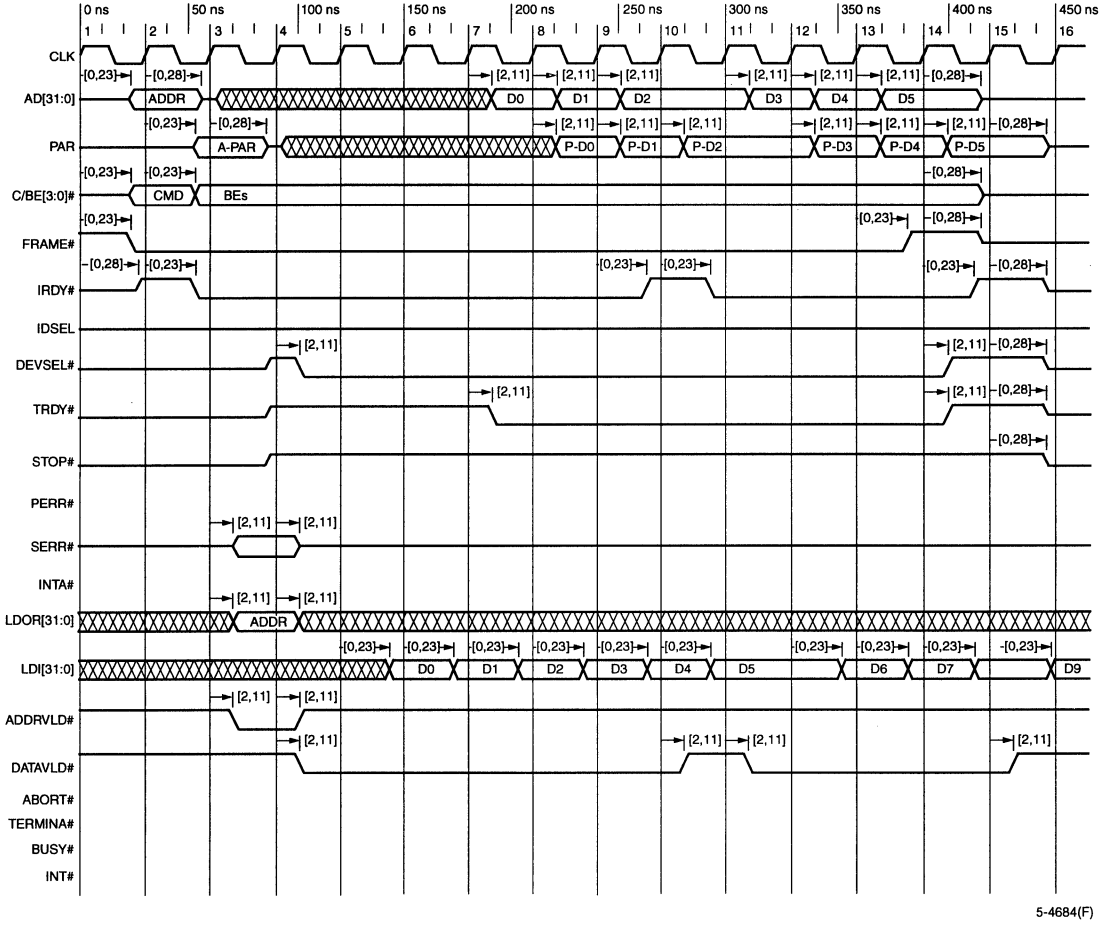


Figure 10. PCI Bus Single-Phase Read with Back-End Signals

VHDL Implementation (continued)

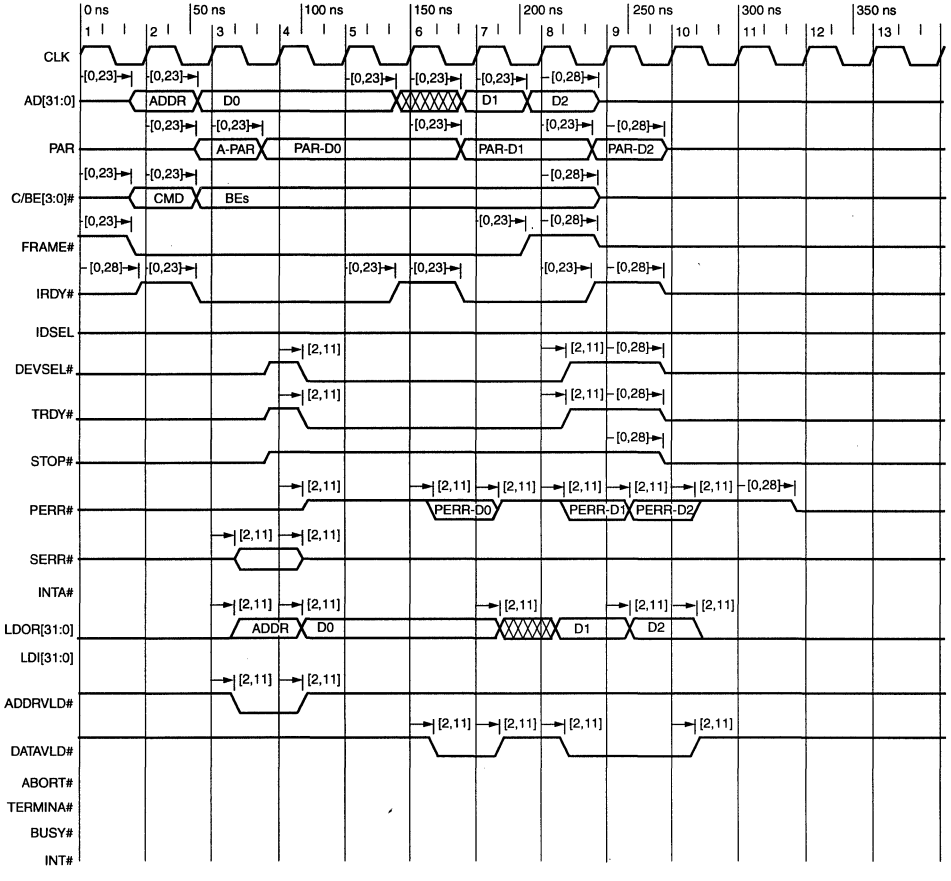


Figure 11. PCI Bus Multiphase Write with Back-End Signals

VHDL Implementation (continued)

Back-End Handshake Signal Timing

Figures 12, 13, and 14 provide timing diagrams for write operations with back-end handshaking. They show that BUSY#, TERMINA#, and RETRY# must arrive well in advance of related events on the back end.

For example, to assert a BUSY# wait-state on D2, BUSY# must be asserted one clock before D2 on the D1 sampling edge. This is four clocks ahead of the clock that would have sampled D2 on the back-end side, if no wait-state had been asserted. The same holds true for TERMINA# (to disconnect) and RETRY#. Because there is a two-stage pipeline between the PCI bus and the back end, the back-end circuitry must anticipate wait-state or busy requirements in order to properly hold off the master.

One of the side effects of the currently designed timing is that RETRY# cannot be applied to the first (D0) data phase. Some options are the following:

- Ignore if there is no need for back-end wait-states or disconnects on write operations.
- Ignore if wait-states or disconnect requirements can be anticipated in the back-end design.
- Add wait-states to all write data phases to give the controller time to respond to back-end events. This involves changing the controller design.
- Put the back-end wait-state/terminate circuitry in the *ORCA* Series FPGA design so it can respond faster.
- Add a holding register to the write pipeline and a multiplexer to the input of the last output in the pipe.
- Some applications may be able to respond to pipelined data after the master goes away, so this is not a problem.

For more detailed timing diagrams, please see the *PCI Bus Target VHDL Source Code* Manual (MN96-018FPGA).

VHDL Implementation (continued)

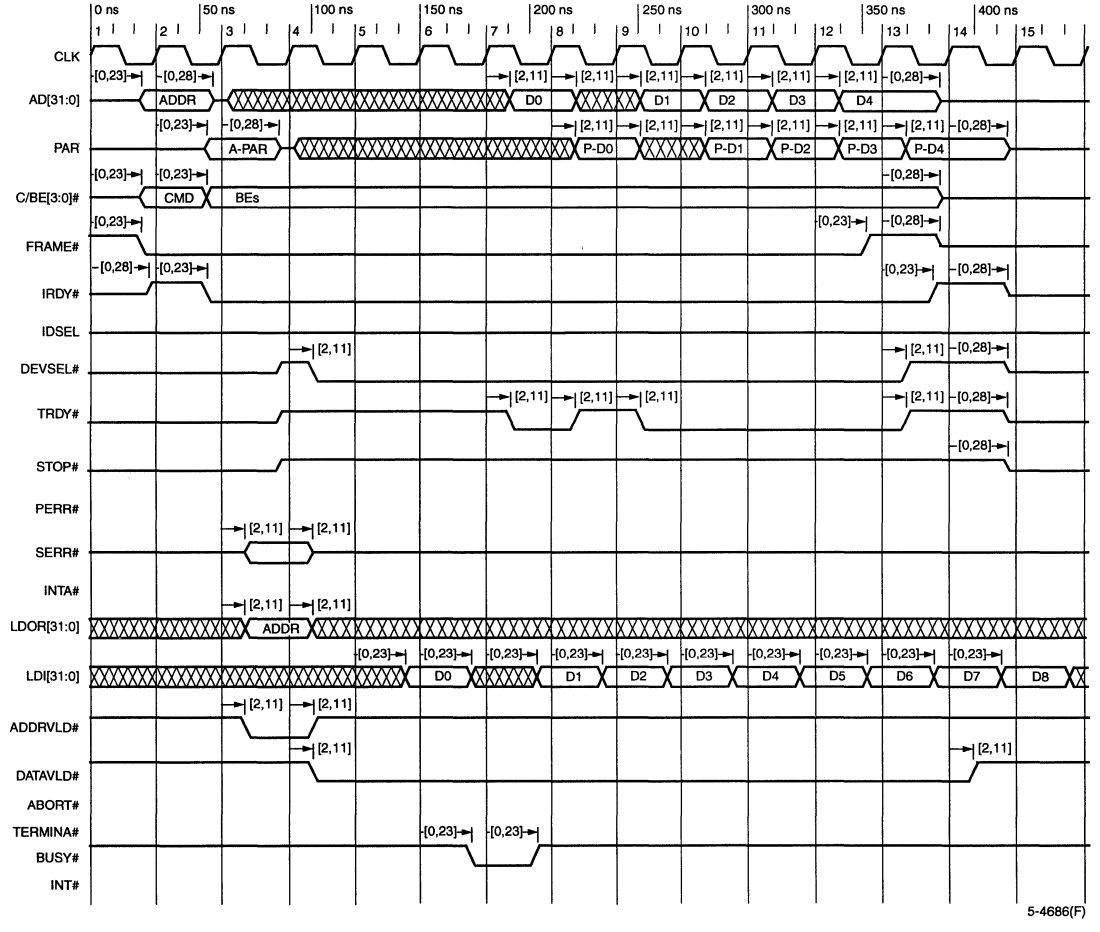


Figure 12. Read Cycle with BUSY#-Imposed Wait-State

VHDL Implementation (continued)

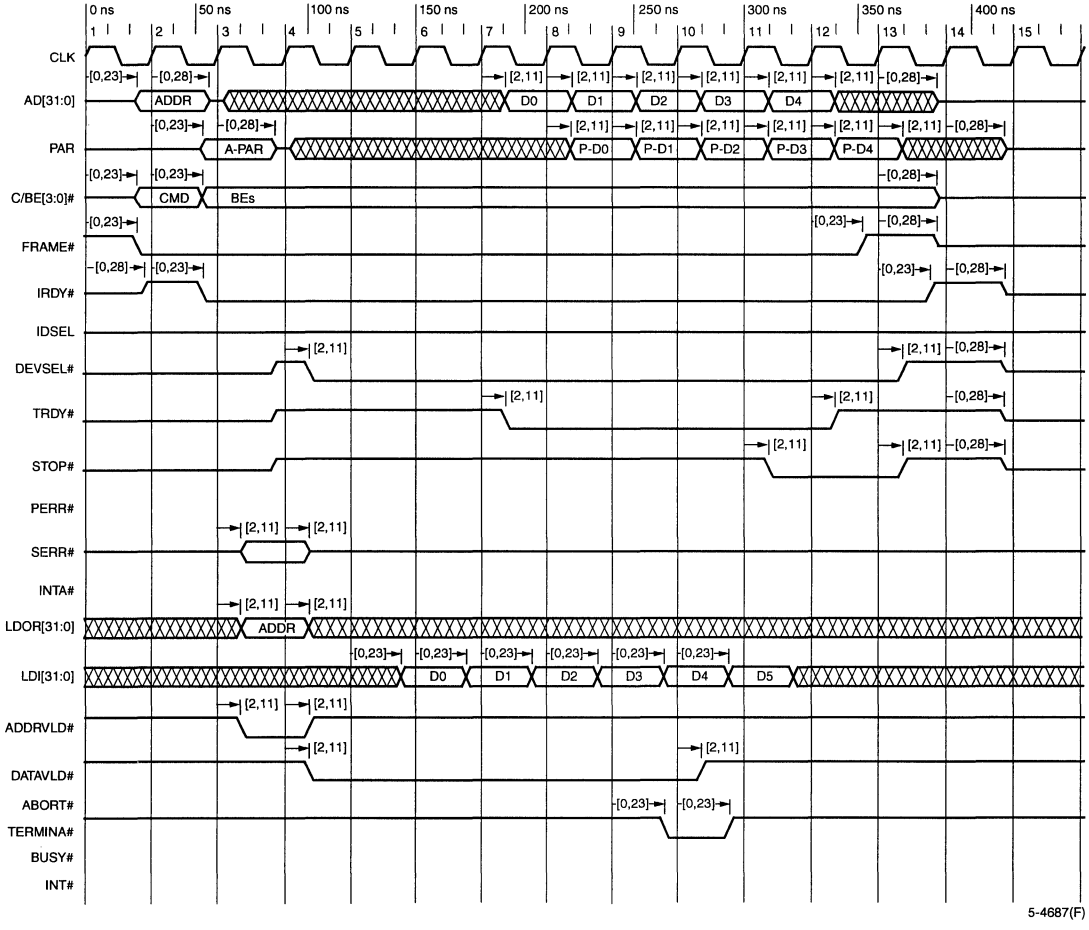


Figure 13. Read Cycle with TERMINATE#-Imposed Disconnect

VHDL Implementation (continued)

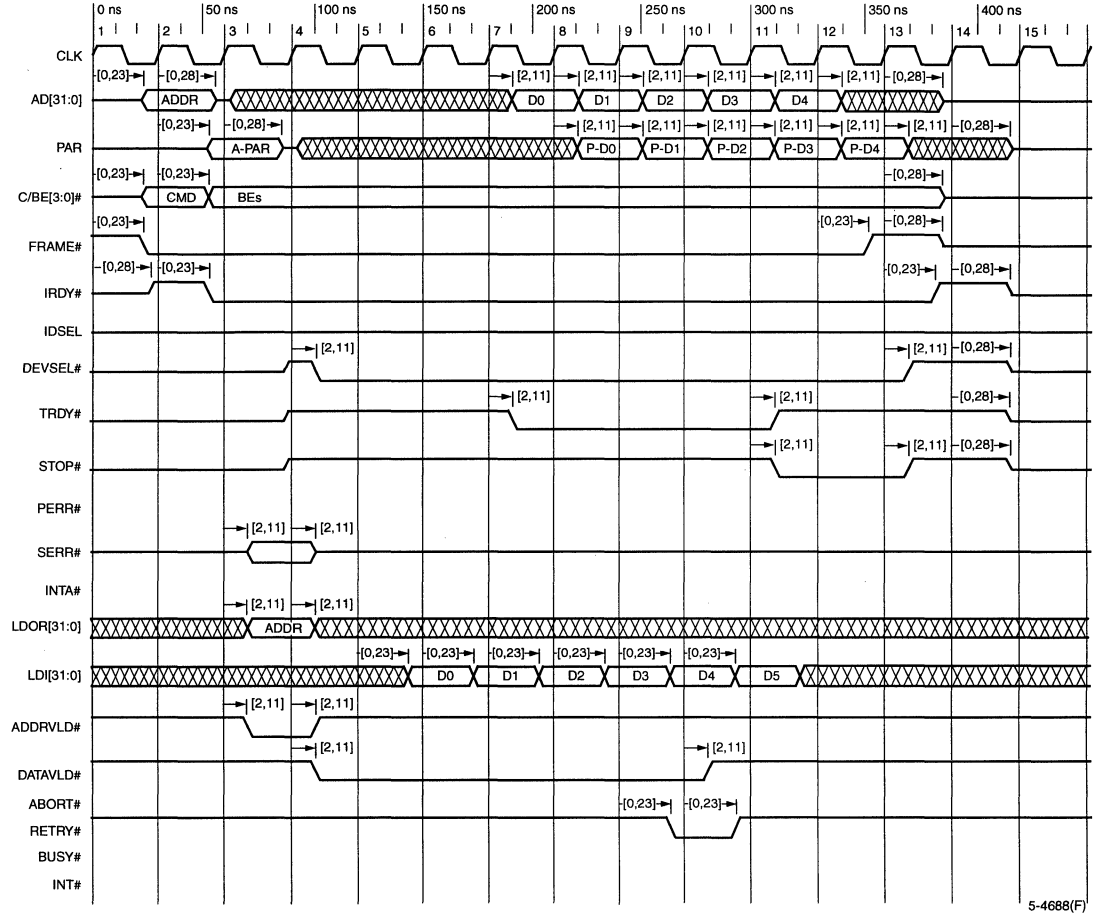


Figure 14. Read Cycle with Retry

VHDL Implementation (continued)

Parity Checking

ADIR[31:0] and CBER[3:0] are routed to a parity checker. The output of the parity checker is XORed with the PAR bit, which is input on the clock following the address or data. The result is ANDed with the parity enable bit from the command register and is routed to the output register. The PERR# signal or SERR# signal is then clocked out of the parity register, depending on whether this is the address phase or a data phase.

Parity Generation

For read operations, parity is generated and routed to the output register that asserts the PAR signal. CBE[3:0], not CBER[3:0], is used in the generation of parity. Using CBER[3:0] would give the CBE value for the wrong cycle.

Target State Machine

Detailed understanding of the PCI bus control mechanism requires reading the VHDL code, whose function is summarized here.

The control mechanism must assert the following signals:

- PCI bus control output enables (OE) for DEVSEL#, TRDY#, and STOP#
- AD bus output enable
- PAR output enable
- PERR# and SERR# output enable
- DEVSEL#, TRDY#, and STOP#
- ADDRVLD#, DATAVLD#, RD/WR#, MEM/IO#

Control OE. The output enable for DEVSEL#, TRDY#, and STOP# must turn on when the state machine has decided to accept a cycle. This occurs when the state machine detects MEMHIT and a memory cycle, IOHIT and an IO cycle, or a configuration cycle and IDSEL. The drive is turned off one cycle after FRAME# is deasserted with IRDY# asserted, which is a unique combination that always means the end of a transaction.

AD Bus OE. The output enable for the AD bus is turned on when the cycle is a read (CBE[0] = 0), one clock after the control OE is turned on. It is turned off when the clock edge FRAME# is first detected deasserted with IRDY# low.

Target Terminations Revisited

PAR OE: This signal follows the AD bus OE by one clock cycle.

PERR# OE.: This signal follows the controls OE by two clock cycles.

DEVSEL#: This signal goes true one cycle after the control OE is asserted, and goes false on the edge that FRAME# is deasserted with IRDY# asserted, or when ABORT# is asserted by the back end.

TRDY#: A number of conditions define the assertion or deassertion of TRDY#:

On read operations, the pipeline must be filled with data, and BUSY# is sampled to ensure that data in the pipeline is valid. Once the pipe is full, and as long as the back end is not holding off with BUSY# asserted, TRDY# is asserted until the master deasserts FRAME#. On the next cycle, TRDY# should be deasserted. If the back end does assert BUSY#, TRDY# is deasserted until valid data can be fetched from the pipe. In this context, TRDY# is based on pipelining BUSY# with the data.

On write operations, BUSY# doesn't affect TRDY# until the pipeline is being filled. Until then, TRDY# can be asserted. TRDY# is asserted with DEVSEL# on write operations and is not deasserted until FRAME# is deasserted, or the back end asserts BUSY#.

In all cases, the TRDY# state machine samples STOP# for a terminate data phase. Since STOP# is used to terminate single-phase transactions to the configuration registers, TRDY# must be able to handle STOP# on the first data phase.

STOP#. This signal is asserted only when the back-end signals a terminate with either the TERMINA# or RETRY# pins, or when the state machine suppresses burst activity on I/O and configuration cycles. If TERMINA# is signaled, STOP# is asserted on the next data phase and deasserted after FRAME# is deasserted. If RETRY# is asserted, STOP# is asserted whether there is a data phase or not. Presumably RETRY# is not asserted unless BUSY# is already in operation and has shut TRDY# off. STOP# is also asserted when ABORT# has been asserted.

ADDRVLD#. This signal is a FRAME# high-to-low transition with IOHIT or MEMHIT, delayed by two clocks, with a duration of one clock.

Target Terminations Revisited

(continued)

DATAVLD#: This signal interacts with BUSY# and IRDY#. On write operations, as data is transferred down the pipe, DATAVLD# must reflect when the pipe doesn't get new data from the master. It can be IRDY# attached as a 33rd bit to the write pipeline, in much the same way that BUSY# controls TRDY# on read operations.

On read operations, DATAVLD# is the AND of the control OE and IRDY#. Qualifying with memory access cycles so that DATAVLD# is asserted for only one cycle ensures that I/O cycles do not prefetch. DATAVLD# is always limited to I/O and MEMORY cycles involving back-end accesses.

RD/WR#: This signal is registered command register bit 0 and can be preserved until the next command is latched in.

MEM/I/O#: This signal is registered command register bit 2 and can be preserved until the next command is latched in.

Interrupt Support

There is no timing definition for the interrupt pin on the PCI bus. The IRQ# signal is synchronized with the clock and retransmitted. No mechanism is provided to turn off the interrupt. This can be done on- or off-chip by the designer. Off-chip, the capability should be mapped into the I/O space. Alternately, a configuration register bit could be added, and a signal used to disable the interrupt routed to the back-end circuitry.

Caveats and Suggestions

Some functions not included in this design are discussed here.

16/8-Bit I/O Support: The I/O is implemented as 32-bit I/O. To implement 16- or 8-bit I/O, AD[1:0] is sampled, and the target aborts if the AD[1:0] and the byte enables asserted for the data phase do not match. While this isn't difficult to implement, requiring additional address decode on a data phase, the test vector permutations are complex.

Byte Enables: Byte enables are usually useful only on write operations. They are not included in the interface to the back end, but can be added. The pipeline is 36-bit instead of a 32-bit, and are added as an extra 4 data bits in the pipeline.

Extra Holding Registers. If FIFOs are the target of burst read operations, additional holding registers are required since the controller has to prefetch considerably ahead of the data stream because of the pipeline requirements.

Design Verification Requirements

The PCI SIG defines the verification process for PCI bus designs in their PCI compliance checklist. We have taken the requirements described in that document and created our simulation vectors around the checklist and the timing requirements of the PCI bus.

Generic Test Parameters and the Compliance Checklist

Below is a summary of the checklist categories. The scenarios for features that we do not support have been lined through because we have no simulation vectors for those scenarios. This is per the compliance checklist rules.

~~2.1: Target reception of an interrupt acknowledge cycle.~~

~~2.2: Target reception of a special cycle.~~

~~2.3: Target reception of address and parity errors on a special cycle.~~

2.4: Target reception of I/O cycles with legal and illegal byte enables.

2.5: Target ignores reserved commands.

2.6: Target receives configuration cycles.

2.7: Target receives I/O cycles with address and data parity errors.

2.8: Target gets configuration cycles with address and data parity errors.

2.9: Target receives memory cycles.

2.10: Target gets memory cycles with address and data parity errors.

2.11: Target gets fast back-to-back cycles (same target writes only).

~~2.12: Target performs exclusive access cycles.~~

2.13: Target gets cycles with IRDY used for data stepping.

Design Verification Requirements
(continued)

The scenarios excluded in this list assume that this implementation doesn't support the following target features/cycle types:

- Interrupt acknowledge
- Cache support
- Special cycles
- 64-bit address/data
- Exclusive access (lock)
- Any master features

Parameters Common to All the Tests

Bus speed: All tests are specified for a bus running with a 30 ns clock cycle (33.333 MHz).

Clock related timing requirements: All vectors are specified to test for parameters relative to the clock.

Inputs: t_su = 7 ns, t_hold = 0 ns.

Outputs: t_valid min = 2 ns, max = 11 ns, t_off max = 28 ns, t_on min = 2 ns

Since all the timing of this bus is oriented to the bus clock, all the test vectors will also be oriented. For each clock, there will be signals asserted by the simulation, which should be timed to challenge the input setup and hold times, and outputs from the PCI bus target model, which should be tested for compliance with the output timing parameters (usually t_val min and max).

Structure of timing model for PCI-SIG specified test scenarios: Any particular set of vectors that tests both setup and hold parameters on the inputs must assert unrealistic waveforms that change state shortly before the clock edge and immediately after it.

For this reason, all test vectors for the PCI-SIG defined test scenarios will be structured as follows for each clock/edge:

@ clk? / –7 ns, assert simulation inputs
@ clk? / +11 ns, test for model output

The following is performed when dealing with signal release (allowing asserted signals to float):

@ clk2 / + 0 ns release AD[31:00] (reads only)
@ clk3 / + 0 ns release PAR (reads only)

Note: *clk2* is the clock edge where FRAME# is first sampled low and the transaction truly begins. Address is sampled on this clock. "Asserted" signals are those asserted by the simulation (e.g., the "stimulus").

AD[31:00] and PAR are left hiZ (released) until 7 ns before the clock edge they are first supposed to be driven on. FRAME# and IRDY# are never released by the simulation because they are s/t/s and always appear high to the device inputs.

Tests for hiZ outputs will occur as follows:

(following block only for full timing test, challenges the t_on min parameter of 2 ns)

@ clk2 / +2 ns DEVSEL#, TRDY#, STOP# s/b hiZ
@ clk3 / +2 ns AD[31:00] s/b hiZ (on reads)
@ clk4 / +2ns PAR s/b hiZ (on reads)
@ clk5 / +2 ns PERR# s/b hiZ (on writes)

(following block is used on all the test scenarios defined by PCI-SIG, tests t_off max of 28 ns)

@ clkf / +1clk+28 ns DEVSEL#, TRDY#, & STOP# s/b hiZ
@ clkf / +28 ns AD[31:00] s/b hiZ (on reads)
@ clkf / +1clk+28 ns PAR s/b hiZ (on reads)
@ clkf / +2clks+28 ns PERR# s/b hiZ (on writes)

Note: *clkf* indicates the clock where FRAME# is false and both TRDY# and IRDY# are true (the clock where the final data word is sampled). For the signals DEVSEL#, TRDY#, STOP#, and PERR#, the clock preceding the hiZ test should find the signal driven high by the model, to support the s/t/s function.

Test Definitions

In each of the scenarios defined below, the purpose of the test is given, followed by the actual steps of the test. Unimplemented scenarios are not included. All of the PCI-SIG specified tests are functional tests where the standard timing indicated above (assert inputs 7 ns before clock edge, test for outputs 11 ns after, etc.) is used throughout. This means that each clock of the transaction(s) in the test will have inputs asserted 7 ns before the clock edge and outputs checked for 11 ns after the clock edge. Testing for standard transaction timing is assumed without being documented to keep the size of this document manageable.

Only signal testing that is specific to the test is mentioned in the definition. The timing that is used for these tests directly relates to this PCI bus target implementation. For example, since this is a slow decode device (e.g., DEVSEL# is asserted on clk4), data to be output on read transactions is provided at the implementation's LDI bus, while data written by the master appears at the implementation's LDO bus. Both buses have pipeline effects that are accounted for in the tests. All tests defined below include checks for the non-PCI bus inputs and outputs supported on the back-end side of this implementation.

Design Verification Requirements

(continued)

The last set of tests are implementation-specific tests that do some timing checks using some of the tests from the memory test section. These tests are functionally equivalent, but will include tests of the t_valid and t_on minimum parameters and the input t_hold parameter. The functionality of the back-end handshake signals is also tested.

Scenario 4: IUT Reception of I/O Cycles

This scenario tests for proper implementation of I/O cycles. Generating target abort or disconnect when byte enables are illegal is not included because this PCI bus target implementation supports all 32 bits of the data path on IO cycles. The PCI SIG test specifications require testing of illegal byte enable handling only if the implementation does not support all 32 bits.

Start with a cycle to establish the I/O base address at 0x33cc000 and enable I/O, using the configuration registers. All the tests following assume that the address asserted is 0x33cc000.

Apply I/O read cycle w/ AD[1:0] = 00b, data = 0xA55A5AA5

Apply I/O write cycle w/ AD[1:0] = 00b, data = 0x5AA5A55A

Scenario 5: IUT Ignores Reserved Commands

This scenario verifies that the IUT doesn't see reserved or unimplemented commands as mapped to it, by verifying that DEVSEL# is false and that the IUT doesn't inappropriately drive any signals. In this implementation, the following seven cycle types are ignored:

CBE	CYCLE TYPE
0000	Interrupt Acknowledge
0001	Special Cycle
0100	Reserved
0101	Reserved
1000	Reserved
1001	Reserved
1101	Dual Address Cycle

In addition, it is verified that memory and I/O commands are ignored when they are not enabled:

0010	I/O Read
0110	Memory Read

Start with a configuration cycle that disables memory and I/O in the command register.

For each of the seven types:

- Apply a read cycle of that type.
- Verify that DEVSEL remains deasserted for at least six clocks.
- Verify that AD, TRDY#, STOP#, PERR#, & SERR# are not driven by the IUT.
- Apply a write cycle of that type.
- Verify that DEVSEL remains deasserted for at least six clocks.
- Verify that AD, TRDY#, STOP#, PERR#, & SERR# are not driven by the IUT.

Scenario 6: IUT Receives Configuration Cycles

This scenario validates that the IUT responds to configuration cycles with the proper bus signals and stores the data in implemented registers and outputs it properly when read back. For all configuration transactions, the local bus should stay inactive.

Note: This IUT handles only type 0 (AD[1:0] = 00b) configuration cycles.

Verify valid configuration cycles work properly for all 64 32-bit registers.

- Apply configuration write cycles of one and two data phases to fill the registers with test values, while AD[1:0] = 00b (use 1, 2, 1 data phases).
- Verify no local bus activity (no address valid or data valid supplied).
- Apply configuration read cycles of one and two data phases to read back content of the registers, while AD[1:0] = 00b (use 2,1,1 data phases).
- Verify return of zeros for data in unimplemented registers/bits.

Verify invalid configuration cycle (IDSEL deasserted) doesn't map to IUT.

- Apply configuration read cycle with AD[1:0] = 00b but IDSEL deasserted.
- Verify that DEVSEL# stays deasserted for at least six clocks and that AD, TRDY#, STOP#, PERR#, and SERR# are not driven by the IUT (this line is the test for "not mapped" and is assumed for the remaining "no IUT response" tests).
- Apply configuration write cycle with AD[1:0] = 00b but IDSEL deasserted, and verify no IUT response.

Design Verification Requirements
(continued)

Verify invalid configuration cycle (AD[1:0] = 01b) doesn't map to IUT.

- Apply configuration read cycle with AD[1:0] = 01b, and verify no IUT response.
- Apply configuration write cycle with AD[1:0] = 01b, and verify no IUT response.

Verify invalid configuration cycle (AD[1:0] = 10b) doesn't map to IUT.

- Apply configuration read cycle with AD[1:0] = 10b, and verify no IUT response.
- Apply configuration write cycle with AD[1:0] = 10b, and verify no IUT response.

Verify invalid configuration cycle (AD[1:0] = 11b) doesn't map to IUT.

- Apply configuration read cycle with AD[1:0] = 11b, and verify no IUT response.
- Apply configuration write cycle with AD[1:0] = 11b, and verify no IUT response.

Scenario 7: IUT Receives I/O Cycles with Address and Data Parity Errors

This scenario verifies that the parity error mechanism on the IUT is working properly. The enable bit for this function must be set in the command register in order for this feature to be implemented, so it is tested both ways.

Start with a cycle to establish the I/O base address at 0x33cc0000, I/O enabled and parity error and system error generation enabled in the command register. All the tests following assume that the address asserted is 0x33cc000.

Validate that IUT generates parity errors when command bit is set.

- Apply a configuration write cycle to enable parity and I/O cycles.
- Apply I/O read cycle with bad address parity, verify that IUT asserts SERR#.
- Apply I/O write cycle with bad address parity, verify that IUT asserts SERR#.
- Apply I/O write cycle with bad data parity, verify that IUT asserts PERR# with proper timing.

Validate that IUT ignores parity errors when command bit is not set.

- This test is buried in the standard I/O and memory tests by asserting a bad parity bit with the address and one phase of data and testing that SERR# and PERR# are not asserted.

Scenario 8: IUT Gets Configuration Cycles with Address and Data Parity Errors

This scenario verifies that IUT generates parity errors on configuration cycles that have them.

- Apply CONFIGURATION write cycle to enable parity error and system error generation.
- Apply CONFIGURATION read cycle with bad address parity, verify that IUT asserts SERR#.
- Apply CONFIGURATION write cycle with bad address parity, verify that IUT asserts SERR#.
- Apply CONFIGURATION write cycle with bad data parity, verify that IUT asserts PERR#, two clock cycles after the bad data phase.
- Apply CONFIGURATION write cycle with bad data parity on the second phase, verify that IUT asserts PERR#, two clock cycles after the bad data phase.

Scenario 9: IUT Receives Memory Cycles

This scenario verifies the many ways that the master can access memory on the IUT. Because of the variations in how memory can work on the PCI bus, this test has many permutations: single and multiple data phase accesses, accesses with linear increment set (allows bursts), and attempts to burst with reserved mode set (target forces single data phase with a disconnect). Note that linear increment or reserved mode is set with the AD[1:0] bits. Accesses that attempt to burst beyond the IUT's end of memory are also attempted. And we verify that the IUT maps Read Multiple, Read Line, and Write and Invalidate into standard read and write cycles. This definition doesn't do any byte lane rotation because the implementation at this writing takes no action on the local bus in relation to the byte enables.

Start with configuration cycles to set memory BAR = 33c00000 and memory enabled in the command register. All cycles below assume (unless otherwise indicated) that an address of 33cc0000 is applied during the address phase of the cycle.

Design Verification Requirements

(continued)

Verify standard memory read cycle, linear-increment mode, single and multiple data phases.

- Apply memory read cycle with BE = 0000b, AD[1:0] = 00b, one data phase, data = 0x5AA5A55A.
- Apply memory read cycle with BE = 0000b, AD[1:0] = 00b, six data phases, data = 0x5AA5A55A, 0x32212E01, 0x65546134, 0xAAAA5555, 0x5555AAAA, 0x69699696.

Verify memory read multiple cycle, linear-increment mode, single and multiple data phases.

- Apply memory read cycle with BE = 0000b, AD[1:0] = 00b, one data phase, data = 0x5AA5A55A.
- Apply memory read cycle with BE = 0000b, AD[1:0] = 00b, six data phases, data = 0x5AA5A55A, 0x32212E01, 0x65546134, 0xAAAA5555, 0x5555AAAA, 0x69699696.

Verify memory read line cycle, linear-increment mode, single and multiple data phases.

- Apply memory read cycle with BE = 0000b, AD[1:0] = 00b, one data phase, data = 0x5AA5A55A.
- Apply memory read cycle with BE = 0000b, AD[1:0] = 00b, six data phases, data = 0x5AA5A55A, 0x32212E01, 0x65546134, 0xAAAA5555, 0x5555AAAA, 0x69699696.

Verify standard memory write cycle, linear-increment mode, single and multiple data phases.

- Apply memory write cycle with BE = 0000b, AD[1:0] = 00b, one data phase, data = 0xAA559967.
- Apply memory write cycle with BE = 0000b, AD[1:0] = 00b, four data phases, data = 0xAA559967, 0xDD55CC67, 0x1088FF9A, 0x01AF6E9D.

Verify memory write and invalidate cycle, linear-increment mode, single and multiple data phases.

- Apply memory write cycle with BE = 0000b, AD[1:0] = 00b, one data phase, data = 0xAA559967.
- Apply memory write cycle with BE = 0000b, AD[1:0] = 00b, four data phases, data = 0xAA559967, 0xDD55CC67, 0x1088FF9A, 0x01AF6E9D.

Verify standard memory read cycle, reserved mode, single and (attempted) multiple data phases.

- Apply memory read cycle with BE = 0000b, AD[1:0] = 00b, one data phase, data = 0x5AA5A55A.
- Apply memory read cycle with BE = 0000b, AD[1:0] = 00b, two data phases, data = 0x5AA5A55A, 0x32212E01. Verify IUT ends transaction with a disconnect after first data phase.

Verify memory read multiple cycle, reserved mode, single and (attempted) multiple data phases.

- Apply memory read cycle with BE = 0000b, AD[1:0] = 00b, one data phase, data = 0x5AA5A55A.
- Apply memory read cycle with BE = 0000b, AD[1:0] = 00b, two data phases, data = 0x5AA5A55A, 0x32212E01. Verify IUT ends transaction with a disconnect after first data phase.

Verify memory read line cycle, reserved mode, single and (attempted) multiple data phases.

- Apply memory read cycle with BE = 0000b, AD[1:0] = 00b, one data phase, data = 0x5AA5A55A.
- Apply memory read cycle with BE = 0000b, AD[1:0] = 00b, two data phases, data = 0x5AA5A55A, 0x32212E01. Verify IUT ends transaction with a disconnect after first data phase.

Verify standard memory write cycle, reserved mode, single and (attempted) multiple data phases.

- Apply memory write cycle with BE = 0000b, AD[1:0] = 01b, one data phase, data = 0xAA559967.
- Apply memory write cycle with BE = 0000b, AD[1:0] = 01b, two data phases, data = 0xAA559967, 0xDD55CC67. Verify that IUT ends transaction with a disconnect after one data phase.

Verify memory write and invalidate cycle, reserved mode, single and (attempted) multiple data phases.

- Apply memory write cycle with BE = 0000b, AD[1:0] = 01b, one data phase, data = 0xAA559967.
- Apply memory write cycle with BE = 0000b, AD[1:0] = 01b, two data phases, data = 0xAA559967, 0xDD55CC67. Verify that IUT ends transaction with a disconnect after one data phase.

Verify standard memory read cycle in burst that steps past end of memory is disconnected by IUT when burst steps across the boundary.

- Apply memory read cycle with BE = 0000b, AD[1:0] = 00b, four data phases, data = 0x5AA5A55A, 0x32212E01, 0x65546134, 0xAAAA5555. Start address should be at end of IUT's memory space, less two DWORDs worth of space. Verify read and then disconnect after third word of data is transferred.

Design Verification Requirements
(continued)

Verify standard memory write cycle in burst that steps past end of memory is disconnected by IUT when burst steps across the boundary.

- Apply memory write cycle with BE = 0000b, AD[1:0] = 00b, four data phases, data = 0xAA559967, 0xDD55CC67, 0x1088FF9A, 0x01AF6E9D. Start address should be at end of IUT's memory space, less two DWORDs worth of space. Verify writes with disconnect after third word of data is transferred.

Scenario 10: IUT Gets Memory Cycles with Address and Data Parity Errors

This scenario does the same test of parity check functionality as the I/O and configuration parity error scenarios with more permutations. Extra permutations account for there being a total of five types of memory cycles for read and write transactions. Validation of not generating errors when command bit is not set is done as part of the tests in 2.10, by applying address and data parity errors during the cycles and verifying that parity error signals are not asserted. It is also verified that parity errors are not generated when the command register bit is set but the applied parity is correct.

Note: The PERR# signal is driven one clk after the PAR bit is asserted, one clk after the data, so PERR# follows the data by two clocks.

Start with configuration cycles to set the memory BAR to 0x33c00000, memory enabled and parity error and system error generation enabled.

- Apply MEMORY read cycle with good address parity. Verify that IUT does not assert SERR#.
- Apply MEMORY read cycle with bad address parity. Verify that IUT asserts SERR#.
- Apply MEMORY read multiple cycle with bad address parity. Verify that IUT asserts SERR#.
- Apply MEMORY read line cycle with bad address parity. Verify that IUT asserts SERR#.
- Apply MEMORY write cycle with bad address parity. Verify that IUT asserts SERR#.
- Apply MEMORY write and invalidate with bad address parity. Verify that IUT asserts SERR#.
- Apply MEMORY write cycle with good data parity. Verify that IUT does not assert PERR#.
- Apply MEMORY write cycle with bad data parity. Verify that IUT asserts PERR#.
- Apply MEMORY write cycle with bad data parity on second phase. Verify that IUT asserts PERR#.
- Apply MEMORY write and invalidate cycle with bad data parity. Verify that IUT asserts PERR#.
- Apply MEMORY write and invalidate cycle with bad data parity on second phase. Verify that IUT asserts PERR#.

Scenario 11: IUT Gets Fast Back-to-Back Cycles

This scenario verifies that the IUT handles fast back-to-back write cycles (where a write is followed by an immediate read or write to the same target and the master, therefore, imposes no idle phase for turn-around time).

Start with configuration cycles to set memory BAR = 33c00000 and memory enabled in the command register. All cycles below assume (unless otherwise indicated) that an address of 33cc0000 is applied during the address phase of the cycle.

Verify fast back-to-back with both transactions on IUT.

- Apply a write to address 0x33cc0000 of data 0x96696996 followed by a fast back-to-back write to address 0x34cc0000 of data 0xA55A5AA5.
- Apply a write to address 0x33cc0000 of data 0x96696996 followed by a fast back-to-back read from address 0x34cc0000 of data 0xA55A5AA5.

Verify fast back-to-back with first transaction to another IUT, followed by a fast back-to-back on this IUT.

- Apply a write to address 0x00000 of data 0x96696996 followed by a fast back-to-back write to address 0x33cc0000 of data 0xA55A5AA5.
- Apply a write to address 0x00000 of data 0x96696996 followed by a fast back-to-back read from address 0x33cc0000 of data 0xA55A5AA5.

Design Verification Requirements

(continued)

Scenario 13: IUT Gets Cycles with IRDY Used for Data Stepping

This scenario verifies that the IUT can handle wait-states imposed by the master. Incorrect data is applied on the clock edge of the wait-state for write operations. The PCI-SIG specification indicates a total of eight permutations based on three data phase transactions by applying four variations of read and write. The four variations are applying a wait-state to data phase 1, 2, or 3 and all 3. Because of the nature of the read pipeline, six data phases are tested for read operations so that the pipeline is fully tested.

Start with configuration cycles to set memory BAR = 33c00000 and memory enabled in the command register. All cycles below assume (unless otherwise indicated) that an address of 33cc0000 is applied during the address phase of the cycle.

Verify Read with wait on phase 1, 2, 3, or all 6.

- Apply six data phase memory read cycle, data = 0xA55A5AA5, 0x32212E01, 0x65546134, 0xAAAA5555, 0x5555AAAB, 0x69969669, and a wait-state on first data phase.
- Apply six data phase memory read cycle, data = 0xA55A5AA5, 0x32212E01, 0x65546134, 0xAAAA5555, 0x5555AAAB, 0x69969669, and a wait-state on second data phase.
- Apply six data phase memory read cycle, data = 0xA55A5AA5, 0x32212E01, 0x65546134, 0xAAAA5555, 0x5555AAAB, 0x69969669, and a wait-state on third data phase.
- Apply six data phase memory read cycle, data = 0xA55A5AA5, 0x32212E01, 0x65546134, 0xAAAA5555, 0x5555AAAB, 0x69969669, and a wait-state on all data phases.

Verify Write with wait on phase 1, 2, 3, or all 3.

- Apply three data phase memory write cycle, data = 0xAA559966, 0xDD55CC66, 0x1088FF9A, and a wait-state on first data phase.
- Apply three data phase memory write cycle, data = 0xAA559966, 0xDD55CC66, 0x1088FF9A, and a wait-state on second data phase.
- Apply three data phase memory write cycle, data = 0xAA559966, 0xDD55CC66, 0x1088FF9A, and a wait-state on third data phase.
- Apply three data phase memory write cycle, data = 0xAA559966, 0xDD55CC66, 0x1088FF9A, and a wait-state on all data phases.

Implementation-Specific Test 1: Full Timing Test

This test verifies memory cycles that challenge t_hold, t_val, and t_on minimums. The six data phase memory reads with wait-states and three data phase memory writes are used. The timing of the test changes as follows:

- The stimulus is asserted on clock edge – 15 ns and taken away at clock edge + 0 ns.
- Each test for model output at clock edge + 11 ns is enhanced by test for prior cycle's data still there at clock edge + 2 ns.
- For signals first being asserted by the model, hiZ is tested for at edge + 2 ns.

Start with configuration cycles to set memory BAR = 33c00000 and memory enabled in the command register. All cycles below assume (unless otherwise indicated) that an address of 33cc0000 is applied during the address phase of the cycle.

- Apply six data phase memory read cycle, data = 0xA55A5AA5, 0x32212E01, 0x65546134, 0xAAAA5555, 0x5555AAAB, 0x69969669, and a wait-state on all data phases.
- Apply three data phase memory write cycle, data = 0xAA559966, 0xDD55CC66, 0x1088FF9A, and a wait-state on all data phases.

Design Verification Requirements

(continued)

Implementation-Specific Test 2: Test Back-End Side Handshake Signals

This test verifies that the back-end handshake signals (Abort, Terminate, and Busy) function correctly. The following is the function of these signals:

- Abort—local logic cannot accept or provide data requested so it tells PCI bus to abort the transaction.
- Busy—local logic slows down access (this makes the IUT perform TRDY# wait-states).
- Terminate—local logic cannot accept or provide the data requested, but prior data in the transaction was provided, so the current data phase is stopped. The master may retry.

Note: Terminate has already been tested by the memory test that attempts a memory burst that crosses the end-of-memory boundary.

The method used is to assert memory read and write transactions with Abort and Busy asserted at appropriate times.

Start with configuration cycles to set memory BAR = 33c00000 and memory enabled in the command register. All cycles below assume (unless otherwise indicated) that an address of 33cc0000 is applied during the address phase of the cycle.

Verify Abort causes Target Abort signals on the PCI bus.

- Apply memory read cycle, data = 0xAAAA5555 and Abort asserted after ADDRVLD. Verify IUT conducts a Target Abort.
- Apply memory write cycle, data = 0xE7C381A5 and Abort asserted after ADDRVLD. Verify IUT conducts a Target Abort.

Verify Busy causes TRDY# wait-states.

- Apply six data phase memory read cycle, data = 0xA55A5AA5, 0x32212E01, 0x65546134, 0xAAAA5555, 0x5555AAAB, 0x69969669 with Busy asserted on the first three phases. Verify IUT asserts corresponding TRDY# wait-states.
- Apply three data phase memory write cycle, data = 0xAA559966, 0xDD55CC66, 0x1088FF9A with Busy asserted on last two data phases. Verify IUT asserts corresponding TRDY# wait-states.

CAE Tools

The remarks in this application note on CAE tools are introductory. Complete details on how to get from the VHDL source file to a working *ORCA*-based PCI controller are contained in the *PCI Bus Target VHDL Source Code* Manual (MN96-018FPGA).

To get to working silicon with a VHDL source file, you must use a tool that synthesizes the source code into a working file that the *ORCA* place-and-route tools can use. The following is a list of tools used for this project.

Synthesis (only one of the two following synthesis tools is required):

- Exemplar—This tool reads the VHDL input and synthesizes logic into a format that the *ORCA* Foundry tool can understand. This tool is available on both PC and workstation platforms.
- *Synopsys*—This tool reads the VHDL input and synthesizes logic into a format that the *ORCA* Foundry tool can understand. This tool is available on a workstation platform.

Map and Place-and-Route:

- *ORCA* Foundry—This tool reads the synthesized logic and performs map and place-and-route, to the *ORCA* FPGAs. This tool is available on both PC and workstation platforms. From the results of the place and route, *ORCA* Foundry produces information for the bit-stream files (used to program the cell array) and timing back-annotation (used to produce a simulation that works with the timing the place-and-route results give you). This tool is available on both PC and workstation platforms.
- *Viewlogic*—This tool is used to perform functional simulation and at-speed simulation of the design. This tool is available on both PC and workstation platforms.

Conclusion

The PCI bus was specified to enable designers to interface to it directly with large-scale ASICs. This objective has produced special parameters that designers must investigate carefully when conducting a component survey. In this case, we have chosen a field-programmable ASIC, the Lucent Technologies *ORCA* ATT2C15/OR2C15A/OR2T15A, because:

- It meets the stringent I/O performance requirements of the PCI bus.
- It meets the timing requirements of the PCI bus.
- It meets the input loading requirements of the PCI bus.
- The availability of the part is very good.
- It has sufficient density to allow this generic PCI bus target to be further enhanced by the end user.
- It is SRAM-based for ease of use in development.
- It has nibble-based PLCs that fit bus-oriented logic very well.
- Its PLCs can be used as 16 x 4 SRAM cells for internal memory requirements.
- It has JTAG boundary-scan capability.
- It can be designed using VHDL, which will allow the design to be retargeted to a volume technology where appropriate.
- The timing of routing passes is repeatable.
- It can be simulated with *ViewSim*.

References

The following publications provide additional information. Also note that the PCI SIG telephone number is (800) 433-5177. The *ORCA* 2C and 2CxxA Series data sheets can be obtained by calling (610) 712-5164.

- PCI Special Interest Group, *PCI Local Bus Specification*; Rev. 2.1; June 1, 1995.
- Lucent Technologies, *PCI Bus Target VHDL Source Code* Manual (MN96-018FPGA), June 1996.
- Lucent Technologies, *Field-Programmable Gate Arrays* Data Book (MN95-001FPGA), February 1995.
- PCI Special Interest Group, *PCI Compliance Checklist*, Rev. 2.0B.

Ordering Information

- *ORCA* PCI Bus Target Customer Solution Core for *Synopsys*/VHDL (OR-PCITCSC-VHDL-SYN)
- *ORCA* PCI Bus Target Customer Solution Core for Exemplar *Galileo*/VHDL (OR-PCITCSC-VHDL-EXG)
- *ORCA* PCI Bus Target Customer Solution Core for *Synopsys*/*Verilog* (OR-PCITCSC-VER-SYN)
- *ORCA* PCI Bus Target Customer Solution Core for Exemplar *Galileo*/*Verilog* (OR-PCITCSC-VER-EXG)

Note: Requires a signed license agreement.

Lucent Technologies
Bell Labs Innovations

ORCA Series FPGAs in PCI Bus Master Applications

Introduction

Lucent Technologies has developed a VHDL/*Verilog* implementation of a peripheral component interface (PCI) bus initiator (master) controller. Because it is implemented in VHDL/*Verilog*, the code can be synthesized to the *ORCA* ATT2Cxx/ATT2Txx/OR2CxxA/OR2TxxA Series FPGAs.

This product brief highlights the features and benefits of using an *ORCA* FPGA device in a PCI bus design. The PCI bus initiator VHDL/*Verilog* source code and a sample OR2C15A implementation are available directly from Lucent Technologies.

Features

- VHDL/*Verilog* design compliant with *IEEE* 1076 and 1076/1164 extensions on VHDL interoperability
- 32-bit PCI initiator interface designed to PCI bus standard, Rev 2.1
- VHDL/*Verilog* code can be synthesized to *ORCA* Series FPGAs using industry standard synthesis tools, such as *Synopsys* and Exemplar's *Galileo*
- *ORCA* Series FPGAs are available in both **0.35 µm** and **0.5 µm** process technologies
- Full 32-bit I/O and memory spaces supported
- Extendible to 64-bit data path
- Full-speed burst support (132 Mbytes/s) in memory space
- Address and data parity generation and checking for I/O, memory, and configuration spaces
- PCI interrupt support
- Simple synchronous local bus interface
- PCI configuration space registers:
  — Device ID, vendor ID
  — Status, command
  — Class code, revision ID
  — Latency timer
  — Interrupt line
  — Interrupt pin
  — MIN_GNT
  — MAX_LAT
- Full 1149.1 JTAG Boundary Scan (optional)

Technical Challenges of a PCI Bus Design

It generally requires 6 to 12 months of development to implement a PCI bus interface in an add-on card design. This is in addition to the design work for the add-on's main functions. One reason for this is the critical electrical specifications that must be met by the design. Because the PCI bus is a high-speed, unterminated CMOS bus, the PCI Bus Specification (Rev 2.1) was written to describe in detail the PCI bus' required I/O performance. The result was a set of critical parameters that could not be met until the introduction of the *ORCA* 2C Series of FPGAs. These parameters include the following:

- ac output drive characteristics defined as I/V curves (equations given for minimum and maximum drive current)
- Stringent input specifications (10 pF, <70 nA leakage current)
- Very high-speed performance:
  — Clock rate ≥33 MHz
  — 7 ns setup and 0 ns hold to system CLK
  — System CLK to output valid delay: 11 ns
- Density and routability to handle 36-bit parity generation and checking, configuration registers, 36-bit input and output pipelines, and PCI bus control
- Sufficient I/O count to handle 50 connections to the PCI bus and 70+ connections to the back end

This application note shows that the *ORCA* family of FPGAs meets these stringent requirements and has other features essential to the success of a project of this scope. The *ORCA* series is also the FPGA family with a working *Verilog* model for a PCI bus master.

Table of Contents

PCI Bus Master Controller Features

This implementation of the bus master controller supports the following features:

- *Verilog* design, compliant with *Open Verilog International*, Rev. 1.0
- 32-bit PCI master interface designed to the PCI Bus Specification, Rev 2.1
- Full-speed burst memory transfer (132 Mbytes/s) as master and configuration support as target
- 16-word deep by 32-bit transfer buffer
- Address and data parity generation and checking
- PCI interrupt support
- Simple interface to back-end application
- Full 1149.1 JTAG boundary scan (optional)
- PCI configuration space registers:
  - — Device ID, vendor ID
  - — Status, command
  - — Class code, revision ID
  - — Latency timer
  - — Minimum grant
  - — Maximum latency
  - — Interrupt line
  - — Interrupt pin

The following features are **not** supported by this implementation but can be added by the user:

- Interrupt acknowledge or special cycles
- Cache support
- Exclusive access (lock)
- Bridge operation

Note: 64-bit operation will be available at a later date.

PCI Bus Design Criteria

This section provides background information on the PCI bus. It also contains detailed information relating to critical design parameters that influence the design, especially in the area of FPGA device selection.

Description of the PCI Bus

The PCI bus is an interconnect for personal computer (PC) and add-on boards that provides substantial performance gains over the usual ISA or EISA expansion slots. Among these are the following:

- High bus bandwidth—132 Mbytes/s on a 33.333 MHz bus at full-burst speed.
- PCI bus-to-system bridges give add-on PCI bus masters a high bandwidth path to main memory.
- Transparent upgrade paths to 64 bits (data and address) and 3.3 V operation.
- Full autoconfiguration of PCI add-ons through uniformly defined configuration registers; jumpers are not required to configure a system.
- Bus electrical specifications designed for direct drive by the FPGA or ASIC, eliminating the need for external bus drivers. This allows significant cost reduction because PCI interfacing can be designed into an add-on ASIC without MSI glue.
- The PCI bus is processor independent. It is not an extension of a processor's bus control scheme. This will extend the life of add-on designs.

Typical PCI applications are add-on boards that require high-speed memory or I/O access, including LAN adapters, video adapters, hard drive controllers, and SCSI cards. Using the PCI bus allows system designers to implement critical system components on a high-bandwidth bus using low-cost ASIC components, enhancing system price/performance. For example, the PCI bus is the bus used in full-performance *Pentium* systems.

There are three main types of devices that operate on the PCI bus:

PCI Bus/System Bridge. Interfaces the PCI bus to the system processor, main memory, etc. This device can act as a PCI bus master. This includes arbitration for systems that allow multiple bus masters.

PCI Bus Add-on Masters. Add-on devices that can operate the bus and may need access to other PCI add-ons or main memory on the system.

PCI Bus Target-only Add-ons. Add-on devices that can only operate as targets. These devices respond to but do not initiate bus cycles.

This implementation is a master device that supports configuration cycles as a target.

PCI Bus Design Criteria (continued)

PCI Bus Signaling Method

For ease of use, this section briefly describes the PCI bus transfer methodology. Refer to the PCI Bus Specification, Revision 2.1, for more detailed information. (Note that the PCI bus standard uses "#" to indicate low-true signals.) Table 1 gives a description of all I/O signals used in this implementation.

Table 1. PCI I/O Signals

Name	Type	Description
PCI Interface		
CLK	I	CLK provides the reference signal for all other PCI interface signals, except RST# and INTA#. The frequency of CLK ranges from dc to 33.333 MHz.
RST#	I	RST# is an input which initializes the FPGA's PCI interface circuitry to a known state. When reset, the PCI output signals are 3-stated, and the open-drain signals, such as SERR#, are floated.
GNT#	I	GNT# indicates to the master that it has control of the PCI bus. Every master has its own GNT# signal.
REQ#	O	REQ# indicates to the arbiter that the master wants to gain control of the PCI bus to perform a transaction.
AD[31:0]	TS	AD[31:0] are time-multiplexed address/data signals, with each bus transaction consisting of an address phase followed by one or more data phases. The FRAME# input signal identifies the start of an address phase. The data phases occur when IRDY# and TRDY# are both asserted.
C/BE[3:0]#	TS*	Command and byte enable inputs are time-multiplexed on C/BE[3:0]#. The bus command (Table 3) is indicated during the address phase of a bus cycle. The byte enables are active during the data phase of a bus transaction.
PAR	TS	PAR is a 3-stated output of even parity calculated on the concatenation of the AD[31:0] and C/BE[3:0]# fields.
FRAME#	S/T/S*	FRAME# is an output from the current bus master that indicates the beginning and duration of a bus operation. When FRAME# is first asserted, the address and command signals are present on AD[31:0] and C/BE[3:0]#. FRAME# remains asserted during the data operation and is deasserted to identify the end of a data operation.
IRDY#	S/T/S*	Initiator ready is output by a bus master to a target to indicate that the bus master can complete a data operation. In write operations, IRDY# indicates that data is on AD[31:0].
DEVSEL#	S/T/S†	Target asserts DEVSEL# as a decode acknowledge that address and bus commands are valid.
TRDY#	S/T/S†	Target ready is a target output that indicates that the current data operation can occur. In a read operation, TRDY# indicates that the target is providing data on AD[31:0].
STOP#	S/T/S†	STOP# is a target output that requests that the bus master stop the current transaction.
IDSEL	I	Initialization device select is a chip select for configuration read or write transactions.
PERR#	S/T/S	Data parity error indicates parity error on a data operation.
SERR#	O/D	SERR# indicates system error and address parity error.
INTA#	O/D	Interrupt A is an active-low interrupt to the host. INTA# must be used for any single-function device requiring an interrupt capability.

* Master out, target in.

† Master in, target out.

Note: See Table 2 for a definition of I/O signal types.

PCI Bus Design Criteria (continued)

Table 1. PCI I/O Signals (continued)

Name	Type	Description
JTAG Signals		
TCK	I	Test clock input used to clock test commands into TMS and test data into TDI.
TMS	I	Test Mode Select is used to specify JTAG boundary-scan instruction or *ORCA*-defined instruction to execute.
TDI	I	Test data input into boundary-scan register, instruction register, or programmable scan ring.
TDO	O	Test data output from bypass register, boundary-scan register, instruction register, or programmable scan ring.
Back-End Signals		
ADRCOMN	I	ADRCOMN is an input from the back-end application used to direct the local access to the address destination register (=1) or the local command/status register (=0).
IRQ	I	IRQ is an active-high signal from the back-end application used to request an interrupt.
LDI[31:0]	I	LDI[31:0] is a 32-bit input data bus from the back-end application.
RAMREGN	I	RAMREGN is an input from the back-end application used to direct the local access to the RAM buffer (=1) or local register set (=0).
RDWRN	I	RDWRN is an input from the back-end application used to indicate that the local access is a read (=1) or a write (=0) operation to the PCI master model.
STROBE	I	STROBE is an active-high input used to indicate a valid cycle is active on the local bus between the back-end application and the PCI master model.
CLKOUT	O	CLKOUT is the buffered output of the PCI CLK input, and as such, ranges from dc to 33.333 MHz.
LDOR[31:0]	O	LDOR[31:0] is a 32-bit data bus to the back-end application.
MASABRT	O	MASABRT identifies that the master issued a master abort resulting from no target responding to the PCI destination address with a valid DEVSEL#.
PCIBUSYR	O	PCIBUSYR is an output used to specify that a master operation is in progress and the back-end interface has been temporarily disabled.
RESETRN	O	RESETRN is an active-low output that is the registered output of the RST# input signal. It is synchronized to CLK.
TRGABRT	O	TRGABRT identifies that the master received a target abort condition during a master transaction.
XFRDONER	O	XFRDONER is an output strobe used to specify that a pending master operation has completed successfully.

* Master out, target in.
† Master in, target out.

Note: See Table 2 for a definition of I/O signal types.

PCI Bus Design Criteria (continued)

Basics

The PCI bus signals consist of a 32-bit multiplexed address/data bus (AD[31:00]) and control signals. The bus is synchronous (all bus devices assert and sample data using the bus clock). Most signals are 3-state and bidirectional and will be driven only when a device is selected. The only purely input pins are CLK, IDSEL (slot select for configuration cycles), RST# (bus reset), and the JTAG interface pins (optional) TDI, TCK, and TMS. The only purely output pins are INTA, B, C, and D# (also optional), which are defined as open-drain signals. In addition, every master has its own set of arbitration signals: REQ# (output only) and GNT# (input only). See Table 2 for a definition of PCI I/O types.

Table 2. PCI I/O Types

Signal Type	Function
I	Standard input.
O	Standard output.
TS	3-stated output or 3-stated bidirectional I/O.
S/T/S	Sustained 3-state is an active-low signal that must be driven high for a minimum of one clock cycle before it is floated. This signal cannot be driven prior to one clock cycle after it has been released. A pull-up resistor is required to sustain the inactive state.
O/D	Open-drain output signals allow multiple outputs to function as a wired-OR.

Signals Sent and Received by Masters

If functioning as a master device, some signals are output-only in relation to the master device. These are REQ# (master desires use of the PCI bus), FRAME# (sampled by all PCI bus devices to detect start and end of a transaction), IRDY# (initiator ready, used to assert master wait-states), and C/BE[3:0]# (command/byte enables, used to classify transactions and identify active byte lanes for a transaction). The master first asserts REQ# to gain control of the bus. After receiving GNT# from the arbiter, the master asserts FRAME# and drives the address on the AD bus. One cycle later, the master asserts IRDY#, indicating it's ready to write data to or read data from the target over the AD bus.

At this stage, the master monitors the control signals driven by the target. When it is the selected device (determined by address/command decoded), a target device will drive the signals DEVSEL# (indicates it has decoded and accepted the transaction), TRDY# (used to assert target wait-states), and STOP# (used to assert target-initiated transaction terminations). When both TRDY# and IRDY# are true, a data word is clocked from a sending to a receiving device. In Figure 1, the signal directions are drawn to illustrate the bus from a master's point of view.

PCI Bus Design Criteria (continued)

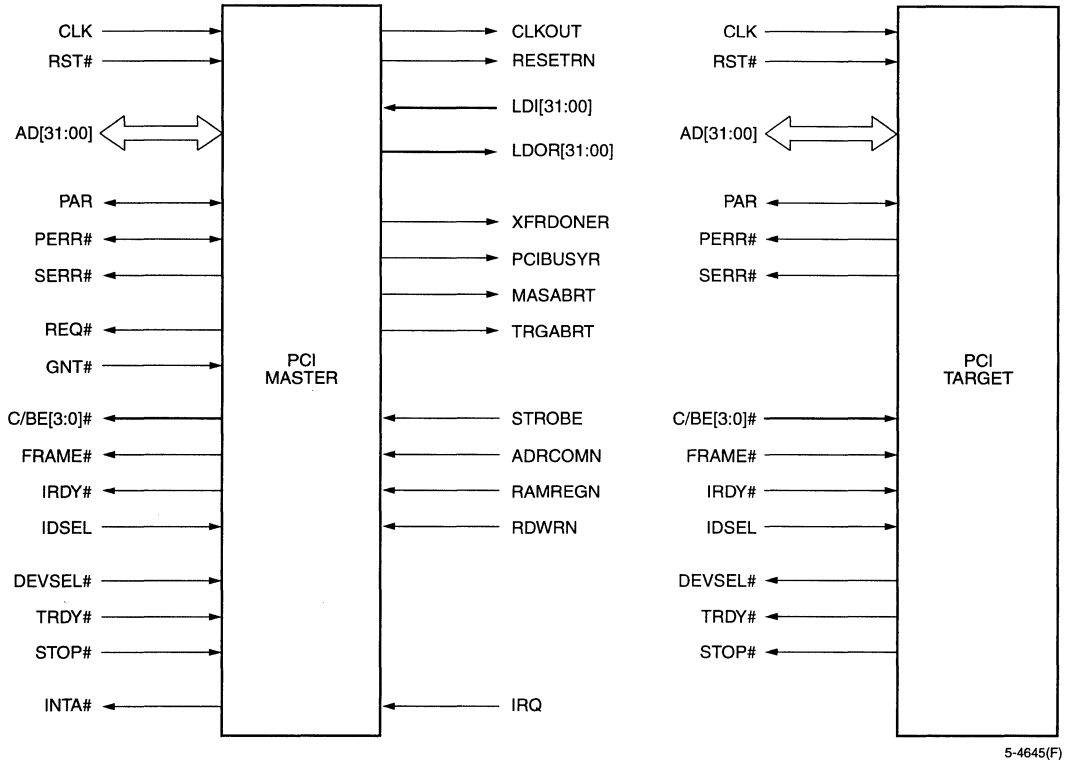


Figure 1. PCI Pin Diagram for Master and Target

Signals Sent and Received by Targets

If functioning as a target device for configuration cycles, some signals are input-only in relation to the target device, since only a master will drive those signals. These are FRAME#, IRDY#, and C/BE[3:0]#. When the target is the selected device, it will drive the signals DEVSEL#, TRDY#, and STOP. These signals are also driven during interrupt acknowledge cycles (not supported in this implementation). When both TRDY# and IRDY# are true, a data word is clocked from a sending to a receiving device. In Figure 1, the signal directions are drawn to illustrate the bus from a target's point of view.

Special 3-State Considerations

The FRAME# signal is sampled by bus targets to detect the start of a transaction. A master that starts a cycle samples DEVSEL# for a response, even though DEVSEL# is not driven until a device actually accepts the cycle. Therefore, some signals that are constantly sampled may also, at times, not be driven. The PCI bus defines these signals (FRAME# and DEVSEL#) in an "off" state with pull-ups.

PCI Bus Design Criteria (continued)

For this approach to work at speed (pull-ups mean slow rise times on open-drain or 3-state outputs), these signals are defined as sustained 3-state (s/t/s). This means that the PCI bus standard requires using one clock cycle to assert the signal false (high) before being 3-stated. The standard also requires that any signal that is being released, such as the master releasing AD after asserting the address on a read operation, is given a full cycle to 3-state before another device can start driving it. This is a turn-around cycle and prevents contention on the bus.

Two output signals, SERR# and INTA#, are defined as open-drain in the PCI standard. This is done by tying the signal to both the input of the 3-state output buffer and the output enable, with the output enable active-low.

Parity

Each cycle asserted on the bus includes parity. Every device that transmits on AD[31:0] must also drive the PAR signal, including masters outputting the address. Since parity on the PCI bus is even, the sum of AD[31:0], C/BE[3:0]#, and PAR must be even. The PAR bit lags the AD bus by one clock.

Parity checking is not required, but can be enabled through the agent's command register. Address parity errors are signaled on the SERR# pin, and data parity errors are signaled on the PERR# pin. Parity errors lag the PAR bit by one clock, and, therefore, the address or data by two clocks.

Bus Commands

The PCI bus has no read/write signals. Control signals are embedded in the command transmitted during the address phase of a cycle. However, a latched C/BE[0]# signal can be used as an RD#/WR indicator. Table 3 indicates which cycles are acknowledged by this implementation and which are not.

In master mode, the PCI model executes the standard memory read and write operations. These commands are valid for both single- and multidata-phase transactions. Memory read line and memory read multiple commands can be included by adding bits to the local control/status register to specify the desired command. These bits would determine the state of the C/BE# signals during the address phase of the master transaction. I/O cycles and master control of configuration cycles are not supported, as these operations are usually reserved for bridges.

Interrupt acknowledge cycles are ignored by this implementation because such cycles are acknowledged only by the device that contains the interrupt controller. Since this is an add-on card implementation, this cycle is not needed. Special cycles are ignored because the specification requires all devices to ignore this type of cycle as far as overt acknowledgment in the form of asserting DEVSEL# is concerned. Special cycles are for broadcast messages from bridge to bus and receiving devices that do anything at all with them will do so by capturing the cycles as they go by. This is easily added to any design, as needed. Dual address cycles are used for transferring 64-bit addresses, which are beyond the scope of this implementation.

The AD[1:0] signals affect how a target responds to a cycle. On configuration cycles, AD[1:0] = 00b indicates a regular configuration cycle, and AD[1:0] = 01b indicates a cycle intended for a bridge to another PCI bus. Multifunction devices with several sets of configuration registers would also decode AD[10:8] on configuration cycles, but this implementation will ignore those bits since it does not implement multiple functions requiring multiple sets of configuration registers.

Table 3. PCI Bus Commands

C/BE[3:0]#	Command	Support by
0000	Interrupt Acknowledge	Ignore
0001	Special Cycle	Ignore
0010	I/O Read	Ignore
0011	I/O Write	Ignore
0100	Reserved	Ignore
0101	Reserved	Ignore
0110	Memory Read	Master
0111	Memory Write	Master
1000	Reserved	Ignore
1001	Reserved	Ignore
1010	Configuration Read	Target
1011	Configuration Write	Target
1100	Memory Read Multiple	Ignore
1101	Dual Address Cycle	Ignore
1110	Memory Read Line	Ignore
1111	Memory Write/Invalidate	Ignore

Exclusive Access

Exclusive access, which allows a master to lock out other masters from accessing a target, is a feature that is required if you implement system memory. However, since this implementation is intended only for add-ons, exclusive access is not needed, so it was not included.

PCI Bus Design Criteria (continued)

PCI Bus Cycle

Figure 2 shows the timing of a basic single data phase PCI bus read cycle. The master first acquires the PCI bus by asserting REQ# and waiting for GNT#. FRAME# low indicates the start of a cycle. Targets latch the address and command on the clock edge on which FRAME# is low and start the decode. In the example given, FRAME# is deasserted and IRDY# is asserted immediately after the address phase, indicating that the master is ready for data. If the master needs more time, IRDY# is delayed and FRAME# is not deasserted until the clock before IRDY# goes true. If the following cycles were multiple data phases, FRAME# would remain asserted until the last data phase.

A target that asserts DEVSEL# will have accepted the transaction. In Figure 2, DEVSEL# is asserted at clock 5, and is sampled by the master on clock 6. This is "fast decode" timing. There are fast (DEVSEL# asserted on clock 5), medium (DEVSEL# asserted on clock 6), and slow (DEVSEL# asserted on clock 7) decode targets. If no DEVSEL# is detected by clock 7, a subtractive decode device may respond. Only one device on the bus can use subtractive decode, in which the decoding device accepts a transaction after detecting that no other device has asserted DEVSEL# by clock 7. This implementation is a "slow decode" device when receiving configuration cycles.

Turn-around cycles are shown in the timing diagram. Because this is a read cycle, the master 3-states AD[31:0] on clock 5, when an address is sampled. The target drives AD[31:0] in clock 6. At the end of the cycle, the data is transferred on clock 7, when both TRDY# and IRDY# are true. (Normally data would be transferred on clock 6, but TRDY# was not true, thereby allowing the target to delay the transfer for a clock cycle.) On this edge, the target releases the AD bus (the data has been sampled by the master), and the master releases C/BE#. PAR is released by the target one clock later, as is DEVSEL#, TRDY# and STOP#. They are driven high on clock 7, so the S/T/S requirement of driving the signal high for one clock before releasing it is met. Though shown in 3-state, IRDY#, TRDY#, DEVSEL#, and STOP# are actually high. As mentioned, pull-up resistors are used to keep S/T/S signals high. They are shown 3-stated to illustrate when devices start and stop driving the signals.

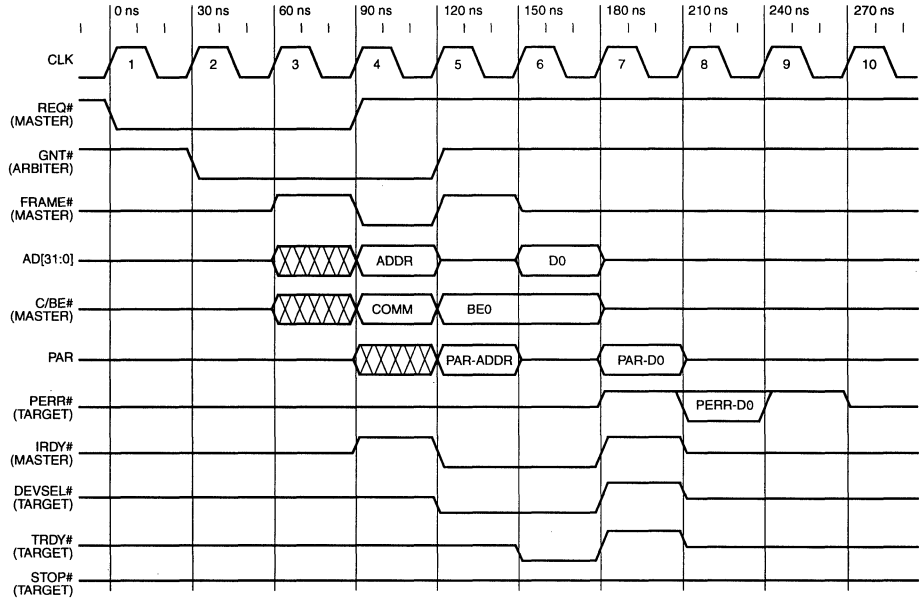


Figure 2. Basic PCI Bus Read Cycle

PCI Bus Design Criteria (continued)

PCI Bus Bursts

Figure 3 illustrates a write operation burst. If both the master and the target support burst operations, the PCI bus can burst data to cause a transfer at each clock cycle, as shown. Note the timing of PERR#, two clocks after the data. Since PERR# can be either 0 or 1, the last data assertion on clock 10 cannot be the "last clock asserted high" cycle for sustained 3-state. One more cycle must be applied high before PERR# can be released. This affects when devices on the next cycle are allowed to drive PERR#, shown here at clock 6.

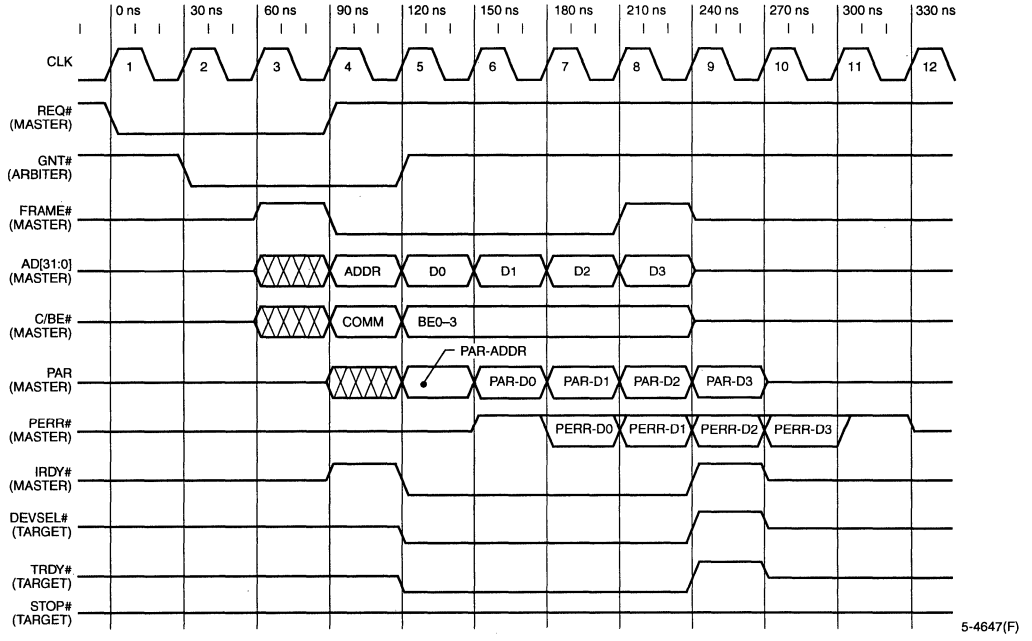


Figure 3. Burst PCI Bus Write Cycle

PCI Bus Design Criteria (continued)

Target Termination

The PCI bus provides a mechanism for targets to disconnect cycles that it cannot support. The three types of termination are disconnect, retry request, and abort, as shown in Table 4. An example of a use of termination is to halt a master's attempt to burst when the target cannot support bursting. Targets **disconnect** by asserting STOP# synchronous with TRDY# deasserting. A **retry** request is initiated when STOP# is asserted with TRDY# already deasserted. The master ends the current data phase without a data transfer occurring and may reattempt the transfer later. A target **abort** request is initiated when STOP# is asserted and DEVSEL# is deasserted on the same clock edge. In a target abort, the target indicates to the master that the transaction should not be attempted again.

Table 4. Target-Initiated Terminations

Termination	DEVSEL#	STOP#	TRDY#	Action
Disconnect	Assert	Assert	Assert	End after current transaction.
Retry	Assert	Assert	Deassert	Data is not transferred. Retry transaction later.
Abort	Deassert	Assert	Don't Care	Fatal error. Data is not transferred.

PCI Bus Configuration Registers

Uniformly defined configuration registers allow the PCI bus add-on cards to be configured by the system without requiring the end user to set jumpers. Configuration registers are accessed via normal PCI bus transaction cycles, but there is no throughput to the back-end application. Configuration cycles have their own command code, as defined in Table 3. The target must detect IDSEL in order for it to accept the cycle.

These configuration registers, shown in Figures 4 through 7, have the following functions:

Address 00h—03h, 08h: Device, Revision and Vendor ID Registers. Read-only registers with ID values that uniquely identify the device to the system and application software. The vendor ID identifies the manufacturer and is assigned by the PCI SIG. The device ID and revision ID are set by the manufacturer.

Address 04h—05h: Command Registers. The command register controls the device, enabling functions such as I/O space, memory space, parity error generation, etc.

Address 06h—07h: Status Register. The status register reports on basic capabilities, events, and errors.

Address 09h—0Bh: Class Code Register. Identifies the type of device to system and application software, with codes that are defined by the standard.

Address 0Dh: Latency Timer Register. Specifies the latency timer count in clock cycles for masters performing burst transactions. The five high-order bits are writable, and the three low-order bits are hardwired to 000b.

Address 2Ch—2Fh: Subsystem Vendor I.D. and Subsystem I.D. These read-only registers are programmed with values to allow differentiation among PCI cards that may use the same PCI controller and, therefore, have the same vendor I.D. and device I.D.

Address 3Ch—3Dh: Interrupt Line and Pin Registers. The interrupt line register is an 8-bit read/write register that is not utilized by the PCI unit itself; rather, the operating system and device drivers transfer information among themselves regarding the interrupt structure via this register. The interrupt pin register is read-only and indicates which PCI bus interrupt pin (INTA#, INTB#, INTC#, or INTD#) the device is using. This application uses INTA#, but it can easily be changed, if desired.

Address 3Eh—3Fh: Minimum Grant and Maximum Latency Registers. These byte wide read-only registers identify time settings in units of 0.25 µs. Minimum grant refers to the length of time that the master needs to use the bus during burst transfers. Maximum latency refers to how often the master needs to use the bus.

PCI Bus Design Criteria (continued)

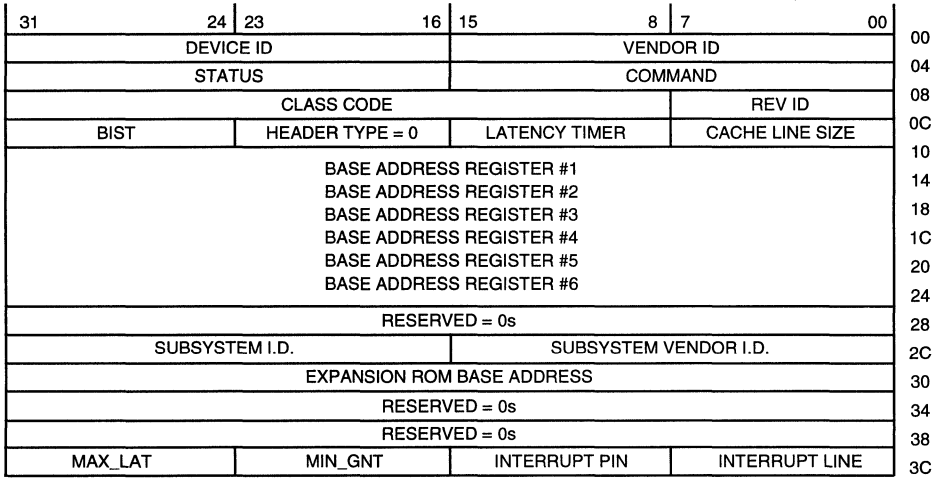


Figure 4. Configuration Registers

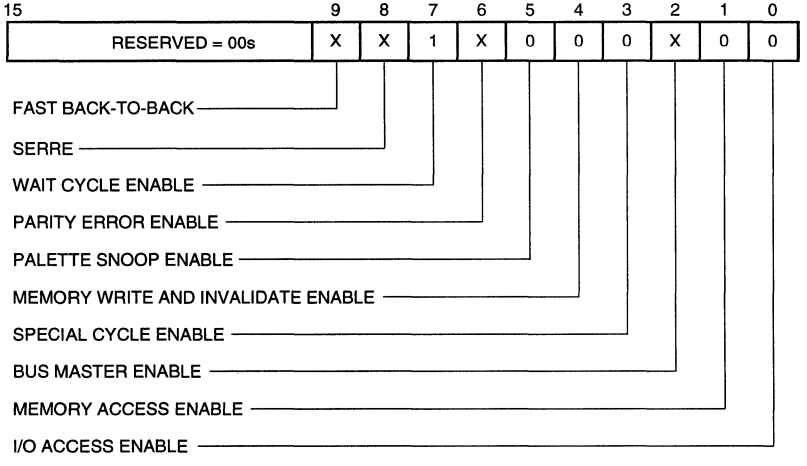


Figure 5. Command Register

PCI Bus Design Criteria (continued)

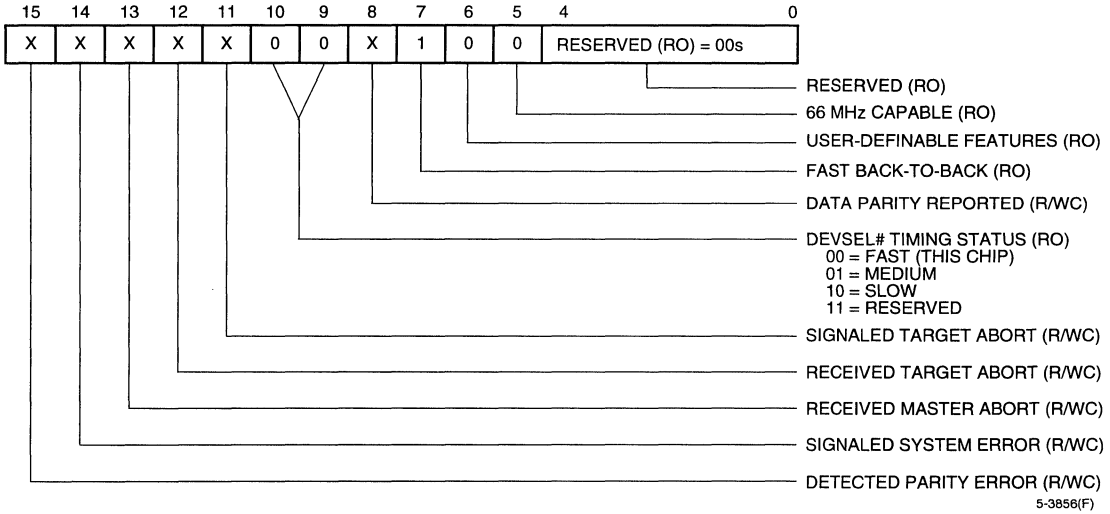


Figure 6. Status Register

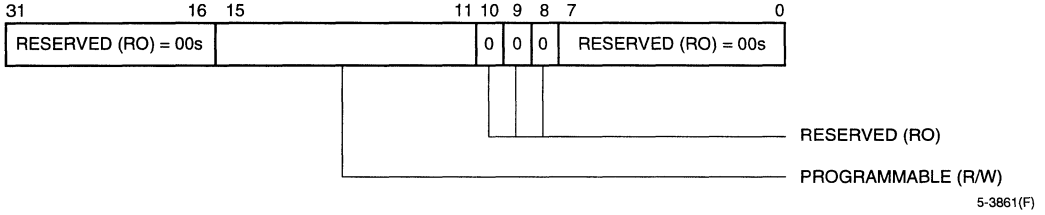


Figure 7. Latency Timer Register

PCI Bus Design Criteria (continued)

PCI Bus I/O Drive Characteristics

The PCI bus is defined as an unterminated CMOS bus. This means that steady-state current is very small, with almost all of it due to the pull-ups on control signals, whereas most current is transient current. The drive characteristics for output drivers on the PCI bus are thoroughly defined for ac as well as dc conditions. A study of Section 4.2 of the PCI bus standard is required to select an FPGA for a PCI bus interface.

The specification of I/O drive characteristics for the PCI bus includes definition of the voltage/current relationship through the driver's active switching range. In Figure 8, the *ORCA* output driver V/I curve is superimposed over the V/I specification for the PCI bus. Many vendors do not provide this data for their drivers or the devices do not meet the specification.

The specification also includes device protection requirements. These relate directly to the unterminated environment and require inputs to handle transients from signal reflections. The specification requires the inputs to withstand an 11 V, 11 ns transient pulse (5.5 V overshoot) and a –5.5 V, 11 ns input undershoot.

While the PCI specification supports driving the bus directly with ASICs and FPGAs, this requires attention to details by the designer. Some FPGA technologies meet the specification, but most do not. For reference, PCI bus 5 V signaling dc and ac requirements as well as the *ORCA* 2C family associated values are indicated in Tables 5 and 6.

PCI Bus Timing Requirements

The PCI clock cycle is defined in Figure 3, with parameter values given in Table 7. To operate at 33.333 MHz, the period cannot exceed 30 ns. In the *ORCA* series, registers are located in programmable logic cells (PLCs). To meet input timing requirements, the *ORCA* 2C Series provides direct inputs to registers, located in the PLCs, from I/O pads. To meet output timing requirements, direct outputs to pads are available from registers in the PLCs to I/O pads.

An effective placement of logic in critical timing paths is essential to meeting PCI timing requirements. The *ORCA* layout tools allow timing to be specified in what is called a preference file. The following excerpts of the preference file (pcimastr.prf) define the critical setup, hold time, and propagation delay requirements.

```
/* CLK FREQUENCY */
FREQUENCY NET CLK 33.3333 MHz;
/* PROP DELAY */
OFFSET OUT COMP "AD<0>" 11.000 NS AFTER
COMP "CLK";
/* SETUP TIME */
OFFSET IN COMP AD<0> 7.000 NS BEFORE COMP
"CLK";
```

The two methods used to verify that the *Verilog* code meets PCI timing requirements are static timing analysis and timing simulation.

The preference file is also used as input into the *ORCA* Foundry *TRACE* static timing analysis tool to identify critical paths to analyze. An example of the timing analysis report file pcimastr.twr is given below.

Design file: pcimastr.ncd

Preference file: pcimastr.prf

Device, speed: or2c15a,3

Report level: error report, limited to 3 items per preference

---

Preference: FREQUENCY NET "manual_CLK" 33.333333 MHz; 8546 items scored, 0 timing errors detected.

Report: 37.594 MHz is the maximum frequency for this preference.

Preference: OFFSET IN COMP "CBEN<0>" 7.000000 NS BEFORE COMP "CLK";17 items scored, 0 timing errors detected.

In the *ORCA* series, speed grades are designated using single-digit prefixes, with faster devices designated by higher numbers. A –3 speed grade is needed to meet the PCI requirements, with other faster speed grades available.

The pcimastr.prf specifies a setup time of 7 ns and a hold time of 0 ns for all PCI input signals. In order to guarantee the 0 ns hold time, the input buffers are placed into a delayed input mode. This still allows the setup time of 7 ns to be met for all *ORCA* 2C devices (the OR2C15A-3 has a system setup time of less than 4.7 ns, for example).

A clock to output valid delay (generally 11 ns) is specified in the pcimastr.prf file for all PCI outputs as well. This is a specification from the system CLK input pin to FFs to output pins which have a 50 pF load. All *ORCA* devices meet this 11 ns specification.

PCI Bus Design Criteria (continued)

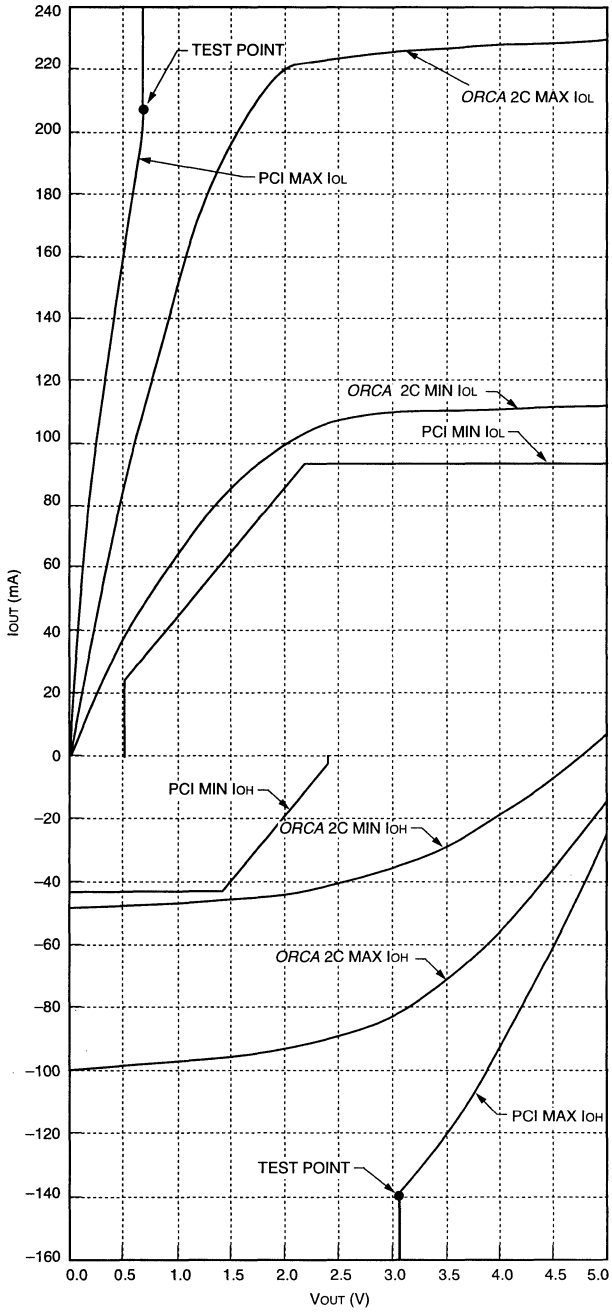


Figure 8. *ORCA* vs. PCI V/I Curves

PCI Bus Design Criteria (continued)

Table 5. PCI Bus I/O dc Specifications

Symbol	Parameter	PCI Specification		*ORCA* 2C		Unit
		Min	Max	Min	Max	
VDD	Supply Voltage	4.75	5.25	4.75	5.25	V
VIH	Input High Voltage	2.0	VDD + 0.5	2.0	VDD + 0.5	V
VIL	Input Low Voltage	–0.5	0.8	–0.5	0.8	V
IIH	Input High Leakage Current	—	70	—	10	µA
IIL	Input Low Leakage Current	—	–70	—	–10	µA
VOH	Output High Voltage	2.4	—	2.4	—	V
VOL	Output Low Voltage	—	0.55	—	0.4	V
CIN*	Input Pin Capacitance	—	10	—	7	pF
CCLK*	Clock Pin Capacitance	5	12	—	7	pF
CIDSEL*	IDSEL Pin Capacitance	—	8	—	7	pF
LPIN	Pin Inductance	—	20	—	†	nH

* Parameter met by *ORCA* but not most other FPGA families.
† This value will vary depending on the package used. Please see the FPGA Data Book (MN95-001FPGA).

Table 6. PCI Bus I/O ac Specifications

Symbol	Parameter	Condition	PCI Specification		*ORCA* 2C		Unit
			Min	Max	Min	Max	
IOH	ac Switching Current High*	0 < VOUT < 1.4 1.4 < VOUT < 2.4 VOUT = 3.1	–44 –44 + (VOUT – 1.4)/0.024 —	— Eq. A† –142	–47 See Fig. 8. —	— See Fig. 8. –84	mA mA mA
IOL	ac Switching Current Low*	VOUT > 2.2 2.2 > VOUT > 0.55 VOUT = 0.71	95 VOUT/0.023 —	— Eq. B† 206	–47 See Fig. 8. —	— See Fig. 8. 116	mA mA mA
ICL	Low Clamp Current	–5 < VIN < –1	–25 + (VIN + 1)/0.015	—	Complies	—	mA
TR	Output Rise Time	0.4 V – 2.4 V	1	5‡	3.1	6.4	V/ns
TF	Output Fall Time	2.4 V – 0.4 V	1	5‡	4.2	9.7	V/ns

* Parameter met by *ORCA* but not most other FPGA families.
† Eq. A: IOH (max) = 11.0 x (VOUT – 5.25) x (VOUT + 2.45); Eq. B: IOL (max) = 78.5 x VOUT x (4.4 – VOUT).
‡ See *PCI Standard*, Revision 2.1, pp. 123—125.

PCI Bus Design Criteria (continued)

Table 7. PCI Bus I/O Timing Specifications

Symbol	Parameter	PCI Specification		*ORCA* 2C		Unit
		Min	Max	Min	Max	
TVAL	Clock to Data Valid*	2	11	—	<10.8	ns
TON	Float to Active Delay	2	—	—	—	ns
TOFF	Active to Float Delay	—	28	—	<28	ns
TSU	Input Setup Time*	7	—	<5	—	ns
TH	Input Hold Time*	0	—	0	—	ns
TCUC	Clock Cycle Time*	30	—	<30	—	ns
THIGH	Clock High Time	12	—	<12	—	ns
TLOW	Clock Low Time	12	—	<12	—	ns

* Parameter met by *ORCA* but not most other FPGA families.

Other PCI Bus Requirements

The standard specifies a maximum trace length of 1.5" from the card edge connector to the PCI device pins for all 32-bit signals. If the optional 64-bit bus is used, the trace length maximum is 2.0". The clock must have a total trace length of 2.5" ± 0.1". Since there are 49 to 51 signals interfacing to the PCI connector, the number of devices should be the minimum. While it might be possible to use two devices co-located adjacent to the PCI connector, this is generally not an effective use of PCB space. Also, the standard requires that only one device load be placed on any signal on the PCI bus. For example, a separate address and data path cannot be used. Most applications should use a single, high pin count device, such as the 208-pin SQFP.

Although not in the scope of this application note, the PCI bus has mechanical and software requirements in addition to the electrical specifications addressed here.

Technology Selection

This section discusses the criteria for the selection of a device for the PCI bus target implementation. In this analysis, gate array, or standard-cell technologies, are not evaluated, although the ability to retarget the design to a low-cost ASIC is a consideration. Only complex programmable logic devices and FPGAs are evaluated in this analysis.

General Considerations

Besides PCI bus compliance, this analysis reviews general issues that affect the suitability of a part.

Density

This design provides the interface between the PCI bus and the back-end application. The density of the 2C Series FPGAs allows large application-specific functionality to be added. About 8,000 of the 15,000 gates in an ATT2C15/OR2C15A have been used to implement the design, and there are currently *ORCA* devices available with >40,000 gates, allowing greater than 80% of the logic to be application-specific.

Tools

The availability of *Verilog* source code allows synthesis using several CAE vendors' *Verilog* synthesis tools.

Functional and Repeatable Routing

This is a qualitative evaluation of a programmable ICs suitability for timing critical design. If the routing's effect on timing delays is not repeatable, then timing requirements may not be met consistently.

Other *ORCA* features include the following:

- Same pinout for all devices from the OR2C04 to the OR2C40 allows package selection based on design requirements. Upgrading to a larger *ORCA* series FPGA has no impact on pinout. This allows the designer to layout the PCB sooner. An added bonus is that a working pinout for the PCI bus is provided with this design, allowing the designer to start board layout even sooner.
- Nibble orientation of PLCs lends itself well to processing 32-bit buses.
- The look-up tables (LUTs) in the PLCs can be used as static RAM. This is useful for configuration registers, internal FIFOs, and/or RAM. Each PLC can be a 16 x 4 RAM.
- Boundary scan (JTAG/*IEEE* 1149.1) capability can be connected to PCI bus JTAG pins.

PCI Bus Parameters to Support

Critical PCI bus parameters that the device must support are as follows:

- Complies with output V/I curve
- Complies with input buffer requirements: 10 pF, <70 µA leakage current
- Input setup time: 7 ns
- Input hold time: 0 ns
- Clock to output valid delay: 2 ns—11 ns
- FMAX: 33.333 MHz

Structurally, the device will need input buffers with one load on any signal. For example, AD[31:0] is input to both an address latch and to parity generation and check circuitry. This must appear as one load to the bus. Also, there needs to be the ability to control the output enables of output buffers with relatively complex combinatorial circuitry.

Verilog Implementation

In this section, the important characteristics of the *Verilog* implementation as they relate to PCI bus features and performance are discussed. In other words, the design is emphasized, not the *Verilog*. This information is fundamentally no different than if the design is implemented using VHDL (also available from Lucent Technologies), equations, or a schematic editor. For details on how to use the *Verilog*, see the *PCI Bus Master* Verilog *Source Code* User's Manual (MN96-008FPGA). Figure 9 is a detailed block diagram of this implementation and should be referred to when reading this section.

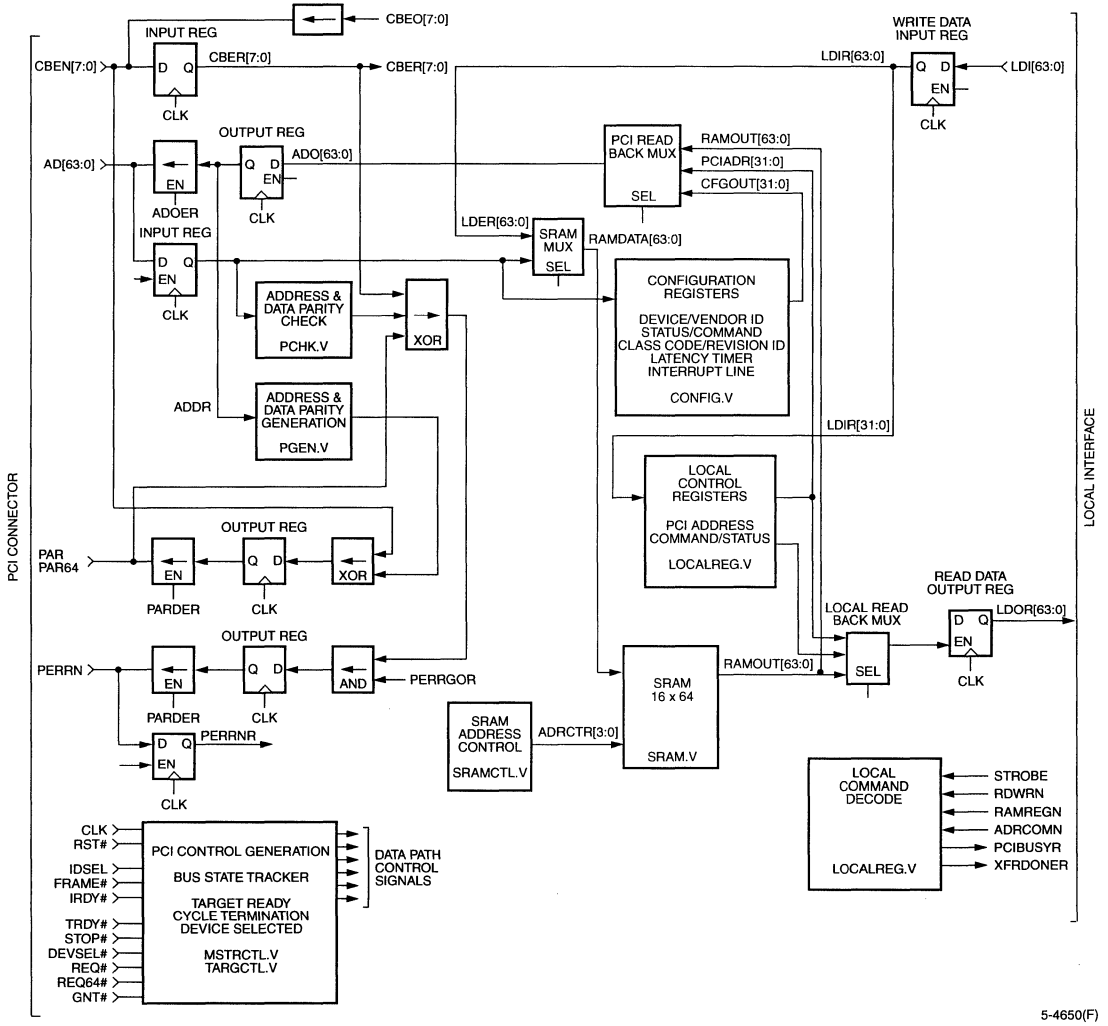


Figure 9. Detailed *Verilog* Master Block Diagram

Verilog Implementation (continued)

Back-End Signals and Data Path Flow

The purpose of this design is to interface the PCI bus to the user's back-end application. The signals provided to the back-end application are given in Table 1.

The back-end application reads data from the LDOR[31:0] bus and writes data to the LDI[31:0] bus. In many applications, these two data buses would interface to logic internal to the FPGA, thereby eliminating the need for a bidirectional data bus. Both buses can be combined into a single bidirectional bus if needed. Data written into the LDI bus is clocked into the LDIR register, where it can load the local register set or be multiplexed through to the RAMINDAT bus to the RAM buffer. Data output from the RAM buffer, RAMOUTDAT, is multiplexed with the local register set to drive the LDO bus. The LDO bus is clocked into the LDOR register, which drives data out to the back-end application.

The local application controls data transfer to the PCI master model through the use of four control signals. These signals facilitate data transfer to or from the PCI model's RAM buffer and local register set. The signals STROBE, ADRCOMN, RAMREGN, and RDWRN are synchronous to the PCI clock. The STROBE signal acts as a chip select to the PCI model. The other three signals determine the source or destination of the access in the PCI model. RDWRN determines whether the access is a read (=1) or write (=0) operation. RAMREGN determines whether the access is to the PCI's RAM buffer (=1) or local register set (=0). If the operation is to the local register set, then ADRCOMN determines whether the access is to the PCI destination register (=1) or the local control/status register. The local interface decode logic is shown below in Table 8.

Table 8. Local Command Decode

CLK	STROBE	ADRCOMN	RAMREGN	RDWRN	MODE
0 or 1	X	X	X	X	—
↑	0	X	X	X	—
↑	1	0	0	0	Com Reg Wr
↑	1	0	0	1	Com Reg Rd
↑	1	1	0	0	Addr Reg Wr
↑	1	1	0	1	Addr Reg Rd
↑	1	X	1	0	RAM Write
↑	1	X	1	1	RAM Read

Key:

↑ Rising edge
X Don't care
Com Reg Wr: Command Register Write
Com Reg Rd: Command Register Read
Addr Reg Wr: Address Register Write
Addr Reg Rd: Address Register Read
RAM Write: RAM Buffer Write
RAM Read: RAM Buffer Read

Verilog Implementation (continued)

Four back-end status signals inform the local processor of the state of the PCI transfer in progress. The signal PCIBUSYR identifies when a PCI master transfer is in progress, which prevents any operation from the local side. XFRDONER is a single cycle pulse that signifies the successful completion of the latest transaction. MASABRT indicates that no target responded with DEVSEL# to the PCI address broadcast at the start of the master's bus access. TRGABRT indicates that the target issued an abort during a data phase of the bus transfer by asserting STOP# and deasserting DEVSEL#. Both of these indicators are pulsed signals that preclude any retry by the PCI master model.

The timing for read and write operations is relatively simple. Figure 10 shows the back-end signals for a multiple data phase write. Figure 11 shows a multiple data phase read operation. The PCI bus signals are not shown, because the two interfaces are disconnected. Transactions at one interface do not have an immediate effect on the other interface.

Write Operations

A master write operation is comprised of two independent transactions. First, the local processor or state machine configures the local register set and fills the RAM buffer. Second, the PCI master model bursts this data over the PCI bus without inserting master wait-states. If the size of the transfer is greater than a single buffer (i.e., 16 words), then the local processor and PCI master would alternate filling then emptying the buffer. This sequence continues until the entire DMA operation is completed or an abort condition occurs.

The local processor starts the procedure by writing the PCI destination to the PCI address register (clock cycle 1 in Figure 10. Next, the local processor sets the local command/status register with the word count of the transfer (bits 31—14), the direction (bit 1 = 0), and the start enable (bit 0 = 1) (clock cycle 2). If the remaining transfer length is 16 words or greater, then the local processor writes the first 16 data words to the RAM buffer. If the remaining transfer length is less than 16 words, then the local processor writes that number of data words to the buffer (clock cycles 3—6).

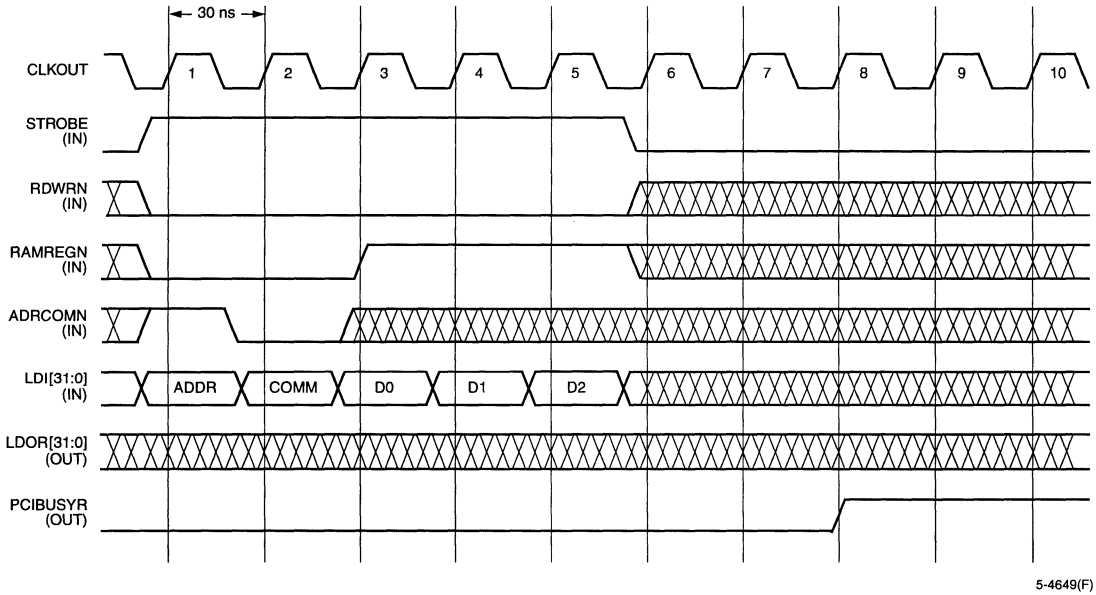


Figure 10. PCI Bus Multidata Phase Write with Back-End Signals

Verilog Implementation (continued)

At this stage, the PCI master model assumes control of the process, as was shown in Figure 3 previously. The master first asserts REQ# to the bus arbiter to gain control of the bus. Upon receiving GNT#, the PCI master model selects the PCI address register as the source for the ADO bus. The master also drives FRAME# (to deasserted state), AD, and C/BE#. The process of driving the bus and control lines for one cycle before asserting FRAME# is known as address stepping. The master model implements address stepping in order to preactivate the bus drivers, since it takes longer to activate a high-impedance line than it takes to switch it.

The address phase occurs during the following cycle when FRAME# is asserted, the ADO bus (containing the PCI address) is clocked through the ADOR register to the AD output pins, and REQ# is deasserted. During this cycle, the AD and C/BE# buses contain the address and memory write command (C/BE# = 0111), respectively. The master model selects the RAM buffer output containing the first data word as the source of the ADO bus.

The first data phase occurs during the next cycle as IRDY# is asserted, and the data is clocked through the ADOR bus to the AD output pins. The model does not insert any master wait-states, so burst transfers operate at the target's maximum bandwidth. A target inserts wait-states by deasserting TRDY#, which disables the clock enable of the ADOR output register. The current data word is then frozen in the output register until TRDY# is asserted.

Every data phase completes when both IRDY# and TRDY# are asserted. This state causes the RAM address to increment, which reads the next word in the RAM buffer. When the burst is almost complete, FRAME# is deasserted to indicate the transaction is in the final data phase. After this phase completes, IRDY# deasserts, and FRAME#, AD, and C/BE# are 3-stated. During the following cycles, IRDY# is 3-stated, and PCIBUSYR is deasserted. The local processor can now continue the DMA transfer by refilling the model's RAM buffer.

Read Operations

A master read operation is comprised of three independent transactions. First, the local processor or state machine configures the local register set. Second, the PCI master model reads this burst data over the PCI bus without inserting master wait-states. Third, the local processor reads the contents of the RAM buffer containing the PCI read data. If the size of the transfer is greater than a single buffer (i.e., 16 words), then the PCI master and local processor would alternate filling then emptying the buffer. This sequence continues until the entire DMA operation is completed or an abort condition occurs.

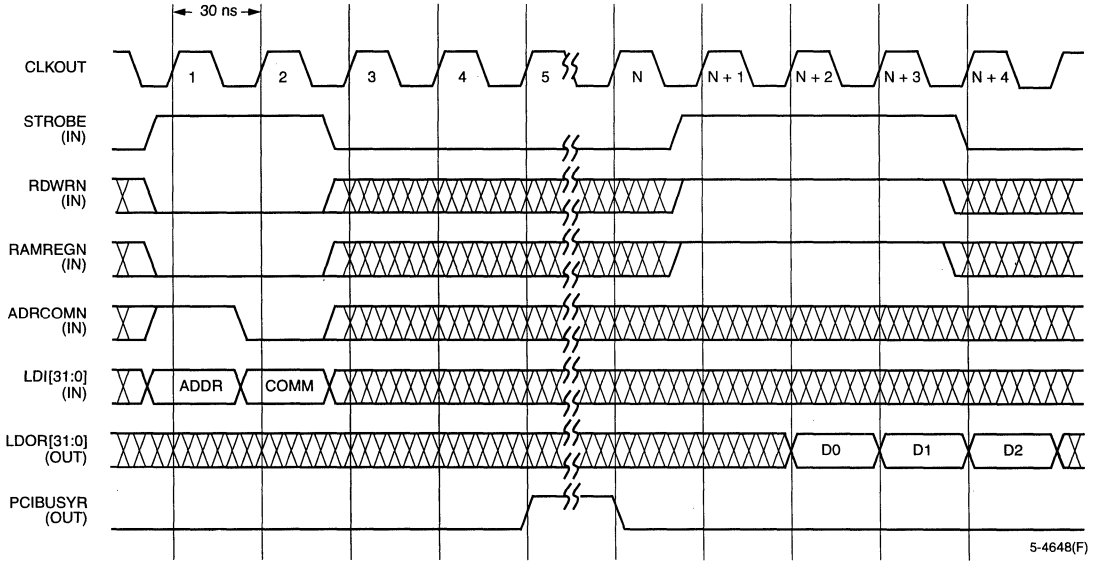


Figure 11. PCI Bus Multidata Phase Read with Back-End Signals

Verilog Implementation (continued)

The local processor starts the procedure by writing the PCI source address to the PCI address register (clock cycle 1 in Figure 11. Next, the local processor sets the local command/status register with the word count of the transfer (bits 31—14), the direction (bit 1 = 1), and the start enable (bit 0 = 1) (clock cycle 2).

At this stage, the PCI master model assumes control of the process, as previously shown in Figure 2. The master first asserts REQ# to the bus arbiter to gain control of the bus. Upon receiving GNT#, the PCI master model selects the PCI address register as the source for the ADO bus. The master also drives FRAME# (to deasserted state), AD, and C/BE#. The address phase occurs during the following cycle, when FRAME# is asserted, the ADO bus (containing the PCI address) is clocked through the ADOR register to the AD output pins, and REQ# is deasserted. During this cycle, the AD and C/BE# buses contain the address and memory read command (C/BE# = 0110) respectively.

The PCI master now performs a turn-around cycle by 3-stating the AD bus and asserting IRDY#. The first data phase can occur as soon as the next cycle if the target asserts TRDY# and drives the data on the AD bus. The model does not insert any master wait-states, so burst transfers operate at the target's maximum bandwidth.

Every data phase completes when both IRDY# and TRDY# are asserted. This state causes the RAM address to increment, which writes the next word into the RAM buffer. When the burst is almost complete, FRAME# is deasserted to indicate the transaction is in the final data phase. After this phase completes, IRDY# deasserts, and FRAME# and C/BE# are 3-stated. During the following cycles, IRDY# is 3-stated, and PCIBUSYR is deasserted. The local processor can now continue the DMA transfer by reading the model's RAM buffer (clock cycles 5—8).

Back-End Signal Timing

Figures 10 and 11 are timing diagrams for write and read operations at the back-end interface. For write operations, STROBE must be active, and ADRCOMN, RAMREGN, and RDWRN must all be valid along with the data on LDI for each clock edge. For read operations, the data on the LDOR bus is valid two clock cycles after STROBE is active. The back-end interface is disabled whenever PCIBUSYR is active.

The four output status signals are all synchronous to the clock edge, meaning that setup and hold time are assured. The interrupt request logic detects a rising edge on IRQ, and is not level sensitive. For more detailed information, consult the *PCI Bus Master* Verilog *Source Code* User's Manual (MN96-008FPGA).

Parity Checking

ADIR[31:0] and C/BE[3:0]# are routed to a parity checker. The output of the parity checker is XORed with the PAR bit, which is input on the clock following the address or data. The result is ANDed with the parity enable bit from the command register and is routed to the output register. The PERR# signal or SERR# signal is then clocked out of the parity register, depending on whether this is the address phase or a data phase.

Parity Generation

For master write and target read operations, parity is generated and routed to the output register that asserts the PAR signal. CBEOUTR[3:0] is used in the generation of parity.

Verilog Implementation (continued)

Master State Machine

Detailed understanding of the PCI bus control mechanism requires reading the *mstrctl.v* module of the *Verilog* code, whose function is summarized here.

The control mechanism must assert the following signals:

- PCI bus control output enables (OE) for FRAME#, C/BE#, and IRDY#
- AD bus output enable
- PAR output enable
- PERR# output enable
- REQ#, FRAME#, and IRDY#
- PCIBUSYR, XFRDONER, MASABRT, and TRGABRT

FRAME# and C/BE# OE. The output enable for FRAME# and C/BE# turns on when the state machine receives GNT# from the arbiter and no other master is currently running a PCI transaction. The drive is turned off one cycle after FRAME# is deasserted with IRDY# asserted, which identifies the end of a transaction. If the arbiter is merely using this master for bus parking, then the output enable turns off one clock after GNT# is deasserted.

IRDY# OE. This signal follows the FRAME# OE by one clock cycle.

AD Bus OE. The output enable for the AD turns on when the state machine receives GNT# from the arbiter and no other master is currently running a PCI transaction. It is turned off on write operations on the clock edge when FRAME# is first detected deasserted with IRDY# low. It is turned off on read operations on the cycle after the address phase (turn-around cycle). If the arbiter is merely using this master for bus parking, then the output enable turns off on the clock edge where GNT# is deasserted.

PAR OE. This signal follows the AD bus OE by one clock cycle.

PERR# OE. This signal follows the start of the transaction by two clock cycles for read operations.

REQ#. This signal goes true for PCI write operations after the local processor has either filled the RAM buffer or reached the terminal count of the DMA operation. It also goes true for PCI read operations after the local processor has either set the start bit in the local control register or read the last word from the RAM buffer. REQ# goes active for all PCI transactions where a retry or latency time-out occurs. The signal deasserts during the same cycle that FRAME# asserts.

FRAME#. This signal goes true one cycle after the registered version of GNT# is asserted when REQ# is active. Several conditions can cause FRAME# to deassert. The conditions include completion of burst transfer, master abort, latency time-out, and target cycle termination by asserting STOP#.

IRDY#. This signal goes true one cycle after FRAME# is asserted. IRDY# goes false on the completion of the last data phase.

PCIBUSYR. This signal informs the local processor that the PCI master is currently performing a transaction on the PCI bus and the local interface is disabled. PCIBUSYR is asserted one cycle after REQ# goes active and is deasserted one cycle after XFRDONER, MASABRT, or TRGABRT assert.

XFRDONER. This signal informs the local processor that the PCI master successfully completed a burst transaction on the PCI bus and the local interface is available. XFRDONER is asserted one cycle after the last data phase and is active for only one clock cycle.

MASABRT. This signal informs the local processor that no target responded to the PCI master's address with an active DEVSEL# within four clock cycles of the address. A counter tracks the number of clock cycles and asserts MASABRT when its terminal count is reached and DEVSEL# has not been asserted. No retry of the transaction to the faulty address is attempted.

TRGABRT. This signal informs the local processor that the target issued a target abort condition by asserting STOP# and deasserting DEVSEL. No retry of the transaction to the target is attempted.

Verilog Implementation (continued)

Target State Machine

Detailed understanding of the PCI bus control mechanism requires reading the targctl.v module of the *Verilog* code, whose function is summarized here.

The control mechanism must assert the following signals:

- PCI bus control output enable (OE) for DEVSEL#, TRDY#, and STOP#
- AD bus output enable
- PAR output enable
- PERR# and SERR# output enable
- DEVSEL#, TRDY#, and STOP#

Control OE. The output enable for DEVSEL#, TRDY#, and STOP# must turn on when the state machine has decided to accept a cycle. This occurs when the state machine detects a configuration cycle and IDSEL. The drive is turned off one cycle after FRAME# is deasserted with IRDY# asserted, which is a unique combination that always means the end of a transaction.

AD Bus OE. The output enable for the AD bus is turned on when the cycle is a read (C/BE[0]# = 0), one clock after the control OE is turned on. It is turned off when the clock edge FRAME# is first detected deasserted with IRDY# low.

PAR OE. This signal follows the AD bus OE by one clock cycle.

PERR# OE. This signal follows the controls OE by two clock cycles for write operations.

DEVSEL#. This signal goes true one cycle after the control OE is asserted, and goes false on the clock edge that FRAME# is deasserted with IRDY# asserted.

TRDY#. This signal goes true one cycle after the control OE is asserted for write operations and two cycles after the control OE is asserted for read operations. TRDY# goes false on the clock edge that FRAME# is deasserted with IRDY# asserted.

STOP#. This signal is asserted when the state machine suppresses burst activity on configuration cycles. STOP# is asserted on the first data phase and deasserted after FRAME# is deasserted.

Interrupt Support

There is no timing definition for the interrupt pin on the PCI bus. The PCI master model detects a rising edge on the IRQ signal and asserts INTA# on the PCI bus. The interrupt is cleared when the master's configuration register space is accessed. There are other mechanisms the designer should consider for deactivating INTA#. For example, a specific configuration register bit could be added, which would reset the interrupt signal.

Caveats and Suggestions

Some functions not included in this design are discussed here.

Byte Enables: Byte enables for write operations can be incorporated by adding four bits to the RAM buffer and the LDI data path. These four bits would be clocked into the CBEOUTR register at the same time as the data word is clocked into the ADOR register. All logic affecting the data bus in the RAM buffer and output registers would also apply to the byte enables.

PCI Bus Controlled Master: If a local processor or state machine is not available, then the local register set can be moved into the configuration space in the *config.v* module. Decode logic would then be needed for these registers. A new state machine would fill or empty the buffer based on status signals such as XFRDONER.

Secondary SRAM Buffer: If the design application requires an effective bandwidth greater than 66 Mbytes/s, then a second SRAM buffer can be added. This implementation would allow the master to burst data across the bus while the local processor fills the other RAM buffer. Currently, the local processor must wait for the master to empty the buffer before refilling it. Additional logic would be needed to switch between the two buffers and multiplex the correct data to the desired interface.

Dual-Port FIFO RAM: Another way to increase the bandwidth is to replace the SRAM buffer with a dual-port FIFO. This allows both the PCI Bus and the local bus to access this buffer at the same time. The control signals IRDY# and PCIBUSYR can then be controlled using FIFO status signals. Dual-port FIFOs and RAMS are available in the OR2CxxA/OR2TxxA devices.

Design Verification Requirements

The PCI SIG defines the verification process for PCI bus designs in their PCI compliance checklist. We have taken the requirements described in that document and created our simulation test bench around the checklist.

Generic Test Parameters and the Compliance Checklist

Below is a summary of the checklist categories. Scenarios 1.1 through 1.14 are for master devices, and scenarios 2.1 through 2.13 are for target devices. Our master device accepts only configuration cycles as a target; therefore, only five of the 13 target scenarios are supported. The scenarios for features we do not support have been lined through. This does not affect our PCI model's compliance to the checklist rules.

1.1 PCI Device Speed and Master Abort Cycles

1.2 PCI Bus Target Abort Cycles

1.3 PCI Bus Target Retry Cycles

1.4 PCI Bus Target Disconnect Cycles

1.5 PCI Bus Multidata Phase Target Abort Cycles

1.6 PCI Bus Multidata Phase Target Retry Cycles

1.7 PCI Bus Multidata Phase Target Disconnect Cycles

1.8 PCI Bus Multidata Phase and TRDY# Cycles

1.9 PCI Bus Data Parity Error Single Cycles

1.10 PCI Bus Data Parity Error Multidata Phase Cycles

1.11 PCI Bus Master Time-out

1.12 Target Lock (not included)

1.13 PCI Bus Master Parking

1.14 PCI Bus Master Arbitration

~~2.1 Target Reception of an Interrupt Acknowledge Cycle~~

~~2.2 Target Reception of a Special Cycle~~

~~2.3 Target Reception of Address and Parity Errors on a Special Cycle~~

~~2.4 Target Reception of I/O cycles with Legal and Illegal Byte Enables~~

2.5 Target Ignores Reserved Commands

2.6 Target Receives Configuration Cycles

~~2.7 Target Receives I/O Cycles with Address and Data Parity Errors~~

2.8 Target Gets Configuration Cycles with Address and Data Parity Errors

~~2.9 Target Receives Memory Cycles~~

~~2.10 Target Gets Memory Cycles with Address and Data Parity Errors~~

2.11 Target Gets Fast Back-to-Back Cycles (Same Target Writes Only)

~~2.12 Target Performs Exclusive Access Cycles~~

2.13 Target Gets Cycles with IRDY Used for Data Stepping

The scenarios excluded in this list assume that this implementation doesn't support the following features/ cycle types:

- Interrupt acknowledge
- Cache support
- Special cycles
- 64-bit address/data
- Exclusive access (lock)
- Memory or I/O target cycles

Timing Sequence for Compliance Testbenches

All tests run at the maximum operating frequency of 33.333 MHz. The testbenches validate functional compliance to the PCI SIG Compliance Checklist and are not designed to verify ac timing parameters. It is crucial that you specify all timing parameters of the PCI Bus Specification in the place-and-route phase of your design process. Static timing analysis will ensure compliance with the PCI timing requirements. An *ORCA* Foundry timing preference file that works with both the place & route software and the static timing analyzer are included in the Lucent Technologies PCI Master Customer Solution Core.

Signals driven by a testbench to the PCI master change on the falling edge of the PCI clock. Both PCI and back-end outputs are strobed 1 ns before the rising edge of the PCI clock. Simulation results are continually compared to data files containing previously generated test vectors. This structure permits *Verilog* simulation and comparison of both pre- and postlayout designs.

Design Verification Requirements
(continued)

Procedures Common to All Master Tests

The master scenarios test both read and write operations executed by the master. The first seven master scenarios test all speeds of the intended target. A fast target responds by asserting DEVSEL# on the following clock after FRAME# is sampled active. A medium target responds two clock cycles after FRAME#, and slow (three clocks) and subtractive decode targets (four clocks) respond accordingly.

The master is initialized before starting each scenario's test sequence. This setup is comprised of a PCI reset and then writing two registers in the master's configuration space. First, the PCI reset signal RST# is asserted for one clock period. Second, the latency timer register is set to a large value to prevent latency time-outs from interfering with test results. Third, the status/control register is written to clear all status bits and to enable master and parity modes.

Test Definitions

In each of the scenarios defined below, the purpose of the test is given, followed by the actual steps of the test. Unimplemented scenarios are not included. Only signal testing that is specific to the test is mentioned in the definition. The timing that is used for these tests directly relates to this PCI bus master/target implementation.

Scenario 1.1—PCI Device Speed and Master Abort Cycles

This scenario tests for proper execution of single data memory cycles to targets of all speeds. For write cycles, the testbench will verify that data written to the PCI master's internal RAM via the local interface is driven onto the PCI bus during the data phase of the cycle. For read cycles, data driven onto the PCI bus by the testbench during the data phase must be read via the local interface from the PCI master's internal RAM on completion of the PCI read transaction. Configuration, I/O, special, and interrupt acknowledge cycles are not tested because they are specific to motherboard-type masters. In addition, master abort cycles are tested, and the "master abort" bit is checked in the configuration status register.

Start with a cycle to set the latency timer register and enable master mode using the configuration registers. All the following tests assume that the address asserted is 0x12345678.

Perform single phase memory write cycle to fast decode target with data = 0x11111111.

Perform single phase memory read cycle to fast decode target with data = 0x5555aaab.

Perform single phase memory write cycle to medium decode target with data = 0x11111111.

Perform single phase memory read cycle to medium decode target with data = 0x5555aaab.

Perform single phase memory write cycle to slow decode target with data = 0x11111111.

Perform single phase memory read cycle to slow decode target with data = 0x5555aaab.

Perform single phase memory write cycle to subtractive decode target with data = 0x11111111.

Perform single phase memory read cycle to subtractive decode target with data = 0x5555aaab.

Perform single phase memory write cycle to slower than subtractive target with data = 0x11111111.

Verify that master abort bit is set in configuration status register, and then clear register.

Perform single phase memory read cycle to slower than subtractive target with data = 0x5555aaab.

Verify that master abort bit is set in configuration status register.

Test Definitions (continued)

Scenario 1.2—PCI Bus Target Abort Cycles

This scenario tests for proper execution of single data memory cycles terminated with target abort. After every test, the target abort bit is checked in the master's configuration status register, which is then cleared. The master must not retry the operation after a target abort is received. Configuration, I/O, and interrupt acknowledge cycles are not tested because they are specific to motherboard-type masters. All tests are executed for both read and write operations.

Start with a cycle to set the latency timer register and enable master mode using the configuration registers. All the tests following assume that the address asserted is 0x12345678.

Perform single phase memory write cycle to fast decode target with data = 0x11111111.

Verify that target abort bit is set in configuration status register, and then clear register.

Perform single phase memory read cycle to fast decode target with data = 0x5555aaab.

Verify that target abort bit is set in configuration status register, and then clear register.

Perform single phase memory write cycle to medium decode target with data = 0x11111111.

Verify that target abort bit is set in configuration status register, and then clear register.

Perform single phase memory read cycle to medium decode target with data = 0x5555aaab.

Verify that target abort bit is set in configuration status register, and then clear register.

Perform single phase memory write cycle to slow decode target with data = 0x11111111.

Verify that target abort bit is set in configuration status register, and then clear register.

Perform single phase memory read cycle to slow decode target with data = 0x5555aaab.

Verify that target abort bit is set in configuration status register, and then clear register.

Perform single phase memory write cycle to subtractive decode target with data = 0x11111111.

Verify that target abort bit is set in configuration status register, and then clear register.

Perform single phase memory read cycle to subtractive decode target with data = 0x5555aaab.

Verify that target abort bit is set in configuration status register.

Scenario 1.3—PCI Bus Target Retry Cycles

This scenario tests for proper execution of single data memory retry cycles to targets of all speeds. The master should receive a retry termination from the target and then reattempt the access with the same address. Data driven on PCI master's local interface must match data driven on the PCI bus during the data phase of the PCI read or write operation. The retry termination should be undetectable by the back-end application. Configuration, I/O, and interrupt acknowledge cycles are not tested because they are specific to motherboard-type masters.

Start with a cycle to set the latency timer register and enable master mode using the configuration registers. All the tests following assume that the address asserted is 0x12345678.

Perform single phase memory write cycle to fast decode target with data = 0x11111111.

Perform single phase memory read cycle to fast decode target with data = 0x5555aaab.

Verify that master abort bit is not set in configuration status register.

Perform single phase memory write cycle to medium decode target with data = 0x11111111.

Perform single phase memory read cycle to medium decode target with data = 0x5555aaab.

Verify that master abort bit is not set in configuration status register.

Perform single phase memory write cycle to slow decode target with data = 0x11111111.

Perform single phase memory read cycle to slow decode target with data = 0x5555aaab.

Verify that master abort bit is not set in configuration status register.

Perform single phase memory write cycle to subtractive decode target with data = 0x11111111.

Perform single phase memory read cycle to subtractive decode target with data = 0x5555aaab.

Verify that master abort bit is not set in configuration status register.

Test Definitions (continued)

Scenario 1.4—PCI Bus Target Disconnect Cycles

This scenario tests for proper execution of single data memory disconnect cycles to targets of all speeds. The master should receive a disconnect termination from the target and then not reattempt the access. Data driven on PCI master's local interface must match data driven on the PCI bus during the data phase of the PCI read or write operation. Configuration, I/O, and interrupt acknowledge cycles are not tested because they are specific to motherboard-type masters.

Start with a cycle to set the latency timer register and enable master mode using the configuration registers. All the tests following assume that the address asserted is 0x12345678.

Perform single phase memory write cycle to fast decode target with data = 0x11111111.

Perform single phase memory read cycle to fast decode target with data = 0x5555aaab.

Perform single phase memory write cycle to medium decode target with data = 0x11111111.

Perform single phase memory read cycle to medium decode target with data = 0x5555aaab.

Perform single phase memory write cycle to slow decode target with data = 0x11111111.

Perform single phase memory read cycle to slow decode target with data = 0x5555aaab.

Perform single phase memory write cycle to subtractive decode target with data = 0x11111111.

Perform single phase memory read cycle to subtractive decode target with data = 0x5555aaab.

Scenario 1.5—PCI Bus Multidata Phase Target Abort Cycles

This scenario tests for proper execution of multidata phase memory cycles terminated with target abort. The termination occurs after the first data phase of the write or read transaction. After every test, the target abort bit is checked in the master's configuration status register, which is then cleared. The master must not retry the operation after a target abort is received. Configuration cycles are not tested because they are specific to motherboard-type masters. Dual address cycles are not tested, because the master has 32-bit addressing only. Memory read multiple, memory read line, and memory write and invalidate cycles are not tested, because cache mode is not supported. All tests are executed for both read and write operations.

Start with a cycle to set the latency timer register and enable master mode using the configuration registers. All the tests following assume that the starting address is 0x12345678.

Perform multidata phase memory write cycle to fast decode target with data = 0x11111111, 0x22222222, 0x33333333.

Verify that master does not retry operation and target abort bit is set in configuration status register, and then clear register.

Perform multidata phase memory read cycle to fast decode target with data = 0x5555aaab.

Verify that master does not retry operation and target abort bit is set in configuration status register, and then clear register.

Perform multidata phase memory write cycle to medium decode target with data = 0x11111111, 0x22222222, 0x33333333.

Verify that master does not retry operation and target abort bit is set in configuration status register, and then clear register.

Perform multidata phase memory read cycle to medium decode target with data = 0x5555aaab.

Verify that master does not retry operation and target abort bit is set in configuration status register, and then clear register.

Perform multidata phase memory write cycle to slow decode target with data = 0x11111111, 0x22222222, 0x33333333.

Verify that master does not retry operation and target abort bit is set in configuration status register, and then clear register.

Perform multidata phase memory read cycle to slow decode target with data = 0x5555aaab.

Verify that master does not retry operation and target abort bit is set in configuration status register, and then clear register.

Perform multidata phase memory write cycle to subtractive decode target with data = 0x11111111, 0x22222222, 0x33333333.

Verify that master does not retry operation and target abort bit is set in configuration status register, and then clear register.

Perform multidata phase memory read cycle to subtractive decode target with data = 0x5555aaab.

Verify that master does not retry operation and target abort bit is set in configuration status register.

Test Definitions (continued)

Scenario 1.6—PCI Bus Multidata Phase Target Retry Cycles

This scenario tests for proper execution of multidata phase memory retry cycles to targets of all speeds. The master should receive a retry termination from the target and then reattempt the access with the same address. Data driven on PCI master's local interface must match data driven on the PCI bus during the three data phases of the PCI read or write transaction. The retry termination should be undetectable by the back-end application. Configuration cycles are not tested because they are specific to motherboard-type masters. Dual address cycles are not tested because the master has 32 bit addressing only. Memory read multiple, memory read line, and memory write and invalidate cycles are not tested because cache mode is not supported.

4

Start with a cycle to set the latency timer register and enable master mode using the configuration registers. All the tests following assume that the starting address is 0x12345678.

Perform multidata phase memory write cycle to fast decode target with data = 0x11111111, 0x22222222, 0x33333333.

Perform multidata phase memory read cycle to fast decode target with data = 0x00000001, 0x00000002, 0x00000003.

Perform multidata phase memory write cycle to medium decode target with data = 0x11111111, 0x22222222, 0x33333333.

Perform multidata phase memory read cycle to medium decode target with data = 0x00000004, 0x00000005, 0x00000006.

Perform multidata phase memory write cycle to slow decode target with data = 0x11111111, 0x22222222, 0x33333333.

Perform multidata phase memory read cycle to slow decode target with data = 0x00000007, 0x00000008, 0x00000009.

Perform multidata phase memory write cycle to subtractive decode target with data = 0x11111111, 0x22222222, 0x33333333.

Perform multidata phase memory read cycle to subtractive decode target with data = 0x0000000a, 0x0000000b, 0x0000000c.

Scenario 1.7—PCI Bus Multidata Phase Target Disconnect Cycles

This scenario tests for proper execution of multidata phase memory disconnect cycles to targets of all speeds. The master should receive a disconnect termination from the target and then continue the access with the next address. Data driven on PCI master's local interface must match data driven on the PCI bus during the three data phases of the PCI read or write transactions. The disconnect termination should be undetectable by the back-end application. Configuration cycles are not tested because they are specific to motherboard-type masters. Dual address cycles are not tested because the master has 32-bit addressing only. Memory read multiple, memory read line, and memory write and invalidate cycles are not tested because cache mode is not supported.

Start with a cycle to set the latency timer register and enable master mode using the configuration registers. All the tests following assume that the starting address is 0x12345678.

Perform multidata phase memory write cycle to fast decode target with data = 0x11111111, 0x22222222, 0x33333333.

Perform multidata phase memory read cycle to fast decode target with data = 0x00000001, 0x00000002, 0x00000003.

Perform multidata phase memory write cycle to medium decode target with data = 0x11111111, 0x22222222, 0x33333333.

Perform multidata phase memory read cycle to medium decode target with data = 0x00000004, 0x00000005, 0x00000006.

Perform multidata phase memory write cycle to slow decode target with data = 0x11111111, 0x22222222, 0x33333333.

Perform multidata phase memory read cycle to slow decode target with data = 0x00000007, 0x00000008, 0x00000009.

Perform multidata phase memory write cycle to subtractive decode target with data = 0x11111111, 0x22222222, 0x33333333.

Perform multidata phase memory read cycle to subtractive decode target with data = 0x0000000a, 0x0000000b, 0x0000000c.

Test Definitions (continued)

Scenario 1.8—PCI Bus Multidata Phase and TRDY# Cycles

This scenario tests for proper execution of multidata phase memory cycles with a variety of target wait-states. Data driven on PCI master's local interface must match data driven on the PCI bus during the four data phases of the PCI read or write transactions. Dual address cycles are not tested because the master has 32 bit addressing only. Memory read multiple, memory read line, and memory write and invalidate cycles are not tested because cache mode is not supported.

Start with a cycle to set the latency timer register and enable master mode using the configuration registers. All the following tests assume that the starting address is 0x12345678.

Perform multidata phase memory write cycle to fast decode target with data = 0x11111111, 0x22222222, 0x33333333, and 0x44444444. Transaction sequence is one target wait-state and then four data phases.

Perform multidata phase memory read cycle to fast decode target with data = 0x00000001, 0x00000002, 0x00000003, and 0x00000004. Transaction sequence is one target wait-state and then four data phases.

Perform multidata phase memory write cycle to fast decode target with data = 0x11111111, 0x22222222, 0x33333333, and 0x44444444. Transaction sequence is one data phase, one target wait-state, and then three data phases.

Perform multidata phase memory read cycle to fast decode target with data = 0x00000005, 0x00000006, 0x00000007, and 0x00000008. Transaction sequence is one data phase, one target wait-state, and then three data phases.

Perform multidata phase memory write cycle to fast decode target with data = 0x11111111, 0x22222222, 0x33333333, and 0x44444444. Transaction sequence is one data phase, two target wait-states, and then three data phases.

Perform multidata phase memory read cycle to fast decode target with data = 0x00000009, 0x0000000a, 0x0000000b, and 0x0000000c. Transaction sequence is one data phase, two target wait-states, and then three data phases.

Perform multidata phase memory write cycle to fast decode target with data = 0x11111111, 0x22222222, 0x33333333, and 0x44444444. Transaction sequence is two data phases, two target wait-states, and then two data phases.

Perform multidata phase memory read cycle to fast decode target with data = 0x0000000d, 0x0000000e, 0x0000000f, and 0x00000010. Transaction sequence is two data phases, two target wait-states, and then two data phases.

Perform multidata phase memory write cycle to fast decode target with data = 0x11111111, 0x22222222, 0x33333333, and 0x44444444. Transaction sequence is alternating one target wait-state and then one data phase for a total of four data phases.

Perform multidata phase memory read cycle to fast decode target with data = 0x00000011, 0x00000012, 0x00000013, and 0x00000014. Transaction sequence is alternating one target wait-state and then one data phase for a total of four data phases.

Perform multidata phase memory write cycle to fast decode target with data = 0x11111111, 0x22222222, 0x33333333, and 0x44444444. Transaction sequence is alternating two target wait-states, and then two data phases for a total of four data phases.

Perform multidata phase memory read cycle to fast decode target with data = 0x00000015, 0x00000016, 0x00000017, and 0x00000018. Transaction sequence is alternating two target wait-states, and then two data phases for a total of four data phases.

Test Definitions (continued)

Scenario 1.9—PCI Bus Data Parity Error Single Cycles

This scenario tests for proper detection and reporting of parity errors during single data memory cycles to a fast target. For write cycles, the testbench will assert PERR# two cycles after the data phase and verify that the master detected the condition. For read cycles, the testbench will assert odd parity on PAR one cycle after the data phase and verify that the master reported the condition. Configuration and I/O cycles are not tested because they are specific to motherboard-type masters.

Start with a cycle to set the latency timer register and enable master mode using the configuration registers. All the tests following assume that the address asserted is 0x12345678.

Perform single phase memory write cycle to fast decode target with data = 0x11111111.

Verify that data parity detected bit is set in configuration status register, and then clear register.

Perform single phase memory read cycle to fast decode target with data = 0x5555aaaa.

Verify that PERR# is active two clocks after data phase. Verify data parity detected bit is set in configuration status register.

Scenario 1.10—PCI Bus Data Parity Error Multidata Phase Cycles

This scenario tests for proper detection and reporting of parity errors during single data memory cycles to a fast target. For write cycles, the testbench will assert PERR# two cycles after the data phase and verify that the master detected the condition. For read cycles, the testbench will assert odd parity on PAR one cycle after the data phase and verify that the master reported the condition. Configuration cycles are not tested because they are specific to motherboard-type masters. Dual address cycles are not tested because the master has 32-bit addressing only. Memory read multiple, memory read line, and memory write and invalidate cycles are not tested because cache mode is not supported.

Start with a cycle to set the latency timer register and enable master mode using the configuration registers. All the tests following assume that the address asserted is 0x12345678.

Perform multidata phase memory write cycle to fast decode target with data = 0x11111111, 0x22222222.

Verify that data parity detected bit is set in configuration status register, and then clear register.

Perform multidata phase memory read cycle to fast decode target with data = 0x5555aaaa, 0x5555aaaa.

Verify that PERR# is active two clocks after each data phase. Verify data parity detected bit is set in configuration status register.

Scenario 1.11—PCI Bus Master Time-Out

This scenario tests for proper master termination of multidata phase memory cycles when the latency timer expires before completion of the transaction. Data driven on PCI master's local interface must match data driven on the PCI bus during the four data phases of the PCI read or write transactions. Dual address cycles are not tested because the master has 32-bit addressing only. Configuration cycles are not tested, because they are specific to motherboard-type masters. Memory read multiple, memory read line, and memory write and invalidate cycles are not tested, because cache mode is not supported.

Start with a cycle to set the latency timer register to 0x8 and enable master mode using the configuration registers. All the tests following assume that the starting address is 0x12345678.

Perform multidata phase memory write cycle to fast decode target with data = 0x11111111, 0x22222222, 0x33333333, and 0x44444444. Transaction sequence is alternating one target wait-state then one data phase for a total of four data phases.

Verify that master terminated transaction before the four data phases completed.

Perform multidata phase memory read cycle to fast decode target with data = 0x00000001, 0x00000002, 0x00000003, and 0x00000004. Transaction sequence is alternating one target wait-state then one data phase for a total of four data phases.

Verify that master terminated transaction before the four data phases completed.

Test Definitions (continued)

Scenario 1.13—PCI Bus Master Parking

This scenario tests for proper parking of the AD and CBEN buses when the master acquires GNT# with REQ# inactive. The master must drive the AD and CBEN buses within eight clocks of GNT# becoming active. The master must then 3-state the AD and CBEN buses one clock after GNT# becomes inactive.

Start with a cycle to set the latency timer register and enable master mode using the configuration registers.

Assert GNT# to PCI master.

Verify that master drives the AD and CBEN buses within eight clocks of GNT# becoming active.

Deassert GNT# to PCI master.

Verify that master 3-states the AD and CBEN buses one clock after GNT# becomes inactive.

Scenario 1.14—PCI Bus Master Arbitration

This scenario tests for proper execution of single phase memory cycles when GNT# is deasserted coincident with FRAME# turning on.

Start with a cycle to set the latency timer register and enable master mode using the configuration registers. The test's starting address is 0x12345678.

Perform single phase memory write cycle to fast decode target with data = 0x11111111.

Verify that master completed transaction successfully even though GNT# was deasserted coincident with FRAME# turning on.

Scenario 2.5—Target Ignores Reserved Commands

This scenario verifies that the target doesn't respond to reserved or dual address cycle commands. The testbench verifies that DEVSEL# remains inactive and that the IUT doesn't inappropriately drive any signals. In this implementation, the following five cycle types are ignored:

C/BE#	Cycle Type
0100	Reserved
0101	Reserved
1000	Reserved
1001	Reserved
1101	Dual Address Cycle

For each of the five cycles above, verify that AD, TRDY#, STOP#, PERR#, & SERR# are not driven by the target and DEVSEL# remains deasserted for at least six clocks.

Scenario 2.6—Target Receives Configuration Cycles

This scenario validates that the target responds to configuration cycles with the proper bus signals and stores the data in implemented registers and outputs it properly when read back. The test also verifies that the target will only respond to cycles with address bits AD[1:0] = 00b, which signifies a type 0 configuration cycles. For all configuration transactions, the local bus should stay inactive.

Apply configuration write cycle of one data phase to fill the latency timer register (address = 0xc) with 0xffffffff.

Apply configuration read cycle of one data phase to read the latency timer register (address = 0xc).

Verify that all writable bits in latency timer register are set and all read-only bits not altered.

Apply configuration write cycles of one data phase to addresses = 0xd, 0xe, 0xf.

Verify that DEVSEL# stays deasserted for at least six clocks and that AD, TRDY#, STOP#, PERR#, and SERR# are not driven by the target.

Scenario 2.8—Target Gets Configuration Cycles with Address and Data Parity Errors

This scenario tests for proper detection and reporting of parity errors during single data configuration cycles. For read cycles, the testbench will assert PERR# two cycles after the data phase and verify that the target detected the condition. For write cycles, the testbench will assert odd parity on PAR one cycle after the data phase and verify that the target reported the condition via PERR#. In addition, address parity will assert odd parity during the address phase and then verify that the target reported the condition via SERR#.

Start with a cycle to enable parity error response for both address and data cycles via the configuration control register.

Perform single phase configuration write cycle to target with odd parity.

Verify that SERR# and PERR# are asserted at the appropriate time by the target to report address and data parity errors, respectively.

Perform single phase configuration read cycle to target.

Verify that SERR# is asserted at the appropriate time by the target to report an address parity error.

Test Definitions (continued)

Scenario 2.11—Target Gets Fast Back-to-Back Cycles

This scenario verifies that the target handles fast back-to-back write cycles (where a write is followed by an immediate read or write to the same target and the master, therefore, imposes no idle phase for turn-around time).

Start with a cycle to enable parity error response for both address and data cycles via the configuration control register.

Perform configuration write cycle to target's latency timer register followed immediately by a read to the same register.

Verify read data matches write data for the latency timer register's writable bits.

Perform configuration write cycle to a different target's latency timer register followed immediately by a write to the target model's latency timer register.

Perform configuration write cycle to a different target's latency timer register followed immediately by a read of the target model's latency timer register.

Verify read data matches write data for the target model's latency timer register.

Scenario 2.13—Target Gets Cycles with IRDY Used for Data Stepping

This scenario verifies that the IUT can handle wait-states imposed by the master. The scenario was originally intended for memory transactions, but we modified it to access configuration registers. The target model issues a disconnect during the first data phase of all configuration cycles, therefore preventing bursts during configuration transactions. The testbench performs a configuration write then read of the target's latency timer register. Both accesses contain master wait-states for data stepping.

Perform configuration write cycle to target's latency timer register followed by a read to the same register. Both transactions contain a wait-state during the first clock cycle following FRAME# to enact data stepping.

Verify read data matches write data for the latency timer register's writable bits.

CAE Tools

The remarks in this application note on CAE tools are introductory in nature. Complete details on how to get from the *Verilog* source file to a working *ORCA*-based PCI controller is contained in the *PCI Bus Master* Verilog *Source Code* User's Manual (MN96-008FPGA).

To get to working silicon with a *Verilog* source file, you must use a tool that synthesizes the source code into a working file that the *ORCA* place-and-route tools can use. The following is a list of tools used for this project.

Synthesis (only one of the following synthesis tools is required):

- Exemplar *Galileo*—This tool reads the *Verilog* input and synthesizes logic into a format that the *ORCA* Foundry tool can understand. This tool is available on both PC and workstation platforms.
- *Synopsys*—This tool reads the *Verilog* input and synthesizes logic into a format that the *ORCA* Foundry tool can understand. This tool is available on a workstation platform.
- Other—Other synthesis tools can also synthesize the *Verilog* source code into *ORCA* FPGAs; these include those available from Viewlogic Systems, Inc.; Synplicity, Inc.; and Mentor Graphics Corporation.

Map and Place-and-Route:

- *ORCA* Foundry—This tool reads the synthesized logic and performs map and place-and-route, to the *ORCA* FPGAs. This tool is available on both PC and workstation platforms. From the results of the place-and-route, *ORCA* Foundry produces information for the bit stream files (used to program the cell array) and timing back-annotation (used to produce a simulation that works with the timing that the place-and-route results give you). This tool is available on both PC and workstation platforms.

Simulation (only one of the following *Verilog* simulation tools is required):

- *Cadence*—This tool is used to perform functional simulation and at-speed simulation of the design. This tool is available on workstation platforms.

Conclusion

The PCI bus was specified to enable designers to interface to it directly with large-scale ASICs. This objective of the bus specification has produced special parameters that designers must investigate carefully when conducting a component survey. The reasons to choose an *ORCA* OR2CxxA/OR2TxxA for this implementation include the following:

- It meets the stringent I/O performance requirements of the PCI bus.
- It meets the timing requirements of the PCI bus.
- It meets the input loading requirements of the PCI bus.
- It has sufficient density to allow this generic PCI bus master to be further enhanced by the end user.
- It is SRAM-based for ease of use in development.
- It has nibble-based PLCs that fit bus-oriented logic very well.
- Its PLCs can be used as 16 x 4 dual-port SRAM cells for internal memory requirements.
- It has JTAG boundary-scan capability.
- It can be designed using *Verilog*, which will allow the design to be retargeted to a volume technology where appropriate.
- The timing of routing passes is repeatable.
- It can be simulated with *Cadence* or some other *Verilog* simulator.

References

The following publications provide additional information:

PCI Special Interest Group, *PCI Local Bus Specification*, Rev. 2.1, June 1, 1995.

PCI Special Interest Group, PCI Compliance Checklist, Rev. 2.0B.

Lucent Technologies, *PCI Bus Master* Verilog *Source Code* Manual (MN96-008FPGA), July 1996.

Lucent Technologies, *Field-Programmable Gate Arrays* Data Book (MN95-001FPGA), April 1995.

Lucent Technologies, ORCA *OR2CxxA FPGAs* Data Sheet (DS96-025FPGA), April 1996.

Ordering Information

- *ORCA* PCI Bus Initiator Customer Solution Core for *Synopsys*/VHDL (OR-PCIMCSC-VHDL-SYN)
- *ORCA* PCI Bus Initiator Customer Solution Core for Exemplar *Galileo*/VHDL (OR-PCIMCSC-VHDL-EXG)
- *ORCA* PCI Bus Initiator Customer Solution Core for *Synopsys/Verilog* (OR-PCIMCSC-VER-SYN)
- *ORCA* PCI Bus Initiator Customer Solution Core for Exemplar *Galileo/Verilog* (OR-PCIMCSC-VER-EXG)

Notes

Parameterized FIR Filters in *ORCA* Series FPGAs

Introduction

The advent of large, fast FPGAs with dedicated arithmetic capabilities, such as Lucent Technologies Microelectronics Group's *ORCA* Series FPGAs, has created an opportunity to perform flexible, reprogrammable digital signal processing functions (DSP) in dedicated logic, rather than sequential, DSP microprocessors. Lucent Technologies' DSP Customer Solution Core (CSC) is produced to meet the customer's need for DSP functionality within an FPGA without forcing the customer to perform low-level DSP design.

The DSP CSC provides a complete design solution: HDL description, script/batch processing files, design samples, simulation vectors, and the *SCUBA* macrofunction generator. Constantly expanding in support of design flows, platforms, and functions, the DSP CSC is a fast and convenient means to shave months off of a design cycle while implementing an optimized function in Lucent Technologies' fast, routable, high-density *ORCA* FPGAs.

This application note describes the process by which the DSP CSC is used to produce a suite of direct-form, linear phase, constant coefficient, finite impulse response (FIR) filters. The filters are generated by a simple automated procedure. Many parameters of the filters are programmable and are explained here.

Filtering, a Tutorial

What Does a Filter Do?

A filter is used to remove unwanted portions of a stream of data. Usually, the unwanted portion is noise, an extra signal, or a section of the frequency spectrum of the data that would interfere with subsequent data manipulations. Filtering is usually dealt with in the frequency domain; that is, looking at what frequencies are present in the data (via a Fourier transform) at a given time, and then creating a mask or filter that will block out the unwanted frequency components and pass the desired frequency components.

Figure 1 shows the filtering process in a frequency sense. In that figure, data with a desired frequency component and undesired noise (jagged) components is applied to a filter. The filter response covers the desired frequency component and does not cover the noise components, resulting in an output data spectrum that only contains the desired frequency components.

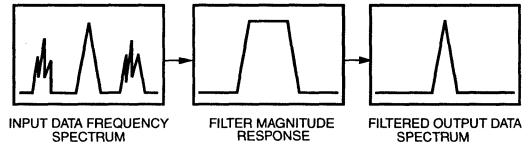


Figure 1. FIR Filter Response Flow

Filtering, a Tutorial (continued)

How Is Filtering Accomplished?

Although filters are most often designed and visually represented in the frequency domain, they are usually implemented in the time domain. There are many different structures or topologies by which filters can be implemented. Common to all of them are numbers called coefficients that are multiplied by the samples in the input data stream. The products of these multiplications are then added in some fashion to produce the filtered output data samples.

Finite impulse response (FIR) filters are a class of filters that exhibit certain properties. The major feature of an FIR filter is referred to by its name—finite impulse response. This means that the filter's output response to an impulse input, or any other type of input signal that eventually goes to and stays at zero, will eventually go to zero.

FIR filters can be physically implemented in a number of ways. The most common topology for a FIR filter is the direct form, or tapped delay line, topology. This topology is nonrecursive; the filter output is only a function of present and past input values, not of past output values. This implies stability, which is one of the main reasons that FIR filters are used extensively in many different applications.

Figure 2 shows an example of a direct-form FIR filter. It has input, x(n), and output, y(n). The filter consists of a series of delay elements, z^{-1}, through which the input data passes. In real terms, these delay elements are equivalent to clocked data registers. The output of each of these delay elements along with the input sample are called taps of the filter. Therefore, an M tap filter retains M – 1 delayed samples and the input. In general, the more taps a filter has the more exactly its response follows the desired ideal response. Each data tap in the filter is multiplied by a coefficient value, h(k); $0 \le k \le M$, corresponding to that tap. The products from all of the multiplied taps are then summed to produce the filtered output sample, y(n), of the filter.

The latency of the filter is the time, usually measured in terms of clock cycles or sample times, between an input sample entering the filter and the corresponding output sample leaving the filter. Latency can be difficult to understand with some types of filters, but for the common subclass of filters known as linear phase FIR filters, latency is straightforward. Linear phase describes a filter with a constant group delay (derivative of the phase with respect to frequency), but in practical terms, it means that there is a symmetry or antisymmetry to the coefficients as expressed below:

Symmetric filter:

$$\text{coefficients } h(k) = h(M - 1 - k),\ 0 \le k \le M - 1$$

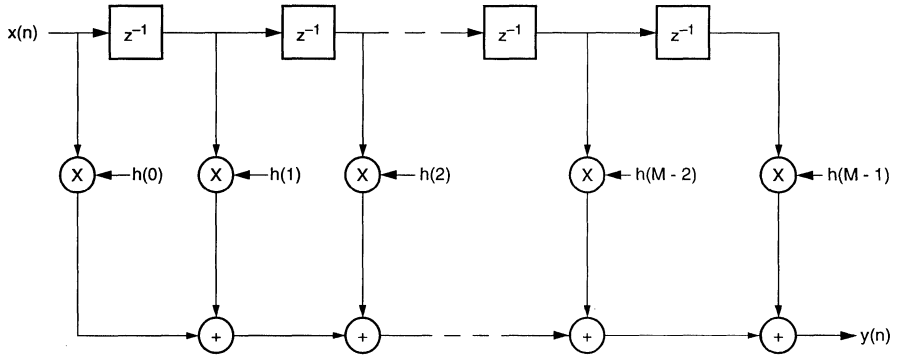
Antisymmetric filter:

$$\text{coefficients } h(k) = -h(M - 1 - k),\ 0 \le k \le M - 1$$

Filters of these types have a latency (excluding pipelining in the implementation) of (M – 1)/2 sample times. This yields an integer sample time latency for an odd tap filter and a fractional sample time latency for an even tap filter.

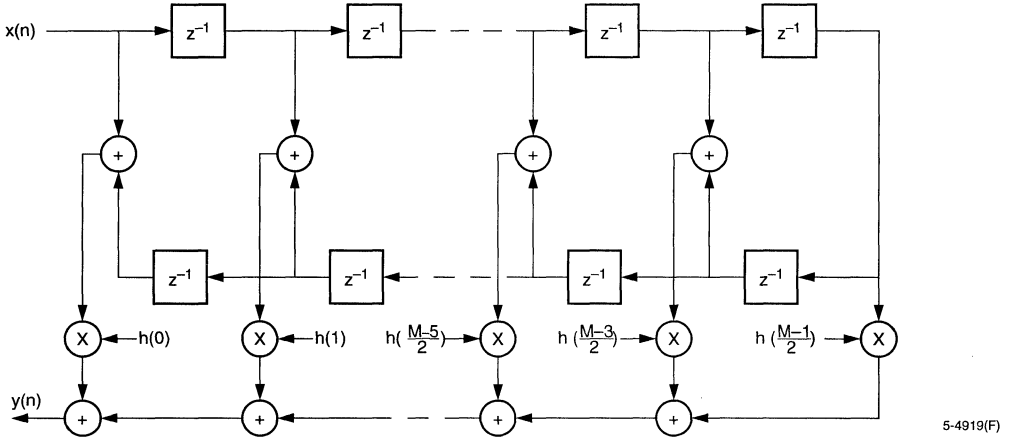
FIR filters that display symmetry or antisymmetry of coefficients can be implemented in a more compact realization of the direct-form topology. Since pairs of coefficients are the same or negatives of each other, data samples which are to be multiplied by the same (or negative) coefficient can be added (or subtracted) prior to multiplication by the coefficient value. This version of the direct-form topology reduces the number of multiplications by a factor of two at the expense of a few additions. Figure 3 shows the direct-form realization for symmetric coefficients.

Filtering, a Tutorial (continued)



Key:
M The number of filter taps.
z^{-1} Represents a unit time delay.

Figure 2. Input Data Sample



Key:
M The number of filter taps.
z^{-1} Represents a unit time delay.

Figure 3. Direct-Form Realization for Symmetric Coefficients

Filtering, a Tutorial (continued)

Design Considerations

Filters that are implemented in dedicated hardware, such as an FPGA or ASIC, are usually implemented using fixed-point numbers as opposed to digital signal processing microprocessors (DSPs) that often use floating-point implementations. There are several practical design considerations to be made for a fixed-point digital filter beyond the basic design specification. These include quantization and scaling of the filter coefficients, quantization of internal data paths (including the use of saturation addition), and quantization of output data.

Quantization of Coefficients

Quantization of coefficients [h(k) in Figures 2 and 3] is the truncation or rounding of a coefficient value into a fixed number of significant bits. Values can be quantized from either floating- or fixed-point numbers. For example, if the decimal 124.6 is quantized into an 8-bit representation (by rounding), it becomes 125. There is not much change because the value 125 falls within the 0 to 255 unsigned range for an 8-bit (decimal) number. If, however, the decimal 283 is quantized to an 8-bit representation, it becomes 255 because that is the largest unsigned value that can be represented using 8 bits. The result of either of these example quantizations introduces error between the calculated response of the filter and the actual response that can be obtained from a physical fixed-point implementation of the filter. The error in the first case is small, and it is impractical to do anything about it. The error in the latter case, though, is large. In this situation, a 9-bit coefficient representation should be used or the filter should be adjusted so that a reasonable response can be obtained using an 8-bit coefficient.

Tap Product Quantization

The products of the multipliers in each filter tap comprise (number of coefficient bits + number of data bits) bits. For example, if each coefficient is 8 bits wide, and the data samples are 16 bits wide, the product of the multiplication of the two values will be up to 8 + 16 = 24 bits wide. The products from all of the taps are summed to produce an output sample. All of the bits in the tap products may not be significant to the design being implemented. If this is the case, the tap product values may be truncated to fewer bits which will reduce the area required for the filter and increase its throughput capability at the expense of some error from the change in precision.

Output Data Summation Quantization (Adder Saturation vs. Wraparound)

Referring to the chain of adders that add the tap products to produce the filter output in Figure 3, notice that if no error is to be introduced in the summation, then each adder must grow by one bit as the chain gets closer to the final addition. This must be done in case an addition sum overflows or underflows the number of bits at the adder's inputs. Careful scaling and quantization of data and coefficients can often eliminate the concern of over/underflowing in the final summation, but often it is impossible to prepare for every instance. Making the data path width grow for each successive addition is not a very viable approach to the design of a filter because of the impact on both area consumed by the filter and speed loss because of wider paths to process.

An alternative approach is to saturate the result of an adder that overflows or underflows. This means that if the adder overflows, the result of the addition is set to its full-scale positive value. If an adder underflows, the result is set to the full-scale negative value. If this is not done and the data path is not widened for each successive add, then the carry-out bit (MSB) of the addition will be lost and the lower, less significant bits will be retained, causing the result to "wraparound" and causing a glitch in the data. Figure 4 shows a graphical representation of saturation vs. wraparound. In this figure, the box in which each signal is shown is the boundary within which the signal can be represented. The original analog signal is shown as overflowing its representable bounds.

Output Data Quantization

Similar to tap product quantization, output quantization is the truncation of nonsignificant bits from the output data stream. The trade-off is error vs. a wider data path to process in the system beyond the filter output.

Filtering, a Tutorial (continued)

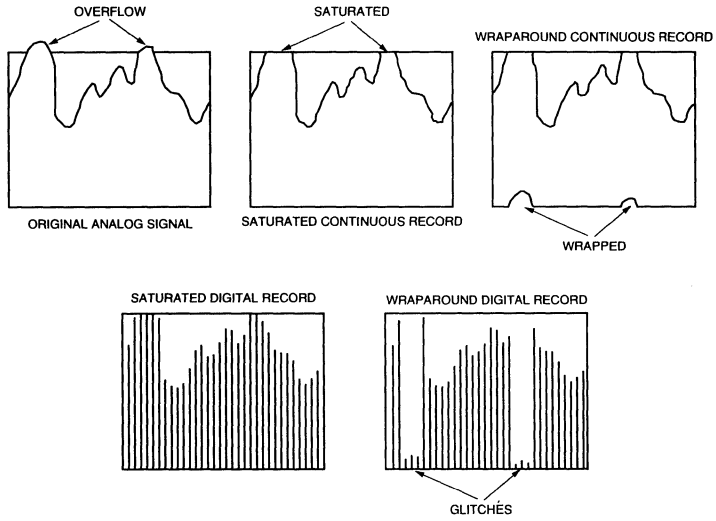


Figure 4. Saturation vs. Wraparound

FIR Filter Suite Features

The filters generated by the DSP CSC may be used as stand-alone filters or as components in user-defined systems. Figure 5 is a graphical representation of the use of the DSP CSC FIR filter in an *ORCA* FPGA.

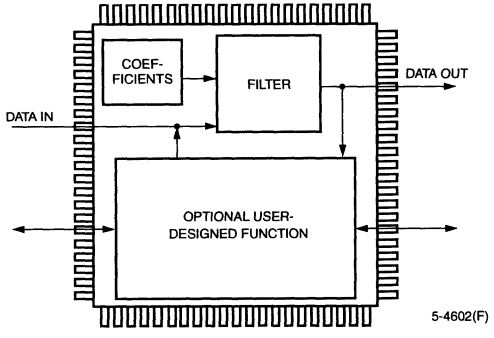


Figure 5. Digital Signal Processing FIR Filter

Features

- User-definable parameters:
  - — Clock polarity
  - — Input data path width
  - — Coefficient data path width
  - — Multiplication product data path width
  - — Output data path width
  - — Number of filter taps
  - — Input data type: two's complement, unsigned, signed magnitude, offset binary
  - — Coefficient data type: two's complement, unsigned, signed magnitude, offset binary
  - — Coefficient data representation: binary, hexadecimal, decimal
  - — Filter symmetry: symmetrical, antisymmetrical
  - — Multiplication pipeline depth
  - — Addition result registering
  - — Addition result control: saturation, overflow
- Easy ASCII text file parameter and coefficient input
- Script files to generate and test filters
- Uses *ORCA*'s fast multiplier mode for tap multiplications
- Forthcoming options to include improved pipelining and ROM-based multipliers for smaller filters

FIR Filter Suite Features (continued)

A description of each of the filter parameters is provided below.

clock polarity Specifies the active edge of clocks used for registers in the filter. 1 = rising edge, 0 = falling edge.

i_data_width Input Data Path Width. The number of bits in the filter input data path. The valid range is 1 to 32.

p_data_width Product Data Path Width. The number of significant bits to retain from the multiplications performed in each filter tap. The maximum amount is (i_data_width + coeff_width + 1). The valid range is 0 to 65, subject to the maximum expressed above. Setting this parameter to 0 will automatically generate the maximum number of significant bits and is the recommended setting for this parameter unless stringent area or speed considerations dictate otherwise.

o_data_width Output Data Path Width. The number of bits in the filter output data path. The valid range is 0 to 32, subject to the constraint that o_data_width is less than or equal to p_data_width. Setting this parameter to 0 will force the output data width path to equal the input data width path value.

data_type The numerical format of the input data. Output data is returned in the same format. The "native" data type of the filter is two's complement. All other data types are converted to two's complement at the input to the filter and converted back again at the output. The valid formats are 0 = unsigned, 1 = two's complement, 2 = signed magnitude, 3 = offset binary.

add_reg Register filter addition results switch. The result of additions prior to tap multiplications, and output sample summations can be registered to improve filter performance in relation to the system clock. 0 = no addition result registering, 1 = register addition results.

mult_depth Multiplier Pipeline Depth. This parameter specifies the number of registered pipeline stages used in each of the filter's tap multipliers. The range for this parameter is 0 to 32.

symmetry The coefficient symmetry type indicator for the filter. 0 = symmetrical coefficients, 1 = antisymmetrical coefficients.

saturate The addition overflow process indicator. Allows for tap summation interim and final values to be saturated to positive full scale on overflow and negative full scale on underflow. 0 = no saturation, 1 = saturation.

taps The number of taps in the filter. The valid range for this parameter is 2 to 256.

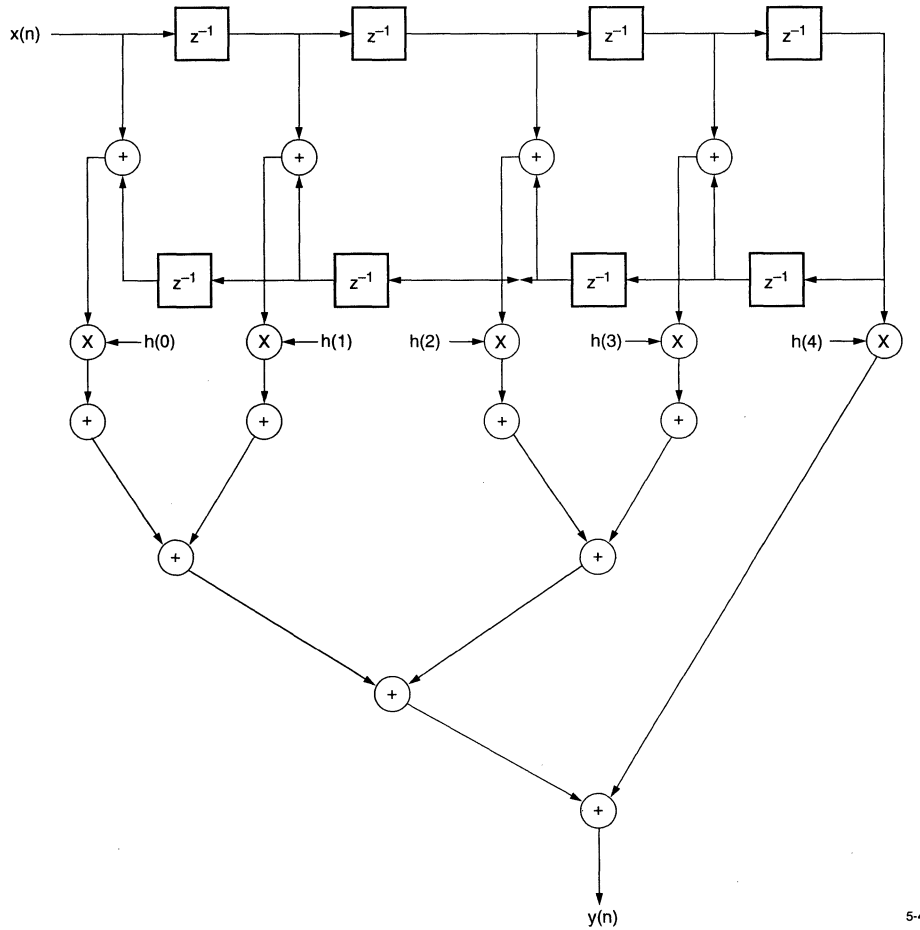
coeff_width Coefficient Data Path Width. The number of bits in the filter coefficient data paths. The valid range is 1 to 32.

coeff_type The numerical format of the coefficient data. The valid formats are 0 = unsigned, 1 = two's complement, 2 = signed magnitude, 3 = offset binary.

FIR Filter Suite Features (continued)

FIR Filter Implementation

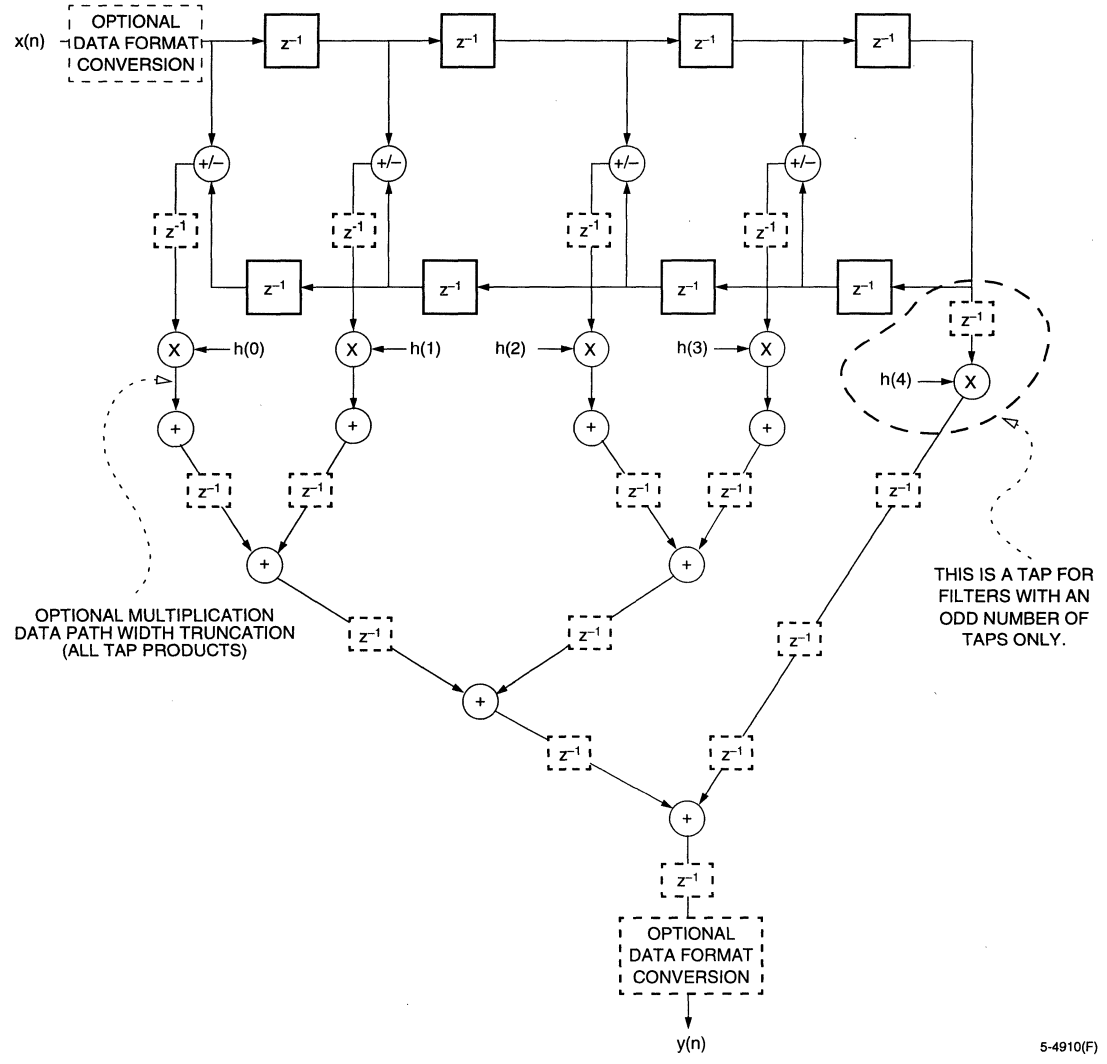
The FIR filter implemented by the DSP CSC has a direct-form, or tapped delay line, topology. A diagram of this structure is shown in Figure 6. The final adder chain of Figure 6 is a modification from the traditional direct-form topology (see Figure 3) to increase the performance of the filter. Figure 7 is the same representation as Figure 6, but it shows where the effects of parameter changes are on the basic direct-form topology. Referring to these diagrams may be useful in understanding the filter features and their implications to the filter design. Multipliers and adders (including two's complementers) for the filter are implemented using the *SCUBA ORCA* optimized macro function generator. This flow ensures that these functions will be optimized to the *ORCA* devices using fast-ripple mode adder and multiplier functions.



Key:
z-1 Represents unit delay, physically a register.
(+) Adder
(X) Multiplier

Figure 6. FIR Filter Implementation

FIR Filter Suite Features (continued)



Key:

- z-1 Represents unit delay, physically a register.
- +/- Adder for symmetry option, subtractor for antisymmetry.
- + Adder with optional saturate on over/underflow.
- z-1 Register for addition result register option.
- X Optional pipelining within multipliers.

Figure 7. FIR Filter with Parameter Changes

FIR Filter Suite Features (continued)

Design Flow

Figure 8 shows the design flow used to create a filter with the customer solution core. Two ASCII text files must be created: a parameter file and a coefficient file. These files are processed by a script file, which generates the necessary *SCUBA* macro functions and synthesizes the filter. The synthesized filter may then be used stand-alone or as a component in a user's design. The design may be functionally simulated at this point in the flow, or the design can be immediately processed in *ORCA* Foundry. The output from *ORCA* Foundry may be back-annotated for simulation and/or downloaded to the target *ORCA* device for use.

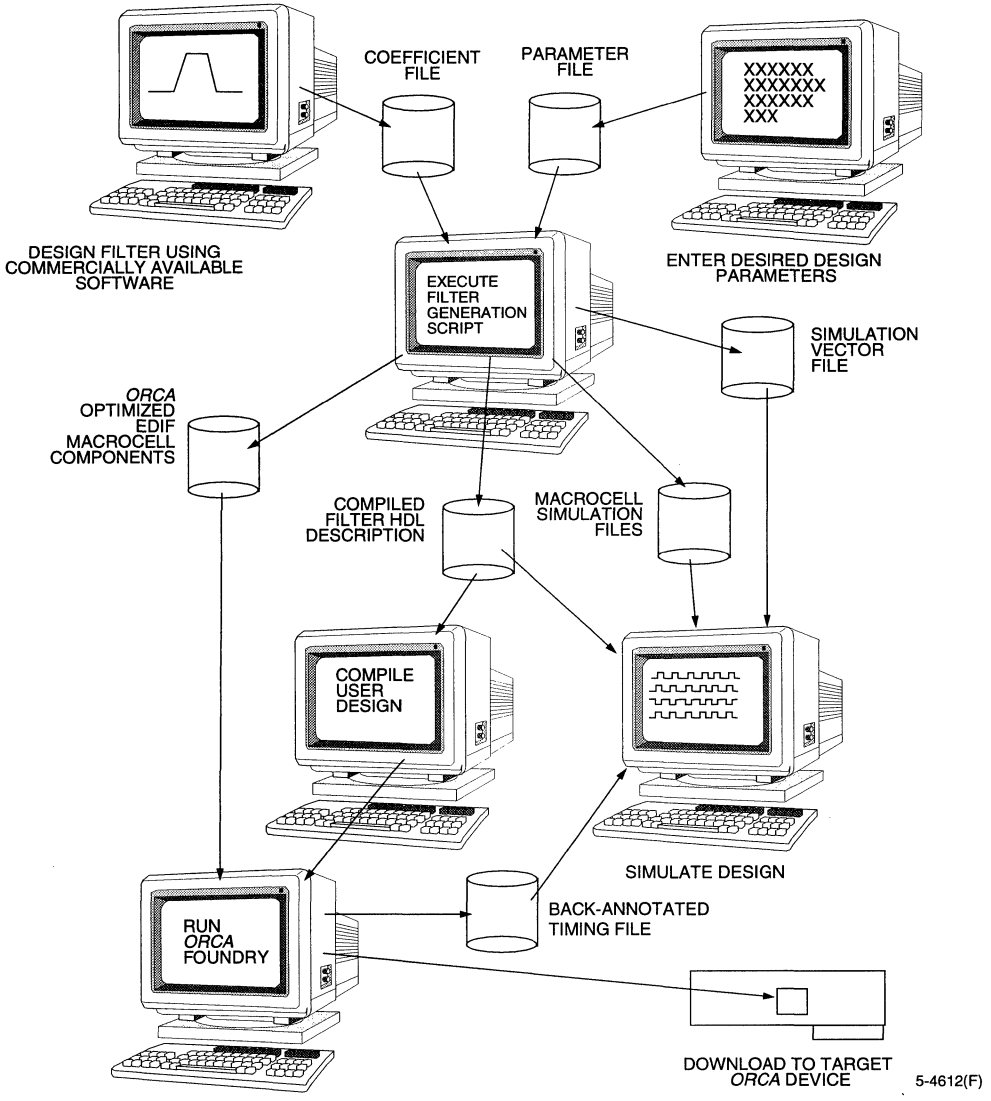


Figure 8. VHDL FIR Filter Design Flow

Required Tools

Synopsys, version 3.4a or later

ORCA Foundry, version 9.0 or later

Model Technology's V-System/PLUS Simulator (to use simulation files provided)

References

Related Application Notes and Product Data

Multipliers in ORCA *OR2CxxA/OR2TxxA FPGAs*, Lucent Technologies Microelectronics Group (AP96-015FPGA), April 1996.

Implementing and Optimizing Multipliers in ORCA *FPGAs*, Lucent Technologies Microelectronics Group (AP94-035FPGA), February 1994.

*Optimized Reconfigurable Cell Array (*ORCA*) OR2CxxA and OR2TxxA Series Field-Programmable Gate Arrays* Data Sheet, Lucent Technologies Microelectronics Group (DS96-140FPGA), August 1996.

DSP and Filtering

Theory and Application of Digital Signal Processing, Rabiner, L. R., Gold, B., Prentice-Hall, NJ, 1975.

Digital Signal Processing in VLSI; Higgins, R. J.; Prentice-Hall, NJ; 1990.

VHDL

A Guide to VHDL Syntax, Bhasker, J., Prentice-Hall, NJ, 1995.

A Guide to VHDL, Mazor, S. and Langstraat, P., Kluwer Academic Publishers, Boston, 2nd Ed., 1993.

Package Information

General Information

Lucent Technologies recognizes that packaging has a dramatic effect on device and system performance when using FPGAs. With today's accelerated development of very fast, high-density devices, packaging technology has been compelled to react swiftly. Higher power ranges and tighter packaging densities are forcing engineers to reevaluate the effectiveness of older, more traditional packaging technologies and styles. Through the research efforts of Lucent Technologies Bell Laboratories, we are spearheading the development and acceptance of new packaging options in order to meet future demands for state-of-the-art devices and system applications.

Packages, such as ball grid arrays, chip carriers, pin grid arrays, and small-outline configurations, bridge the gap of advancements in chip technology and the developments in automated circuit-board assembly processes to lower circuit-board costs and enhance system performance. Currently, we offer a choice of packages in both through-hole and surface-mount technologies. These packages accommodate high packaging densities with proven reliability.

As a member of JEDEC and its committees, Lucent Technologies has been instrumental in setting the standards for new packaging technologies. Our ongoing participation in JEDEC is your assurance that many of our packages not only meet industry standards, but in some cases actually help establish those standards.

Package Thermal Characteristics

When silicon die junction temperature is below the recommended junction temperature of 125 °C, the temperature-activated failure mechanisms are minimized. There are four major factors that affect the thermal resistance value: silicon device size/paddle size, board mounting configuration (board density, multilayer nature of board), package type and size, and system airflow over the package. The values in the table below reflect the capability of the various package types to dissipate heat at given airflow rates. The numbers represent the delta °C/W between the ambient temperature and the device junction temperature.

To test package thermal characteristics, a single package containing a 0.269 in. sq. test IC of each configuration is mounted at the center of a printed-circuit board (PCB) measuring 8 in. x 13 in. x 0.062 in. The assembled PCB is mounted vertically in the center of the rectangular test section of a wind tunnel. The walls of the wind tunnel simulate adjacent boards in the electronic rack and can be adjusted to study the effects of PCB spacing. Forced air at room temperature is supplied by a pair of push-pull blowers which can be regulated to supply the desired air velocities. The air velocity is measured with a hot-wire anemometer at the center of the channel, 3 in. upstream from the package.

A typical test consists of regulating the wind tunnel blowers to obtain the desired air velocity and applying power to the test IC. The power to the IC is adjusted until the maximum junction temperature (as measured by its diodes) reaches 115 °C to 120 °C. The thermal resistance Θ_{JA} (°C/W) is computed by using the power supplied to the IC, junction temperature, ambient temperature, and air velocity:

$$\Theta_{JA} = \frac{T_J - T_A}{Q_C}$$

where:

T_J = peak temperature on the active surface of the IC

T_A = ambient air temperature

Q_C = IC power

The tests are repeated at several velocities from 0 fpm (feet per minute) to 1000 fpm.

The definition of the junction-to-case thermal resistance Θ_{JC} is:

$$\Theta_{JC} = \frac{T_J - T_C}{Q_C}$$

where:

T_C = temperature measured to the thermocouple at the top dead center of the package

The actual Θ_{JC} measurement performed at Lucent Technologies, Θ_{J-TDC}, uses a different package mounting arrangement than the one defined for Θ_{JC} in MIL-STD-883D and SEMI standards. Please contact Lucent for a diagram.

The maximum power dissipation for a package is calculated from the maximum allowed junction temperature (T_{Jmax}, 125 °C), the maximum ambient temperature (T_{Amax}), and the junction-to-ambient thermal characteristic for the given package (Θ_{JA}). The maximum power for the package is calculated as follows:

$$\text{Max. Power (Watts)} = (125\ ^\circ\text{C} - T_{Amax}) \times (1/\Theta_{JA})$$

In Table 1 and Table 2, a maximum power dissipation for each package is shown with T_{Amax} = 70 °C for the commercial temperature range and the Θ_{JA} used is for 0 feet per minute of air flowing over the package. If your application does not correspond to these parameters, the maximum power dissipation should be recalculated using the formula above.

Once the power dissipated by the FPGA has been determined, the maximum junction temperature of the FPGA can be found. This is needed to determine if speed derating of the device from the 85 °C junction temperature used in all of the delay tables is needed. Using the maximum ambient temperature, T_{Amax}, and the power dissipated by the device, P, the maximum junction temperature is given by:

$$T_{Jmax} = T_{Amax} + (P \bullet \Theta_{JA})\ ^\circ\text{C}$$

Table 1 lists the plastic package thermal characteristics for the ATT3000 and the *ORCA* Series FPGAs. Table 2 lists the ceramic package thermal characteristics for the *ORCA* Series FPGAs.

Package Thermal Characteristics (continued)

Table 1. Plastic Package Thermal Characteristics for the ATT3000 and *ORCA* Series*

Package	ΘJA (°C/W)			ΘJC (°C/W)	Max Power (70 °C—0 fpm)
	0 fpm	200 fpm	400 fpm		
44-Pin PLCC	49	43	40	—	1.12 W
68-Pin PLCC	43	38	35	11	1.28 W
84-Pin PLCC	40	35	32	9	1.38 W
100-Pin QFP	81	67	64	11	0.68 W
100-Pin TQFP	61	49	46	6	0.90 W
132-Pin PPGA	22	18	16	—	2.50 W
144-Pin TQFP	52	39	36	4	1.06 W
160-Pin QFP	40	36	32	8	1.38 W
175-Pin PPGA	23	20	17	—	2.39 W
208-Pin SQFP	37	33	29	8	1.49 W
208-Pin SQFP2	16	14	12	2	3.43 W
240-Pin SQFP	35	31	28	7	1.57 W
240-Pin SQFP2	15	12	10	2	3.66 W
256-Pin PBGA[1,3]	22	14	17	3 (est.)	2.62 W
256-Pin PBGA[2,3]	26	23	21	TBD	1.97 W
304-Pin SQFP	33	30	27	6	1.67 W
304-Pin SQFP2	12	10	8	2	4.58 W
352-Pin PBGA[1,3]	18	15	13	2 (est.)	3.06 W
352-Pin PBGA[2,3]	25	21	19	TBD	2.20 W
432-Pin EBGA	13	10	9.3	<1	5.50 W
600-Pin EBGA	12	9	8.3	<1	6.88 W

1. With thermal balls connected to board ground plane.
2. Without thermal balls connected to board ground plane.
3. Mounted on 4-layer board with two power/ground planes.

* Packages for the ATT1700A Series Serial ROMs are not included because operation is guaranteed over all specified temperature ranges.

Table 2. Ceramic Package Thermal Characteristics for the *ORCA* Series

Package	ΘJA (°C/W)			ΘJC (°C/W)	Max Power (70 °C—0 fpm)
	0 fpm	200 fpm	400 fpm		
364-Pin CPGA	18	16	14	2.3	3.05 W
428-Pin CPGA	18	16	14	2.3	3.05 W

Package Coplanarity

Table 3. Package Coplanarity

Package Type	Coplanarity Limit (mils)
DIP	TBD
EBGA	8.0
PBGA	7.5
PLCC	4.0
PPGA	NA
QFP	4.0
SOIC	TBD
SQFP/SQFP2	4.0 (240 and 304 only) 3.15 (all others)
TQFP	3.15

Package Parasitics

The electrical performance of an IC package, such as signal quality and noise sensitivity, is directly affected by the package parasitics. Table 4 lists eight parasitics associated with the ATT1700A, ATT3000, and *ORCA* packages. These parasitics represent the contributions of all components of a package, which include the bond wires, all internal package routing, and the external leads.

Four inductances in nH are listed: L_{SW} and L_{SL}, the self-inductance of the lead; and L_{MW} and L_{ML}, the mutual inductance to the nearest neighbor lead. These parameters are important in determining ground bounce noise and inductive crosstalk noise. Three capacitances in pF are listed: C_M, the mutual capacitance of the lead to the nearest neighbor lead; and C_1 and C_2, the total capacitance of the lead to all other leads (all other leads are assumed to be grounded). These parameters are important in determining capacitive crosstalk and the capacitive loading effect of the lead.

The parasitic values in Table 4 are for the circuit model of bond wire and package lead parasitics. If the mutual capacitance value is not used in the designer's model, then the value listed as mutual capacitance should be added to each of the C_1 and C_2 capacitors. The PGAs contain power and ground planes that will make the inductance value for power and ground leads the minimum value listed. The PGAs also have a significant range of parasitic values. This is due to the large variation in internal trace lengths and is also due to two signal metal layers that are separated from the ground plane by different distances. The upper signal layer is more inductive but less capacitive than the closer, lower signal layer.

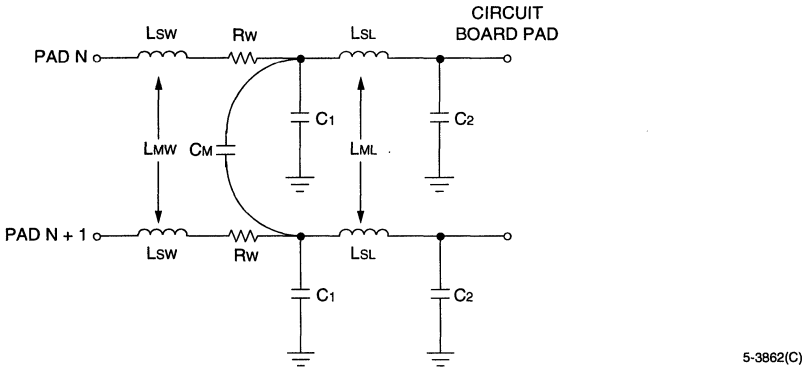


Figure 1. Package Parasitics

Package Parasitics (continued)

Table 4. Package Parasitics

Package Type	L_{SW}	L_{MW}	R_W	C_1	C_2	C_M	L_{SL}	L_{ML}
8-Pin DIP	3	1	140	0.4	0.4	0.2	2—3	1
8-Pin SOIC	3	1	140	0.4	0.3	0.2	2	1
20-Pin PLCC	3	1	140	0.4	0.4	0.2	4—5	2
44-Pin PLCC	3	1	140	0.5	0.5	0.3	5—6	2—2.5
68-Pin PLCC	3	1	140	0.5	0.5	0.4	6—9	3—4
84-Pin PLCC	3	1	140	1	1	0.5	7—11	3—6
100-Pin QFP	3	1	160	1	1	0.5	7—9	4—5
100-Pin TQFP	3	1	150	0.5	0.5	0.4	4—6	2—3
132-Pin PPGA	3	1	150	1	1	0.25	4—10	0.5—1
144-Pin TQFP	3	1	140	1	1	0.6	4—6	2—2.5
160-Pin QFP	4	1.5	180	1.5	1.5	1	10—13	6—8
175-Pin PPGA	3	1	150	1	1	0.3	5—11	1—1.5
208-Pin SQFP	4	2	200	1	1	1	7—10	4—6
208-Pin SQFP2	4	2	200	1	1	1	6—9	4—6
240-Pin SQFP	4	2	200	1	1	1	8—12	5—8
240-Pin SQFP2	4	2	200	1	1	1	7—11	4—7
256-Pin PBGA	5	2	220	1	1	1	5—13	2—6
304-Pin SQFP	5	2	220	1	1	1	12—18	7—12
304-Pin SQFP2	5	2	220	1	1	1	11—17	7—12
352-Pin PBGA	5	2	220	1.5	1.5	1.5	7—17	3—8
364-Pin CPGA	2	1	1000	1—2	1—2	0.5—1	2—11*	1—4
428-Pin CPGA	2	1	1000	1—2	1—2	0.6—1.2	2—11*	1—4
432-Pin EBGA	4	1.5	500	1	1	0.3	3—5.5	0.5—1
600-Pin EBGA	4	1.5	500	1	1	0.4	3—6	0.5—1

* Leads designated as ground (power) can be connected to the ground plane, reducing the trace inductance to the minimum value listed.

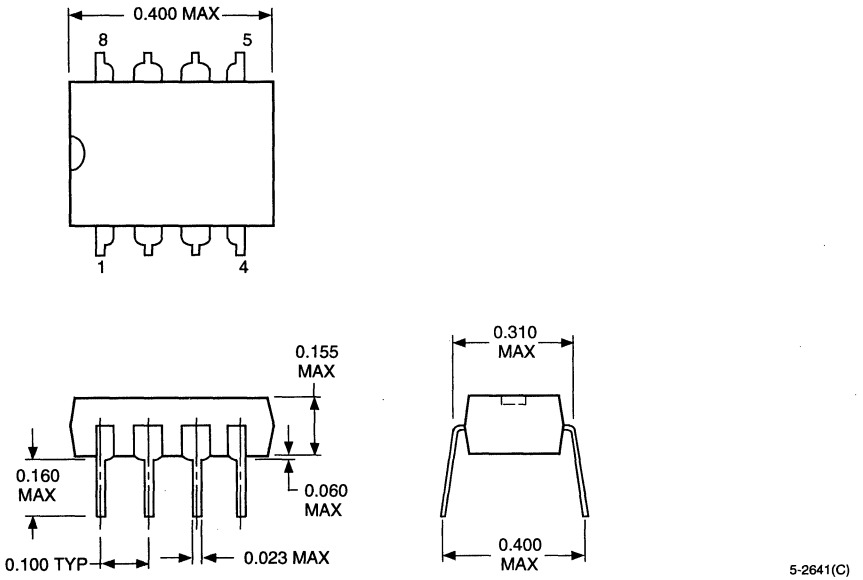
Outline Diagrams*

Terms and Definitions

Basic Size (BSC):	The basic size of a dimension is the size from which the limits for that dimension are derived by the application of the allowance and the tolerance.
Design Size:	The design size of a dimension is the actual size of the design, including an allowance for fit and tolerance.
Typical (TYP):	When specified after a dimension, indicates the repeated design size if a tolerance is specified or repeated basic size if a tolerance is not specified.
Reference (REF):	The reference dimension is an untoleranced dimension used for informational purposes only. It is a repeated dimension or one that can be derived from other values in the drawing.
Minimum (MIN) or Maximum (MAX):	Indicates the minimum or maximum allowable size of a dimension.

8-Pin DIP (ATT1700A Only)

Dimensions are in inches.

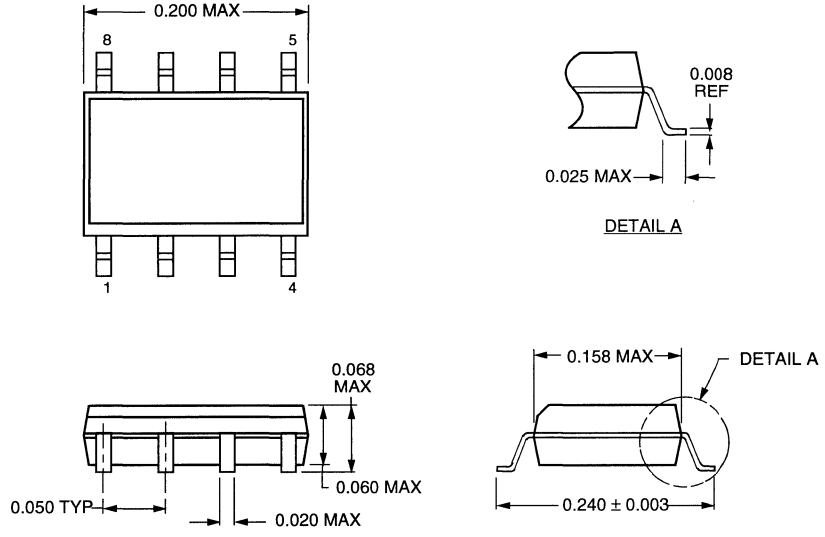


* Please note that the measurements provided in the outline diagrams in this section are intended for informational purposes only. For detailed schematics to assist your design efforts, please contact your Lucent Technologies sales representative.

Outline Diagrams (continued)

8-Pin SOIC (ATT1700A Only)

Dimensions are in inches.

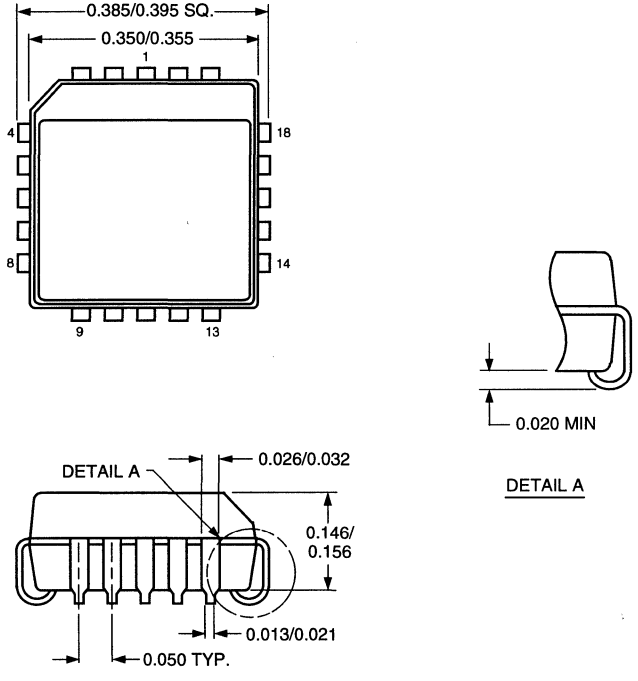


5-3979(C)

Outline Diagrams (continued)

20-Pin PLCC (ATT1700A Only)

Dimensions are in inches.

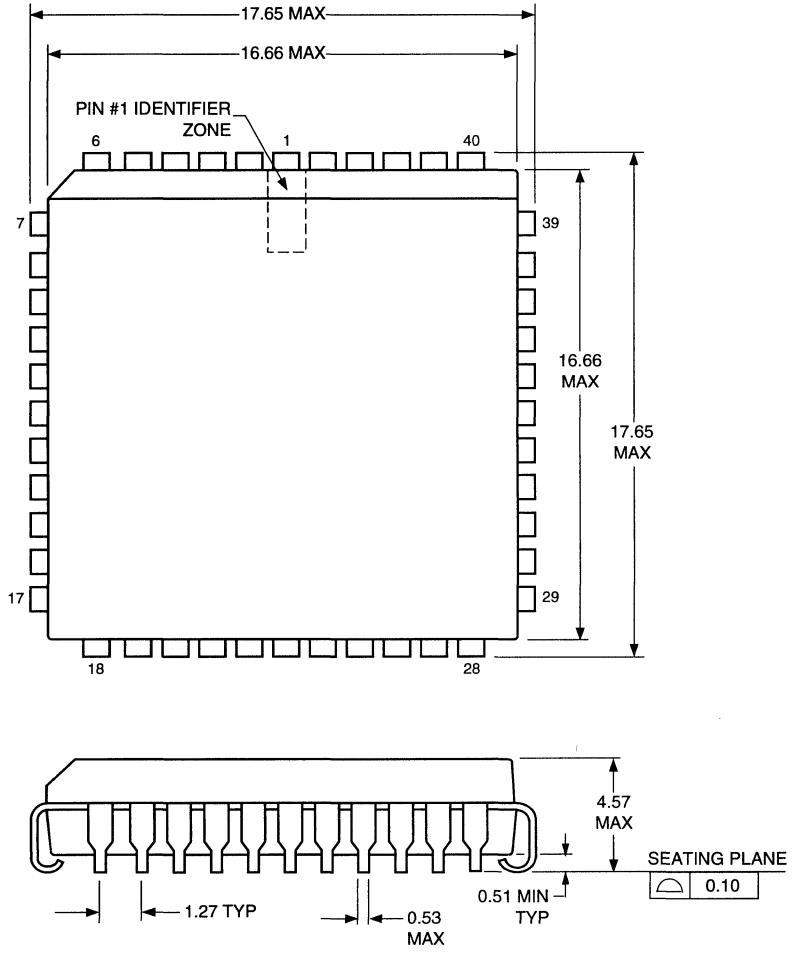


5-2035(C)

Outline Diagrams (continued)

44-Pin PLCC

Dimensions are in millimeters.

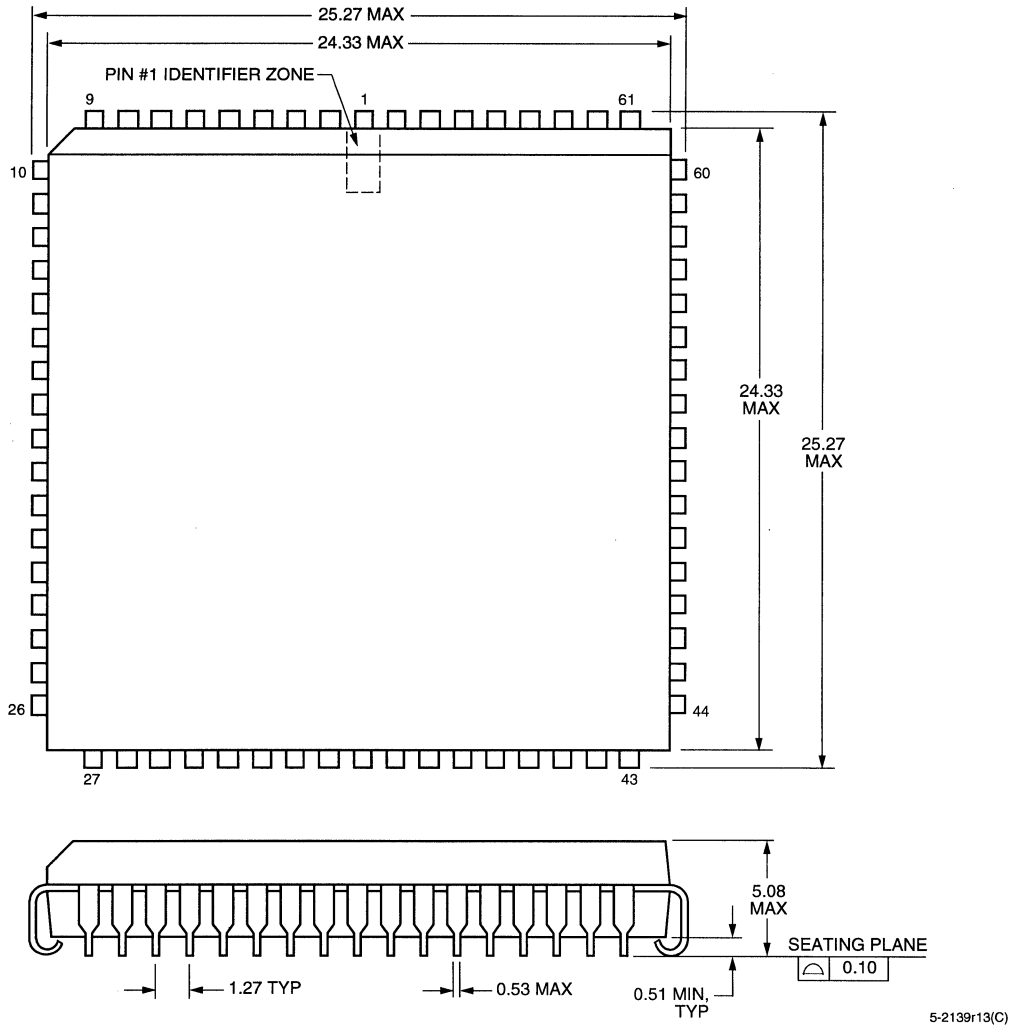


5-2506r7(C)

Outline Diagrams (continued)

68-Pin PLCC

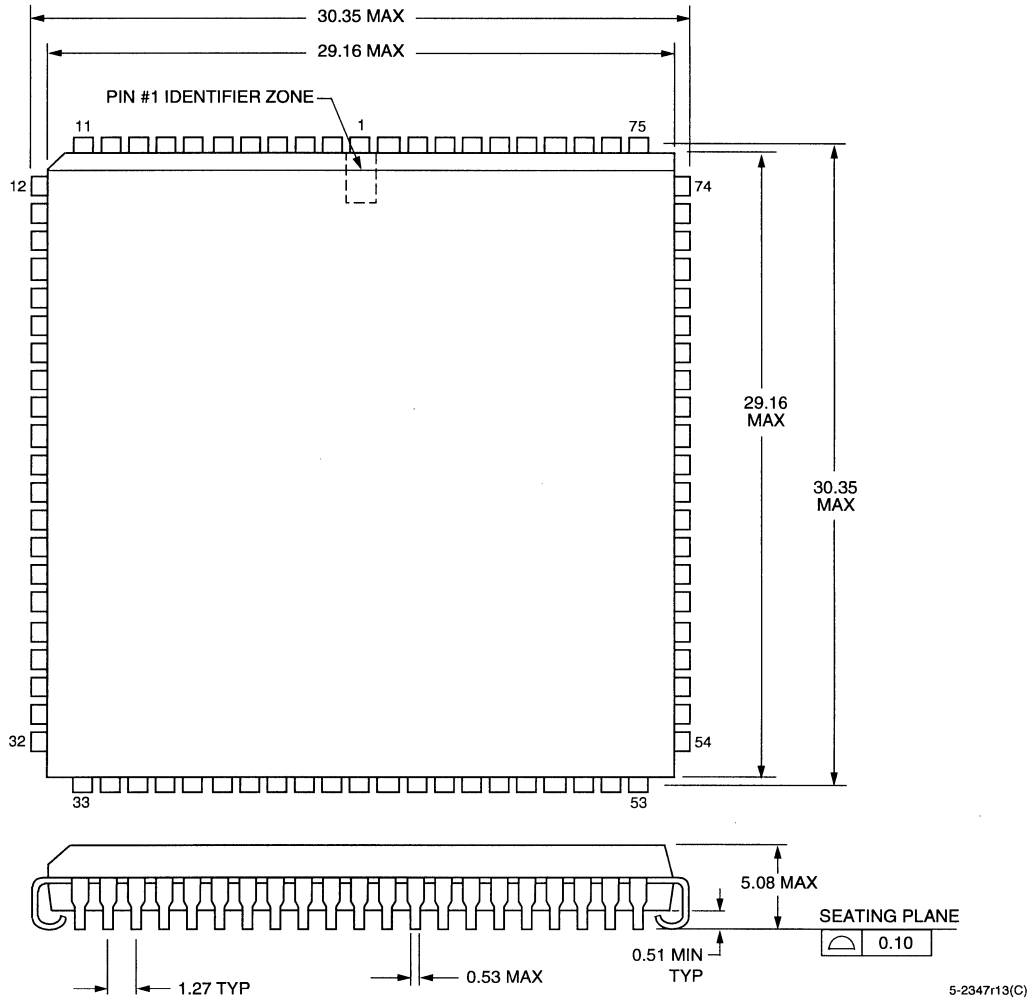
Dimensions are in millimeters.



Outline Diagrams (continued)

84-Pin PLCC

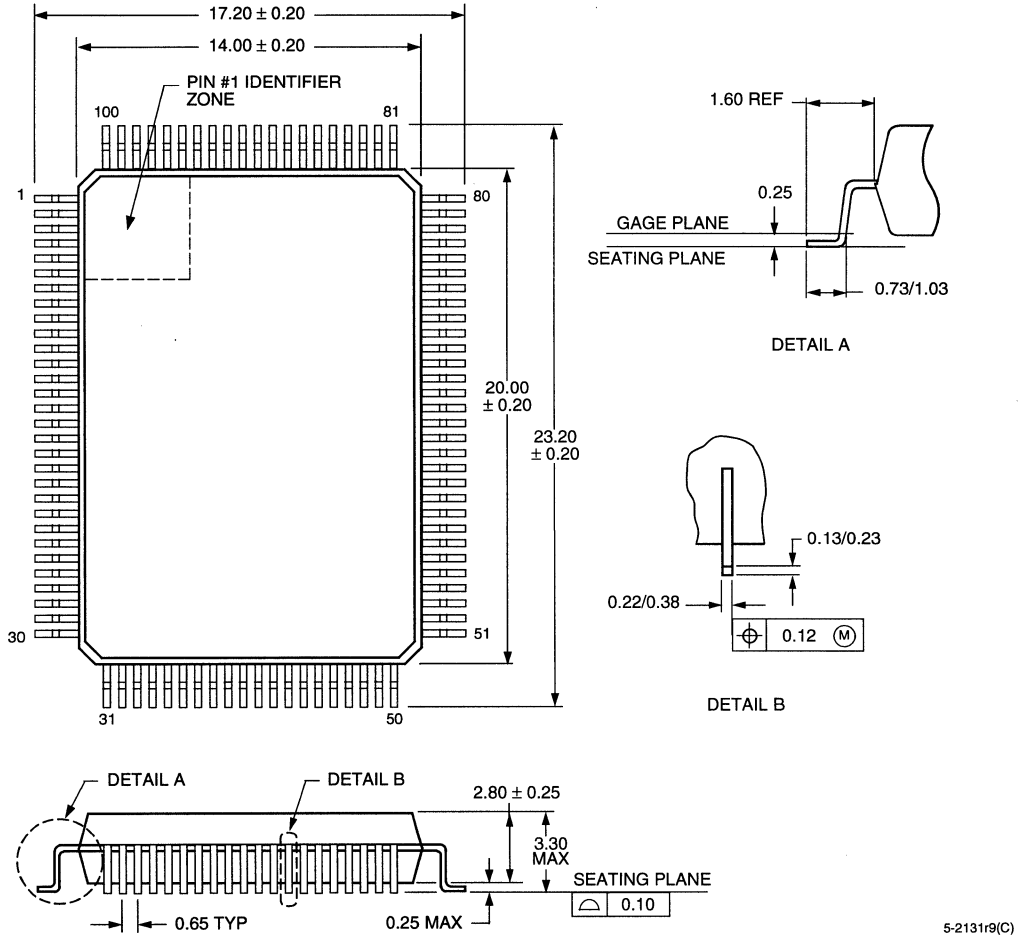
Dimensions are in millimeters.



Outline Diagrams (continued)

100-Pin QFP

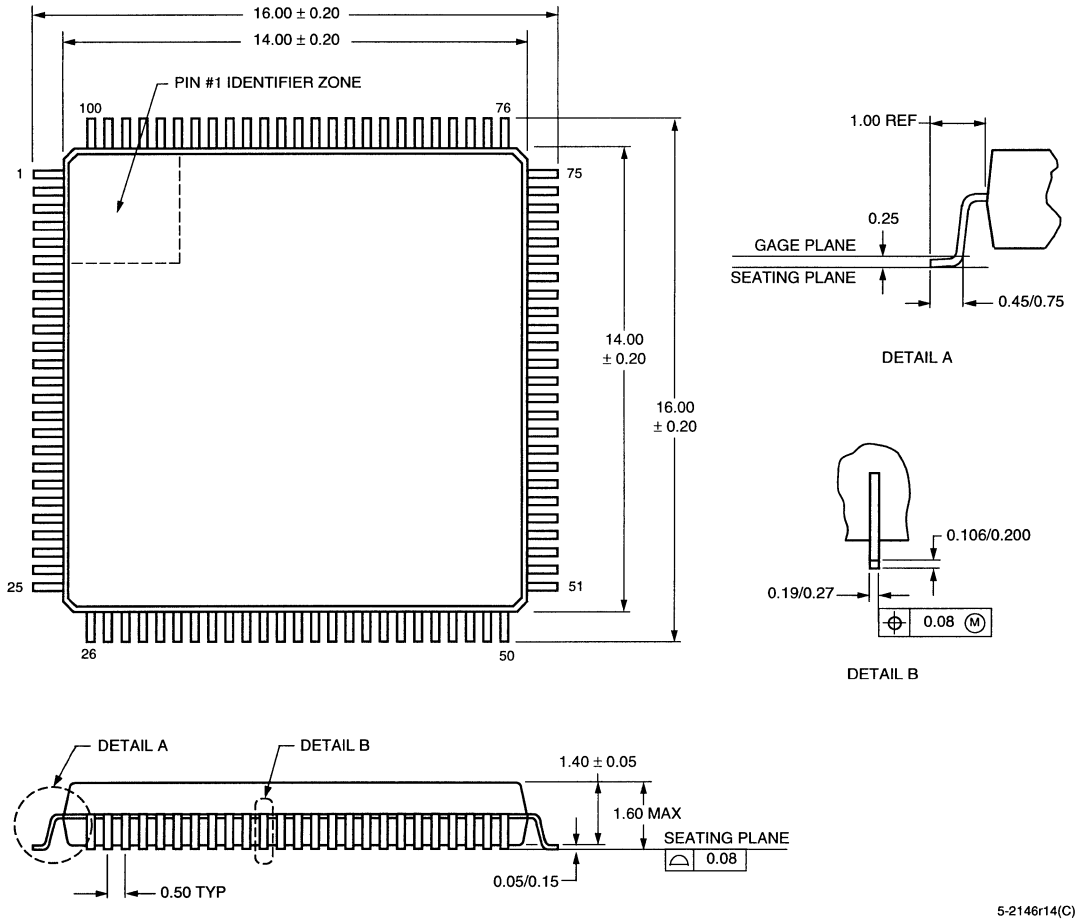
Dimensions are in millimeters.



Outline Diagrams (continued)

100-Pin TQFP

Dimensions are in millimeters.

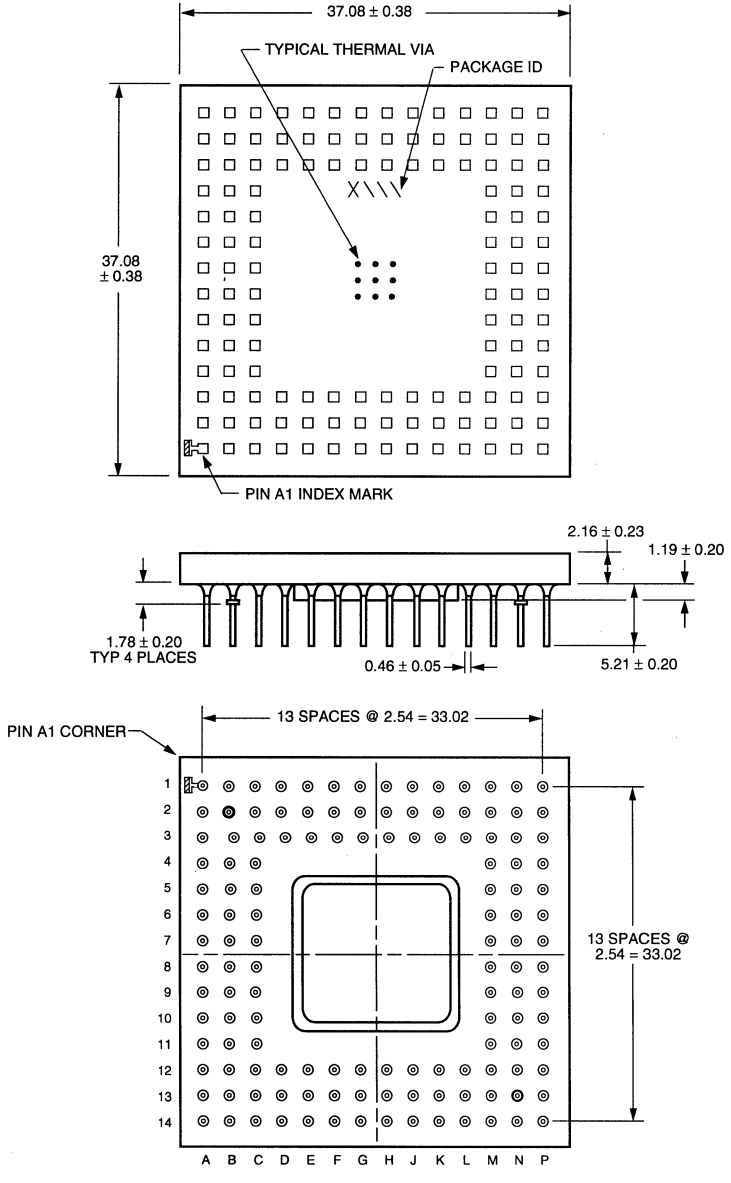


5-2146r14(C)

Outline Diagrams (continued)

132-Pin PPGA

Dimensions are in millimeters.

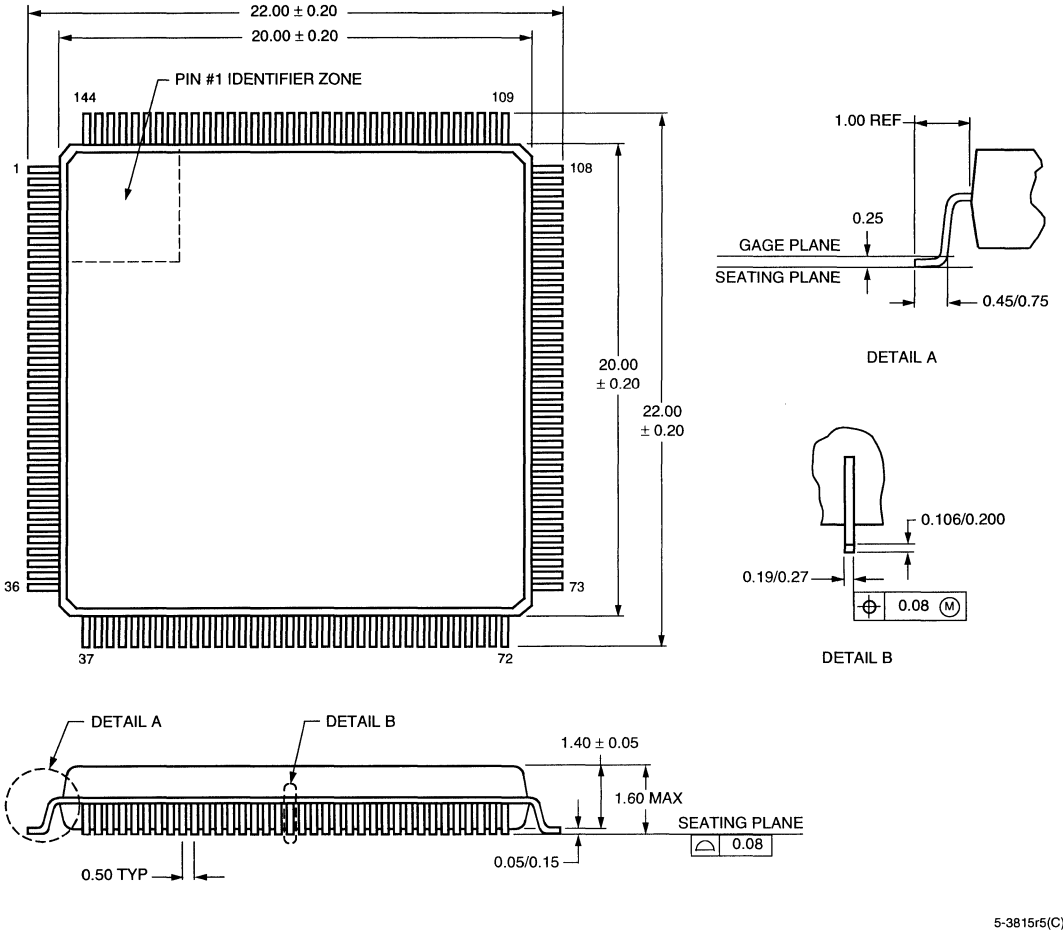


5-2115(C)

Outline Diagrams (continued)

144-Pin TQFP

Dimensions are in millimeters.

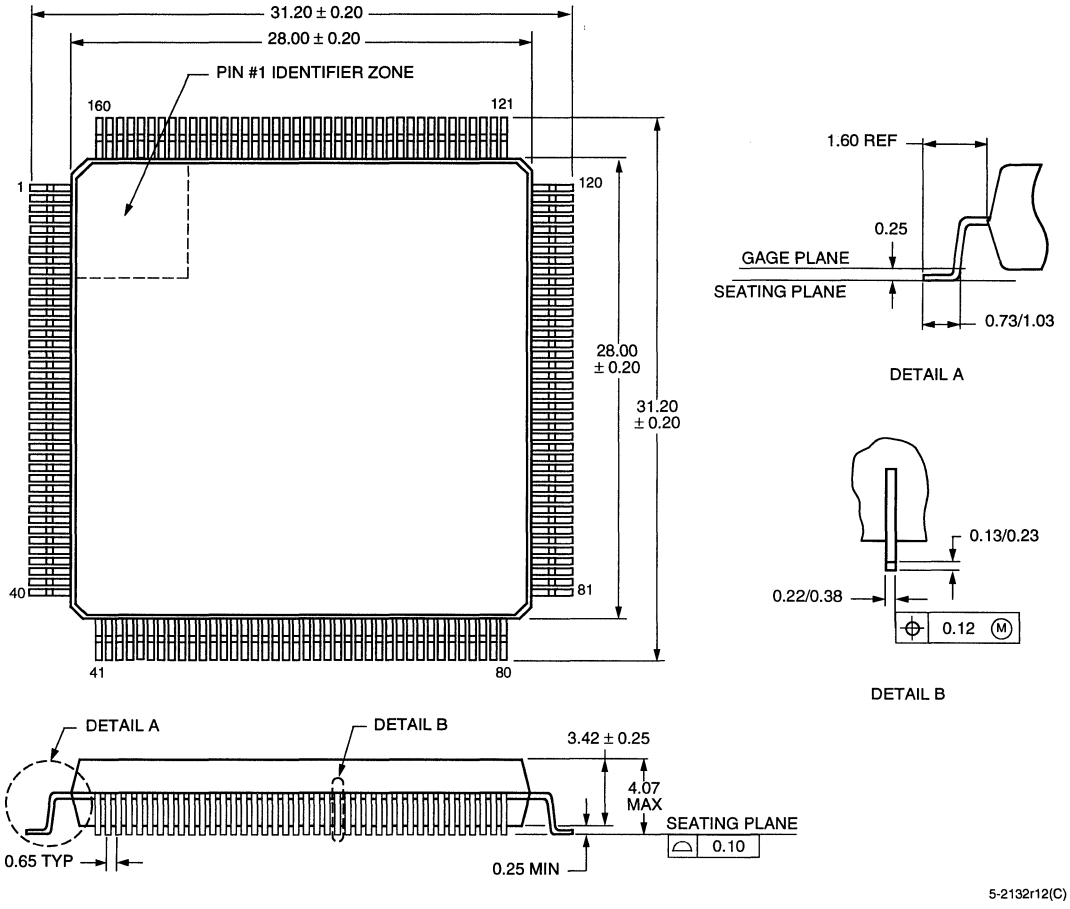


5-3815r5(C)

Outline Diagrams (continued)

160-Pin QFP

Dimensions are in millimeters.

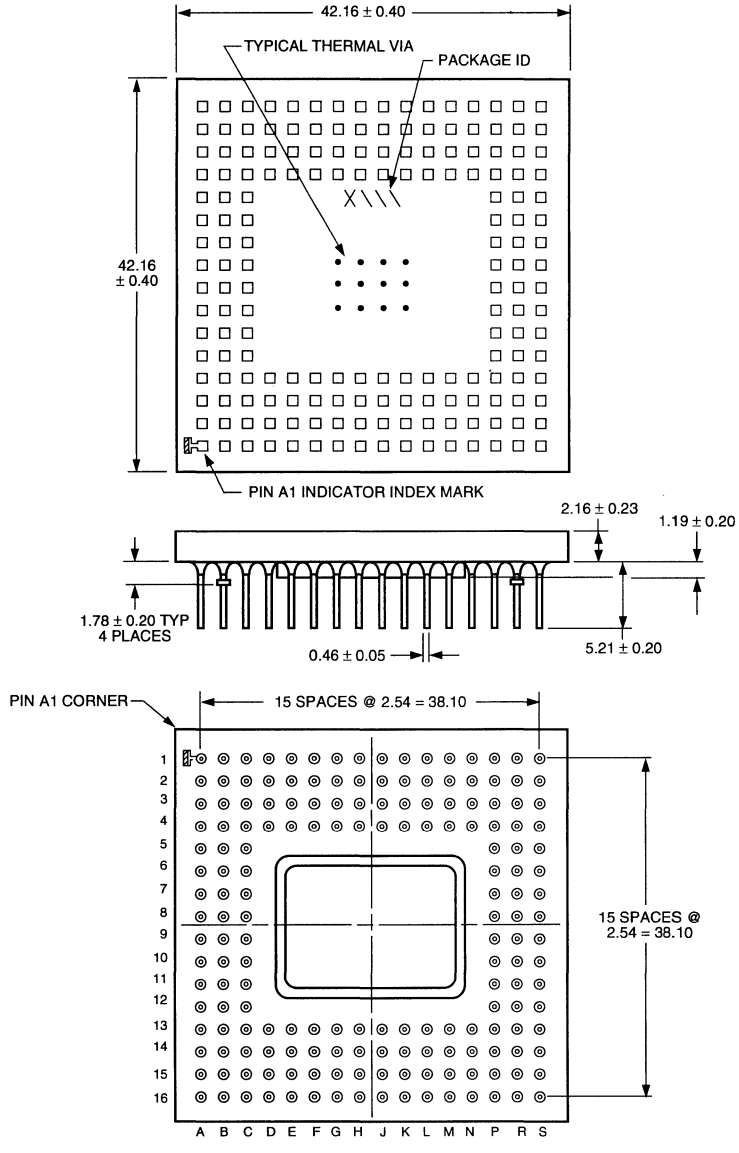


5-2132r12(C)

Outline Diagrams (continued)

175-Pin PPGA

Controlling dimensions are in inches.

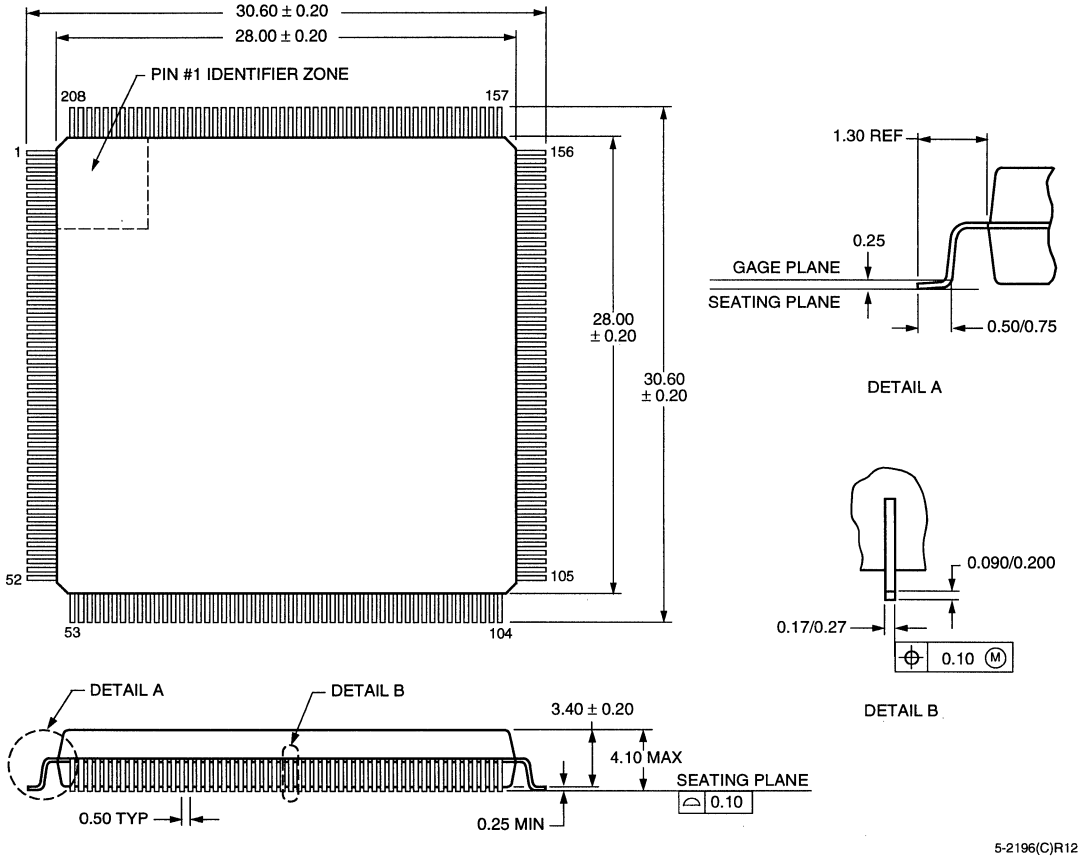


5-2116(C)

Outline Diagrams (continued)

208-Pin SQFP

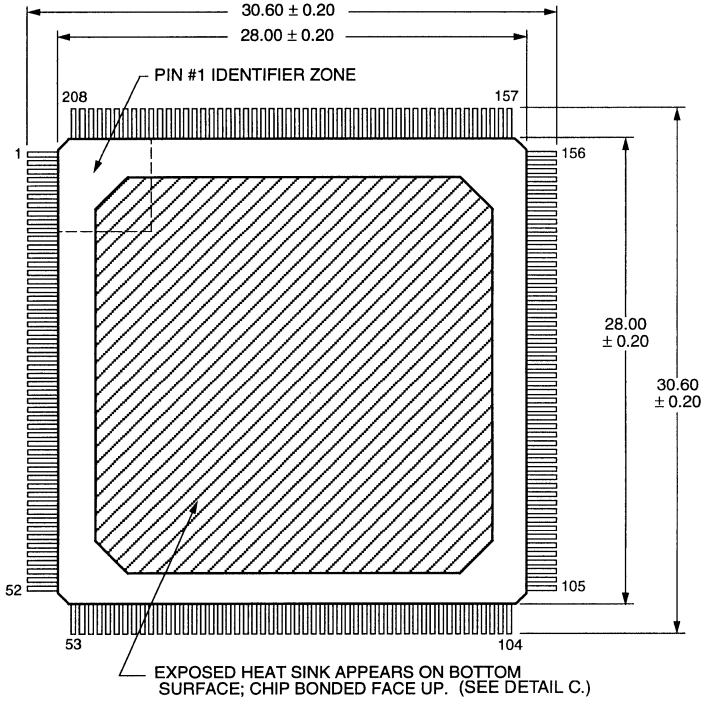
Dimensions are in millimeters.

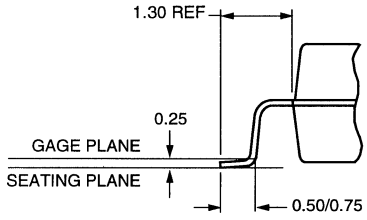


Outline Diagrams (continued)

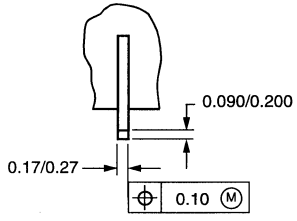
208-Pin SQFP2

Dimensions are in millimeters.

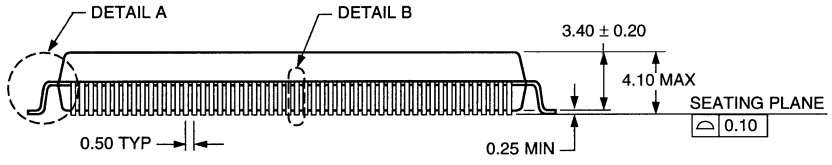




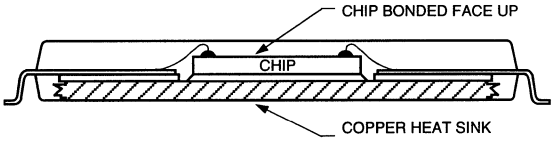
DETAIL A



DETAIL B



5-3828(C)R4



DETAIL C (SQFP2 CHIP-UP)

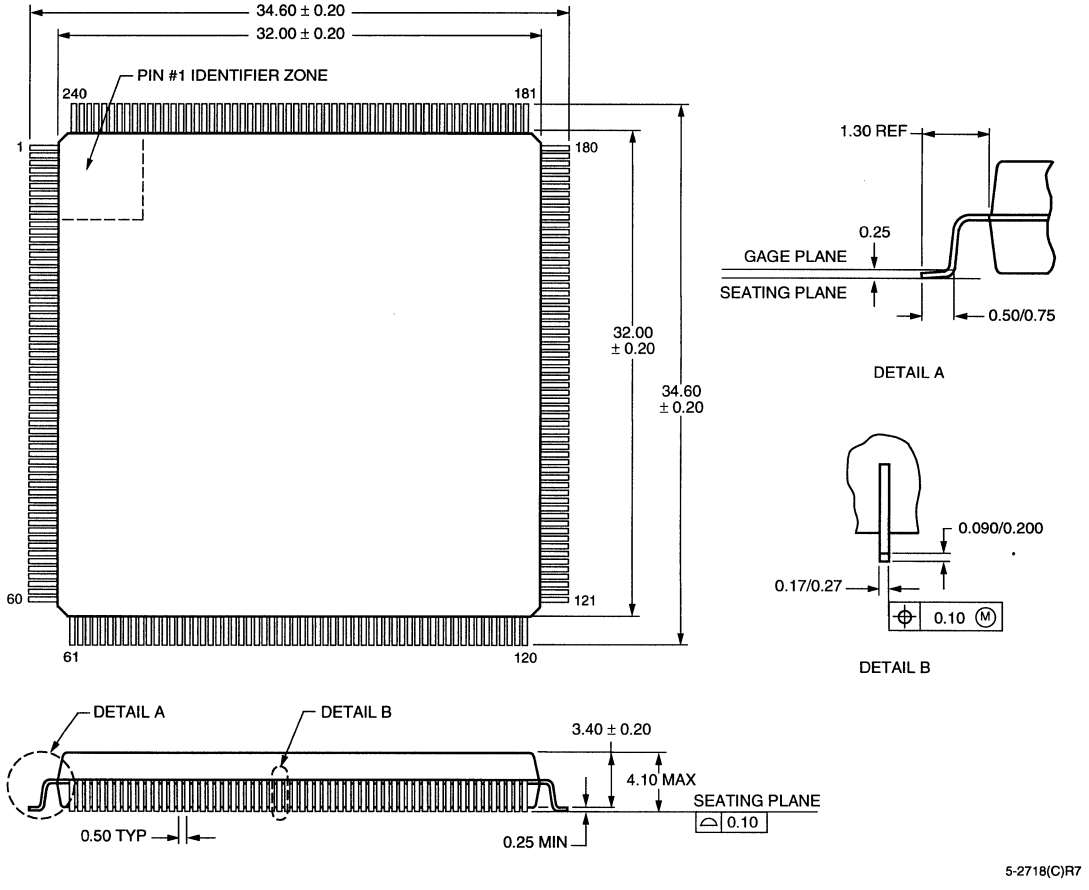
Note: Heat sink appears on top surface for ATT2C40 device (chip bonded face down).

5-4946(C)

Outline Diagrams (continued)

240-Pin SQFP

Dimensions are in millimeters.

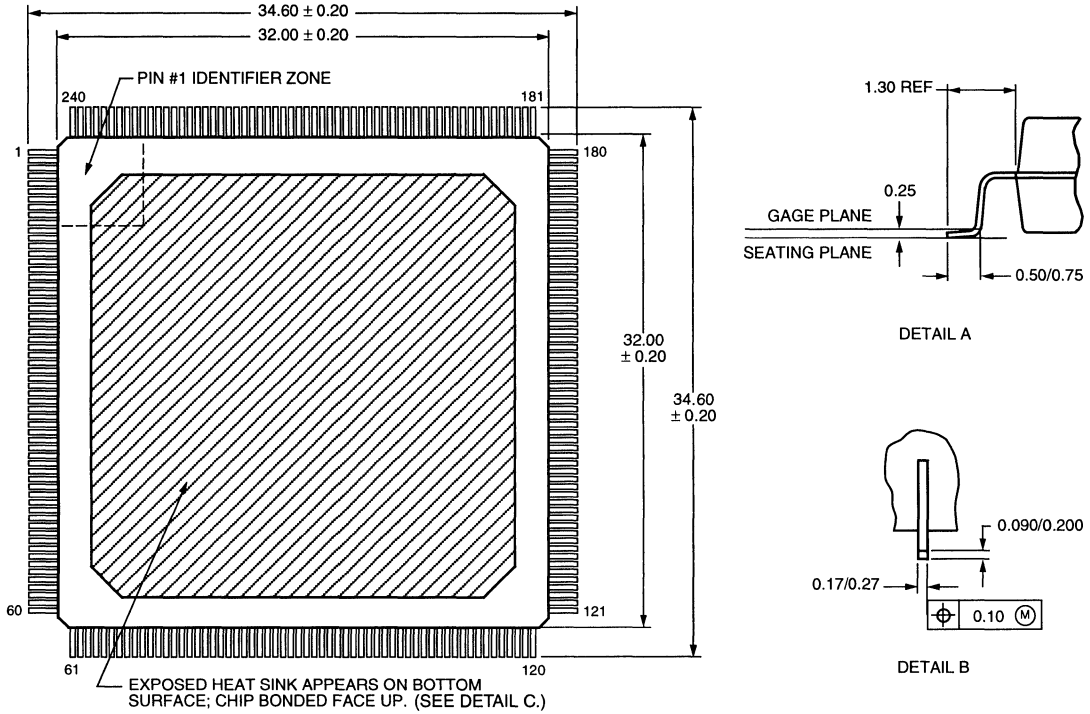


5-2718(C)R7

Outline Diagrams (continued)

240-Pin SQFP2

Dimensions are in millimeters.



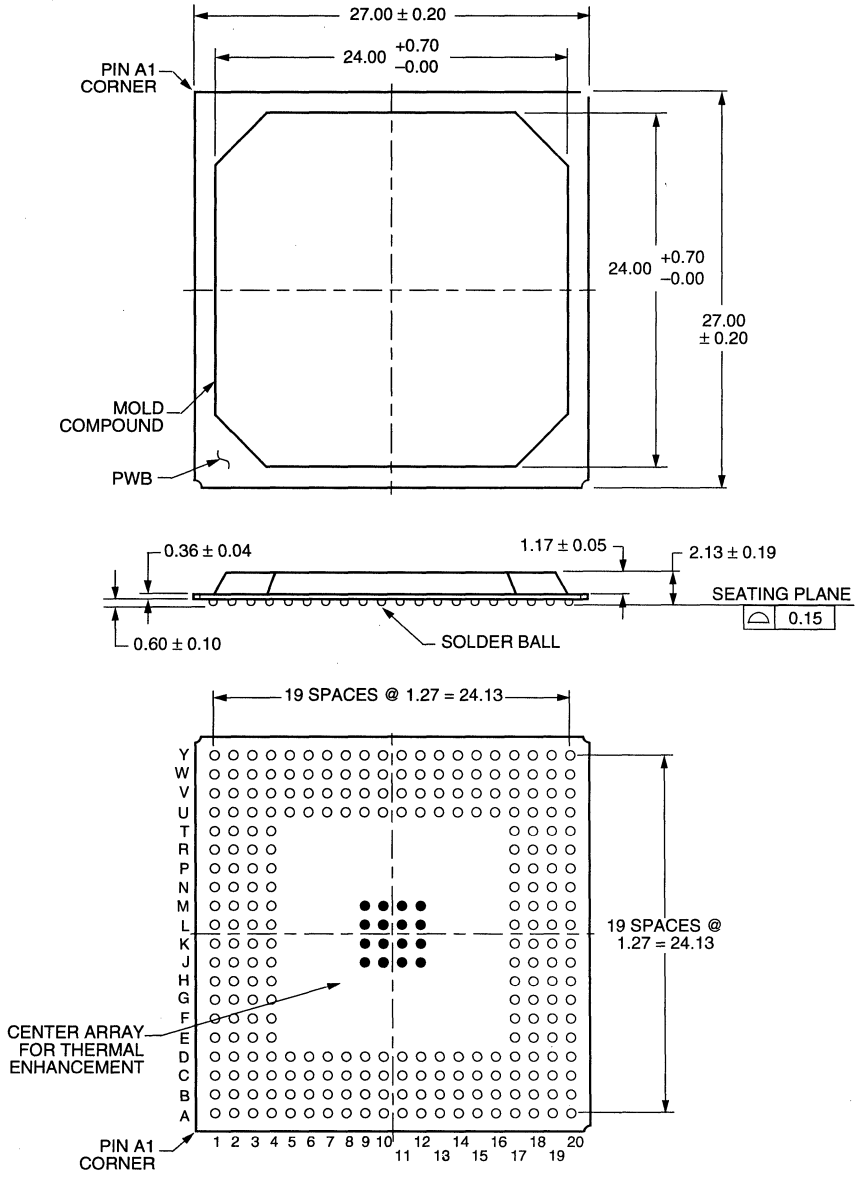
Note: Heat sink appears on top surface for ATT2C40 device (chip bonded face down).

5-4946(C)

Outline Diagrams (continued)

256-Pin PBGA

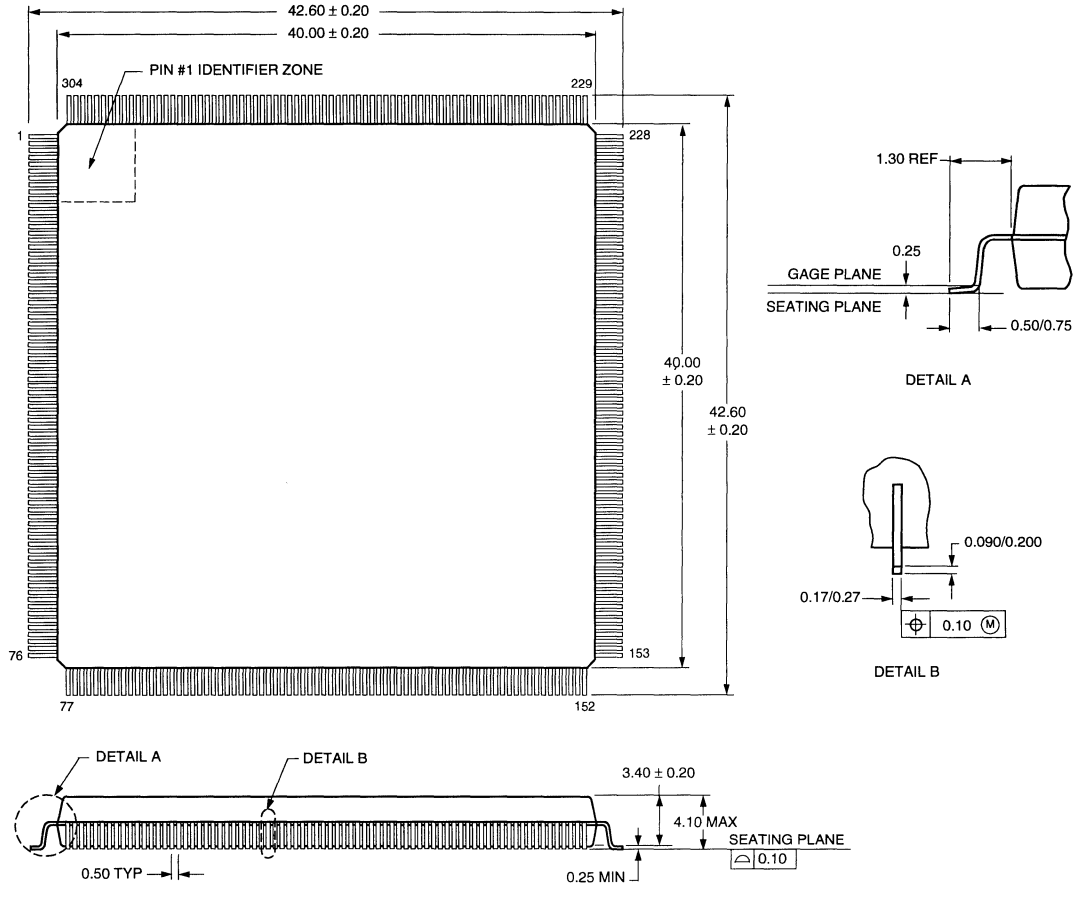
Dimensions are in millimeters.



Outline Diagrams (continued)

304-Pin SQFP

Controlling dimensions are in millimeters.

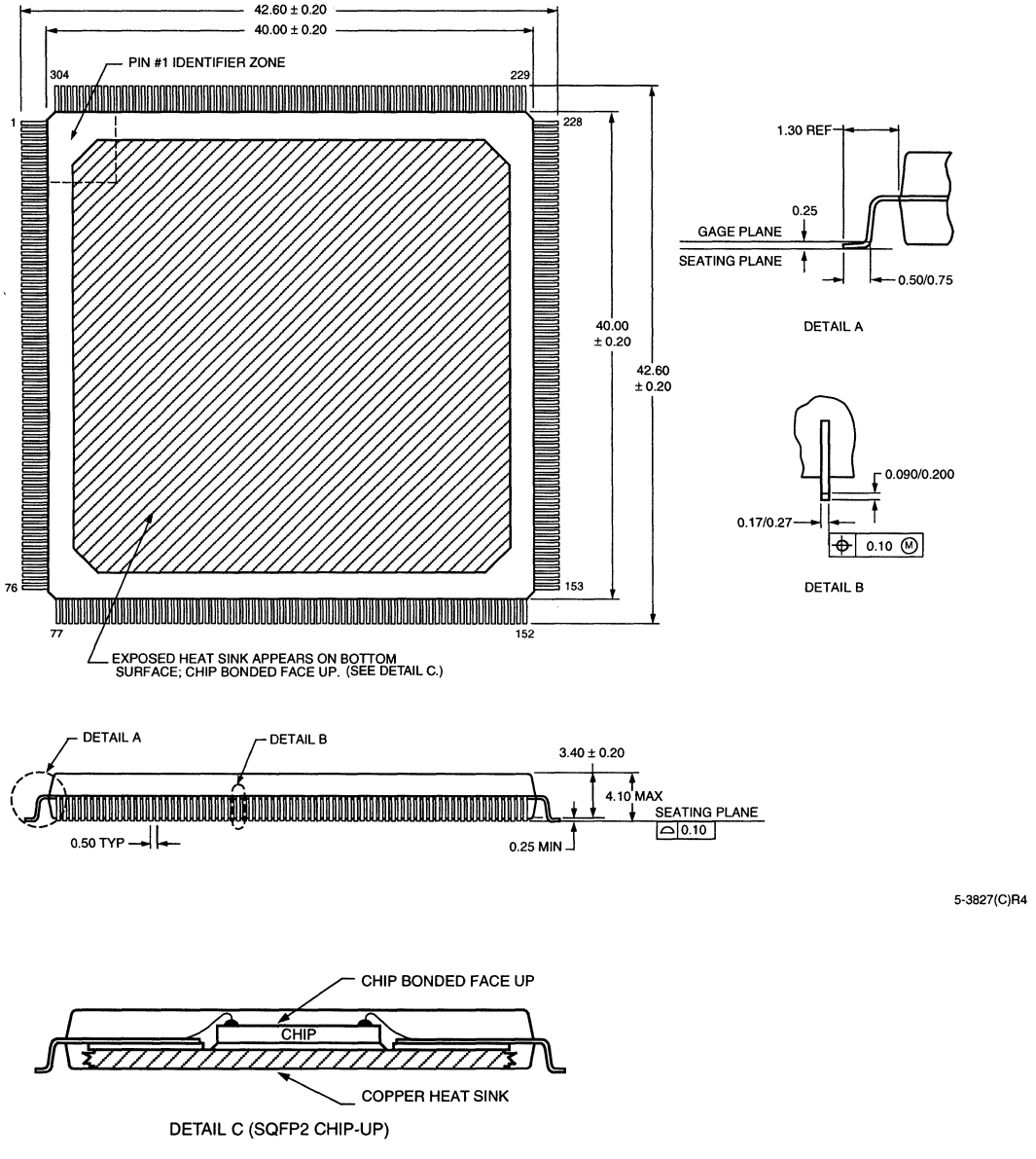


5-3307(C)R7

Outline Diagrams (continued)

304-Pin SQFP2

Dimensions are in millimeters.



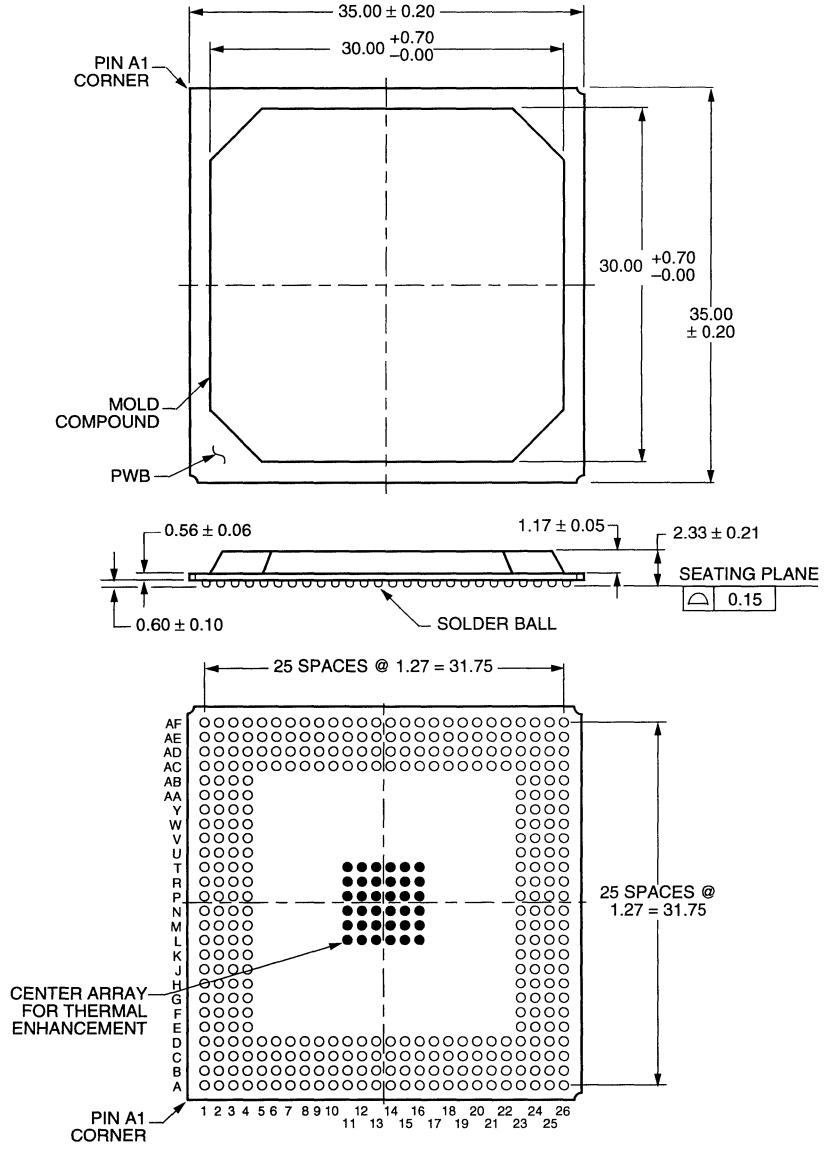
Note: Heat sink appears on top surface for ATT2C40 device (chip bonded face down).

5-4946(C)

Outline Diagrams (continued)

352-Pin PBGA

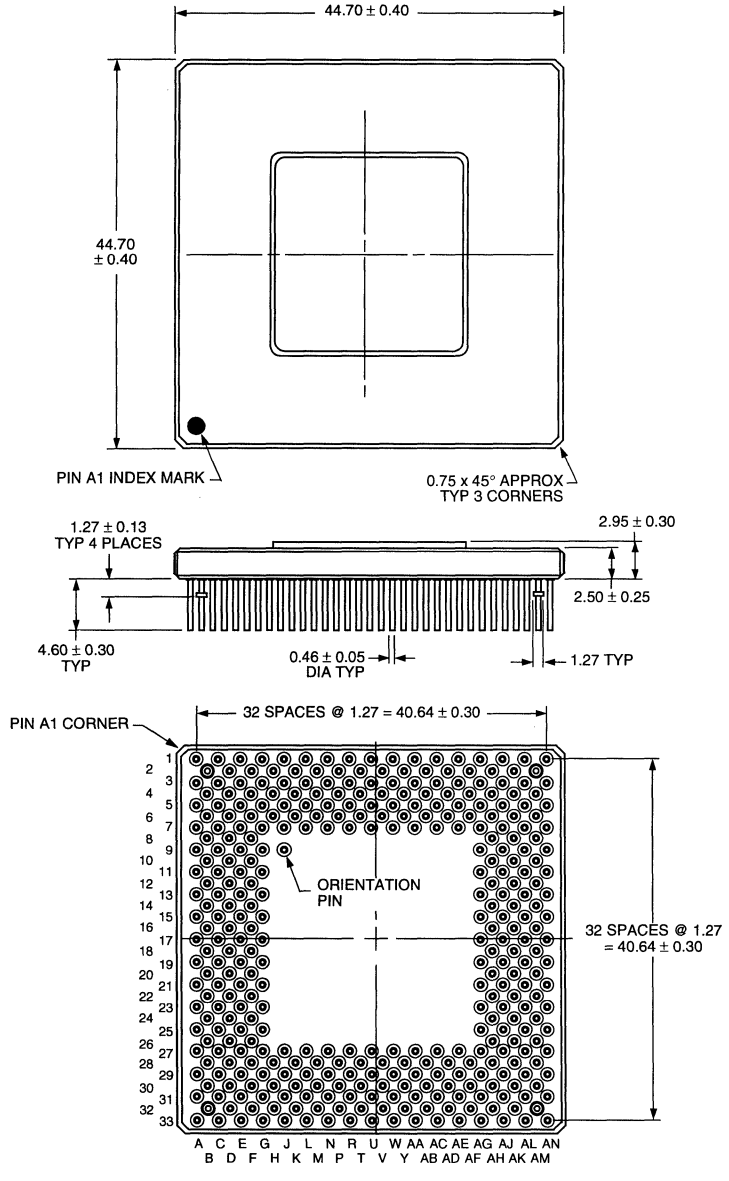
Dimensions are in millimeters.



Outline Diagrams (continued)

364-Pin CPGA

Dimensions are in millimeters.

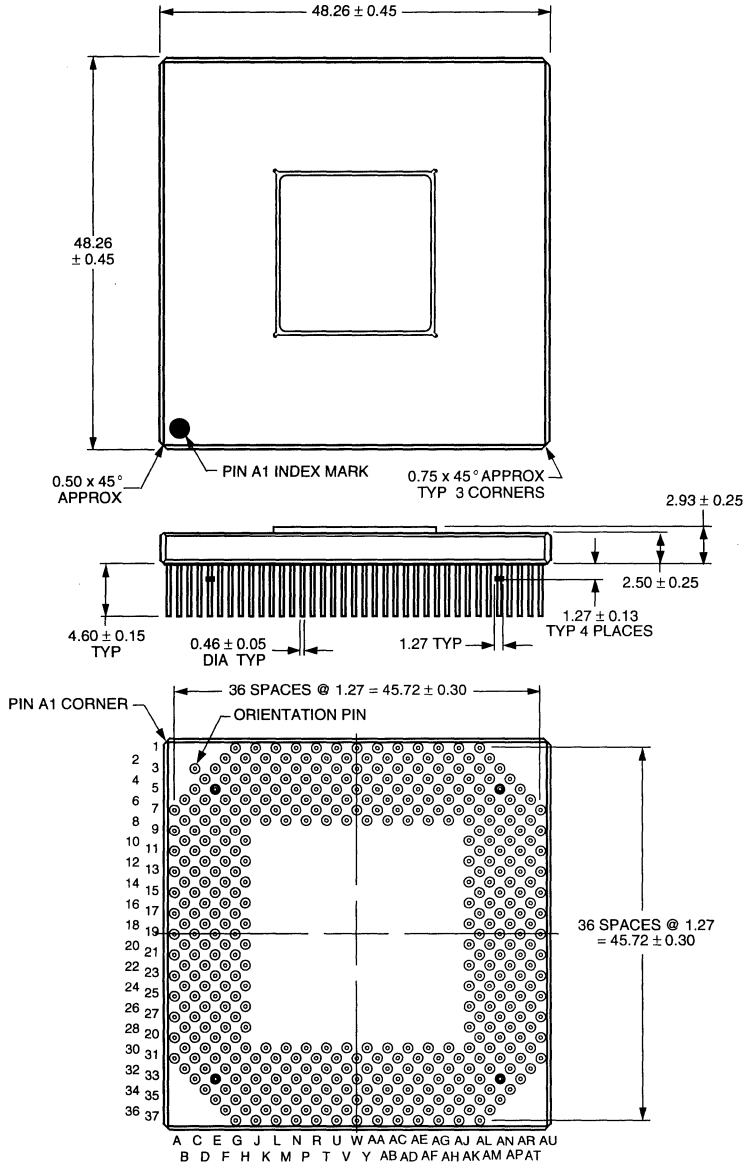


5-3318r5(C)

Outline Diagrams (continued)

428-Pin CPGA

Dimensions are in millimeters.

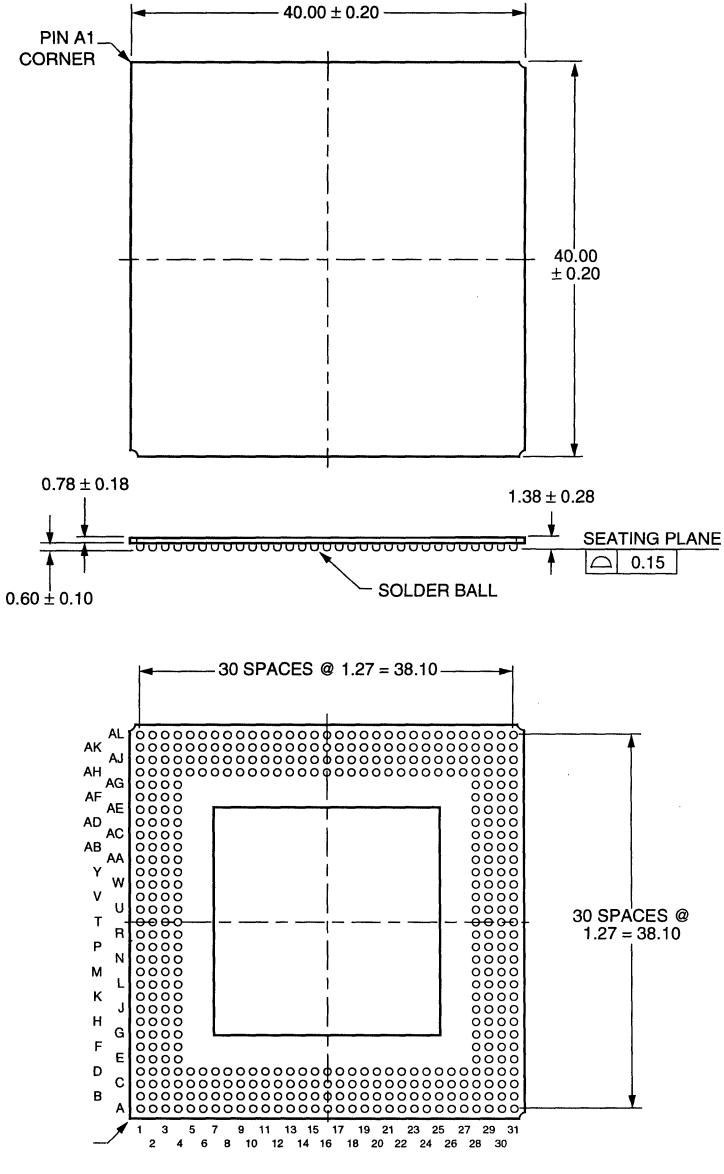


5-3826r3(C)

Outline Diagrams (continued)

432-Pin EBGA

Dimensions are in millimeters.

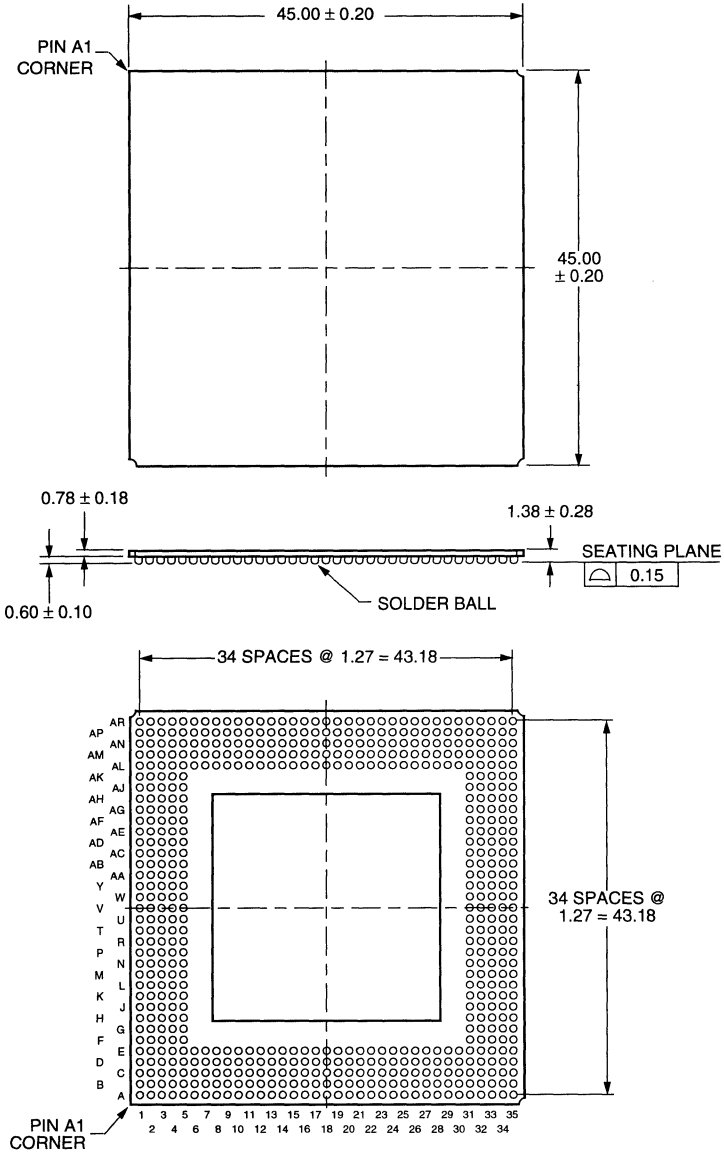


5-4409(C)

Outline Diagrams (continued)

600-Pin EBGA

Dimensions are in millimeters.



5-4408(C)

Lucent Technologies' Packing Methods

Dry Packing

Lucent Technologies packs moisture-sensitive components in compliance with the IPC-SM-786 Standard. All of our devices have been assigned moisture sensitivity ratings (levels 1 through 6). Products that require dry packing (i.e., products rated at levels 2 through 6) have their sensitivity rating levels identified on the dry pack bag (please refer to A88AL1005 for detailed information on moisture sensitivity classification procedures). Level 1 products, such as DIPs and the smaller SOICs and PLCCs, are considered nonmoisture-sensitive and are not dry packed. Customers may request dry packing of nonmoisture-sensitive products whenever necessary.

Moisture Sensitivity Levels

Descriptions for each of the moisture sensitivity levels with which Lucent rates its packages are presented below, followed by a table that summarizes the test and storage conditions associated with each moisture level.

Level 1: Devices meet all lot tolerance percent defective requirements after 168 hours (–0/+4 hours) of 85 °C (±2°) at 85% (±5%) relative humidity moisture preconditioning, followed by class and THB. This simulates unlimited storage at up to 90% RH and 30 °C prior to board mounting. Devices do not require dry packing.

Level 2: Devices meet all lot tolerance percent defective requirements after 168 hours (–0/+4 hours) of 85 °C (±2°) at 60% (±5%) relative humidity moisture preconditioning, followed by class and THB. This simulates 1 year of storage at up to 60% RH and 30 °C prior to board mounting. Devices should be dry packed or baked before use.

Level 3: Devices meet all lot tolerance percent defective requirements after 192 hours (–0/+4 hours) of 30 °C (±2°) at 60% (±5%) relative humidity moisture preconditioning, followed by class and THB. This simulates 168 hours of storage at up to 60% RH and 30 °C prior to board mounting. Devices should be dry packed or baked before use.

Level 4: Devices meet all lot tolerance percent defective requirements after 78 hours (–0/+4 hours) of 30 °C (±2°) at up to 60% (±5%) relative humidity moisture preconditioning, followed by class and THB. This simulates 72 hours of storage at up to 60% RH and 30 °C prior to board mounting. Devices should be dry packed or baked before use.

Level 5: Devices meet all lot tolerance percent defective requirements after 30 hours (–0/+4 hours) of 30 °C (±2°) at 60% (±5%) relative humidity moisture preconditioning, followed by class and THB. This simulates 24 hours of storage at up to 60% RH and 30 °C prior to board mounting. Devices should be dry packed or baked before use.

Level 6: Devices meet all lot tolerance percent defective requirements after 6 hours (–0/+0.5 hours) of 30 °C (±2°) at 60% (±5%) relative humidity moisture preconditioning, followed by class and THB. This simulates 6 hours of storage at up to 60% RH and 30 °C prior to board mounting after a 125 °C, 8-hour bake. Devices do not require dry packing.

Table 5. Moisture Sensitivity Levels Summary

Test Conditions					Storage Conditions		
Level	**Hours**	**Temp. (°C)**	**Humidity (% RH)**	**Dry Pack**	**Temp. (°C)**	**Humidity (% RH)**	**Time**
1	168	85	85	No	30	90	Unlimited
2	168	85	60	Yes	30	60	1 year
3	192	30	60	Yes	30	60	168 hours
4	78	30	60	Yes	30	60	72 hours
5	30	30	60	Yes	30	60	24 hours
6	6	30	60	No	30	60	6 hours after bake

Lucent Technologies' Packing Methods (continued)

Table 6. Package Moisture Sensitivity Levels

Package	Level
8-Pin DIP*	—
8-Pin SOIC	1
20-Pin PLCC	1
44-Pin PLCC	1
68-Pin PLCC	1
84-Pin PLCC†	1
100-Pin QFP	3
100-Pin TQFP	3
132-Pin PPGA*	—
144-Pin TQFP	3
160-Pin QFP	3
175-Pin PPGA*	—
208-Pin SQFP	3
208-Pin SQFP2	3
240-Pin SQFP	3
240-Pin SQFP2	3
256-Pin PBGA	4
304-Pin SQFP	3
304-Pin SQFP2	3
352-Pin PBGA	4
364-Pin CPGA*	—
428-Pin CPGA*	—
432-Pin EBGA	4
600-Pin EBGA	4

* The 8-pin DIP and all plastic or ceramic pin grid arrays (PPGAs or CPGAs) assume board mounting via wave soldered or socket mount.

† Moisture sensitivity default is Level 1, although some devices (ATT3064 and ATT3090) have been assigned a default of Level 3.

Tape-and-Reel Packing

All product is taped in compliance with EIA-481 Standards (please refer to A95AL0056 for detailed information on taping procedures). 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.

JEDEC Tray Packing

All of the trays used by Lucent comply with JEDEC standards. We are now using thin-style trays for all plastic package types (BQFP, QFP, SQFP, and TQFP) and only high-temperature trays are being used. In general, all tray packed products are shipped in stacks of 4 trays (5, with the cover). For detailed information of the standard packing multiples for tray-packed products, please see the table provided in A93AL1146, the JEDEC specification on Primary Packing Materials and Quantities.

Valid Packing Options for MOS Devices in Plastic Packages

The following information is intended to document valid packing options:

- BQFP, QFP, SQFP, and TQFP devices can only be shipped in trays:
  - — Bakable tray standard packing (-BT)
  - — Bakable tray dry pack (-DB)
- SOIC and PLCC devices can be shipped either in tubes or in tape and reel:
  - — Tube standard packing (blank)
  - — 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 (blank)

Lucent Technologies Packing Methods (continued)

Table 7. Packing Options

Package	Capacity/Carrier	Carrier Number	Total
8-Pin DIP	48/Tube	PS-25455	70 Tubes/Box
8-Pin SOIC	92/Tube 2500/Reel	PS25407/6ZPE00001 PS25528	100 Tubes/Box —
20-Pin PLCC	46/Tube 1000/Reel	6ZPF020100 —	60 Tubes/Box —
44-Pin PLCC	27/Tube 500/Reel	PS-25388/6ZPF04402 PS-25445/PS-25467	40 Tubes/Box —
68-Pin PLCC	18/Tube 250/Reel	6ZPF06803 PS-25444	25 Tubes/Box —
84-Pin PLCC	15/Tube 200/Reel	PS-25410/6ZPF08410 PS-25443	25 Tubes/Box —
100-Pin QFP	66/Tray	—	4 Trays/Box
100-Pin TQFP	90/Tray	—	4 Trays/Box
132-Pin PPGA	20/Chipboard Box	PS-25477	8 Boxes/Carton
144-Pin TQFP	60/Tray	—	4 Trays/Box
160-Pin QFP	24/Dry-packed, Bakable Trays	—	4 Trays/Box
175-Pin PPGA	16/Chipboard Box	PS-25477	8 Boxes/Carton
208-Pin SQFP	24/Tray	—	4 Trays/Box
240-Pin SQFP	24/Tray	—	4 Trays/Box
256-Pin PBGA	40/Tray	—	4 Trays/Box
304-Pin SQFP	12/Tray	—	4 Trays/Box
352-Pin PBGA	24/Tray	—	4 Trays/Box
364-Pin CPGA	10/Chipboard Box	PS-25477	8 Boxes/Carton
428-Pin CPGA	10/Chipboard Box	PS-25477	8 Boxes/Carton
432-Pin EBGA	21/Tray	—	4 Trays/Box
600-Pin EBGA	12/Tray	—	4 Trays/Box

Lucent Technologies
Bell Labs Innovations

Qualification Information

Lucent Technologies' Quality Policy

Policy

Quality excellence is the foundation for the management of our business and the keystone of our goal of customer satisfaction. Therefore, it is 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 at Lucent Technologies. It lies at the heart of everything we do.

Through active planning of 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.

Responsibility

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 surpass 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.

Lucent Technologies' Approach to Quality

It is the objective of Lucent Technologies to be rated by our customers as their number one vendor. It is our purpose to provide quality products made with advanced technologies at competitive prices.

At Lucent Technologies, we strive to:

- Understand customers' expectations and consistently meet and exceed them.
- Establish high quality standards and engineer processes to ensure conformance to those standards.
- Work toward continuous improvement of all processes by actively involving every employee supported by informed management leadership.
- Target errors or defects and work toward root cause elimination.
- Adopt perfection as an organizational goal.

In this way, Lucent Technologies will continue to improve the quality of product design and the quality of the manufacturing process.

Our emphasis is on prevention. By utilizing engineering tools, we are striving to eliminate defects and reduce variation in final product.

Education and training in the values and techniques of quality manufacturing extends to all levels of the work force. Through quality and reliability improvements, using tools such as statistical process control and advanced failure mode analysis, Lucent Technologies is driving toward total customer satisfaction.

Lucent Technologies' quality and reliability programs and resources 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, Lucent Technologies' emphasis on customer service and feedback ensures prompt response and timely solutions in case of device shortcomings after delivery.

Lucent Technologies' quality process addresses all elements of product realization and continues with after-sale support.

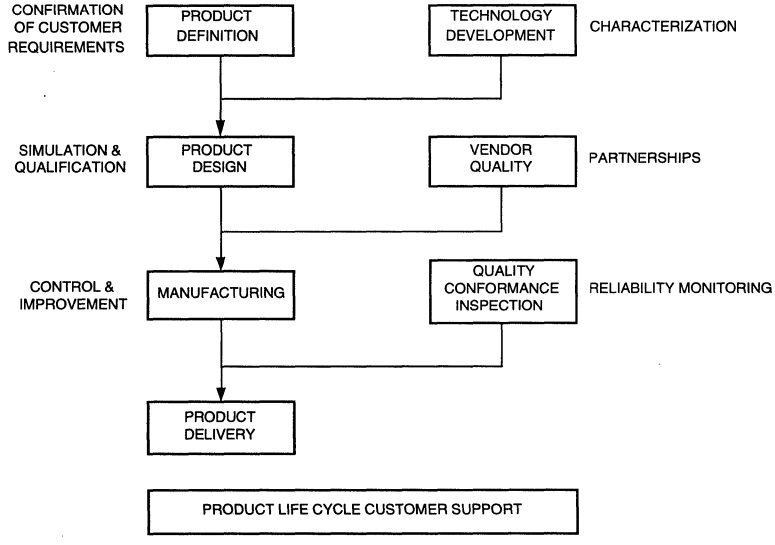


Figure 1. MOS Quality Process

Lucent Technologies' Quality Plan

Zero Defects and 100% On-Time Delivery, Meeting Our Customers' Requirements

The strategy for achieving this goal is for the Quality Council to lead employees in the implementation of Statistical Process Control (SPC) to improve customer satisfaction and achieve business objectives.

The Quality Council is composed of Marketing, R&D, Manufacturing, Finance, and Quality Executives, and is chaired by the Chief Operating Officer. Two steering committees, the Customer Satisfaction and the Quality-in-Design and Development groups, lead the process management teams in a Total Quality Management system.

Policy deployment methodology translates objectives into operational plans executed by Quality Improvement Teams. The Customer-Supplier Model focuses the teams' improvement efforts on customer satisfaction.

SPC is defined in its broadest sense—total employee involvement in improving operations. Teams are trained in the use of quality tools and quality metrics to control processes in a driving effort to reduce variations.

The quality system is patterned after the ISO 9000 quality elements. It uses the unabridged Malcolm Baldrige National Quality Award criteria as an objective, comprehensive assessment of our Total Quality Management system. We consider this assessment a direct measure of the effectiveness of our quality plans and, in particular, our commitment to customers.

The Malcolm Baldrige Award criteria provides us with an integrated systems approach to addressing:

- Customer needs and expectations
- Product and service quality
- Strategic quality planning
- Information management
- Quality assurance
- Human resource management and development
- Leadership for quality

We use the feedback from the assessments as a framework for our quality improvement plans and have been awarded the internal Lucent Technologies Network Systems quality award based on the Malcolm Baldrige Award criteria.

Reliability Education and Training

Education and training are crucial to our reliability improvement effort, and they extend to all levels of the work force. Our manufacturing staff attends training programs covering electrostatic discharge (ESD) control, just-in-time (JIT) manufacturing concepts and methodology, teamwork, and statistical process control. SPC-In-Action training is an operator-level training course in statistical process control, including math basics as well as team problem-solving. The course is designed to give operators hands-on training with actual work examples.

Our development employees receive reliability training courses such as the Statistical Reliability Workshop, Robust Design Workshop, and Control of Electrostatic Discharge Workshop. In these courses, our staff becomes more highly skilled in using statistical techniques to build higher reliability into our products. They learn to use computer-based tools to analyze reliability data more effectively. They also become more familiar with reliability models, graphical methods, maximum likelihood estimation, accelerated life testing, and development of life tests and field-tracking studies.

Lucent Technologies' Product Qualification Process

The process of qualifying a new product at Lucent Technologies is divided into two major efforts: qualification of a new technology and qualification of new device types or designs in the new technology.

Both of these qualifications are described in detail in this section.

Technology Qualification Plan

Lucent Technologies' Process Qualification Plan for both new wafer fabrication and package technologies involves rigorous environmental, mechanical, and electrical testing to confirm technology robustness. Industry-standard tests are applied to test devices. Specialized tests to examine electrostatic discharge and latch-up parameters are also incorporated.

Prior to qualification and manufacture of devices using a new process technology, thousands of ICs from different device types involving multiple wafer lots are subjected to extended reliability tests, and extensive generic reliability tests are conducted.

After the technology is introduced into manufacture, a continuous quality improvement program is begun. Improvements are introduced into the technology, with new issues of the technology put into effect at a rate of about one per year.

Reliability and yield studies are performed on a standard evaluation circuit (SEC).

Testing

Much of the evaluation of a new technology is done with an SEC. The SEC is tested with a very rigorous methodology, intended to ensure the device functions well beyond its intended operating range.

A voltage stress test is included to detect breakdown of any weak oxides. The device is required to be operational during this stress test. Functional testing is done before and after stress testing. Other tests are performed to detect low-level transistor leakage; these include a variety of hold time tests with various patterns. The same test routine is performed in at least three manufacturing stages: at wafer probe, after packaging, and at the end of reliability testing. Hold time and voltage guard band limits are built into the testing.

Intrinsic Reliability Data

Electromigration

Electromigration is a well-known failure mechanism which affects continuity or isolation of interconnections. Interconnections are allowed a maximum failure budget of 10 FITS at a junction temperature of 85 °C over 40 years.

This failure mechanism follows a log-normal distribution. Activation energies, standard deviations, and median times to failure are measured separately for each level in the interconnect structure. For failure rate calculations, the die dissipates about 2 W of power. Measurements are made with current levels 10 to 20 times higher than the maximum allowed by the current density design rules.

Hot Electron Effects

Device aging typically shows greater than 10 years to 10% Gm degradation under worst-case substrate current conditions. Actual devices should not be affected by charge injection mechanisms.

Mobile Ion Contamination

Mobile ion contamination is checked at wafer level using a standard shop procedure.

Time-Dependent Dielectric Breakdown (TDDB)

Extensive time-dependent dielectric breakdown measurements are made on specially designed large area test structures taken from the line monitor circuit. These devices are packed and put under test at electric fields as high as 6 mV/cm and temperatures as high as 150 °C. Both the thermal and voltage characteristics of the oxide failures are studied. The thermal activation energy is shown to be approximately 0.6 eV at an electric field of 6 mV/cm and a linear electric field acceleration parameter of three decades/(mV/cm). Correlations have been made with wafer-level tests using a gate oxide zone tester. This tester is used to routinely monitor oxide quality.

Lucent Technologies' Product Qualification Process (continued)

Lucent Technologies' Product Qualification Plan

Lucent Technologies' product qualification plan involves rigorous environmental, mechanical, and electrical testing to confirm product soundness.

Industry-standard tests, generally derived from military methods, are applied to test devices; additional specialized tests are administered to determine electrostatic discharge and latch-up sensitivity.

Table 1 presents a menu of tests from which a qualification plan for a new device is developed. It gives the sample size and lot tolerance percent defective (LTPD) for each test. The qualification process is administered by the Qualification Review Board (QRB) described in this section.

Depending upon whether the qualification is for a package, die, or process change—or for a change in fabrication facilities—the QRB determines the tests from Table 1 to be administered. Tests normally required for a new qualification can be satisfied by a reference to recent successful testing on a similar product under appropriate conditions. The QRB maintains a database called the Qualification Testing Results System (QTRS) that contains accurate records of each qualification.

The QRB is also responsible for the requalification of any changes in design, fabrication, or packaging of integrated circuit products. The requalification process implemented by the QRB is shown in Figure 2. Note that the QRB determines the seriousness of the change and supervises the requalification. Note also that the customer is involved in the requalification process.

Lucent Technologies' Product Qualification Process (continued)

Table 1. IC Qualification Tests

Test No./ Symbol	Test Description	Method	Sample Size	LTPD
1. LT–1 or LT–2	High-Temperature Operating Bias, 125 °C High-Temperature Operating Bias, 150 °C	M-1005 M-1005	195 100	2 4
2. CL	CLASS (Component Lead Assembly Simulation)	L-757214	132	5
3. BH	Temperature-Humidity Bias	L-757679	129	3
4. SB	Steam Bomb	L-757680	105	5
5. TC	Temperature Cycling	M-1010	105	5
6. TS	Thermal Shock	M-1011	25	15
7. MR	Moisture Resistance	M-1004	38	10
8. LK	Gross/Fine Leak	M-1014	38	10
9. SA	Salt Atmosphere	M-1009	15	15
10. WV	Internal Water Vapor	M-1018	5	50
11. RT	Low-Temperature Aging	L-757203	77	5
12. SE	Soft Error Rate	M-1018	10	—
13. PS	Photo Sensitivity	—	10	22
14. FL	Flammability and O_2 Index	*UL* 94 and ASTM 2863-77	—	—
15. SR	Solvent Resistance	M-2015	8	—
16. IV	Internal Visual	M-2014	5	—
17. PD	Physical Dimensions	M-2016	15	—
18. SD	Solderability	M-2003	22	10
19. MS	Mechanical Shock	M-2002	38	10
20. VF	Variable Frequency Vibration	M-2007	38	10
21. CA	Const. Acceleration	M-2001	38	10
22. SQ	Mechanical Sequence	—	38	10
23. LI	Lead Integrity	M-2004	15	15
24. BS	Bond Strength	M-2011	15	15
25. DS	Die Shear Strength	M-2019	5	50
26. XR	X-Ray	M-2012	5	50
27. TQ	End Torque	M-2024	15	15
28. ES	Electrostatic Discharge (ESD)	X-19435	—	—
29. LU	Latch-up	L-757185	9	—

Qualification Review Board

All new IC devices fabricated in any manufacturing process technology, as well as any changes in a process technology, must be qualified under the administration of the Qualification Review Board (QRB).

The QRB is a committee of Lucent Technologies Bell Labs Innovations and Lucent Technologies representatives who deal with the procedures and issues related to qualification or requalification of silicon integrated circuit devices or processing technologies. Such a qualification must take place before the device or processing technology can be shipped.

A QRB is formed each time a new device type or process change is to be qualified. Each QRB consists of a core group, which defines the qualification needs, plus several additional members who act as auditors of the qualification and provide further expertise. The QRB may ask any member of Lucent Technologies to participate in a qualification review, should the need arise.

Every QRB includes experts in reliability engineering, design, or processing. It may also include specialists in packing, quality assurance, product engineering, processing, and electrostatic discharge phenomenon. All members must be satisfied with the completion of a qualification plan before they sign the completed plan. Control of product shipment prior to qualification, usually in the form of models, is the responsibility of the quality assurance representative of the QRB.

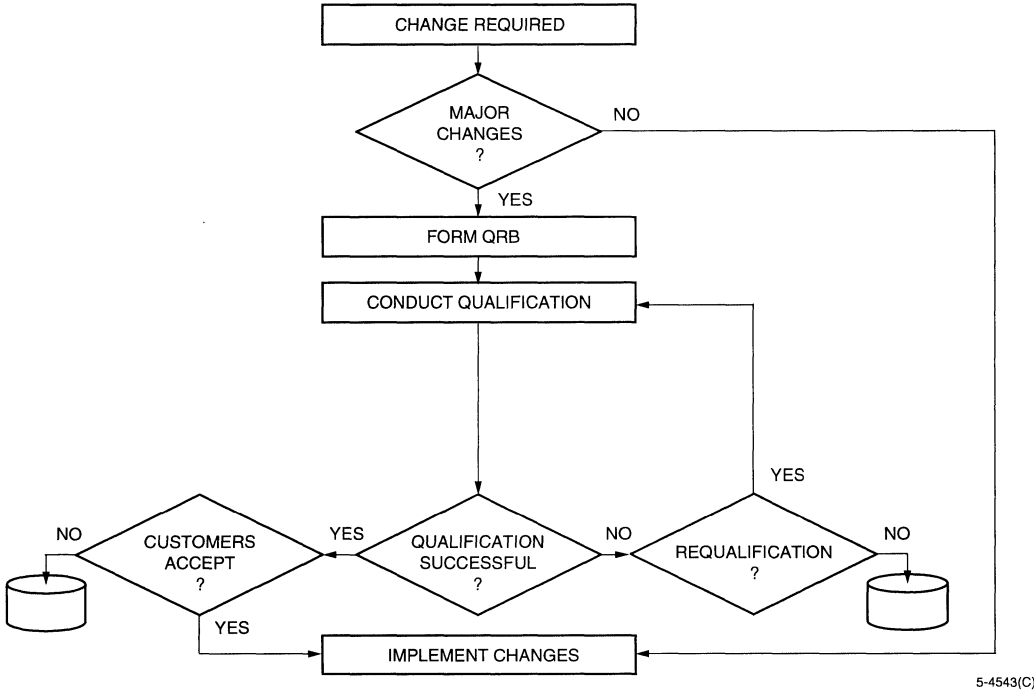


Figure 2. Requalification Process Flow

FPGA Product Qualification Plan

Lucent Technologies' product qualification plan for *ORCA* and 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.

Lucent Technologies' FPGAs are qualified by a three-phase process: Lucent Technologies' 0.6 µm/0.55 µm/ 0.5 µm CMOS processes, the design of the device itself, and the performance of a given die in a given package.

Since all Lucent Technologies FPGAs are manufactured using the Lucent Technologies 0.6 µm or 0.55 µm or 0.5 µm CMOS processes, they are carefully monitored and tested. Regarding device design, since each of the devices in each series of FPGAs (ATT3000, ATT1Cxx, ATT2Cxx, ATT2Txx, OR2CxxA, or OR2TxxA) is a matrix of repetitive elements, the 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 CMOS designs in the same technology.

For example, if a prior qualification of a 0.6 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.

However, if the die to be qualified was the largest 0.6 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.

0.5 µm, 0.55 µm, and 0.6 µm CMOS Process Qualifications

Introduction

This section presents quality and reliability information for Lucent Technologies' 0.5 µm , 0.55 µm, and 0.6 µm advanced CMOS processes. These processes employ N- and P-channel LDD MOS transistors. The 0.5 µm process also uses three levels of metal, and the 0.55 µm and 0.6 µm processes use two levels of metal.

These technologies have been rigorously tested for reliability and manufacturability. Prior to qualification and manufacture of devices using these process technologies, 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 technologies. New improvements are introduced into the technologies, with new issues put into effect at a rate of about one per year.

As part of the processes' qualification, the following tests are performed.

High-Temperature Operating Bias (HTOB)

The main vehicle for studying HTOB is the standard evaluation circuit (SEC). After testing, the failed parts were examined with physical FMA techniques. On the SEC, electrical failure modes are easily correlated with failure locations. The failing parts were taken through electrical and physical FMA. Gate-level defects are caused by silicon particles at the gate level. Several programs are now in place to attack gate-level defects.

Temperature Humidity Bias (THB)

Temperature humidity bias testing was performed on the SEC per Lucent Technologies' requirements of static bias at 85 °C and 85% relative humidity. THB is known to induce mechanical stress effects due to expansion of the plastic due to water absorption.

Temperature Cycling (TC)

Temperature cycling was performed per MIL-STD-883C, Method 1010, Condition C.

Thermal Shock (TS)

Thermal shock was performed per MIL-STD-883, Method-1011, which requires –65 °C to +125 °C, liquid-to-liquid, for 100 cycles.

Steam Bomb (SB)

Steam bomb testing was run on the SEC test chips.

Bond Strength (BS) and Die Shear (DS)

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. Die shear strength was also measured on these devices.

Summary

The latest issues of the 0.5 µm, 0.55 µm, and 0.6 µm CMOS technologies meet or exceed requirements for qualification in all critical areas.

Infant mortality and long-term failure rates have decreased 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.

1.2 µm EEPROM CMOS Process Qualification

Introduction

This section presents quality and reliability information for the 1.2 µm EEPROM CMOS process used to fabricate the ATT1700A family of serial PROMs. As part of process qualification, the following tests are performed.

High-Temperature Operating Bias (HTOB)

The main vehicle for studying HTOB is the standard evaluation circuit (SEC). After testing, the failed parts were examined with physical FMA techniques. On the SEC, electrical failure modes are easily correlated with failure locations. The failing parts were taken through electrical and physical FMA. Gate-level defects are caused by silicon particles at the gate level. Several programs are now in place to attack gate-level defects.

Temperature Humidity Bias (THB)

Temperature humidity bias testing was performed on the SEC per Lucent Technologies' requirements of static bias at 85 °C and 85% relative humidity. THB is known to induce mechanical stress effects due to expansion of the plastic due to water absorption.

Temperature Cycling (TC)

Temperature cycling was performed per MIL-STD-883C, Method 1010, Condition C.

Thermal Shock (TS)

Thermal shock was performed per MIL-STD-883, Method-1011, which requires –65 °C to +125 °C, liquid to liquid, for 100 cycles.

Steam Bomb (SB)

Steam bomb testing was run on the SEC test chips.

Bond Strength (BS) and Die Shear (DS)

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. Die shear strength was also measured on these devices.

Summary

This latest issue of the 1.2 µm CMOS technology meets or exceeds requirements for qualification in all critical areas.

Device Qualification Testing

Overview

Lucent Technologies' product quality program for the ATT3000 and *ORCA* 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 Operating Bias (HTOB) 150 °C, 6.2 V
- Component Lead Assembly Simulation Sequence or CLASS (CL)
- 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 (MR)
- Salt Atmosphere or Corrosion (SA)

Mechanical Tests

- Flammability and O2 Index (FL)
- Solvent Resistance (SR)
- Physical Dimensions (PD)
- Solderability (SD)
- Bond Strength (BS)
- Die Shear Strength (DS)
- X-Ray (XR)

Electrical Tests

- Electrostatic Discharge (ES[D]) for Each Pin
- Latch-up (LU)

The specific tests used to qualify devices such as the Lucent Technologies' 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.

Test results are presented in the following section.

Device Qualification Testing (continued)

Environmental Tests

High-Temperature Operating Bias (HTOB)

HTOB testing, also called dynamic life testing, is performed at 150 °C and 6.2 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 or CLASS (CL)

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 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 units/85 units for 96 hours.

Autoclave (SB)

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. 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 (TC)

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 (TS)

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 –65 °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 100 cycles, thermal shock end-point testing is performed.

Moisture Resistance (MR)

This test evaluates, on an accelerated schedule, how well component parts and constituent materials resist deterioration from the high temperature and humidity that is 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-883C (method 1004).

Salt Atmosphere or Corrosion (SA)

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-883C (method 1009).

Low-Temperature Aging (RT)

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.

Device Qualification Testing (continued)

Mechanical Tests

Flammability and O2 Index (FL)

This test follows *UL* 94 and ASTM 2863-77 methods.

Solvent Resistance (SR)

This test is performed per MIL-STD-883C (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.

Physical Dimensions (PD)

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-883C (method 2016).

Solderability Test (SD)

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-883C (method 2003).

Bond Strength (BS)

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-883C (method 2011).

Die Shear Strength (DS)

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-883C (method 2019).

X-Ray (XR)

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-883C (method 2012).

Electrical Tests

ESD Human-Body Model as per Lucent Technologies 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 Lucent Technologies 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 Lucent Technologies 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 I_{CC} current exceeds the manufacturer's maximum I_{CC} current and remains at that level after the stress is removed.

Device Qualification Testing (continued)

Table 2. Latch-Up Qualification Summary

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

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

Details of Electrostatic Discharge (ESD) Tests

ESD Test Methods and Requirements

Source: Lucent Technologies 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 Lucent Technologies drawings:

Drawing Number	Figure 3-n	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 Lucent Technologies 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 process, design, packaging, or specification changes. 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.

The power supply pins include VDD, VCC, VSS, VBB, GND, +VS, –VS, and VREF. 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	500 V	2000 V
200 V	1000 V	4000 V

Voltage levels for CDM ESD testing include:

100 V	500 V	2000 V
200 V	1000 V	

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

Details of ESD Tests (continued)

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 3 and Figure 4, respectively.

Specific Test Procedure: HBM

Insert the first device under test (DUT) into the socket as shown in Figure 3. 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.

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.

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 ESD Tests (continued)

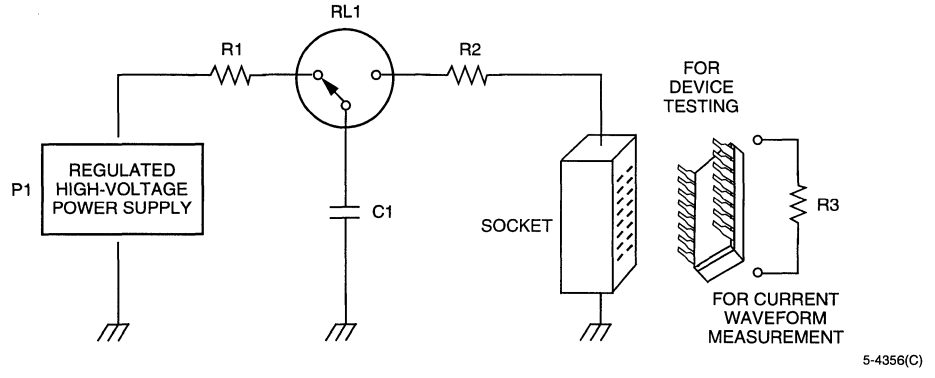


Figure 3. Circuit Schematic of Human-Body ESD Simulator

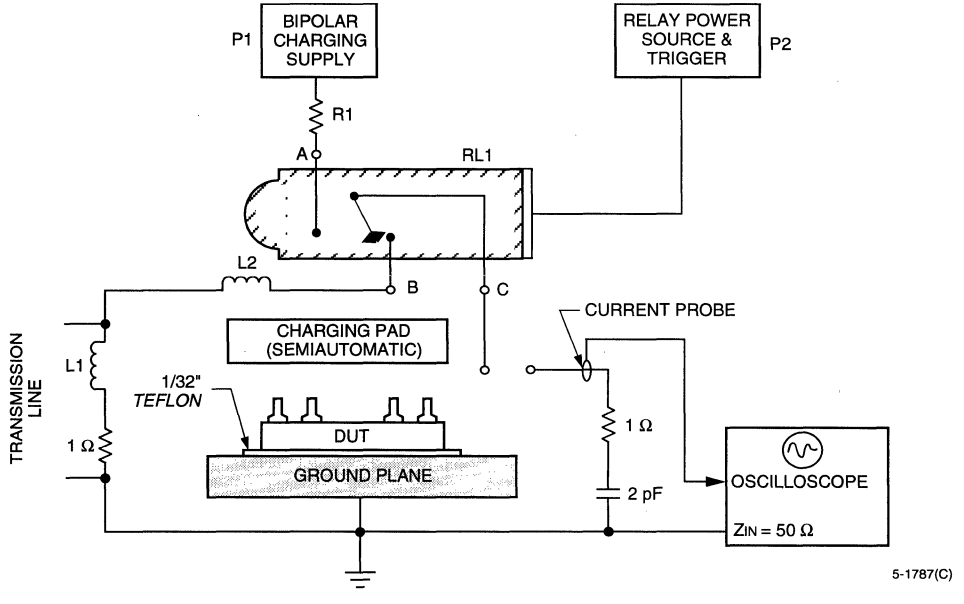


Figure 4. Circuit Schematic for CDM Simulator and Waveform Measurement

Details of Latch-Up Tests

Integrated Circuit Latch-Up Test Procedure per Lucent Technologies Method A88AL1006

Purpose

This section describes the testing method Lucent Technologies 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 $I_{CC} > I_{CC}$ (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-V_{CC}/V_{SS} 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 $I_{CC} > I_{CC}$ 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 5 illustrates equipment hookup.

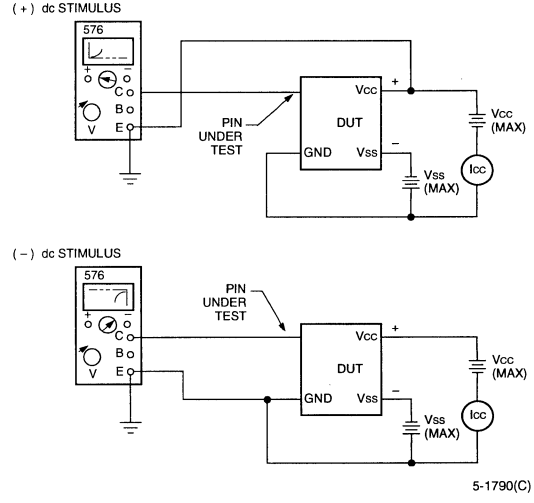


Figure 5. Test 1: Equipment Hookup: + or – dc Stimuli

Details of Latch-Up Tests (continued)

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.

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 ($\Delta V/\Delta t$) 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 voltage rate 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 5 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 $\Delta V/\Delta t$ 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 Icc (active) mode.
4. Twelve times with the device in 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.

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 7 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.

Details of Latch-Up Tests (continued)

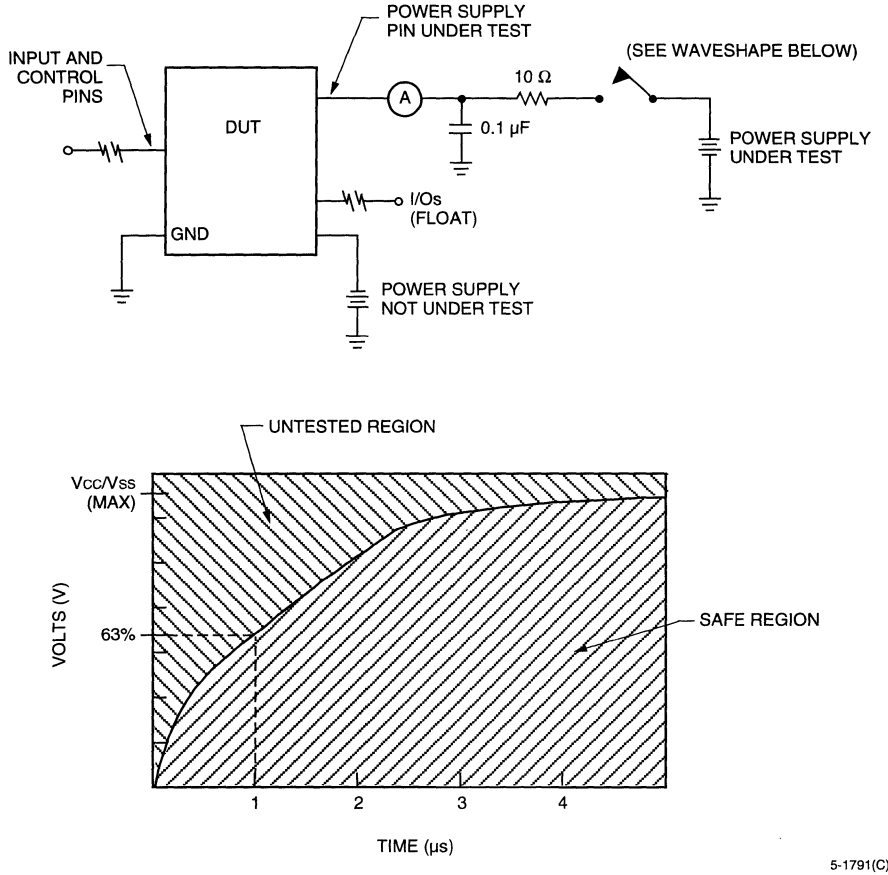


Figure 6. Test 2: Power Supply (ΔV/Δt) Limit

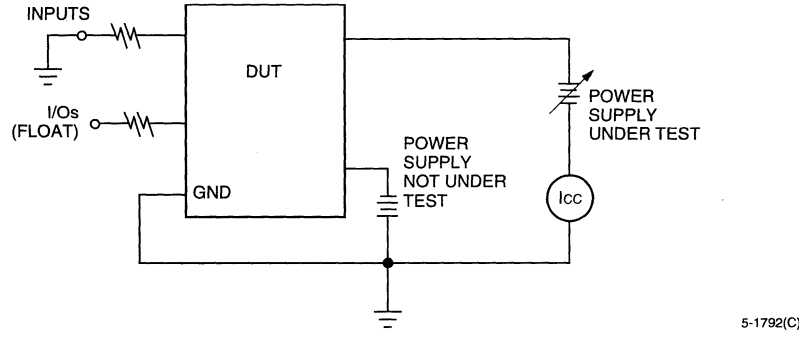


Figure 7. Test 3: Power Supply Overvoltage Test Equipment Hookup

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 3. Limits Employed in Testing

VCC = manufacturer's maximum operating VCC voltage; VSS = manufacturer's maximum operating VSS voltage.

Susceptibility Level	Test 1		Test 2	Test 3
	Current	Voltage	Slew Rate	PS Overvoltage
0	<50 mA	VCC + 5 V VSS + 5 V	>0.63 VDD in 10 µs	<1.50 VDD
1	>50 mA <150 mA	VCC + 5 V VSS + 5 V	>0.63 VDD in 5 µs	>1.50 VDD <1.75 VDD
2	>150 mA <250 mA	VCC + 5 V VSS + 5 V	>0.63 VDD in 1 µs	>1.75 VDD <2.00 VDD
3	>250 mA <500 mA	VCC + 5 V VSS + 5 V	>0.63 VDD in 500 µs	>2.00 VDD <2.25 VDD
4	>500 mA	VCC + 5 V VSS + 5 V	>0.63 VDD in 100 µs	>2.25 VDD

Package Qualification

Introduction

This section will document the qualification test procedures and results for the devices in the Lucent Technologies FPGA product family. The QRB determined the qualification requirements of each die in a given package. Test results are presented in Table 4 for the ATT3000 Series (0.6 µm), in Table 5 for the ATT3000 Series (0.55 µm), in Table 6 for the *ORCA* 1C Series, in Table 7 for the *ORCA* 2C series of FPGAs, and in Table 8 for the ATT1700A series of EEPROMs. **For more detailed qualification information, please see the qualification test procedures and results at the end of this section.**

ESD and latch-up tests have been done on all of the FPGA families, with varying results for each device in each package. For the ATT3000 Series, the ESD results for the human-body model (ESD-HBM) varied from >2000 V to >3500 V, and the ESD results for the charged-device model (ESD-CDM) varied from >2500 V to >3000 V. Most devices passed LU Class IV, with a few passing only LU Class III. For the 1C Series of FPGAs, both ESD-HBM and ESD-CDM passed for >2000 V, which was the highest value tested. These devices also passed LU Class IV. For the 2C Series of devices, ESD-HBM passed for >2000 V and ESD-CDM passed for >1000 V, which, again, were the highest values tested. These devices also passed LU Class IV.

Qualification Status

Table 4. Qualification Status of ATT3000 Series FPGAs (0.6 µm)

Device	Qual No.	References	Tests Performed	Status
3020-68PLCC	—	Q92055, Q89166	—	Fully Qualified
3020-84PLCC	—	Q92055, Q89166	—	Fully Qualified
3020-100QFP	—	Q90001 or Q90111	—	Fully Qualified
3030-44PLCC	—	Q92055, Q89166	—	Fully Qualified
3030-68PLCC	—	Q92055, Q89166	—	Fully Qualified
3030-84PLCC	—	Q92055, Q89166	—	Fully Qualified
3030-100QFP	—	Q90001 or Q90111	—	Fully Qualified
3030-100TQFP	—	Q92055	—	Fully Qualified
3042-84PLCC	—	Q92055, Q89166	—	Fully Qualified
3042-100TQFP	QRB92.51	Q92055	ESD, LU	Fully Qualified
3042-100QFP	QRB93.2	Q90001 or Q90111	ESD, LU	Fully Qualified
3042-132PPGA	—	Q90137	—	Fully Qualified
3064-84PLCC	—	Q92055, Q89027	—	Fully Qualified
3064-132PPGA	—	Q90137	—	Fully Qualified
3064-160QFP	—	Q89129.1, Q90140	—	Fully Qualified
3090-84PLCC	Q92055	Q90148, Q89027	LT, ESD, LU	Fully Qualified
3090-160QFP	—	Q89129.1, Q90140	—	Fully Qualified
3090-175PPGA	Q92054	Q90137	LT, ESD, LU	Fully Qualified
3090-208SQFP	—	MR120, Q92055	—	Fully Qualified

Note: Status reflected is as of 2Q96.

Package Qualification (continued)

Table 5. Qualification Status of ATT3000 Series FPGAs (0.55 μm)

Device	Qual No.	References	Tests Performed	Status
3020-68PLCC	—	Q95273, Q89166	—	Fully Qualified
3020-84PLCC	—	Q95273, Q89166	—	Fully Qualified
3020-100QFP	—	Q95273, Q90111	—	Fully Qualified
3030-44PLCC	—	Q95273, Q89166	—	Fully Qualified
3030-68PLCC	—	Q95273, Q89166	—	Fully Qualified
3030-84PLCC	—	Q95273, Q89166	—	Fully Qualified
3030-100QFP	—	Q95273, Q90111	—	Fully Qualified
3030-100TQFP	—	Q95273, Q92055	—	Fully Qualified
3042-84PLCC	Q95273	MR109, Q89166	LT, ESD, LU	Fully Qualified
3042-100QFP	—	Q95273, Q90111	—	Fully Qualified
3042-100TQFP	—	Q95273, Q92055	—	Fully Qualified
3042-132PPGA	—	Q95273, Q90137	—	Fully Qualified
3064-84PLCC	—	Q95273, Q89027	—	Fully Qualified
3064-132PPGA	—	Q95273, Q90137	—	Fully Qualified
3064-160QFP	—	Q95273, Q90140	—	Fully Qualified
3090-84PLCC	Q96098	Q95273, Q89027	ESD, LU	Fully Qualified
3090-160QFP	—	Q95273, Q90140	—	Fully Qualified
3090-175PPGA	—	Q95273, Q90137	—	Fully Qualified
3090-208SQFP	Q96099	Q95273, MR120	ESD, LU	Fully Qualified

Note: Status reflected is as of 2Q96.

Package Qualification (continued)

Table 6. Qualification Status of *ORCA* 1C Series FPGAs

Device	Qual No.	References	Tests Performed	Status
ATT1C03-84PLCC	—	Q92167, Q92168	—	Fully Qualified
ATT1C03-100TQFP	—	Q92167, QRB92.51	—	Fully Qualified
ATT1C03-132BQFP	—	Q92167, Q92168	—	Fully Qualified
ATT1C03-208SQFP	—	Q92167, Q92168	—	Fully Qualified
ATT1C03-225CPGA	—	Q92167, Q92161	—	Fully Qualified
ATT1C03-225PPGA	—	Q92054, Q92167	—	Fully Qualified
ATT1C05-84PLCC	—	Q92167, Q92168	—	Fully Qualified
ATT1C05-100TQFP	—	Q92167, QRB92.51	—	Fully Qualified
ATT1C05-132BQFP	—	Q92167, Q92168	—	Fully Qualified
ATT1C05-208SQFP	—	Q92167, Q92168	—	Fully Qualified
ATT1C05-225PPGA	—	Q92054, Q92167	—	Fully Qualified
ATT1C05-225CPGA	—	Q92167, Q92161	—	Fully Qualified
ATT1C05-240SQFP	Q92167	—	All major tests	Fully Qualified
ATT1C07-208SQFP	—	Q92167, Q92168	—	Fully Qualified
ATT1C07-240SQFP	—	Q92167, Q92168	—	Fully Qualified
ATT1C07-280CPGA	—	Q92167, Q92161	—	Fully Qualified
ATT1C07-304SQFP	Q94044	Q92167, Q92168	ESD	Fully Qualified
ATT1C09-208SQFP	—	Q92167, MR120	—	Fully Qualified
ATT1C09-240SQFP	—	Q92167, MR141	—	Fully Qualified
ATT1C09-304SQFP	—	Q94044	—	Fully Qualified

Note: Status reflected is as of 2Q96.

Package Qualification (continued)

Table 7. Qualification Status of *ORCA* 2C Series FPGAs

Device	Qual No.	References	Tests Performed	Status
ATT2C04-84PLCC	Q94291	Q93234, T92222, MR109	None	Fully Qualified
ATT2C04-144TQFP	—	Q93234, Q93181	None	Fully Qualified
ATT2C04-160QFP	Q95319	Q93234, MR119	ESD, LU	Fully Qualified
ATT2C04-208SQFP	—	Q93234, Q94124	None	Fully Qualified
ATT2C06-84PLCC	—	Q93234, T92222	None	Fully Qualified
ATT2C06-144TQFP	—	Q93234, Q93181	None	Fully Qualified
ATT2C06-160QFP	—	Q93234, MR119, Q95319	None	Fully Qualified
ATT2C06-208SQFP	—	Q93234, Q94124	None	Fully Qualified
ATT2C06-240SQFP	—	Q93234	None	Fully Qualified
ATT2C08-160QFP	—	Q93234, MR119, Q95319	None	Fully Qualified
ATT2C08-208SQFP	—	Q93234, Q94124	None	Fully Qualified
ATT2C08-240SQFP	—	Q93234	None	Fully Qualified
ATT2C08-256PBGA	—	Q96068, Q95013	None	Fully Qualified
ATT2C08-304SQFP	—	Q93234, Q94046	None	Fully Qualified
ATT2C10-160QFP	—	Q93234, MR119, Q95319	None	Fully Qualified
ATT2C10-208SQFP	—	Q93234, Q94124	None	Fully Qualified
ATT2C10-240SQFP	—	Q93234	None	Fully Qualified
ATT2C10-256PBGA	Q96068	Q95013	TC, SB, TS, PD, ESD, LU	Fully Qualified
ATT2C10-304SQFP	Q95041	Q94046	SB, X-Ray	Fully Qualified
ATT2C12-208SQFP	—	Q93234, Q94124	None	Fully Qualified
ATT2C12-240SQFP	—	Q93234	None	Fully Qualified
ATT2C12-256PBGA	—	Q96068, Q95013	None	Fully Qualified
ATT2C12-304SQFP	—	Q95041	None	Fully Qualified
ATT2C12-364CPGA	—	Q93231	None	Fully Qualified
ATT2C15-208SQFP	Q94290	Q93234, Q94124	ESD, LU	Fully Qualified
ATT2C15-240SQFP	Q93234	Q94046	LT, CL, BH, SB, TC, TS, ESD, LU	Fully Qualified
ATT2C15-304SQFP	Q94046	—	CL, BH, SB, TC, ESD, LU	Fully Qualified
ATT2C15-364CPGA	Q93231	Q94046, Q94290	LT, TC, TS, SQ, ESD, LU	Fully Qualified
ATT2C26-208SQFP-PQ2	—	Q94227	None	Fully Qualified
ATT2C26-240SQFP-PQ2	—	Q94227, Q95007	None	Fully Qualified
ATT2C26-304SQFP-PQ2	Q94227	—	LT, CL, BH, SB, TC, TS, ESD, LU, PD	Fully Qualified
ATT2C40-208SQFP-PQ2	QRB95.17	Q95007	PD	Fully Qualified
ATT2C40-240SQFP-PQ2	Q95007	Q94227	CL, BH, TC, PD, ESD, LU	Fully Qualified
ATT2C40-304SQFP-PQ2	—	Q95007, Q94227	None	Fully Qualified

Note: Status reflected is as of 2Q96.

Package Qualification (continued)

Table 8. Qualification Status of ATT1700A Series EEPROMS

Device	Qual No.	References	Tests Performed	Status
ATT1700A-8DIP	Q94362	—	LT, BK, BH, HAST, SB, TC, TS, MR, SA, RT, SR, PD, SD, LI, HBM, CDM, LU	Fully Qualified
ATT1700A-8SOIC	Q94363	Q94362	BH, HAST, SB, TC0, TC, TS, MR, SA, RT, SR, PD, SD, LI, HBM, CDM, LU	Fully Qualified
ATT1700A-20PLCC	Q94364	Q94362	BH, HAST, SB, TC, TS, MR, SA, RT, SR, PD, SD, LI, HBM, CDM, LU	Fully Qualified

Note: Status reflected is as of 2Q96.

Vendor Quality Monitoring

Overview

Quality of all purchased raw material, parts, and components used in IC manufacture is monitored by the Purchased Material Inspection Section. Partnerships are formed with suppliers of critical materials.

Inspection

A flow chart of the incoming inspection process is shown in Figure 8. Purchase orders for materials prescribe complete and precise specifications. Inspection and testing requirements are included and, where required, requests are made for evidence of chemical, physical, and electrical tests in the form of certificates. Quality Control produces monthly summaries of the results of all inspections, including established and new suppliers. The individual inspections involve various personnel:

- Silicon and Wafers—performed by resident Quality Assurance (QA) inspectors.
- Piece Parts—performed by the inspectors of the Purchased Material Inspection Section.
- Chemicals—incoming chemicals are sampled and inspected by the Analytical Lab.
- Gases—a certificate of inspection and analysis is required with each shipment of compressed gases. Each certificate is reviewed by the Purchased Materials Inspector for completeness and compliance with specifications.

Vendor Qualification

New vendors are qualified by a team that includes representatives from product engineering, quality control, and purchasing. The qualification process may include evaluation of product samples and an audit of the supplier's facility to assess quality programs, practices, and manufacturing capabilities. Approved vendors and products are listed on the Approved Vendors List maintained by Quality Control.

Vendor Commodity Teams

Vendor partnerships are created with the intent of establishing and maintaining close working relationships with key suppliers. This formal program is administered by Vendor Commodity Teams composed of members from Purchasing, Materials Management, Quality, Bell Laboratories, and Product Engineering.

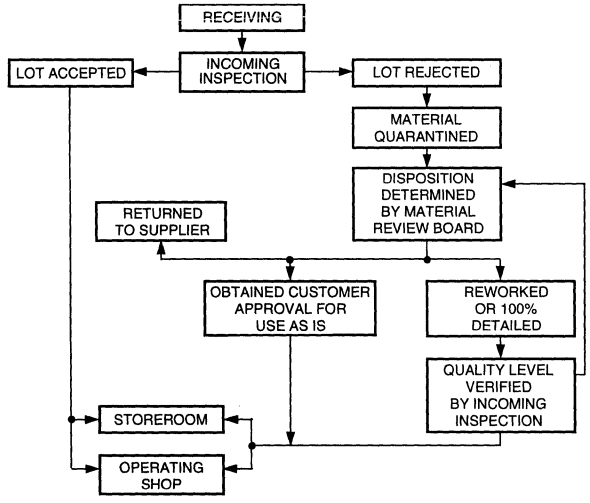


Figure 8. Flow Diagram for Purchased Material

Manufacturing Control and Improvement

Wafer level and quality improvement is an aggressive, proactive program aimed at improving the quality and reliability of the MOS manufacturing product lines.

An example of this process is the use of a line monitor to enhance the control of CMOS fabrication lines. While all IC fabrication lines use zone monitors and process monitors, the CMOS fabrication lines also employ a line monitor that is used to achieve tighter process control for newer technologies.

This program, called the Yield Enhancement System (YES), involves 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. Lucent Technologies' 1.2 µm and 0.9 µm CMOS technologies benefit from the YES program.

Key tools designed for the YES program include a line monitor, zone monitors, a data management system, and a yield model. A key part of the data management system is a rigorous statistical process control program. 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 Monitor (PM).

Lucent Technologies' 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 patterns for measuring defect densities to arrive at a correlation between direct measurement of defect density and SEC yield. It also contains structures 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 that 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 provided by this model also help to prioritize defect reduction projects.

Statistical Process Control (SPC)

Statistical Process Control is a key component of Lucent Technologies' quality program.

As the flow chart in Figure 9 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 9 and Table 10. 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.

Lucent Technologies' goal is to become a 6 sigma manufacturer. 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 Lucent Technologies 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.

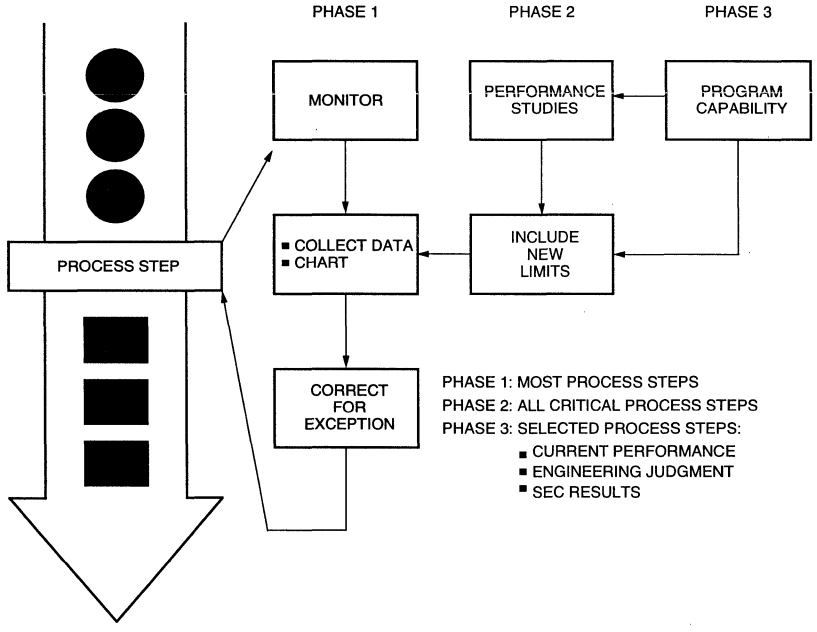


Figure 9. The SPC Process Flow

Statistical Process Control (continued)

Table 9. Examples of Statistical Process Control Monitors in the MOS Fabrication Area

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	Line Size	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	Twice Per Shift	One (1) Control
RTA Ti-Aluminum	Sheet Resistance	Each Lot	One (1) Control
Sputter Aluminum 1	Aluminum Thickness	Twice Per Shift	One (1) Control
Aluminum Photoresist	Line Size	Each Lot	Three (3) Wafers
Sputter Aluminum 2	Aluminum Thickness	Twice Per Shift	One (1) Control
Metal 2 Photoresist	Line Size	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

Statistical Process Control (continued)

Table 10. Examples of Statistical Process Control Monitors in the Assembly Area

Operation	Inspection	Frequency	Sample Size
Wafer Saw	Visual Inspection	Twice Per Machine Per Shift	20 Units
	Machine Inspection	Once Per Machine Per Shift	Each Machine
Die Attach	Visual Inspection	Every Magazine	Three Strips
	Epoxy Resistivity	Once Per Month	Two Glass Slides
	Die Shear	Once Per Machine Per Shift	Three Units
	Epoxy Void	Once Per Machine Per Shift	10 Units
	Machine Inspection	Once Per Machine Per Shift	Each Machine
	Oven Cure	Once Per Machine Per Shift	Each Machine
Wire Bond	Visual Inspection	Every 10 Strips	One Strip
	Bond Pull Strength	Once Per Machine Per Shift	Five Wires Per Unit
	Ball Shear Strength	Twice Per Machine Per Shift	Five Wires Per Unit
	Machine Inspection	Once Per Machine Per Shift	Each Machine
Mold	Visual Inspection	Every 20 Mold Runs	One Mold Run
	X-Ray	Twice Per Machine Per Shift	12 Units
	Machine Inspection	Once Per Machine Per Shift	Each Machine
Code Mark	Visual Inspection	Every 120 Strips	Three Strips
	Permanency	One Out of Five Lots	Four Units
	Machine Inspection	Once Per Day	Each Machine
Trim and Form	Visual Inspection	Every 20 Tubes	20 Units
	Lead Spread	Twice Per Machine Per Shift	Five Units
	Coplanarity	Twice Per Machine Per Shift	Two Units
Solder	Visual Inspection	One Lot Per Package Type Per Shipment	22 Units
	Ionic Contamination	Each Shipment	Two Strips
	Solderability (Incoming from Plater)	Twice Per Week	12 Units
	Thickness (COFC)	Each Shipment	One in Three Lots
	Outgoing Solderability	Once Per Week	Four Units

Document Control

The overall procedures and standards by which Lucent Technologies exercises its control of the manufacture, quality, and reliability of ICs are depicted in a series of A-drawings. A-drawings are interpreted and implemented by engineering organizations using shop instructions (SIs), inspection layouts (ILs), training documents (TDs), and test equipment requirements (TERs). Each of these secondary documents has a provision for adding information via supplementary information forms (SIFs). SIFs must be referenced to the interpretive drawing (for example, SIF-IL5349-009).

The document control group's responsibility is to maintain the current issues of each of these documents and to control the procedures for document changes. The current issue of each document is kept in the Manufacture Information Distribution System (MIDS) database. The document control group places paper copies of current documents applicable to the efforts of a given work area in manufacturing information books, and maintains these books. For example, all drawings applicable to reliability monitoring are contained in a book (MI-098-RI) kept in the reliability laboratory.

Changes in any document are made by engineering staff responsible for the specific process. Revisions must be approved by the appropriate members of the engineering staff and by Quality Control before being implemented.

QC confirms that appropriate qualification of changes has been conducted and, when required, notifies the customer and obtains approval.

Quality Conformance Inspection

This section summarizes the completed product audit for MOS products and describes final inspection procedures used by Quality Assurance (QA). When manufacturing is completed and product is presented for shipment, QA selects samples and confirms that they meet specifications. Sampling procedures apply to electrical and mechanical inspection.

Electrical sampling is performed on a device type basis, using a two-tier sampling plan as shown in Figure 10. Normal inspection is performed lot by lot at 0.1% Acceptable Quality Level (AQL). For example, in 1990, QA measured fewer than 25 parts per million (ppm) defective during electrical audit. Devices with exceptional quality (no defects found in the last 10 lots sampled) receive skip lot inspection, performed on one of every four lots 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 device type.

Sampling procedures specify that sticks (or tubes) of devices are randomly selected until the required sample size is reached. To avoid mixing of products in final inspection, only samples from one lot are inspected at a time, and are then returned to the parent lot before choosing samples from the next lot.

Devices may move to skip-lot inspection after 10 consecutive lots pass the procedure lot by lot. Any rejected lot causes the device or package type to return (or remain) on the lot-by-lot sampling approach.

Specific procedures exist for tighter sampling and inspection procedures when QA 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 ship for corrective action.

QA reports the results of the electrical and mechanical/visual inspections as measures of outgoing quality expressed in ppm. These reports are generated from a computerized database which gathers information entered by inspectors on each lot as inspection is performed. MOS quality performance has shown a 35% improvement rate per year, as illustrated in Figure 11.

Quality Conformance Inspection (continued)

Specific procedures exist for tighter sampling and inspection procedures when QA 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 measures of outgoing quality expressed in ppm. These reports are generated from a computerized database which gathers information entered by inspectors on each lot as inspection is performed. MOS quality performance has shown a 35% improvement rate per year, as illustrated in Figure 11.

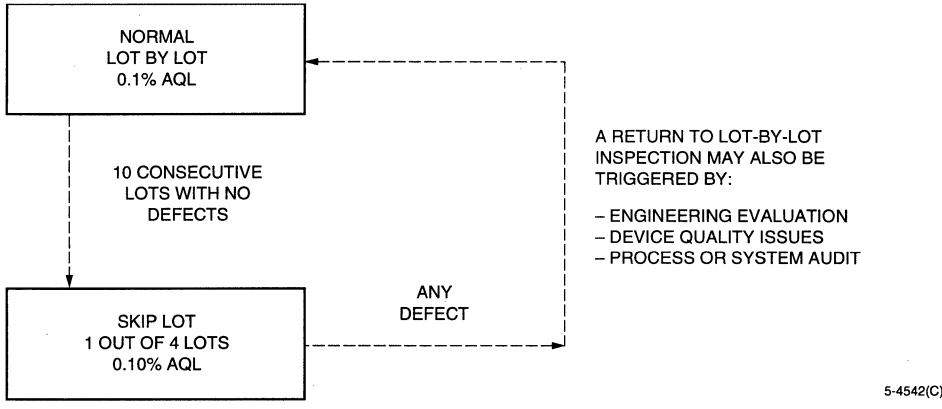


Figure 10. Electrical Test Sampling Level and Flow (Device Code Basis)

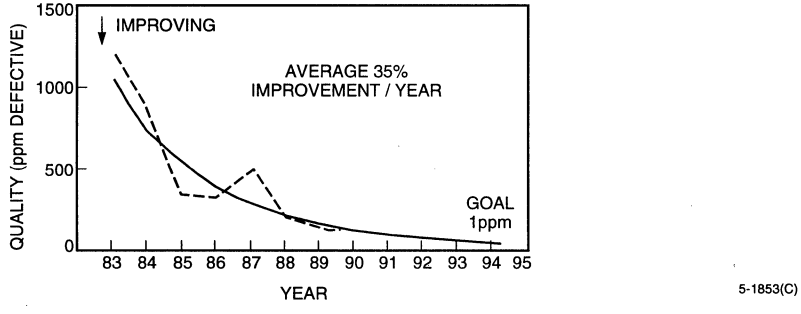


Figure 11. Product Quality Improvement Rate

Reliability Monitoring Program

Introduction

In this section, we present the Lucent Technologies Reliability Model and the Lucent Technologies Reliability Monitoring Program. The section begins with a review of reliability concepts, followed by the device failure model, concepts of accelerated testing, and the Lucent Technologies Reliability Monitoring Program as administered by the Reliability Review Board.

Reliability is the ability of an electronic system to operate satisfactorily over a period of time. Since electronic systems consist of electronic components (devices), system reliability depends on the reliability of each component in the application environment.

A generally accepted definition of reliability is the probability that an item will perform a required function under stated conditions for a stated period of time.

The required function includes both a definition of satisfactory and unsatisfactory operation (failure). The stated conditions are the total physical environment, including mechanical, thermal, and electrical conditions. The stated period of time is the time during which satisfactory operation is desired.

Device Failure Model

The Lucent Technologies Reliability Model is presented in terms of the failure rate, λ (t). Historically, failure rates have been modeled in the bathtub curve shown in Figure 12.

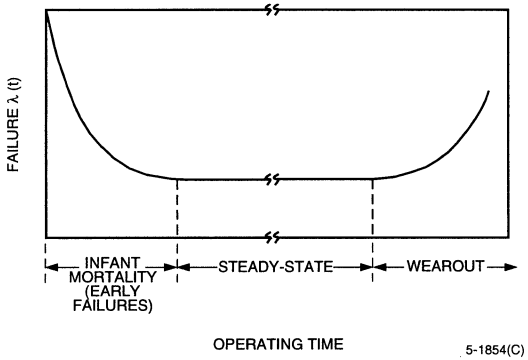


Figure 12. Reliability Bathtub Curve

This curve has three regions with distinct characteristics. The regions are associated with infant mortality, steady-state operation, and wear-out.

In most cases, the bathtub curve model is only partly appropriate for electronic devices. Although such devices exhibit infant mortality and steady-state periods, generally those supplied by reputable suppliers exhibit no wear-out during intended device life.

The failure rate in the infant mortality region of the bathtub curve is modeled by a Weibull distribution. The Weibull failure rate can be expressed as:

$$\lambda\,(t) = \lambda(1)t^{-a}$$

One feature of this distribution is that the failure rate is a straight line when plotted on log-log paper. In such a plot, the slope is –a and the intercept at t = 1 hour is l1. Beyond the infant mortality period, the failure rate is assumed to be determined by the exponential distribution.

Two distinct sources of information exist on device reliability: accelerated life tests and performance actually monitored in the factory or field. These sources provide two quite different measures of reliability. Accelerated life tests provide information about expected reliability in the very long term (decades). Factory or field data gives a real measure of actual device reliability in the short term (less than a few years).

Measures of Reliability

Reliability is usually measured in number of defects or percentage of defects per unit of time. This definition relates more to the unit's failure rate than to its reliability function. A common unit for describing failure rate behavior is the FIT: failures in 109 device hours of operation.

Accelerated Stress Testing

Because devices are so reliable, failure-accelerating stresses are needed to observe failure distributions within a reasonable time period. The accelerating effect of the stress must be well understood to interpret the results of accelerated stress testing.

The relationship between stress and time to failure for a given product is determined by the activation energies of the failure mechanisms which are dominant in that product. The activation energies are determined from extensive accelerated stress testing, usually done at the time the failure mechanism is first discovered.

Reliability Monitoring Program

(continued)

When evaluating the impact of specific problems on device life expectancy, it is important to treat the different failure mechanisms that may occur within a sample independently; they may be accelerated differently, and extrapolations from combined data can be very misleading. (This emphasizes the need for failure analysis to identify the failure mechanism.)

Effects of Operating Voltage on Failure Rates

The dielectric breakdown of oxide film can be accelerated by applied field, particularly for MOS devices. Extensive investigations have established a voltage-dependent acceleration factor that could be applied to MOS devices in which a voltage stress in excess of nominal voltages is applied. This research has led to the relationship

$$A_V = \mathrm{EXP}\left[\frac{C}{tox}(V_1 - V_2)\right]$$

where:

C = voltage acceleration constant in A/V

tox = oxide thickness in Å

V_1 = stress voltage in V

V_2 = operating voltage in V

For Lucent Technologies MOS products, these values are:

Technology	C/tox
0.5 µm	2.3
0.55 µm	2.3
0.6 µm	2.0
1.25 µm	1.4
1.75 µm	1.2
>2 µm	0.6

Time-Temperature Relationship (Arrhenius Equation)

Many of the chemical and physical processes leading to failure are accelerated by temperature in a way that can be readily modeled and reproduced. This makes temperature a very useful accelerating stress.

The equation describing the temperature acceleration factor, A_T, is found from the Arrhenius relationship. It is written as:

$$A_T = \mathrm{EXP}\frac{Ea}{ko}\left[\frac{1}{T_0} - \frac{1}{T_1}\right]$$

where:

Ea = activation energy (in eV)

ko = Boltzman constant

$$8.6 \times 10^{-5}\ \frac{eV}{°K}$$

T_1 = stress ambient temperature in °K, and

T_0 = operating ambient temperature in °K

Assumed Activation Energies

In many cases, device reliability can be approximated by using a composite activation energy. Studies of the infant mortality period indicate a very low activation energy for these failure mechanisms. Recent data suggests that a 0.55 eV activation energy is the most appropriate for establishing this time-temperature trade-off in screening for infant mortality. An activation energy of 0.7 eV is generally assumed as an average activation energy for times beyond the infant mortality period. Lucent Technologies often uses 0.7 eV and 55 °C operating temperature for steady-state reliability calculations. With these assumptions, temperature acceleration factors become:

A_T = 78—for 125 °C stress temperature

A_T = 260—for 150 °C stress temperature

Product Reliability Monitoring Plan

Lucent Technologies' product reliability monitoring plan is based on our fundamental pursuit of quality in every device within each product family we offer. Thus, we monitor our devices constantly, examining their performance over time as we search for and identify opportunities to further increase their reliability.

Reliability Monitoring Program

(continued)

Three-Tier Testing Program

A keystone of Lucent Technologies' reliability plan is its three-tier program of stress testing. Level 1 represents Lucent Technologies' 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 Lucent Technologies' normal production monitoring program in which a sample of devices is chosen from each testing group and stressed for a period approximating 10 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.

Level 0 (HTOB testing) represents a test point of 24 hours, and is intended as an infant mortality measurement. It should be noted that some product families (particularly memory) omit Level 0 testing for HTOB.

Reliability Testing Procedures

Specific Procedures for HTOB Testing

Sample devices, chosen according to the protocol in Table 11, 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 the testing universes shown in Table 12 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 then retested.

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 13. 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 database. 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.

Reliability Monitoring Program (continued)

Table 11. Sampling Plan: High-Temperature Operating Bias (HTOB)

Test Level	Test Hours	Sampling Frequency	Stress Conditions	Sample Size	LTPD (%)
0	48	Biweekly	125 °C	195	2.0
2	160	Biweekly	125 °C	195	2.0
1	1,000	Monthly	125 °C	100	3.0
0	24	Biweekly	150 °C	100	3.0
2	160	Biweekly	150 °C	100	4.0
1	1,000	Monthly	150 °C	58	4.0

Table 12. Sampling Plan: Temperature-Humidity Bias (THB)

Test Level	Test Hours	Sampling Frequency	Stress Conditions	Sample Size	LTPD (%)
2	240	Monthly	85 °C/85% RH	130	3.0
1	1,000	Monthly	85 °C/85% RH	76	3.0

Table 13. Sampling Plan: Temperature Cycle (TC)

Test Level	Test Cycles	Sampling Frequency	Stress Conditions	Sample Size	LTPD (%)
2	240	Monthly	85 °C/85% RH	105	5.0

6

Product Families and Sampling Coverage

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

- ASIC (including FPGAs)
- HPIC
- Microprocessors (including digital signal processors)
- Communication

Sample Selection and Sampling Universes

Level 2 samples are selected from the testing universes shown in Table 14 and Table 15. For HTOB, 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.

160-hour samples are extended 24-hour samples. Every other 160-hour sample is extended to 1,000 hours.

If a sample has more failures than permitted by the LTPD standards shown in Table 11, Table 12, and Table 13, a special RRB meeting is called to address the issue.

Reliability Monitoring Program

(continued)

Table 14. 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
C	1.75 CT	Standard Cell Digital
	1.75 C	Analog/Custom Digital
	1.75 C	Standard Cell (1P 1M)
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
	0.6 LC	Analog
F	0.9 CT	Standard Cell Digital
	0.6 CT	Standard Cell Digital
G	0.6 HD	High-Density CMOS
G.1	0.55 HD	High-Density CMOS
H	0.5 CT	Standard Cell Digital
I	0.35 CT	Standard Cell Digital
HPIC		
—	Linear Bipolar CBIC-M CBIC-L CBIC-R CBIC-S	—
—	Digital Bipolar SFOXL OXL SBC BEST	—
—	High-Voltage IC	—
—	Solid-State Relays	—
—	GaAs	—
NMOS		
J	3.5 N	Standard Cell Digital
	5.0/7.5 N	Analog
	5.0/7.5 N	Standard Cell Digital
K	1.7/1.9/2.8 N	Custom Digital

Table 15. Sampling Universes by Technology: THC 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	All
	Multiple In Line	All
G	Small Outline	SOJ
		SOG
H	Ceramic Leadless CC	All
I	Plastic Leadless CC	All

Reliability Monitoring Program

(continued)

Calculating Failure Rates

Infant Mortality

Lucent Technologies measures devices after 24 hours of stress at 150 °C. With the following assumptions, 24-hour failure rates approximate the percentage of ICs that could be expected to fail in the first month of continuous use:

Ea = 0.4 eV
To = 55 °C

Early Life

With the same assumptions as for infant mortality, Lucent Technologies' 160-hour failure rate approximates the percentages of failures that could be expected to fail in the first half-year of continuous use.

Steady-State Life

IC instantaneous failure rates rapidly decrease to a low, relatively constant level. Steady-state instantaneous failure rates are often expressed in units of FITs, failures in 1 billion hours of operation. The FIT rate can be calculated from the equation:

$$\frac{X^2 (10)^9}{\text{2 x sample size x hours of stress x acceleration factor}}$$

X^2 can be found from the probability table using (2f+2) degrees of freedom. For example, if two ICs in a sample of 500 fail after 1000 hours of life testing at 150 °C:

Instantaneous failure rate =

$$\frac{6.212\ (10)^9}{2 \times 500 \times 1000 \times 260} = 24 \text{ FITs}$$

Assumptions:

Ea = 0.7 eV

To = 55 °C

X^2 @ 60% probability

Failure Mode Analysis (FMA)

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

Lucent Technologies' 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.

Reliability Monitoring Program

(continued)

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.

Analytical Lab

When more sophisticated analytical techniques are needed, the Lucent Technologies-Bell Labs Innovations 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.

Corrective Action

In cases in which a reliability sample has a significant number of failures (exceeding the allowed sample limit) such that the lot fails, corrective action of some form is necessary. Corrective action is also necessary when there are customer returns having unique failure mechanisms or when shop yield loss becomes excessive.

Corrective action in these cases is carried out by a Reliability Review Board (RRB). A Reliability Review Board may be convened by anyone with a device issue. The membership of an RRB is composed of those technical experts needed to determine the nature of the device issue. RRB membership is flexible but usually consists of product engineers, reliability engineers, process engineers, and customer technical support engineers. Other people with technical expertise are added to RRB membership as the need arises.

The first obligation of an RRB is to protect the customer. To this end, every effort is made to confine the device issue to a wafer lot or an assembly lot and isolate the offending material. If material has escaped and represents a threat to the customer, the RRB is obligated to notify all customers receiving the affected product. Secondly, the RRB is responsible for obtaining careful failure analysis to determine the root cause of failure. With an accurate determination of root cause in hand, the RRB determines the corrective action to be implemented. The ownership of the corrective action is assigned, and implementation is carried out under the direction of the process owner.

The RRB must then determine if the corrective action has been effective. Only after sufficient data are gathered to verify the effectiveness of the corrective action is the work of the RRB completed and the RRB disbanded.

Corrective action is documented in several different ways. The RRB may keep minutes of their meetings. If minutes are not kept, the reliability engineer (who is a member of the RRB) is obligated to maintain a running summary of the activities of the RRB. This summary is reviewed periodically for progress toward stated goals. In addition, a RRB Corrective Action report is made to management when the RRB is formed. It is updated periodically and reissued when the RRB disbands.

INQUIRE Database

INQUIRE Database and Its Role in Quality Assurance

Lucent Technologies maintains a number of databases as part of its Quality Information Systems Architecture, organized into a system called INQUIRE. Figure 13 shows the INQUIRE system architecture.

INQUIRE is an on-line system which provides access to component quality information. It provides access to data produced as a result of qualification, reliability, and final inspection testing performed throughout Lucent Technologies. INQUIRE also features access to an index of documents available on manufactured devices, and documents can be easily ordered via an on-line request feature.

INQUIRE collects information from several internal Lucent Technologies databases used to manage and track the quality of manufactured product. Its menu-driven system offers query functions, help options, report and graphics generation, and a bulletin board. It also features a single point of contact for user questions, and the on-line documentation ordering function.

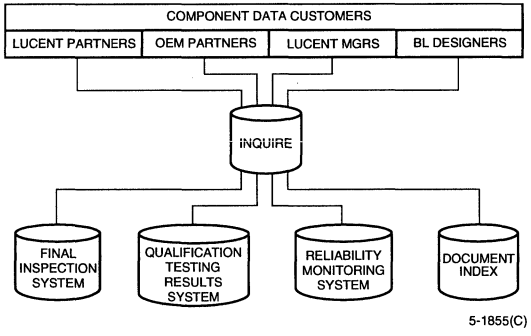


Figure 13. INQUIRE System Architecture

Qualification Test Data

Lucent Technologies performs qualification testing on devices in many product families. The testing is performed to demonstrate the reliability of both new technologies and changes to existing technologies.

Product qualification results are tracked throughout Lucent Technologies' production facilities by the Qualification Testing Results Systems (QTRS) and fed to INQUIRE in twice-weekly consolidated updates. The qualification information available from INQUIRE can be displayed in query or report format and includes qualification test results by device type.

Historical bases for the individual qualifications such as new process qualifications and new code qualifications are also available. As the qualification process of a product evolves, its status can be tracked. Available reports include a qualification device summary list.

Reliability Monitoring Data

Lucent Technologies also performs reliability testing on devices in most product families, in order to ensure that reliability objectives are met on an ongoing basis. These results are tracked by the Reliability Monitoring System (RELMS) and fed to INQUIRE in weekly updates.

INQUIRE can display this data in query, report, or graphic format. Data includes reliability test results summarized by device code, test universe, or technology family. All Failure Mode Analysis (FMA) data and related corrective actions are also available.

Available reports include a reliability test summary report, a test universe summary, and a reliability device level summary report. Graphic output includes charts summarizing percent defective per technology family and percent defective per test universe.

Lucent Technologies product quality data is collected on a lot-by-lot basis for all manufactured device codes and used to calculate and report the performance of product families.

INQUIRE Database (continued)

Final inspection test results are tracked throughout Lucent Technologies by the Final Inspection System (FIS) and fed to INQUIRE in monthly consolidated updates. Product performance is reported in ppm defective, and represents the Average Outgoing Quality (AOQ) of the product.

Report data is available through INQUIRE in query, report, or graphic format. Final inspection test results can by summarized by Strategic Business Unit, such as MOS; by product family, such as ASICs, or by individual device type. Results can also be organized by current month, three-month interval, and 12-month interval.

Document Availability Information

Lucent Technologies provides documents for most of its devices. These documents take the form of data sheets, user manuals, application notes, and product briefs, for example.

Document availability is tracked by INQUIRE's Document Index Database. Users can order documents on-line and can query document availability on-line as well as obtain printed reports.

Applications

System designers and component selection engineers can use INQUIRE to choose components for their specific application. Users can input a specific device type into INQUIRE, and extract the information they need to make a choice.

After they identify components for specific applications, users can confirm that the components have been appropriately qualified by accessing the Qualification Data Query screen, which displays the qualification document number plus qualification results.

As these users continue the component selection process, they can access the Reliability Data Query Screen and the Final Inspection Data Query Screen to identify ongoing reliability data, as well as the quality performance of each device in order to make an informed decision. System designers who need more information can order data sheets electronically from INQUIRE as well.

Customer Support

Customer Service and the Device Quality Issue (DQI) Process

Customer satisfaction is a top priority in the Lucent Technologies quality effort. Lucent Technologies utilizes a philosophy of having a central point of contact for customers to obtain information, gain access to additional resources, and log complaints. The Customer Technical Support Center (CTSC) is the central point of contact for quality issues, and is responsible for receiving and resolving formal and informal complaints of a technical nature. All of our customer contact employees (salespersons, engineers, managers, etc.) are instructed to direct informal complaints to the CTSC to ensure that all complaints are aggregated into quality issues.

Formal quality complaints are generally received and managed via our Device Quality Issue (DQI) process shown in Figure 14. The process incorporates documentation of the pertinent facts surrounding the issue and the provision of samples for analysis. Complaints originating from customers unfamiliar with our DQI system are converted to DQIs to provide a common internal path for all quality complaint management. These customers also receive information about our DQI system to aid any future complaint resolution. A comprehensive database is maintained to aggregate and track DQI issues.

All pertinent facts related to the issue are identified, including the source of the issue (incoming inspection, factory failure, and field failure), the generic type of issue (conformance with specifications, performance of existing specifications, customer-induced failure, and no trouble found), and a brief description of the mode of failure and corrective action.

In addition, the CTSC maintains a telephone log of customer comments and general issues. A routine report analyzes the DQI database to identify trends. The report is distributed to all responsible management.

Customer Support (continued)

In addition to the DQI system, a Return Material Approval (RMA) system, managed by the Customer Service Organization, is monitored by CTSC personnel. The RMAs are reviewed to determine if a quality-related problem caused the return. If so, a DQI is issued to investigate and report on the issue.

The customer is provided with an initial response within 48 hours of the receipt of a quality issue. When appropriate, the response reports on the results of the initial retest of the product. In any case, the 48-hour response notifies the customer that the quality issue has been received and is being addressed.

The goal is to provide a final, written report within ten working days of initiation of the issue. The report provides the initial retest results, the failure-mode analysis findings, and the corrective action to be implemented. When more than ten days are required to complete the investigation and the report, a written report of the proposed procedure necessary to complete the investigation is provided within ten days. The customer is the final arbiter in any investigation regarding the satisfactory nature of the investigation of the report.

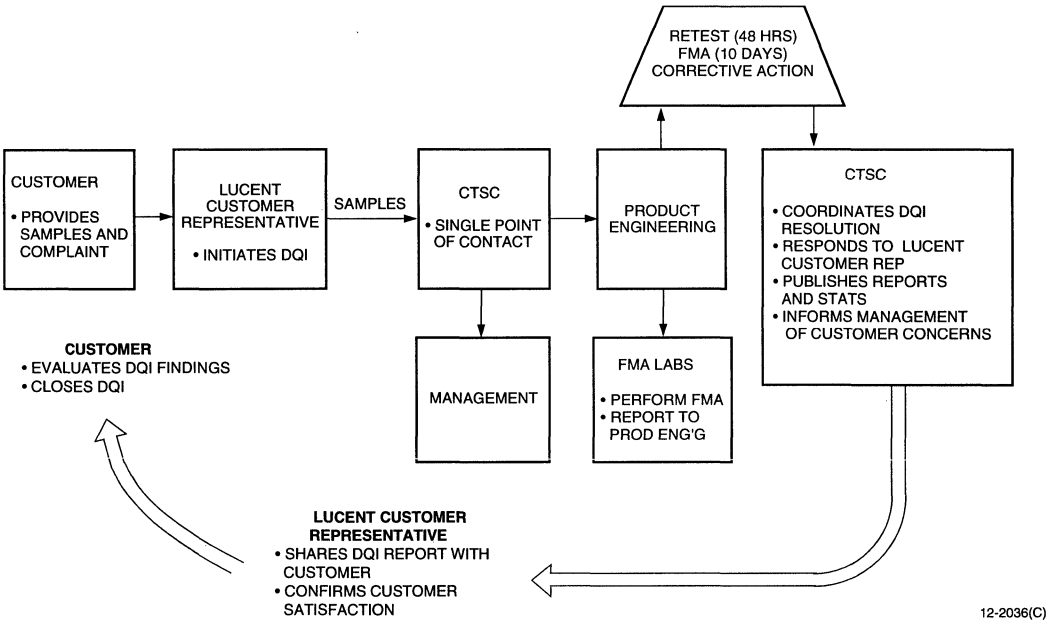


Figure 14. DUI Full-Circle Process Flow

Device Qualification Test Procedures and Results

The purpose of this section is to document the qualification test procedures and results of the ATT3000 and *ORCA* Series FPGA product families. The qualification plan for these FPGA product families 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 testing. 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.

ATT3000 Series Qualification

There are five arrays in the ATT3000 Series FPGA product family: ATT3020, ATT3030, ATT3042, ATT3064, and ATT3090. These arrays are then assembled in a number of different plastic packages. These packages are plastic-leaded chip carrier (PLCC), plastic-pin grid array (PPGA), metric-quad flat pack (MQFP), thin-quad flat pack (TQFP), and shrink-quad flat pack (SQFP).

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 using the 0.6 µm and 0.55 µm CMOS technologies. Lucent Technologies' 0.6 µm and 0.55 µm CMOS processes employ N- and P-channel LDD MOS transistors. They also use two levels of metal and one level of polycide. Source and drain regions are silicided for low resistance. These technologies have been rigorously tested for reliability and manufacturability and are 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, needed tests were performed on the ATT3090 die in the 84-pin PLCC package for 0.6 µm and on the ATT3042 die in the 84-pin PLCC package for 0.55 µm.

Package: The QRB determined the qualification requirements of each die in a given package. The results are given in Table 16 (0.6 µm) and Table 17 (0.55 µm).

ATT3000 Series Qualification (continued)

Table 16. ATT3000 Series (0.6 µm)

Qualification Information	Device						
	3090-84PLCC (Q92055)	3090-175PPGA (Q92054)	3042-100TQFP (QRB92.51)	3042-100QFP (QRB93.2)	3090-84PLCC (0.9 µm) (Q90148)	3090-160QFP (0.9 µm) (Q90140)	3090-175PPGA (0.9 µm) (Q90137)
ESD-HBM[1]	>2000 V	>2500 V	>2000 V	>1500 V	>3500 V	>2000 V	>3000 V
ESD-CDM[1]	>2000 V	>3000 V	>3000 V	>3000 V	>3000 V	>2000 V	>2500 V
1000 hrs. HTOB	1/105[2] Pass	1/105[3] Pass	—	—	—	—	0/98 Pass
1000 hrs. THB	—	—	—	—	0/133 Pass	1/134[4] Pass	1/135[6] Pass
CLASS	—	—	—	—	0/134 Pass	0/134 Pass	0/135 Pass
Autoclave (96 hrs. SB)	—	—	—	—	0/105 Pass	2/105 Pass	—
100 c/s TC	—	—	—	—	0/105 Pass	2/105[5] Pass	0/105 Pass
15 c/s TS	—	—	—	—	0/25 Pass	0/25 Pass	0/125 Pass
Moisture Resistance	—	—	—	—	—	—	0/38 Pass
Corrosion	—	—	—	—	—	—	0/15 Pass
Solvent Resistance	—	—	—	—	—	—	0/8 Pass
Physical Dimensions	—	—	—	—	—	Pass	0/15 Pass
Solderability	—	—	—	—	—	—	Note 7
Bond Strength	—	—	—	—	—	—	Pass
Die Shear Strength	—	—	—	—	—	—	Pass
X-Ray	—	—	—	—	—	—	0/5 Pass
LU Class	IV	IV	IV	IV	IV	IV	IV
Status	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 6-14 for test descriptions.
2. FMA reported a gate-level defect.
3. Defective package. 5 Ω short from VSS to VDD.
4. FMA reported gate-level defect found and isolated. Changes in the silicide process have been made to address this failure mechanism.
5. FMA reported lifted ball bond and package stress crack.
6. Defective package.
7. Leads to be solder dipped before shipment.

ATT3000 Series Qualification (continued)

Table 16. ATT3000 Series (0.6 μm) (continued)

Qualification Information	Device			
	1042L.BR (Q89166)	**409AT (Q89027)**	***WE*®-DSP32CF32 (Q89129.1)**	**409ATX (Q90001)**
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	100-pin MQFP
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
Status	Fully Qualified	Fully Qualified	Fully Qualified	Fully Qualified

ATT3000 Series Qualification (continued)

Table 16. ATT3000 Series (0.6 µm) (continued)

Qualification Information	Device	
	S1116X (Q90111)	**MR120 (Q92055)**
Chip Size (µm)	6830 x 7040	8130 x 8130
Package	100-pin MQFP	208-pin SQFP
Steam Bomb	0/105 Pass	0/105 Pass
1000 hrs. HTOB	1/105 Pass	0/105 Pass
Class	2/132 Pass	0/132 Pass
1000 hrs. THB	0/129 Pass	0/129 Pass
100 c/s TC or 300 c/s TC	1/105 Pass	1/105 Pass
15 c/s TS or 100 c/s TS	0/25 Pass	1/25 Pass
Moisture Resistance	0/38 Pass	—
Corrosion	0/15 Pass	—
Solvent Resistance	0/8 Pass	—
Physical Dimension	0/15 Pass	0/15 Pass
Solderability	0/22 (3 devices) Pass	—
LI	0/15 Pass	—
Bond Strength	—	—
Die Shear Strength	—	—
X-Ray	0/5 Pass	—
Status	Fully Qualified	Fully Qualified

ATT3000 Series Qualification (continued)

Table 17. ATT3000 Series (0.55 μm)

Qualification Information	Device		
	3042-84PLCC (Q95273)	3090-84PLCC (Q96098)	3090-208SQFP (Q96099)
ESD-HBM[1]	>1000 V	>1000 V	>2000 V
ESD-CDM[1]	>1000 V	>500 V[2]	>1000 V
1000 hrs. HTOB	0/105 Pass	—	—
1000 hrs. THB	—	—	—
CLASS	—	—	—
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	II	III	II
Status	Fully Qualified	Fully Qualified	Fully Qualified

1. HBM = human-body model, and CDM = charged-device model; see page 6-14 for test descriptions.
2. The corner pins pass >1000 V.

ATT3000 Series Qualification (continued)

Table 17. ATT3000 Series (0.55 μm) (continued)

Qualification Information	Device	
	MR109 (MR109)	**MR120 (MR120)**
Chip Size (μm)	10280 x 10120	8130 x 8130
Package	84-pin PLCC	208-pin SQFP
Steam Bomb	0/45 Pass	0/105 Pass
1000 hrs. HTOB	0/135 Pass	0/105 Pass
Class	0/132 Pass	0/132 Pass
1000 hrs. THB	0/132 Pass	0/129 Pass
100 c/s TC or 300 c/s TC	1/105[3] Pass	1/105 Pass
15 c/s TS or 100 c/s TS	0/25 Pass	1/25 Pass
Moisture Resistance	0/38 Pass	—
Corrosion	0/15 Pass	—
Solvent Resistance	0/8 Pass	—
Physical Dimension	0/15 Pass	0/15 Pass
Solderability	0/22 Pass	—
LI	0/15 Pass	—
Bond Strength	0/15 Pass	—
Die Shear Strength	0/5 Pass	—
X-Ray	0/5 Pass	—
Status	Fully Qualified	Fully Qualified

3. Aluminum short caused by collet damage.

ORCA 1C Series Qualification

There are four arrays in the *ORCA* 1C Series FPGA product family: ATT1C03, ATT1C05, ATT1C07, and ATT1C09. These arrays are then assembled in a number of different packages. These packages are plastic-leaded chip carrier (PLCC), metric-quad flat pack (MQFP), thin-quad flat pack (TQFP), shrink-quad flat pack (SQFP), and ceramic-pin grid array (CPGA).

Qualification Strategy

The qualification of *ORCA* 1C Series FPGAs can be broken into three specific groups: process technology, silicon design, and package.

Process Technology: These arrays are processed using the 0.6 µm CMOS technology. Lucent Technologies' 0.6 µm CMOS process employs N- and P-channel LDD MOS transistors. 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 four arrays in the *ORCA* 1C 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 four arrays. Therefore, needed tests were performed on the ATT1C05 die in the 240-pin SQFP package.

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

ORCA 1C Series Qualification (continued)

Table 18. *ORCA* 1C Series (0.6 μm)

Qualification Information	Device				
	ATT1C05-240SQFP (Q92167)	ATT3090-175PPGA (Q92054)	ATT1C07-304SQFP (Q94044)	ATT3090-175CPGA (Q92161)	1051CN (Q92168)
ESD—HBM[1]	>2000 V	>2500 V	>2000 V	>1000 V	>1000 V
ESD—CDM[1]	>3000 V	>3000 V	>2000 V	>2000 V	>2000 V
1000 hrs. HTOB	0/109 Pass	1/105 Pass[3]	—	0/133 Pass	1/105 Pass[7]
1000 hrs. THB	1/132 Pass[2]	—	—	—	—
Class	0/132 Pass	—	—	—	—
Steam Bomb	1/105 Pass[4]	—	—	—	—
100 c/s TC or 300 c/s TC	1/105 Pass[5]	—	—	0/105 Pass	—
15 c/s TS or 100 c/s TS	1/15 Pass[6]	—	—	—	—
Moisture Resistance	0/38 Pass	—	—	—	—
Corrosion	—	—	—	—	—
Solvent Resistance	—	—	—	—	—
Physical Dimension	0/15 Pass	—	—	—	—
Solderability	0/22 Pass	—	—	0/3 Pass	—
Bond Strength	—	—	—	—	—
Die Shear Strength	—	—	—	—	—
X-Ray	0/5 Pass	—	—	—	—
Latch-up	IV	IV	III	IV	IV
Status	Fully Qualified	Fully Qualified	Fully Qualified	Fully Qualified	Fully Qualified

1. HBM = human-body model, and CDM = charged-device model; see page 6-14 for test descriptions.
2. One marginal functional failure after 432 hours (acceptable). No other devices failed after 1000 hours.
3. Defective package. 5 Ω short from VSS to VDD.
4. No defect found.
5. One open via found.
6. One mechanically broken wire found.
7. Particles at polysilicon along edge of gate due to poor spacer.

ORCA 1C Series Qualification (continued)

Table 18. *ORCA* 1C Series (0.6 µm) (continued)

Qualification Information	Device		
	3042-100TQFP (QRB92.51)	MR120 (Q92055)	MR141[1]
ESD—HBM[2]	>2000 V	>1000 V	—
ESD—CDM[2]	>3000 V	>1000 V	—
1000 hrs. HTOB	—	0/105 Pass	0/105 Pass
1000 hrs. THB	—	0/129 Pass	0/131 Pass
Class	—	0/132 Pass	—
Steam Bomb	—	0/105 Pass	0/105 Pass
100 c/s TC or 300 c/s TC	—	0/105 Pass	0/105 Pass
15 c/s TS or 100 c/s TS	—	1/25 Pass[3]	0/25 Pass
Moisture Resistance	—	—	0/38 Pass
Corrosion	—	—	0/15 Pass
Solvent Resistance	—	—	0/8 Pass
Physical Dimension	—	0/15 Pass	—
Solderability	—	—	0/22 Pass
Bond Strength	—	—	0/15 Pass
Die Shear Strength	—	—	0/15 Pass
X-Ray	—	—	0/5 Pass
Latch-up	IV	II	—
Status	Fully Qualified	Fully Qualified	Fully Qualified

1. Also passed lead integrity (0/15).
2. HBM = human-body model, and CDM = charged-device model; see page 6-14 for test descriptions.
3. Gate-level defect found under polysilicon gate.

ORCA 2C Series Qualification

There are eight arrays in the *ORCA* 2C Series FPGA product family: ATT2C04, ATT2C06, ATT2C08, ATT2C10, ATT2C12, ATT2C15, ATT2C26, and ATT2C40. These arrays are then assembled in a number of different packages. These packages are plastic-leaded chip carrier (PLCC), metric-quad flat pack (MQFP), plastic-quad flat pack (QFP), plastic-ball grid array (PBGA), thin-quad flat pack (TQFP), shrink-quad flat pack (SQFP), and ceramic-pin grid array (CPGA).

Qualification Strategy

The qualification of *ORCA* 2C Series FPGAs can be broken into three specific groups: process technology, silicon design, and package.

Process Technology: These arrays are processed using the 0.5 µm CMOS technology. Lucent Technologies' 0.5 µm CMOS process employs N- and P-channel LDD MOS transistors. The process also uses three levels of metal. This technology has been rigorously tested for reliability and manufacturability and is fully qualified.

Silicon Design: Since the eight arrays in the *ORCA* 2C 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 eight arrays. Therefore, needed tests were performed on the ATT2C15 die in the 304-pin SQFP package.

Package: The QRB determined the qualification requirements of each die in a given package. The results are given in Table 19.

ORCA 2C Series Qualification (continued)

Table 19. *ORCA* 2C Series (0.5 μm)

Qualification Information	Device					
	ATT2C04-160QFP (Q95319)	ATT2C10-256BGA (Q96068)	ATT2C10-304SQFP (Q95041)	ATT2C15-208SQFP (Q94290)	ATT2C15-240SQFP (Q93234)	ATT2C15-304SQFP (Q94048)
ESD—HBM[1]	>2000 V	>2000 V	>2000 V	>2000 V	>2000 V	>2000 V
ESD—CDM[1]	>1000 V	>1000 V	>1000 V	>1000 V	>1000 V	>1000 V
1000 hrs. HTOB	—	—	—	—	1/60 Pass[2]	1/60 Pass[2]
1000 hrs. THB	—	—	—	—	1/120 Pass[3]	1/132 Pass[4]
Class	—	—	—	—	0/133 Pass	0/133 Pass
Steam Bomb	—	0/46 Pass	0/45 Pass	—	0/105 Pass	0/105 Pass
100 c/s TC or 300 c/s TC	—	0/47 Pass	—	—	0/105 Pass	0/105 Pass
15 c/s TS or 100 c/s TS	—	0/16 Pass	—	—	0/16 Pass	0/15 Pass
Moisture Resistance	—	—	—	—	—	—
Corrosion	—	—	—	—	—	—
Solvent Resistance	—	—	—	—	—	—
Physical Dimension	—	0/15 Pass	—	—	—	—
Solderability	—	—	—	—	—	—
Bond Strength	—	—	—	—	—	—
Die Shear Strength	—	—	—	—	—	—
X-Ray	—	—	0/5 Pass	—	—	—
Latch-up	II	II	II	II	II	II
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 16 for test descriptions.
2. Electrical overstress.
3. Metal 1 short.
4. FMA found failure due to horizontal crack, which created EOS damage.

ORCA 2C Series Qualification (continued)

Table 19. *ORCA* 2C Series (0.5 µm) (continued)

Qualification Information	Device						
	ATT2C26-304SQFP-PQ2 (Q94227)	ATT2C15-364CPGA (Q93231)[4]	ATT2C40-208SQFP-PQ2 (QRB95.17)	ATT2C40-240SQFP-PQ2 (Q95007)	1159J (Q94124)	T92020S (Q93181)	1042BG (T92222)
ESD—HBM[1]	>2000 V	—	—	>2000 V	>2000 V	>2000 V	>2000 V
ESD—CDM[1]	>1000 V	>1000 V	—	>1000 V	>1500 V	>1000 V	>2000 V
1000 hrs. HTOB	0/65 Pass	1/60 Pass[2]	—	—	0/97 Pass	1/105[3] Pass	0/105 Pass
1000 hrs. THB	0/100 Pass	—	—	0/77 Pass	0/132 Pass	0/130 Pass	0/132 Pass
Class	0/100 Pass	—	—	0/80 Pass	0/132 Pass	0/132 Pass	0/132 Pass
Steam Bomb	0/45 Pass	—	—	—	0/105 Pass	0/105 Pass	0/105 Pass
100 c/s TC or 300 c/s TC	0/50 Pass	0/45 Pass	—	0/46 Pass	0/105 Pass	0/105 Pass	0/105 Pass
15 c/s TS or 100 c/s TS	0/15 Pass	0/15 Pass	—	—	0/25 Pass	0/25 Pass	0/25 Pass
Moisture Resistance	—	—	—	—	—	—	—
Corrosion	—	—	—	—	—	—	—
Solvent Resistance	—	—	—	—	—	—	—
Physical Dimension	0/15 Pass	0/15 Pass	0/15 Pass	0/15 Pass	—	0/15 Pass	—
Solderability	—	—	—	—	—	—	—
Bond Strength	—	—	—	—	—	—	—
Die Shear Strength	—	—	—	—	—	—	—
X-Ray	—	—	—	—	—	—	—
Latch-up	II	—	—	II	IV	II	IV
Status	Fully Qualified	Fully Qualified	Fully Qualified	Fully Qualified	Fully Qualified	Fully Qualified	Fully Qualified

1. HBM = human-body model, and CDM = charged-device model.
2. Conductive particle in interlevel dielectric.
3. No failure found in FMA.
4. Mechanical sequence test was also passed (0/38).

ORCA 2C Series Qualification (continued)

Table 19. *ORCA* 2C Series (0.5 µm) (continued)

Qualification Information	Device		
	MR109 (MR109) 84-PLCC	MR119 (MR119)	ATT2C26-388BGA (Q95013)[7]
ESD—HBM[1]	>3000 V	—	>2000 V
ESD—CDM[1]	>1000 V	—	>500 V[6]
1000 hrs. HTOB	0/135 Pass	1/103[3] Pass	0/59 Pass
1000 hrs. THB	0/132 Pass	0/130 Pass	0/48 Pass
Class	0/132 Pass	—	0/89 Pass
Steam Bomb	0/45 Pass	0/105 Pass	0/50 Pass
100 c/s TC or 300 c/s TC	1/105[2] Pass	2/105[4] Pass	0/50 Pass
15 c/s TS or 100 c/s TS	0/25 Pass	0/25 Pass	0/15 Pass
Moisture Resistance	0/38 Pass	1/38[5] Pass	0/38 Pass
Corrosion	0/15 Pass	0/15 Pass	0/15 Pass
Solvent Resistance	0/8 Pass	0/8 Pass	0/15 Pass
Physical Dimension	0/15 Pass	—	0/15 Pass
Solderability	0/22 Pass	0/22 Pass	—
Bond Strength	0/15 Pass	—	—
Die Shear Strength	0/5 Pass	—	—
X-Ray	0/5 Pass	0/5 Pass	0/5 Pass
Latch-up	IV	—	II
Status	Fully Qualified	Fully Qualified	Fully Qualified

1. HBM = human-body model, and CDM = charged-device model.
2. Aluminum short caused by collet damage.
3. No defect found.
4. FMA found failure was due to ball bond pullout.
5. EOS damage.
6. Corner pins pass >1000 V.
7. Ball integrity test was also passed (0/15).

ATT1700A Series Qualification

The ATT1700A Series of EEPROMs are assembled in three different packages. These packages are plastic-leaded chip carrier (PLCC), plastic dual-in-line package (DIP), and small-outline integrated circuit (SOIC).

Qualification Strategy

The qualification of the ATT1700A Series EEPROMs was done by performing a full set of tests on the device in each package, with one exception: the HTOB and Bake device tests were only performed on the 8-pin DIP.

Table 20. ATT1700A Series EEPROMs (1.2 µm)

Qualification Information	Device		
	ATT1700A-8DIP (Q94362)	ATT1700A-8SOIC (Q94363)	ATT1700A-20PLCC (Q94364)
ESD—HBM[1]	>2000 V	>2000 V	>2000 V
ESD—CDM[1]	>2000 V	>2000 V	>2000 V
1000 hrs. HTOB	6/768 Pass[2]	—	—
1000 hrs. THB	2/3564 Pass[3]	0/2865 Pass	5/2443 Pass[4]
150 °C Bake	0/899 Pass	—	—
Steam Bomb	0/46 Pass	0/46 Pass	0/3322 Pass
100 c/s TC or 300 c/s TC	0/644 Pass	0/1012 Pass	1/907 Pass[5]
15 c/s TS or 100 c/s TS	0/46 Pass	0/46 Pass	6/878 Pass[6]
Moisture Resistance	0/46 Pass	0/46 Pass	0/46 Pass
Corrosion	0/15 Pass	0/15 Pass	0/15 Pass
Solvent Resistance	0/15 Pass	0/15 Pass	0/15 Pass
Physical Dimension	0/2 Pass	0/2 Pass	0/2 Pass
Solderability	0/22 Pass	0/22 Pass	0/22 Pass
Lead Integrity	0/15 Pass	0/15 Pass	0/15 Pass
HAST	0/122 Pass	0/122 Pass	0/281 Pass
Latch-up	II	II	II
Status	Fully Qualified	Fully Qualified	Fully Qualified

1. HBM = human-body model and CDM = charged-device model; see page 6-14 for test descriptions.
2. Bad column in EEPROM array.
3. IDD standby.
4. Three failed from single bit charge loss, one from leakage, one from extended VDD range.
5. Double row failure.
6. Single/multiple bit charge loss.

Lucent Technologies
Bell Labs Innovations

Masked Array Conversion for *ORCA* ("MACO")

Overview

Lucent Technologies provides two families of field-programmable gate arrays (FPGAs) that offer the ability to perform high-performance, high-density digital functions. These two families of devices are the *ORCA* Series and ATT3000 Series. Since each of these families is programmable, they allow the designer to finalize a design through any number of programmable iterations without lengthy implementation delays or needless expense.

After implementation and initial production of the design, the FPGAs can possibly be replaced with a mask-programmed device, such as a gate array. This would only be done for cost-reduction purposes in high-volume applications after the functionality of the design has been proven. In this way, short-term time-to-market requirements can be fulfilled using the FPGA, and long-term cost requirements can be satisfied using the mask-programmed device.

Lucent Technologies provides a design methodology and the tools to automate this translation process from the *ORCA* Series of FPGAs. This process is called Masked Array Conversion for *ORCA* ("MACO") and allows the user to convert an *ORCA* FPGA to a lower-cost device that has been optimized to emulate the *ORCA* FPGAs. This process allows the user to seamlessly target a MACO device using the final verified design files from *ORCA* Foundry.

A second translation process is available for the conversion of the ATT3000 Series of FPGAs to the Lucent Technologies ATT656 Series of CMOS gate arrays. This methodology provides a low-risk path for creating pin-compatible devices that work as direct replacements for the ATT3000 Series of FPGAs.

This manual is divided into three sections dealing with FPGA migration to gate arrays and their implementation:

- *ORCA* FPGA Migration to MACO or a Lucent Technologies Gate Array
- Masked Array Conversion for *ORCA* ("MACO")
- Preventing FPGA Migration Timing Issues

Although ATT3000 series migration is not specifically covered, this path is also available. More specific information regarding FPGA to gate array migration from either the *ORCA* Series or ATT3000 Series is available directly from Lucent Technologies.

ORCA FPGA Migration to MACO or a Lucent Technologies Gate Array

Introduction

Lucent Technologies has developed a methodology to enable our customers to migrate an *ORCA* FPGA design into one of our high-volume products that provides a high-quality, low-cost migration path. There are several options available to allow our customers to choose the most advantageous path for their needs.

The FPGA design will be translated and optimized in the target technology. Optional scan path insertion allows for higher fault coverage and better quality gate arrays to be delivered. Our wide variety of package offerings makes a pin-for-pin compatible migration possible.

The models and production parts will be fabricated in our ISO- and QML-certified lines and packaged in the same package as the original FPGA, assuring ease of introduction into manufacture.

Design the FPGA with Migration in Mind

When migrating a design from the FPGA to a gate array or MACO device, much time and effort will be saved if the FPGA designer has put some effort into designing with the future migration in mind. When the migration effort is taking place, the original FPGA designer may be unavailable for consultation, so taking into account the need to migrate the design at a future date will save the time and effort necessary to relearn the circuit and uncover all of the potential concerns from a gate array design point of view. Good documentation and functional vectors will also help shorten the gate array/MACO design schedule.

Use Good Digital Design Techniques

One of the biggest problems in digital designs is asynchronous circuits. While a successful FPGA development might include asynchronous circuitry, this same circuitry may cause the migrated device to be untestable, have low yield, or have unstable performance. While every design will not allow it, complete implementation of the following points will yield a cleaner, more predictable design with good performance and high yield:

1. Make your design completely synchronous. A single clock, synchronous design eliminates asynchronous interfaces between clock domains. This makes circuit analysis, vector generation, and testing much easier and the device more reliable.
2. Don't divide clocks or gate them with other signals for use as clock or data inputs to registers or latches.
3. Use synchronous circuits instead of asynchronous loops or ripple counters.
4. Synchronize any asynchronous primary inputs to the main chip clock.
5. Don't purposely insert analog delays in a path to make the timing right for a particular circuit. Using long routing wires to implement a delay will be lost when the FPGA netlist is translated, causing timing problems in the gate array. If delays are needed, use digital circuits, such as a string of inverters, but document this clearly because the delay elements will otherwise be optimized away during resynthesis.
6. Don't implement circuits that will create spikes since spike width may vary and cause undesired behavior in the gate array. Use enable signals to eliminate spikes.
7. Ensure that all internal buses are actively driven at all times.
8. Provide for initialization of all flip-flops (FFs) to minimize vector count and increase testability.

ORCA FPGA Migration to MACO or a Lucent Technologies Gate Array

(continued)

Design the FPGA for Testability

The FPGA to be migrated should be designed with testability in mind. A mask-programmed device needs to have high fault coverage, and some circuits can be made untestable if care is not taken to recognize and address them ahead of time. Testability could always be added later but would probably add pins, making a pin-for-pin compatible migration impossible.

A design is completely testable when you can control and observe every node in the circuit from package pins with a reasonable number of vectors. Designing for testability means making the circuit nodes more controllable and observable from primary I/O. Some ways to achieve this are:

1. Use unused pins to control or observe a deeply imbedded node.
2. Break up long counter chains into smaller pieces that won't require as many vectors to exercise.
3. Initialize every flip-flop (FF) using a global set/reset signal. Every gate array needs to start from a known state when testing it in the factory. Initializing all FFs accomplishes this with a minimum number of vectors.
4. MUX input and output pins to increase the controllability and observability of internal nodes.

Need for Functional Test Vectors

Lucent Technologies' objective is to maximize the first-time success rate of our customers' boards. This can be accomplished with devices that pass vectors which functionally mimic the system's interaction with the chip being designed. These test vectors are called functional vectors and should stress all critical paths, asynchronous interfaces, and I/O specifications while being run at system clock speeds. These vectors need to be written by someone with a detailed understanding of the intended chip operation, normally the FPGA logic designer.

The silicon vendor does not have this functional knowledge and can only write fault coverage vectors that will toggle nodes in the chip to detect stuck at faults introduced in the manufacturing process. These vendor written vectors are not written in such a way that they verify chip functionality, but simply make each net in the circuit change from a one to a zero and back in such a manner that the result can be viewed at the output pins of the chip. Therefore, in order to maximize the first-time success of the device in the system, Lucent Technologies strongly suggests that functional at-speed vectors be supplied for every hard-mask design.

To generate a high level of fault coverage, the vendor will normally use a scan methodology that introduces extra logic into the chip. The extra logic will marginally slow the maximum operating speed of the device, but, since the gate array device can operate at a higher rate of speed than the FPGA, the effect is generally negligible.

Often an FPGA development cycle takes the form of specifying the circuit either through schematic capture or high-level language specification. A netlist is then extracted and physical layout is implemented. The resulting part is plugged into the target board, and lab testing is done to verify proper operation/implementation of the device. If the device doesn't work when tested in the board, the cycle is repeated until the desired operation is achieved.

This method of design is viable only because the FPGA can be reprogrammed as many times as is necessary. This method also requires that the circuit be small enough that the designer can gain a good idea of the specific part of the circuit that is not implemented correctly just by analyzing the board behavior. As FPGAs get more complex, this method of design becomes less and less successful, and a more traditional ASIC approach becomes necessary. This includes simulating functional at-speed vectors to determine correct implementation of the circuit.

ORCA FPGA Migration to MACO or a Lucent Technologies Gate Array

(continued)

While it is not necessary in some FPGA design cycles today to write and simulate functional at-speed vectors, it is desirable to do so for those FPGA designs which will be migrated to a gate array in the future. A design can be migrated to a gate array just by using fault coverage vectors for manufacturing screening, but this does not give as high a level of assurance that the translated design works logically or meets all of the timing specifications. In this methodology, the fault coverage vectors will only screen out manufacturing defects and may result in delivering initial models that will not work on the board; or worse, initial models may work, but sometime during production, all of the chips (that pass this kind of silicon manufacturing test) may fail at board level. This, in turn, will cause production of the board to stop and will mean that some engineering effort will be required.

The following is a list of some of the issues fault coverage vectors **cannot** specifically check:

1. Timing at interfaces between multiple clock domains.
2. Any critical timing paths between FFs.
3. Any I/O timing specifications.
4. Any paths that used (either intentionally or unintentionally) excessive layout parasitics to ensure timing was met in the FPGA.

   Similarly, any delays added by using digital elements such as inverter/buffer strings or nand/nor gates with control inputs tied to VDD/VSS will probably get optimized away during the resynthesis process unless clearly documented.
5. Races that may have been introduced in the gate-array design.
6. Mistakes that would otherwise be propagated from the FPGA to the gate array.

In short, a good set of functional test vectors is imperative to design and test the gate array. Otherwise the chip may not be tested thoroughly, and the device may not be manufacturable over the full process variation of the silicon or board manufacturing lines. This set or sets of functional vectors should be developed when the FPGA is designed because months or years later, when the FPGA is being migrated, the original designer who is familiar with the details of the chip may be reassigned to other projects or may not even be at the same company.

Vector Writing Guidelines

Vectors are a set or sets of input stimuli and output expected responses used to exercise the chip for simulation and manufacturing test purposes. They are most often represented as a series of 1s and 0s for the logic levels and 3s and Zs for the unknown and Hi-Z levels. There are two types of vectors generally used in the design of a chip: functional vectors and fault coverage vectors.

Functional vectors are a set or sets of vectors used to exercise the function of the chip in a manner consistent with the way it will be used on the board. These vectors are typically simulated at system speeds and in a manner consistent with the I/O specification of the chip.

Fault coverage vectors are a set or sets of vectors whose sole purpose is to toggle as many of the internal nodes of a chip as possible in both directions and be able to observe that change on a primary output. Fault coverage vectors are usually simulated and tested at slower speeds than the system requires, since they do not necessarily represent the function of the chip, and running them at system speeds would probably result in overdesigning the chip.

The fault coverage of a chip is a measure of the number of faults that can be detected by observing a different behavior at the chip's primary outputs from what is expected for a fault-free device when input vectors are applied. Good fault coverage numbers are typically in the 90th percentile. This reduces the number of undetected faults that can slip through the manufacturing tests to an acceptable range.

Vectors need to conform to the test hardware limitations in a way that will not introduce artificial timing constraints, such as races between the clock and data leads. Vectors should be written to conform to the system stimuli as much as possible. Additionally, the timing as applied should stress all I/O specs so that they can be verified during simulation, but the vectors also need to be written to set up the logic levels appropriately to do this.

ORCA FPGA Migration to MACO or a Lucent Technologies Gate Array

(continued)

To be successful at writing stimuli for test vectors, discipline must be employed. Valid test vectors that conform to test equipment constraints use only a subset of the possible stimuli which can be described. Venturing out of that subset will force reworking of stimuli. To stay in the test vector subset of stimuli mentioned above, the following guidelines must be followed:

1. **Do not waste cycles.** An important testing resource is vector memory. Long sets of vectors also take longer to simulate. Very long vector sets waste test resources and simulation/verification time. Taking a little time to pack vector sets full of function will pay off. Similarly, only simulate inputs which will eventually affect outputs. The toggling of internal nodes, while visible in simulation, cannot be detected on the test set.
2. **Use only synchronous, deterministic sequences.** Some stimuli generation schemes rely on system simulation. This can work, but it carries a risk along with it. Some systems use asynchronous handshaking to communicate. This can put signal transitions at a different point of time in several different vector periods. Most ASIC test generation schemes, including Lucent Technologies', do not have this capability. In addition, some system simulations will generate nondeterministic vectors. In this case, the device under test (DUT) requires a carefully timed input response to one of its outputs. Again, most test machines, including Lucent Technologies', do not have this capability.
3. **Organize vectors into self-initializing sets with specific purposes.** Test vectors are typically organized into sets. Each set consists of sampled values for each DUT pin during each vector cycle. Each set also includes timing and waveform information to be associated with each pin and sampled column of data. For debugging purposes, it is quite useful for each set to have a known purpose. For example, some sets should only exercise specific modules in the design. In this way, if only some sets fail, only specific modules need to be considered. Each primary I/O pin must be present in every set and must have the same column assignment in every set.
4. **Drive every input.** Even when tests only exercise a portion of the circuit, every input should have some known value. It may be a constant 1 or 0, but it should not be left unknown.
5. **Use only permitted waveforms.** Figure 1 summarizes the shapes of waveforms that are acceptable to TPG2 (the Lucent Technologies proprietary test program generator) and *Advantest* (the test machines used by Lucent Technologies). Any of the waveforms in Figure 1 can optionally be delayed in a given test period.

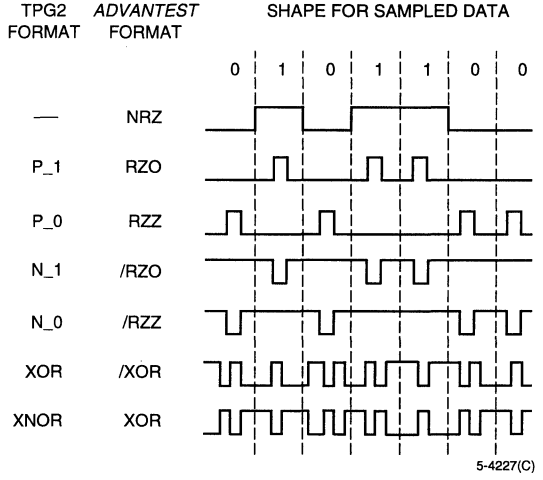


Figure 1. Allowed Stimulus Waveform Shapes

6. **Do not change a signal's waveform within a vector set.** The waveforms described in Figure 1 can be used for any signal. However, in any vector set, each signal may use exactly one waveform shape. This is a limitation of the *Advantest* test machines used by Lucent Technologies.
7. **Do not change a signal's waveform timing within a vector set.** The waveforms described in Figure 1 can be timed or delayed with parameters. However, in any vector set, each signal may use exactly one timing set. This is a limitation of the test generation program, TPG2.
8. **Do not use pulses of less than 10 ns.** No pulses on any stimulus should be shorter than 10 ns in duration because an *Advantest* test machine cannot reliably create pulses shorter than this.
9. **Use no more than 19 input edges in a clock period.** *Advantest* testers share timing generators between pins. There are only 19 available timing generators. Therefore, the parameters which delay and time the waveforms in the above must be limited to 19 in any vector set.

ORCA FPGA Migration to MACO or a Lucent Technologies Gate Array

(continued)

10. **Use no more than 2 bidirectional driver enables.** *Advantest* testers devote two special timing generators to time when bidirectional pins are switched between input and output modes. Bidirectional turnaround can therefore occur at only two different places in a vector period. One of these times is usually the vector boundary.
11. **Maintain fault coverage of 90% (required) to 95% (recommended) or better.** Stimuli must employ the high-fault coverage BIST patterns on memory blocks. It is also important to use **all** of the same stimuli during device testing that are used for in-circuit or board testing.

Waveform Auditing

Auditing the waveforms serves two purposes:

1. Checking compliance with test vector guidelines.
2. Recording signal relationships and timing parameters.

Lucent Technologies encourages test vector development methods that are correct by construction. This means that vector development test beds, by their very construction, should only generate waveforms that satisfy the audits.

Waveform Audits

Waveforms should be audited for many of the guidelines discussed in the previous section. In addition, output and control waveforms need to be audited. (Each goal below applies to every vector set individually, unless otherwise noted.)

1. Input waveform audits:
   — Every input driven?
   — Used only permitted waveforms?
   — Did not change waveforms?
   — Did not change waveform timing?
   — Maximum number of different edges in period N less than or equal to 19?
2. Output audits:
   — Masked strobes in transition regions?
   — Strobed only during stable time common to worst-case slow delay and unit delay?
   — Every output drives steady 0 and steady 1 in at least one vector set?
   — Did not change signal strobe times in period?
3. Bidirectional/3-state control audits:
   — Every bidirectional pin has a corresponding control vector/signal?
   — Maximum number of bidirectional driver enable timings less than or equal to 2?
   — At least 15 ns between strobing of bidirectional and driving to input value?
   — Avoided opposite state contention between DUT and test machine?
   — Every 3-statable output pin has a corresponding control vector/signal?
   — Every 3-statable output pin goes Hi-Z in at least one vector set?

To perform the audits above manually, waveforms can be examined in the waveform display tools supplied by many third-party CAD companies. Auditing the waveforms before long simulation verifications is usually worthwhile.

Functional vectors should be simulated at system speeds to ensure the proper operation of the chip, but actual device testing will be performed at 10 MHz or the system speed, whichever is less.

Masked Array Conversion for *ORCA* ("MACO")

Features

- Seamless conversion from *ORCA* FPGA to mask-programmed devices for fast time-to-volume production
- Reduced cost for high-volume applications
- Same 0.5 µm/0.35 µm process as corresponding *ORCA* devices
- Same 5 V/3.3 V options as *ORCA* devices
- Same I/O pin buffer characteristics as corresponding *ORCA* devices
- Same packages/pinouts as corresponding *ORCA* devices
- MACO meets or exceeds all *ORCA* performance specifications
- Same preference file used to control *ORCA* Foundry map/place/route, and static timing is used to specify timing for the MACO device
- Greater than 50% reduction in power required for MACO device vs. an *ORCA* FPGA
- Provides *ORCA* pin emulation for most programming pins
- Conversion made directly from *ORCA* Foundry output includes any and all design changes
- Easy conversion regardless of design flow
- Internal scan path logic allows full fault coverage
- Fault coverage vectors automatically generated
- Allows the user to prove-in their system with *ORCA* FPGAs before committing to high-volume production

Description

Masked Array Conversion for *ORCA* (MACO) is an application-specific, mask-programmed array implementation that provides a fast and easy migration path from any *ORCA* FPGA to low-cost, high-volume production. Lucent Technologies provides both the *ORCA* FPGA and the MACO array which means that the I/O buffer characteristics are identical, the parts are manufactured using the same foundry, and they can be placed in the same packages. It also means there is only one supplier with whom the designer must interface.

The transfer of a design from *ORCA* to MACO is as simple as providing the output files generated by *ORCA* Foundry. The user first implements a working *ORCA* design that has been system-verified. For a MACO migration to begin, the simple pre-hand-off procedure shown in Figure 2 is required. The post-hand-off procedure shown in Figure 3 is then performed by the MACO team.

Additional checks are performed by the MACO team to ensure that no race conditions exist and that asynchronous behavior (if any) is understood. With this input, the MACO team can process the design, providing evaluation parts in a matter of weeks.

Table 1. Lucent Technologies' *ORCA* FPGA Migration-Capable Devices

Device*	84-Pin PLCC	100-Pin TQFP	144-Pin TQFP	160-Pin QFP	208-Pin EIAJ SQFP/ SQFP2	240-Pin EIAJ SQFP/ SQFP2	256-Pin PBGA	304-Pin EIAJ SQFP/ SQFP2	352-Pin PBGA	428-Pin EBGA	600-Pin EBGA
2x04	X	X	X	X	X	—	—	—	—	—	—
2x06	X	X	X	X	X	X	X	—	—	—	—
2x08	X	—	—	X	X	X	X	—	—	—	—
2x10	X	—	—	X	X	X	X	—	X	—	—
2x12	X	—	—	—	X	X	X	X	X	—	—
2x15	X	—	—	—	X	X	X	X	X	X	—
2x26	—	—	—	—	X	X	—	X	X	X	X
2x40	—	—	—	—	X	X	—	X	—	X	X

* Includes the devices in the ATT2Cxx, OR2CxxA, and OR2TxxA series FPGA families.

Note: The OR2TxxA series is not offered in the 304-pin SQFP/SQFP2 packages.

Masked Array Conversion for *ORCA* ("MACO") (continued)

Pre T = 0 Design Process

In order to convert an *ORCA* FPGA to a MACO device, the starting point is a design database of an *ORCA* FPGA that is currently working in the targeted system. As shown in Figure 2, this database consists of three main files that are generated by *ORCA* Foundry:

1. The .ncd design database,
2. The .prf preference file, and
3. The .pad pinout file.

As shown in Figure 2, functional test vectors are also preferred. This provides the highest level of confidence that the migrated FPGA will work in the board the first time it is plugged-in as a MACO device, the reasons having been discussed in detail previously.

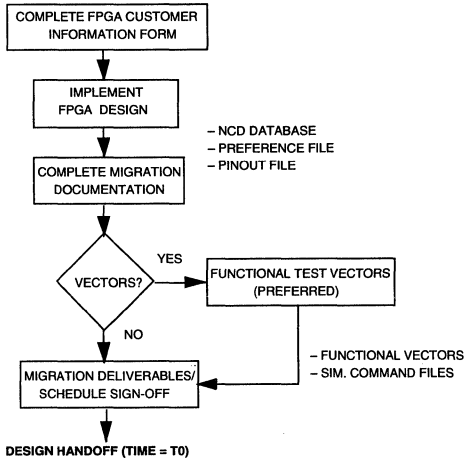


Figure 2. Pre-Hand-Off Migration Flow

In addition to these files, two documents must also be completed before design hand-off (known as T = 0 or Time = T0):

1. **Migration Checklist**: A checklist that ensures that all required information is transferred to the Lucent Technologies design team.
2. **Hand-Off List**: A questionnaire about the FPGA design and customer contacts.

Post T = 0 Design Process

After receiving the final design database and other documentation, a detailed schedule with both customer and Lucent deliverables will be presented for sign-off by the customer and a Lucent representative.

As shown in Figure 3, the design is then converted to a MACO device and verified. Once this verification is complete, a customer review will be performed to make sure the MACO device meets all of the customer's requirements. If no issues are found, then the customer will be required to sign a mask order request form, known as the Design Verification Form, which will begin the fabrication process.

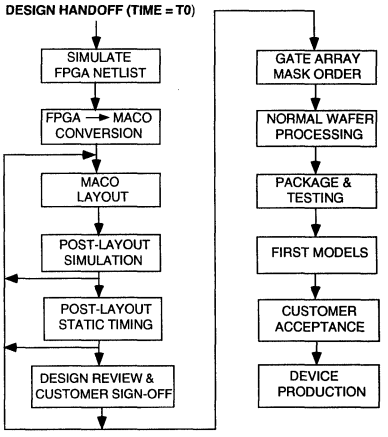


Figure 3. Post-Hand-Off Migration Flow

Once fabrication has finished and the devices have been packaged and tested, the first models will be shipped to the customer. The customer then signs a Final Design Approval Form, which permits the creation of production devices.

All of the forms that are required during this process are included in the Appendix of this manual (note that these forms are occasionally updated, so please see your Lucent representative for the latest version):

Appendix A: MACO Migration Checklist
Appendix B: MACO Hand-Off List
Appendix C: MACO Design Verification Form
Appendix D: MACO Final Design Approval Form

Masked Array Conversion for *ORCA* ("MACO") (continued)

Hand-Off Options

There are three hand-off options that can be selected during a migration:

1. **Preferred Option.** This option includes functional test vectors supplied by the designer. Once a migration feasibility study has been completed, there are no restrictions for this option.
2. **Base-Line Option.** This option does not include functional test vectors. Contact your Lucent representative to determine if there are any restrictions for this option.
3. **Special Requests Option.** Special requests may be accommodated on a case-by-case basis. Some examples include special configuration emulation or the combining of multiple *ORCA* FPGAs into one MACO device.

For each of these options, the following caveats must be observed:

1. Evaluation of programming circuits (DONE pin, etc.). Some understanding of how these are used in the system is required when migrating. Although emulation of many of the programming pins is an option, the exact timing or functionality of the FPGA programming pins may not be duplicated in the MACO device. See the Emulation of *ORCA* Programming Pins section for more information.
2. Limited support for initial value RAMs.
3. If any FPGA in a programming daisy chain is migrated, then all the following FPGAs in the daisy chain must also be migrated or programmed in a different manner.
4. Readback mode cannot be supported in the MACO device.

Boundary Scan and Testability

Boundary Scan

MACO migration supports the standard boundary scan instructions required by *IEEE* standards: extest, sample, and bypass. The MACO instruction register has these commands located at the same addresses as our *ORCA* FPGAs. Both preconfiguration and postconfiguration boundary-scan modes are emulated in MACO. Any vectors that previously ran in either of these two modes will work in the MACO device. As a result, customers do not have to change any of the board assembly tests that utilize these instructions when migrating to MACO.

However, some of the features that are supported in *ORCA* are not logically supported in MACO. These features are RAM Write (the device cannot be programmed through the TAP controller) and RAM Read (the programming information cannot be "read back" through the TAP controller).

In addition, there are two user-defined instructions that *ORCA* supports but that are not supported by MACO; they are PLC Scan Ring 1 and PLC Scan Ring 2. Table 2 summarizes the MACO/*ORCA* boundary-scan instructions.

Table 2. MACO/*ORCA* Boundary-Scan Instructions

Instruction	Address	Supported in *ORCA*?	Supported in MACO?
Extest	000	Yes	Yes
PLC Scan Ring 1	001	Yes	No
RAM Write (RAM_W)	010	Yes	No
Reserved	011	—	—
Sample	100	Yes	Yes
PLC Scan Ring 2	101	Yes	No
RAM Read (RAM_R)	110	Yes	No
Bypass	111	Yes	Yes

Masked Array Conversion for *ORCA* ("MACO") (continued)

Automatic Test Program Generation (ATPG)

To reduce device test issues to an acceptable level, Lucent Technologies created fault coverage vectors for each MACO device to bring fault coverage up to the mid- to high-90th percentile range. These vectors are generated using ATPG tools. Most migrated *ORCA* designs will be able to have an ATPG vector generator run on the MACO counterpart to generate these vectors. This is done without the added overhead of scan insertion or other structured circuit methodologies to help increase fault coverage.

Scan Insertion

In some cases, the ATPG tool may not be able to raise the fault coverage high enough without adding structured test circuitry to the chip in question. This is due to the complexity of the circuit or its size. In these cases, Lucent Technologies will add full scan circuitry to achieve the desired fault coverage.

RAM BIST

Designs that include RAMs will have RAM BIST circuitry inserted in them to accommodate the fault coverage of the RAMs.

Functional Vectors

Any functional vectors that the customer provides need to meet the guidelines discussed in the Vector Writing Guidelines section. This will ensure test set compatibility and will allow Lucent to run these vectors on the test set. Exercising these vectors on the test set will generally ensure that the paths exercised are operating correctly (at speed) in the silicon. Fault coverage vectors generated by Lucent will not test asynchronous interface nor tight timing paths and will not be run at speed.

Verification

Functional Simulations

The MACO methodology includes simulating all functional vectors provided in the T = 0 hand-off package as long as they meet the guidelines set out in the Vector Writing Guidelines section. In addition, they need to be created in one of the following platforms: *Verilog*, VHDL, *Viewlogic* (PC or Workstation), Mentor, or ATTSIM. All MACO simulations will be done in *Verilog-XL* as the golden simulator.

Static Timing Analysis

A static timing analysis of every path on the chip is also performed independently from any simulations completed. This analysis is performed with postlayout capacitance and RC delay back-annotated into the analysis database. The results are then provided to the customer prior to mask order sign-off for final approval. These results include every possible input setup and hold time, every possible output propagation delay path, and the frequency analysis (maximum frequency) of every clock domain. This allows customers who do not provide complete functional vectors to check the performance of every path.

Clock Issues

Prior to layout, an analysis is done of each clock on the chip to allow the design and insertion of an appropriate-sized clock tree to ensure that clock slope on the chip is appropriate to prevent problems with individual flip-flop operation and also to ensure that required clock-to-output delay is met. After layout is complete, every clock on the chip is analyzed to ensure that the maximum skew requirement imposed by the technology is met.

Preference File

The customer preference file provided at T = 0 hand-off is verified in the completed *ORCA* .ncd database that is also provided at T = 0. If all preferences are met, then these preferences are checked in the final simulation and static timing analysis results to ensure proper timing on the critical paths specified. **Any preference not met in the *ORCA* device will be documented and ignored.**

Masked Array Conversion for *ORCA* ("MACO") (continued)

Emulation of *ORCA* Programming Pins

Migrated MACO devices contain circuitry to emulate the functionality of many *ORCA* programming pins. These pins are emulated during all states of operation: powerup, initialization, configuration, start-up, and user operation. Since the MACO device is not actually configured with a bit stream, the order of these events is maintained, but the time it takes to complete each state of operation is different for the MACO device than for the original *ORCA* device.

At powerup, the device goes through an initialization process. First, an internal power-on-reset circuit is triggered when power is applied. A time-out delay is initiated to allow the power supply voltage to stabilize. The $\overline{\text{INIT}}$ and DONE outputs are low. At powerup, if V_{DD} does not rise from 2.0 V to V_{DD} in less than 25 ms, the user should delay configuration by inputting a low into $\overline{\text{INIT}}$, $\overline{\text{PRGM}}$, or $\overline{\text{RESET}}$ until V_{DD} is greater than the recommended minimum operating voltage.

The active-low, open-drain initialization signal, $\overline{\text{INIT}}$, is released and pulled high by an internal resistor when initialization is complete. If $\overline{\text{INIT}}$ is held low, the MACO device remains in the initialization state. When $\overline{\text{INIT}}$ is released to go high, the MACO device enters the configuration state.

If configuration has begun, an assertion of $\overline{\text{RESET}}$ or $\overline{\text{PRGM}}$ initiates an abort, returning the MACO device to the initialization state. The $\overline{\text{PRGM}}$ and $\overline{\text{RESET}}$ pins must be pulled back high before the FPGA will enter the configuration state. During the start-up and operating states, only the assertion of $\overline{\text{PRGM}}$ causes the device to return to the initialization state.

All I/Os that are not used during the initialization and configuration states are 3-stated with internal pull-ups, and the internal latches/FFs are held set/reset. After configuration, the MACO device enters the start-up, or transition, phase between the configuration and operational states. The system design issue in the start-up phase is to ensure the user I/Os become active without inadvertently activating devices in the system or causing bus contention. A second system design concern is the release of the global set/reset of the internal latches/FFs.

In the default implementation of the MACO devices, both of these events (the release of the I/O 3-states and the release of the internal set/reset) occur when the external DONE pin rises. Done is an open-drain bidirectional pin that may include an optional (enabled by default) pull-up resistor to accommodate wired ANDing. The open-drain DONE signals from multiple devices (both FPGAs and MACOs) can be tied together (ANDed) with a pull-up (internal or external) and used as an active-high ready signal or reset to other parts of the system. These ANDed DONE pins can be used to synchronize the other two start-up events for all devices, since they can be synchronized to the same external signal. This external DONE signal will not rise until all devices release their DONE pins, allowing the signal to be pulled high.

If any of the other start-up options are required, this is considered a special request that must be submitted on the *MACO Migration Checklist* (see Appendix A). Please contact your Lucent representative for a copy of this document and for more information.

Another function that is emulated in the MACO devices is the TS_ALL function, which functions in exactly the same manner as in the *ORCA* devices to 3-state all I/Os under user control. Additional information on this function is available in any of the *ORCA* Series FPGAs data sheets.

In summary, the following pins and their associated functionality are emulated in the MACO device: $\overline{\text{PRGM}}$, $\overline{\text{RESET}}$, $\overline{\text{INIT}}$, DONE, HDC, and $\overline{\text{LDC}}$. In addition, the functionality of 3-stating I/Os, the set/reset of latches/FFs during initialization/configuration, and the TS_ALL function may also be emulated.

Migration of RAMs from *ORCA* to MACO

Asynchronous RAMs

The asynchronous RAMs found in all *ORCA* devices require that complicated timing requirements be met to ensure operation. These requirements include minimum pulse widths on the write-enable signal and setup/hold checks from data and address to the write-enable signal. To ensure correct operation, both static timing analysis and timing simulation is run for both WCS and WCF processing with these timing checks included. RAM BIST is also included to ensure high fault coverage.

Synchronous RAMs

Since the new synchronous RAMs found in the OR2CxxA/OR2TxxA devices provide a simpler interface that is synchronous to a given clock, no special testing is required to ensure operation. RAM BIST is included to ensure high fault coverage.

Masked Array Conversion for *ORCA* ("MACO") (continued)

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 operations sections of this data sheet. Exposure to absolute maximum ratings for extended periods can adversely affect device reliability.

MACO devices include circuitry designed to protect the chips from damaging substrate injection currents and prevent accumulations of static charge. Nevertheless, conventional precautions should be observed during storage, handling, and use to avoid exposure to excessive electrical stress.

Parameter	Symbol	Min	Max	Unit
Storage Temperature	T_{stg}	–65	150	°C
Supply Voltage with Respect to Ground	V_{DD}	–0.5	7.0	V
V_{DD5} Supply Voltage with Respect to Ground (3.3 V MACO)	V_{DD5}	V_{DD}	7.0	V
Input Signal with Respect to Ground	—	–0.5	V_{DD} + 0.3	V
Signal Applied to High-impedance Output	—	–0.5	V_{DD} + 0.3	V
Maximum Soldering Temperature	—	—	260	°C

Note: During powerup and powerdown sequencing, V_{DD} is allowed to be at a higher voltage level than V_{DD5} for up to 100 ms.

Recommended Operating Conditions

Mode	5 V MACO		3.3 V MACO		
	Temperature Range (Ambient)	Supply Voltage (V_{DD})	Temperature Range (Ambient)	Supply Voltage (V_{DD})	Supply Voltage (V_{DD5})
Commercial	0 °C to 70 °C	5 V ± 5%	0 °C to 70 °C	3.0 V to 3.6 V	V_{DD} to 5.25 V
Industrial	–40 °C to +85 °C	5 V ± 10%	–40 °C to +85 °C	3.0 V to 3.6 V	V_{DD} to 5.25 V

Note: The maximum recommended junction temperature (T_J) during operation is 125 °C.

Masked Array Conversion for *ORCA* ("MACO") (continued)

Electrical Characteristics

Table 3. Electrical Characteristics

5 V Commercial: V_{DD} = 5.0 V ± 5%, 0 °C ≤ T_A ≤ 70 °C; 5 V Industrial: V_{DD} = 5.0 V ± 10%, –40 °C ≤ T_A ≤ +85 °C.
3.3 V Commercial: V_{DD} = 3.0 V to 3.6 V, 0 °C ≤ T_A ≤ 70 °C; 3.3 V Industrial: V_{DD} = 3.0 V to 3.6 V, –40 °C ≤ T_A ≤ +85 °C.

Parameter	Symbol	Test Conditions	5 V MACO		3.3 V MACO		Unit
			Min	Max	Min	Max	
Input Voltage: High Low	 V_{IH} V_{IL}	Input configured as CMOS (Includes 3.3 V MACO)	 70% V_{DD} GND – 0.5	 V_{DD} + 0.3 20% V_{DD}	 80% V_{DD} GND – 0.5	 V_{DD} + 0.3 15% V_{DD}	 V V
Input Voltage: High Low	 V_{IH} V_{IL}	Input configured as TTL (Not valid for 3.3 V MACO)	 2.0 –0.5	 V_{DD} + 0.3 0.8	 — —	 — —	 V V
Output Voltage: High Low	 V_{OH} V_{OL}	 V_{DD} = Min, I_{OH} = 6 mA or 3 mA V_{DD} = Min, I_{OL} = 12 mA or 6 mA	 2.4 —	 — 0.4	 2.4 —	 — 0.4	 V V
Input Leakage Current	I_L	V_{DD} = Max, V_{IN} = VSS or V_{DD}	–10	10	–10	10	µA
Standby Current	I_{DDSB}	5 V MACO (T_A = 25 °C, V_{DD} = 5.0 V) 3.3 V MACO (T_A = 25 °C, V_{DD} = 3.3 V) no output loads, inputs at V_{DD} or GND	—	4.1	—	2.7	mA
Data Retention Voltage	V_{DR}	T_A = 25 °C	2.3	—	2.3	—	V
Input Capacitance	C_{IN}	5 V MACO (T_A = 25 °C, V_{DD} = 5.0 V) 3.3 V MACO (T_A = 25 °C, V_{DD} = 3.3 V) Test frequency = 1 MHz	—	9	—	9	pF
Output Capacitance	C_{OUT}	5 V MACO (T_A = 25 °C, V_{DD} = 5.0 V) 3.3 V MACO (T_A = 25 °C, V_{DD} = 3.3 V) Test frequency = 1 MHz	—	9	—	9	pF
DONE Pull-up Resistor	R_{DONE}	—	100K	—	100K	—	Ω
M3, M2, M1, and M0 Pull-up Resistors	R_M	—	100K	—	100K	—	Ω
I/O Pad Static Pull-up Current	I_{PU}	V_{DD} = 5.25 V, V_{IN} = VSS, T_A = 0 °C	14.4	50.9	14.4	50.9	µA
I/O Pad Static Pull-down Current	I_{PD}	V_{DD} = 5.25 V, V_{IN} = V_{DD}, T_A = 0 °C	26	103	26	103	µA
I/O Pad Pull-up Resistor	R_{PU}	V_{DD} = 5.25 V, V_{IN} = VSS, T_A = 0 °C	100K	—	100K	—	Ω
I/O Pad Pull-down Resistor	R_{PD}	V_{DD} = 5.25 V, V_{IN} = V_{DD}, T_A = 0 °C	50K	—	50K	—	Ω

Masked Array Conversion for *ORCA* ("MACO") (continued)

Table 4. Power Consumption

Parameter	Test Conditions	5 V MACO (Max)	3.3 V MACO (Max)	Unit
Internal Cell	Fan-out = 2	3.70	1.57	µW/MHz
Flip-Flop Power	Fan-out = 1	9.60	4.00	µW/MHz
Input Block (per pin)	P_{CMOS} = P_{TTL} = 1.18 mW +	0.20 0.20	0.09 0.09	mW/MHz mW/MHz
Output Block (per pin)	C_L = 15 pF	132.00	57.50	µW/MHz

Preventing FPGA Migration Timing Issues

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. 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.

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 manual.

Figure 4 shows an example of one of the many different race conditions that can occur. In the FPGA, the gated clock's AND gate has one input with 50 ns of routing delay; the other has 10 ns of routing delay. The intervals before and after the routing delay for one of the inputs are shown as A and A'. Likewise, the intervals before and after the routing delay for the other input are B and B'.

Even though A's routing delay is 50 ns, a glitch doesn't occur at the output and the chip is not clocked. The reason is that one or the other of the inputs (A' or B') to the AND gate is always low.

However, in a gate array or hard-wired FPGA, those same delays could both become 5 ns. Instead of a 50 ns delay at A' and a 10 ns delay at B', the gate array timing will have both A' and B' high at the same time since both now have 5 ns routing 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 have similar characteristics, but each problem has its own unique variations that adversely affect a gate array conversion.

The following examples show that the FPGA routing delays can be considerably larger than those of gate arrays and can easily be larger than the gate delays in a typical signal path.

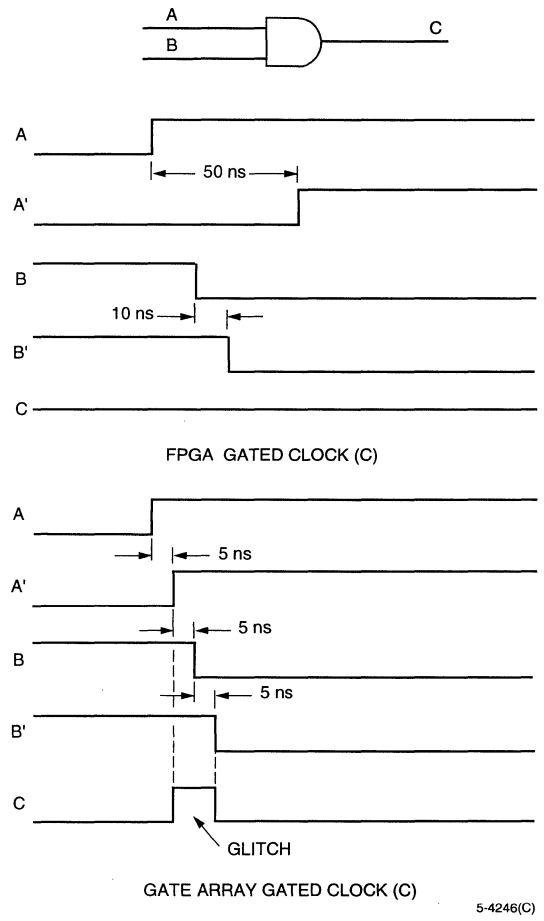
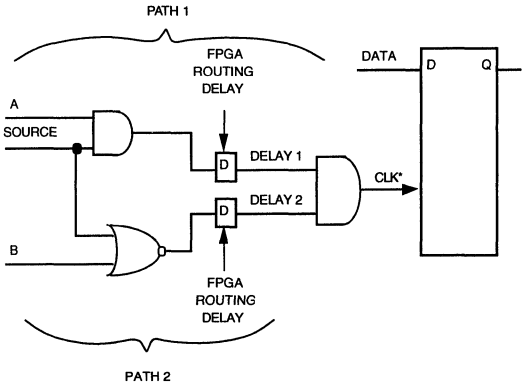


Figure 4. Example Race Conditions

Preventing FPGA Migration Timing Issues (continued)

Gated Clocks

Gated clocks present many issues during FPGA to gate array migrations. Figure 5 shows a gated clock that can take two different paths to get to the clock input on the flip-flop. 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.

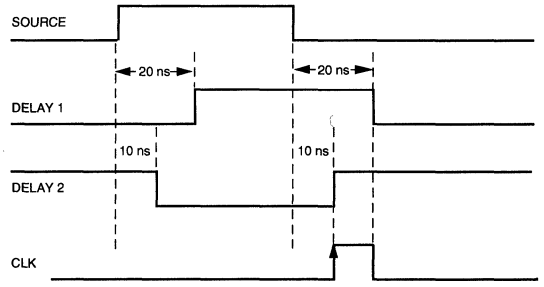


* CLK = (A * SOURCE) * (~ (B + SOURCE)). If A = 1 and B = 0, CLK can be affected by both Path 1 and Path 2.

Figure 5. Gated Clock

As shown in Figure 5, Path 1 and Path 2 have different routing delays, as modeled by the shown delay cells. The nets labeled DELAY 1 and DELAY 2 follow the delay cells and are used in the following diagrams. If Input A is 1 and Input B is 0, the clock can be affected by both Path 1 and Path 2 when the SOURCE input changes.

In the waveform shown in Figure 6, the FPGA design has a Path 1 delay of 20 ns and a Path 2 delay of 10 ns. This results in a 10 ns wide pulse on the falling edge of SOURCE for the CLK signal.

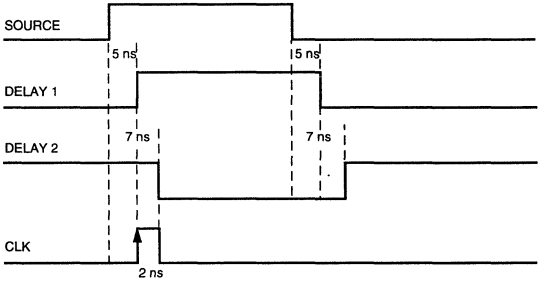


Note: DELAY 1 = 20 ns and DELAY 2 = 10 ns; therefore, there will be a 10 ns clock pulse on the falling edge of SOURCE.

Figure 6. Gated Clock—FPGA Design

However, in the gate array implementation shown in Figure 7, the Path 1 delay becomes 5 ns and the Path 2 delay becomes 7 ns. In effect, the two have skewed in opposite directions with Path 1 now faster than Path 2.

The gate array therefore experiences a 2 ns wide pulse on the rising edge of SOURCE, possibly clocking the flip-flop. Since the flip-flop will be clocked on the opposite edge of SOURCE, the appropriate data may not be at the data pin, and therefore will not be latched.



Note: DELAY 1 = 5 ns, DELAY 2 = 7 ns. The gate array now has an unwanted glitch on the rising edge of SOURCE, while the pulse from the falling edge of SOURCE is gone.

Figure 7. Gate Array Implementations

Preventing FPGA Migration Timing Issues (continued)

Figure 8 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 SOURCE and goes to the input of the OR gate. These two paths could conceivably have 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 is not necessarily the case with an FPGA.

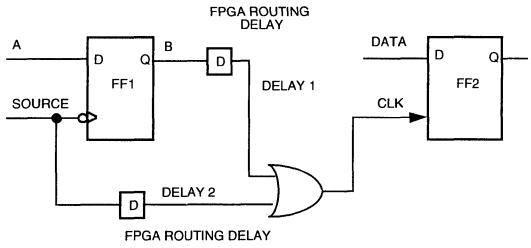


Figure 8. Clock with Reconvergent Sources

To explain, assume that in the waveform in Figure 9 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 ORed together, when SOURCE falls in the FPGA implementation, one of the inputs to the OR gate is always high. Therefore, there is no glitch on the falling edge of SOURCE.

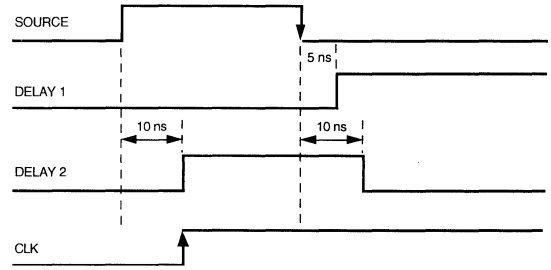


Figure 9. SOURCE Clock without Glitch—FPGA Design

As shown in Figure 10, this may not be true for the gate array design. Figure 10 shows DELAY 1 remaining at 5 ns and DELAY 2 reduced to 2 ns. Since DELAY 1 is now larger than DELAY 2, the clock going to flip-flop 2 will glitch on the falling edge of SOURCE.

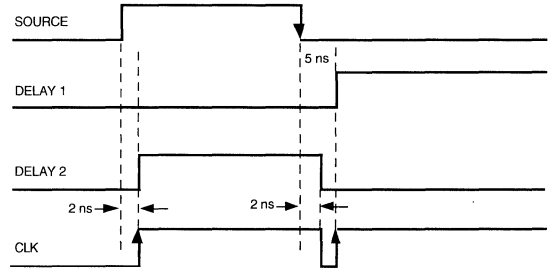


Figure 10. SOURCE Clock with Glitch—Gate Array Design

Preventing FPGA Migration Timing Issues (continued)

Clock/Data Races

Figure 11 shows an asynchronous circuit involving flip-flops having different clocks and three different routing 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 routing 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.

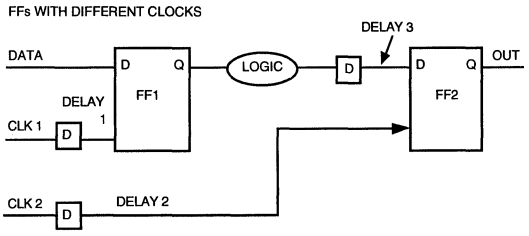


Figure 11. Clock/Data Race

The waveforms for an example FPGA implementation and gate-array implementation are shown in Figure 12 and Figure 13. Since CLK2's delay (DELAY 2) in the gate array is speeded up, data to flip-flop 2 does not get there in time to be clocked by DELAY 2, and OUT remains a 0 instead of going high, which is what happens in the FPGA.

Also, asynchronous presets and clears pose the same problems as the clocks on flip-flops, as explained in Figures 2—9. 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.

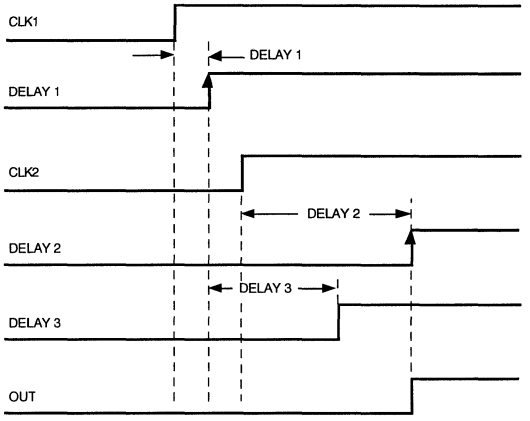


Figure 12. Clock/Data Race FPGA Design

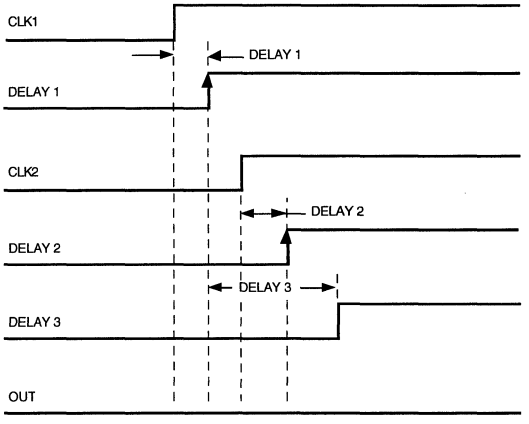


Figure 13. Clock/Data Race Gate Array Design

Preventing FPGA Migration Timing Issues (continued)

Figure 14 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.

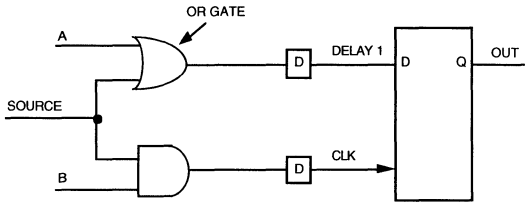


Figure 14. Mixed Data and Clock

For the FPGA implementation, Figure 15 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 16, 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.

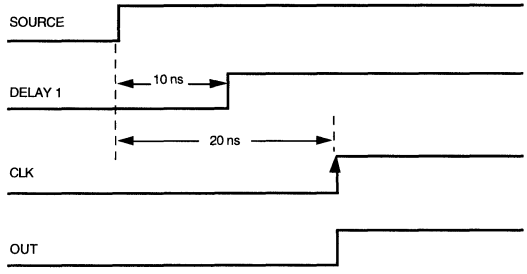


Figure 15. Mixed Data and Clock FPGA Design

Other Timing Issues

Routing delay problems can also affect the timing of asynchronous loops used in an FPGA, which are generally included in a circuit for their timing characteristics. Sometimes asynchronous loops are added as ring oscillators, as timing chains, or as long delay paths to get FPGAs to function properly. 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.

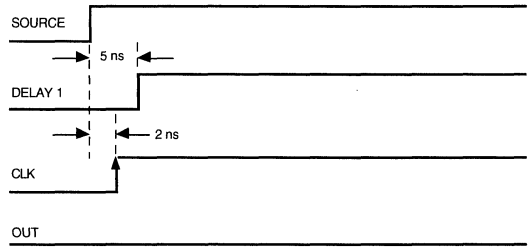


Figure 16. Mixed Data and Clock Gate Array Design

Conclusion

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 generally 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 array working the first time. The test vectors from these simulations can then be used for testing the gate array during manufacturing.

Notes

Lucent Technologies
Bell Labs Innovations

Masked Array Conversion for *ORCA* ("MACO") Migration Checklist (Version 2.0)

Item	Format
_____ A. Pinout Definition	Use **file.pad** from *ORCA* Foundry
_____ B. Design Description	Completed **handoff.lis** including device environment (voltage, package, frequency, clock strategy, etc.)
_____ C. Design Database	Use **file.ncd** from *ORCA* Foundry
_____ D. Functional Vectors (optional but preferred)	Many industry-standard formats are supported (with simulation command files)

General Information Requirements

1. If vectors are supplied, do all vector sets meet the test set guidelines as specified in the Migration Section of the FPGA Data Book as required? Yes ___ No ___ If no, **please attach an explanation.**
2. What platform did you use for design entry? _____
   What platform did you use for synthesis? _____
   What platform did you use for simulation? _____
3. Is the RESET pin used to set/reset all PLC FFs globally:
   During user operation? Yes ___ No ___ Is it active-high _____ or active-low ____?
4. Is the $\overline{\text{RESET}}$ pin used to reinitialize the FPGA during configuration? Yes ___ No ___
5. Are any programmable functions used? Yes ___ No ___
   If yes, which ones:
   CCLK* _____ DONE _____ Start-up† _____
   HDC _____ $\overline{\text{INIT}}$ _____ $\overline{\text{RESET}}$ _____
   LDC _____ Readback _____
   RD_CFGN as TS_ALL after configuration _____
6. Do any of these functions need to be emulated in the migrated device? Yes ___ No ___
   If yes, which ones:
   CCLK* _____ DONE _____ Start-up† _____
   HDC _____ $\overline{\text{INIT}}$ _____ $\overline{\text{RESET}}$ _____
   LDC _____ Readback _____
   RD_CFGN as TS_ALL after configuration _____
7. Is the TS_ALL function driven by an internal signal? Yes ___ No ___
8. Does the DONE pin require an internally supplied pull-up? Yes ___ No ___
9. Was the oscillator used, and is it needed in the masked array? Yes ___ No ___
   If so, which speed? 10 MHz nominal ___ 1.25 MHz nominal ___
10. Were internal 3-state bus drivers used? Yes ___ No ___
    Are these buses always driven as required in *ORCA*? Yes ___ No ___

* Used only for programming master mode/daisy-chained devices.
† Is the sequence or timing of start-up events important?

11. Are any PLCs used as delay elements? Yes ___ No ___
12. Are routing delays intentionally added in the FPGA? Yes ___ No ___
13. Were any paths routed by hand for timing requirements? Yes ___ No ___
    If so, **attach a list of these paths** if they need priority in the masked array.
14. Do any signals that had to be given routing priorities in the FPGA need the signal names maintained in the masked array implementation? Yes ___ No ___
    If so, **attach a list of these signals.**
15. Is the boundary-scan circuitry used? Yes ___ No ___
    Is it needed in the masked array? Yes ___ No ___
    If yes, preconfiguration _____ or postconfiguration _____?
    Are the internal user scan paths used? Yes ___ No ___
    Are the dedicated boundary-scan pins used for another purpose after configuration? Yes ___ No ___
16. Do you have any minimum output propagation delay specifications? Yes ___ No ___
    If yes, **please provide a list of the pins and their specifications.**
17. What is the top level circuit name? _____
18. Are there any LUTs implemented as RAM? Yes ___ No ___
    If so, list the number of RAMs and their sizes.

    _____ _____

    _____ _____

    _____ _____
19. Is this device a master _____, middle slave _____, or trailing slave _____?
20. Are there any required relationships between multiple clock signals (*Remember*, any signal that clocks a PFU is considered a separate clock signal) as described in the Migration Section of the FPGA Data Book? Yes ___ No ___
    If yes, can this relationship be described as the following: CLK A will occur a minimum of ___ ns and a maximum of ___ ns before CLK B?

Name _____

Company _____

Date _____

microelectronics group

Lucent Technologies
Bell Labs Innovations

Masked Array Conversion for *ORCA* ("MACO") Hand-Off List (Version 1.0)

Device Name: _____

FPGA Device (e.g., ATT2C15): _____

Package (e.g., 160 EIAJ): _____

Customer Design Contact: _____

Phone: _____

Lucent Design Contact: _____

Phone: _____

Number of models contracted for: _____

Name and address to ship models to: _____

_____

_____

_____

1. Design Type (select all that apply):

   ____ *ORCA* 1C FPGA Drop-In Replacement (pin-pin compatible)
   ____ *ORCA* 2C FPGA Drop-In Replacement (pin-pin compatible)
   ____ ATT3000 FPGA Drop-In Replacement (pin-pin compatible)
   ____ *ORCA* 2T FPGA Drop-In Replacement (pin-pin compatible)
   ____ Board Consolidation/Cost Reduction (contact Lucent Technologies for availability)
   ____ Other Migration

2. One Paragraph Description of Design Intent and Device Function:

   _____
   _____
   _____
   _____
   _____

3. Device Operating Conditions:

   Power Supply Voltage: _____ Vdc ± _____%
   Maximum Power Supply Current: _____ mA or Maximum Power Dissipation: _____ mW
   Minimum Ambient Temperature: _____ °C
   Maximum Ambient Temperature: _____ °C
   Minimum Junction Temperature: _____ °C
   Maximum Junction Temperature: _____ °C

Note: If there is not enough space to complete a section, please attach additional pages and make a note in the appropriate section.

4. Required Hand-off Items:

   The following items must be attached:

   a. The completed *ORCA* Customer Information form (this form will be completed by Lucent FPGA marketing).
   b. The completed migrate.chk form (to be completed by the customer).

   What version of *ORCA* Foundry was used? _____

	File Name	Version/Date
ORCA Foundry NCD database file (.ncd):	_____	_____
ORCA Foundry preference file (.prf):	_____	_____
ORCA Foundry pin assignment file (.pad):	_____	_____
Functional vector set(s):	_____	_____
	_____	_____
	_____	_____
	_____	_____
	_____	_____
Simulation command file(s):	_____	_____
	_____	_____
	_____	_____
	_____	_____
	_____	_____

5. Are there any changes/additions to the preference file timing? Yes ____ No ____

   (For example, path delays, clock frequencies, I/O setup time, I/O hold time, I/O propagation delays, and signal offsets.) All timing preferences must have been met in the original FPGA; i.e., they must pass *ORCA* TRACE:

   If so, list them below (must be in *ORCA* Foundry preference format):

	Mod #	Timing Specification
1.	_____	_____
2.	_____	_____
3.	_____	_____
4.	_____	_____
5.	_____	_____

6. Simulation Vectors for Bidirectional Pins:

   Please list for all bidirectional buffers the corresponding control column in the vector set that indicates when the buffer is in INPUT mode. (Note: Only 1s or 0s may appear in the control columns.)

Bidirectional Output(s)	CLMN # (s)	Control CLMN #	Logic Value of Control Column when in INPUT Mode
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____

7. Tristatable Pins:

   Please list for all tristatable buffers the corresponding control column in the vector set that indicates when the buffer is tristated. (Note: Only 1s or 0s may appear in the control columns.)

Tristate Output(s)	CLMN # (s)	Control CLMN #	Logic Value of Control Column when in Tristate

8. Clock Definitions:

   List all signals that clock memory elements (latches, FFs, memory WREN).

Clock(s)	Primary Input? (Y or N)	If No, Primary Inputs Clock is Derived from:

9. Oscillator:

   If oscillator cells are included that require external I/O pins, list the pin numbers, pin names, and pin functionality (ATT3000 series only).

Pin #	Pin Name	Pin Functionality

10. Optional Hand-off Items:

    Critical path specification(s) (max 5): (i.e., FFx to FFy)

    Requests for deviation from methodology:

I hereby approve this design for post-T0 work at Lucent Technologies.

Name: _____

Company: _____

Title: _____

Signature: _____

Date: _____

Notes

microelectronics group

Lucent Technologies
Bell Labs Innovations

Masked Array Conversion for *ORCA* ("MACO") Design Verification Form

Company Name _____ **Customer Name** _____

Address _____

City _____ **State** _____ **Zip Code** _____

Telephone (___) ___ – _____ **Customer Part (SIF) Number** _____

Original Database Files:

.NCD File and Time Stamp: _____

Simulation Vectors File and Time Stamps: _____
(If applicable)

Optional Test Circuitry:

Is boundary-scan emulation needed?	[ ] No	[ ] Yes
Is internal scan circuitry present?	[ ] No	[ ] Yes

Simulations Completed Successfully:

Functional Vectors:	[ ] NA	[ ] Unit Delay	[ ] Layout Annotated Timing
Fault Coverage Vectors:	[ ] NA	[ ] Unit Delay	[ ] Layout Annotated Timing
Boundary-Scan Vectors:	[ ] NA	[ ] Unit Delay	[ ] Layout Annotated Timing

Attached Files:

Pinout File Name: _____

I/O Timing Overview File Name: _____

Clock Overview File Name: _____

Preset/Clear Overview File Name: _____

Customer Acceptance:

[ ] I certify that the above listed Simulation Vectors (if any) indicate required functionality of the device in system design.

[ ] I certify that the indicated pinout is correct.

[ ] I certify that the indicated preference file is correct.

[ ] I certify that the indicated NCD file is correct.

[ ] I have reviewed the attached I/O Timing Overview and have determined that there are no issues that will be a problem in the system.

[ ] I have reviewed the attached Clock Overview and have determined that there are no issues that will be a problem in the system.

[ ] I have reviewed the attached Preset/Clear Overview and have determined that there are no issues that will be a problem in the system.

[ ] I authorize Lucent Technologies to start the MACO mask making and fabrication process.

Customer Name _____ **Signature** _____

Date _____

microelectronics group

Lucent Technologies
Bell Labs Innovations

Masked Array Conversion for *ORCA* ("MACO") Final Design Approval Form

Company Name ____________________ **Customer Name** ____________________

Address ____

City ____________________ **State** ____________ **Zip Code** ____________

Telephone (___) ___ – _______ **MACO Part (SIF) Number** ____________________

****

The following information was copied from the *MACO Design Verification Form* (OT96-114FPGA):

Original Database Files:

.NCD File and Time Stamp: ____

Simulation Vectors File and Time Stamps: ______________________________
(If applicable)

Optional Test Circuitry:

Is boundary-scan emulation needed? [ ] No [ ] Yes

Is internal scan circuitry present? [ ] No [ ] Yes

Review Files:

Pinout File Name: ____________________

I/O Timing Overview File Name: ____________________

Clock Overview File Name: ____________________

Preset/Clear Overview File Name: ____________________

Customer Preference File: ____________________

****

Customer Acceptance:

[ ] I have tested models of the above MACO device and have verified that the device meets our requirements.

[ ] I request that Lucent Technologies transfer this design to its manufacturing facility.

Authorized by:

Name _____ **Signature** _____

Title _____ **Date** _____

Phone _____

Multipliers in *ORCA* OR2CxxA/OR2TxxA Series Field-Programmable Gate Arrays

Introduction

This application note describes the new *ORCA* OR2CxxA/OR2TxxA FPGA family multiplier mode and its application to high-speed, parallel multipliers. The add/pass-shift method of parallel multiplication—the basis for the *ORCA* multiplier mode—is outlined, followed by an architectural description of the multiplier mode, implementation issues, and other circuits that can be optimized using the multiplier mode. These concepts are then combined to produce a method by which an n x m bit multiplier can be developed to meet a target operating frequency. For further information on the multiplier mode, refer to the *ORCA* OR2CxxA/OR2TxxA data sheets or contact *ORCA* FPGA technical support.

Multiplier Quick Tutorial

The *ORCA* multiplier mode is based on the add/pass-shift form of parallel multiplier. In short, this type of multiplication sequentially adds the multiplicand to the partial product if the multiplier bit corresponding to the given partial product is a logical one, or passes the partial product without addition if the multiplier bit is a zero.

The new partial product is then shifted right, effecting an implicit multiply by two that maintains the proper power of 2 of the multiplier bit corresponding to that step. The process is sequential in the sense that the LSB of the multiplier operates on the first level, generating a partial product (equal either to the multiplicand or zero for the first level) that is then operated on by the next LSB of the multiplier, and so on. The process may or may not be sequential in reference to a process clock, depending on the amount of pipelining added. The final product of the multiply may occupy up to n + m bits. m + 1 bits of the result will be output from the final multiplier stage (including the carry-out), and n – 1 bits will be the LSBs that have been right-shifted during previous levels. The following example demonstrates 26 x 28 = 728 (decimal).

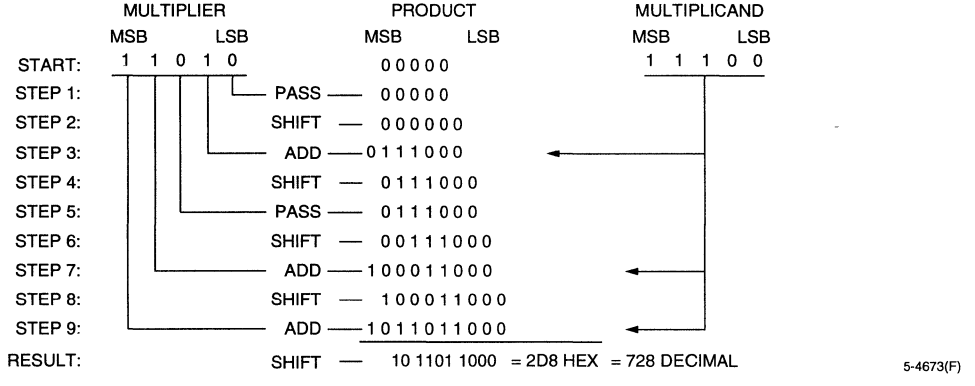


Figure 1. 26 x 28 = 728 (Decimal) Example

ORCA OR2CxxA/OR2TxxA Family Multiplier Mode

Figure 2 is a diagram of an *ORCA* FPGA PFU in multiplier mode. The large box delineates the PFU, the four small boxes represent PFU flip-flops that (optionally) may be used to register the multiplier output, and the box containing the adder and multiplexer symbols represents the PFU look-up table logic in multiplier mode. A single PFU can effect a 1 x 4 bit multiply and sum with a partial product. The multiplier bit is input at A4, and the multiplicand bits are input at B[3:0], where B3 is the MSB. A[3:0] contains the partial product (or other input to be summed) from a previous stage. If A4 is logical 1, the multiplicand is added to the partial product. If A4 is logical 0, 0 is added to the partial product, which is the same as passing the partial product. CIN can input the carry-in from the less significant PFUs if the multiplicand is wider than four bits, and COUT outputs any carry-out from the addition, which may be used as part of the product or routed to another multiplier PFU for multiplicand width expansion. Shifting of the bits is done implicitly through routing between PFUs. The next section shows how PFUs are arrayed to create large, and, in particular, pipelined multipliers.

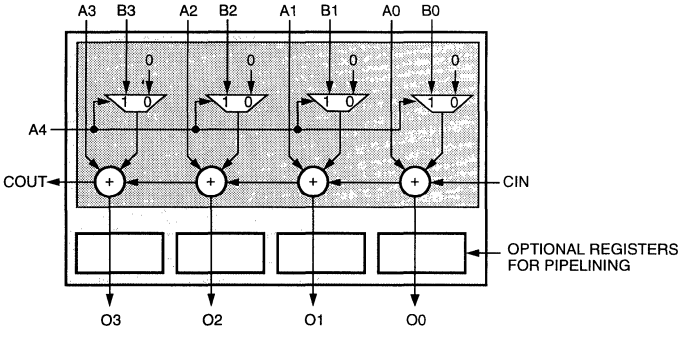


Figure 2. *ORCA* OR2CxxA/OR2TxxA PFU in Multiplier Mode

Creating Large, Pipelined Multipliers

Pipelining entails the insertion of one or more levels of clocked registers in the multiply process. By breaking the multiply up into segments in time, a higher clock rate may be used for the multiplication at the expense of a latency before the product is generated. The latency is equal to the number of clock periods corresponding to the number of pipeline levels used. If new multiplication factors are applied at each clock cycle, a new product will be generated each clock cycle following the initial pipeline latency.

The following diagrams show several pipelined multiplier configurations for the *ORCA* OR2CxxA/OR2TxxA architecture. Each large block represents a PFU. The multiplicand bits are applied to the B LUT inputs for each multiplier bit level, but are only shown at the input to the first stage for clarity. The arrows between internal levels of the multiplier represent bits of the partial products that are applied to the A[3:0] inputs of each PFU (except for the first level, where A[3:0] are implicitly zero). Multiplier bits are shown on the right and are input into A4 of each level for each PFU on that level, although, for clarity, they are only shown to enter the rightmost PFU in the diagram. Note that multiplier bits that operate on pipeline-delayed levels are also registered corresponding to the pipeline delay for that level. The particular PFU(s) in which the multiplier bits are to be registered are not addressed by the pipelining algorithm (described later) and are placed in the diagrams where they reflect the pipeline stage with which the delay is associated. Note also that the PFUs labeled as unused are not used for the multiplier but are usable for the other logic that may be in the designer's circuit.

Creating Large, Pipelined Multipliers (continued)

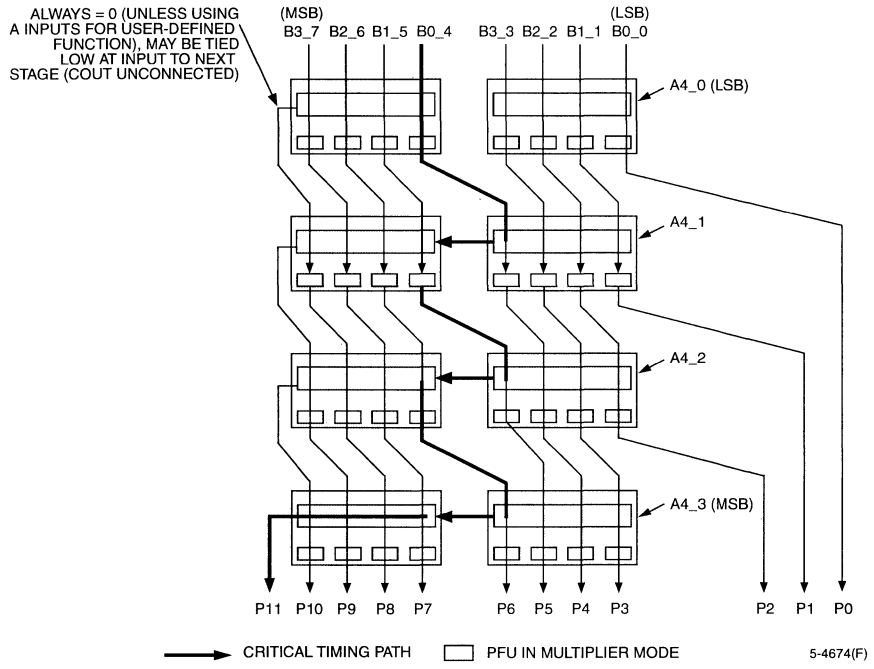


Figure 3. 4 x 8 Multiplier, Unpipelined

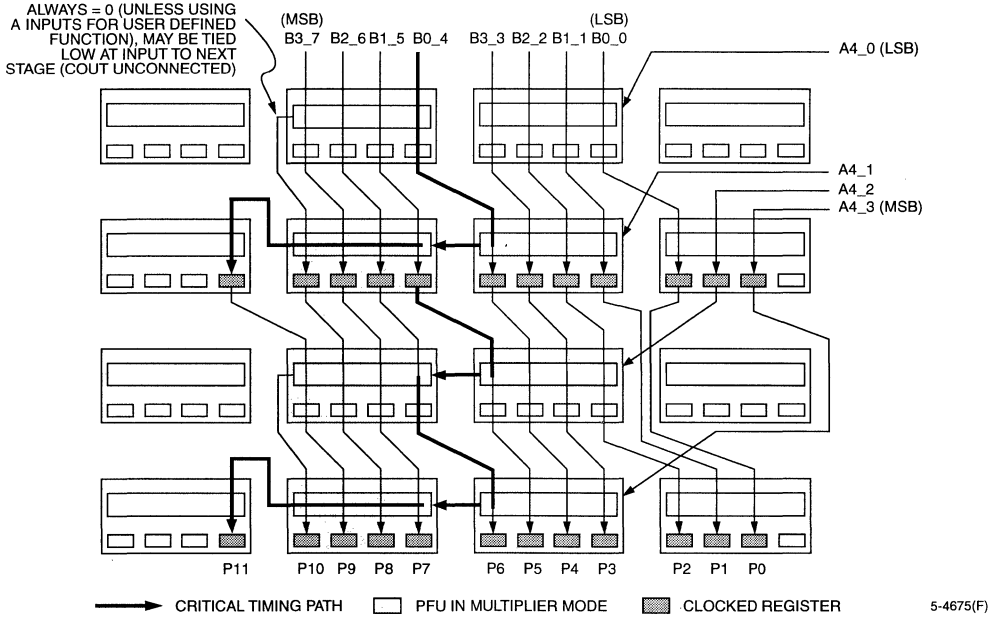


Figure 4. 4 x 8 Multiplier with 2 Pipelined Stages, Each 2 Multiplier Bits Deep

Creating Large, Pipelined Multipliers (continued)

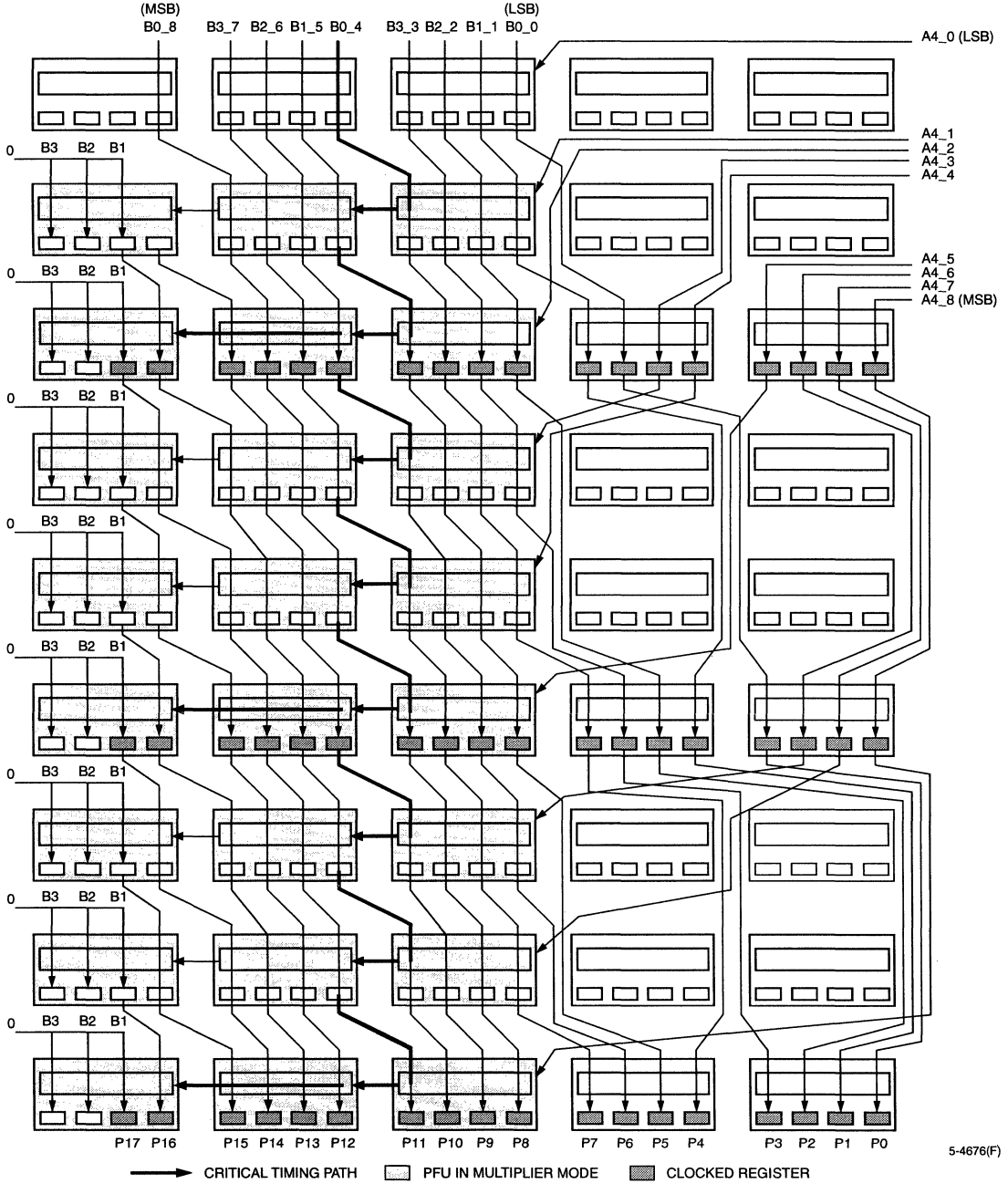


Figure 5. 9 x 9 Multiplier with 3 Pipelined Stages, Each with 3 Multiplier Bits per Stage

Creating Large, Pipelined Multipliers (continued)

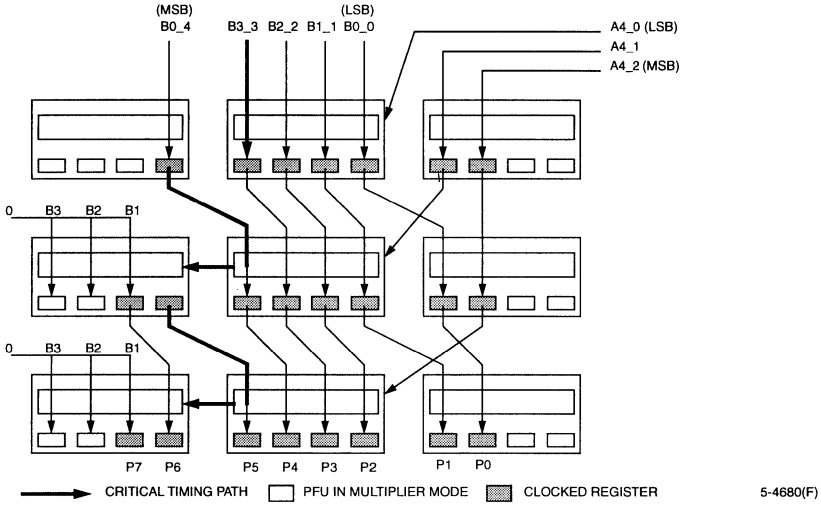


Figure 6. 3 x 5 Multiplier with 3 Pipelined Stages

Implementation Issues

This document addresses n x m bit multipliers with n being the multiplier and m being the multiplicand. In general, one would intuitively treat the larger size factor as the multiplicand so that the depth of the operation would be minimized in a time sense. But other conditions may dictate an alternate approach.

For instance, a user may need to access a partial product, at a given time relative to the inputs, that is available in an m x n configuration but not an n x m configuration. It must also not be assumed that forcing the largest factor to be the multiplicand will offer the highest throughput. An example of this might be a 4 x 32 multiplier at a desired clock rate that makes it impossible to process the 32-bit multiplier ripple in one clock period, even if the partial product is registered at all four levels. It is possible, however, that a 32 x 4 multiplier would work at the desired clock rate, although it might be at the expense of pipeline latency.

Other Multiplier Mode Capabilities

Consider again the diagram of the single PFU in multiplier mode (Figure 2), and you will notice that the architecture has other potential uses. The PFU can be used as a selective adder (or subtracter using two's complement arithmetic) adding (subtracting) the B[3:0] inputs to the A[3:0] inputs only when A4 is a logical 1. An example of the use of this feature is in scaling a value. Suppose a process always operates on a 4-bit value. Under certain conditions, flagged by a single bit, an offset or scaling value needs to be added or subtracted from the normal 4-bit value. This process may be implemented simply by routing the normal 4-bit value to the A[3:0] inputs, the conditional offset value to the B[3:0] inputs, and the control bit to the A4 input. Bit width expansion is possible using the carry-in and carry-out bits.

Parallel Multiplier Generation Algorithm

It is a fairly simple task for a designer to partition a multiplier into pipelined stages, given the operating frequency for various n x m stages (see Tables 1 and 2). To do so, simply select a stage depth that meets the required operating frequency and divide the multiplier into stages accordingly. The automated division of a multiplier into pipeline stages, however, can be somewhat less straightforward.

The following algorithm attempts to create a nonpipelined, registered output, n x m bit multiplier that will meet a target operating frequency. If the target operating frequency cannot be met, pipelined multiplier stages will be added until the desired target frequency is achieved. If the target frequency is unattainable (regardless of the amount of pipelining), the algorithm is aborted.

Pipeline stages are added so that only the minimum amount of pipeline delay is incurred, while maintaining the largest timing margin necessary to accommodate the target frequency. For instance, if a 4 x 8 bit multiplier (multiplier = 4) cannot meet its target without two stages of pipelining, the multiplier will be broken into two stages of 2 bits each (as long as both 2-bit sections will meet the target frequency). This will be the case, even if a 3-bit and 1-bit split might work, because the number of pipeline stages is the same, and a 2-bit x 2-bit split yields a better timing margin. In the following algorithm, the number of multiplier bits per pipeline stage is referred to as the depth of the stage. Calculations involving the width of the multiplier refer to the number of bits (m) in the multiplicand.

Algorithm (Text)

Pipeline delay = 1 automatically due to registered output
Calculate the nonpipelined, registered output operating frequency of the n x m multiplier
If the calculated frequency is less than the target frequency (pipelining necessary)
    for each possible pipeline depth (in PFUs), 0 < depth < n+1
        calculate if first and last/internal pipeline stage frequencies meet target
    if the first or last pipeline stage for depth of 1 doesn't meet target
        failure, impossible to meet target
    do until the first pipeline stage meets the target frequency
        increment the pipeline delay
        set the first stage depth to next smaller value ((n + 1)/pipeline delay)
        set the last pipeline stage depth equal to the first pipeline stage depth
    do until the last pipeline stage meets the target frequency
        decrement the last pipeline stage depth
    set the internal pipeline stage depth to the number of multiplier bits not covered by the first and last pipeline stages, n_int
    do until all internal pipeline stages meet the target frequency
        set the internal depth to next smaller value ((n_int + pipeline delay – 3) / (pipeline delay – 2))
        increment the pipeline delay
    for each internal pipeline stage (total pipeline delay – 2) set the depth value from above step
    set final internal pipeline stage depth to any remaining bits not evenly divisible by depth value from above

Multiplier Speed and Area Characteristics

Table 1. Multiplier Speed for First Pipeline Block (MHz)

# Bits Deep/Reg. (D)	# Bits Wide (W) (Multiplicand)							
	4	6	8	9	12	16	24	32
2	113.64	98.04	87.72	81.97	74.63	64.94	51.55	42.74
3	66.67	51.81	48.78	46.95	44.44	40.82	35.09	30.77
4	47.17	35.21	33.78	32.89	31.65	29.76	26.60	24.04
5	36.50	26.67	25.84	25.32	24.57	23.42	21.41	19.72
6	29.76	21.46	20.92	20.58	20.08	19.31	17.92	16.72
7	25.13	17.95	17.57	17.33	16.98	16.42	15.41	14.51
8	21.74	15.43	15.15	14.97	14.71	14.29	13.51	12.82
9	19.16	13.53	13.32	13.18	12.97	12.64	12.03	11.48
10	17.12	12.05	11.88	11.76	11.60	11.34	10.85	10.40

Table 2. Multiplier Speed for Internal/Final Pipeline Block (MHz)

# Bits Deep/Reg. (D)	# Bits Wide (W) (Multiplicand)							
	4	6	8	9	12	16	24	32
1	163.93	133.33	114.94	105.26	93.46	78.74	59.88	48.31
2	81.30	60.24	56.18	53.76	50.51	45.87	38.76	33.56
3	54.05	38.91	37.17	36.10	34.60	32.36	28.65	25.71
4	40.49	28.74	27.78	27.17	26.32	25.00	22.73	20.83
5	32.36	22.78	22.17	21.79	21.23	20.37	18.83	17.51
6	26.95	18.87	18.45	18.18	17.79	17.18	16.08	15.11
7	23.09	16.10	15.80	15.60	15.31	14.86	14.03	13.28
8	20.20	14.04	13.81	13.66	13.44	13.09	12.44	11.85
9	17.95	12.45	12.27	12.15	11.98	11.70	11.17	10.70
10	16.16	11.19	11.04	10.94	10.80	10.57	10.14	9.75

Table 3. Multiplier Resource Required (# PFUs in Multiplier Mode)

# Multiplier Bits	# Multiplicand Bits							
	4	6	8	9	12	16	24	32
2	1	2	2	3	3	4	6	8
3	2	4	4	6	6	8	12	16
4	3	6	6	9	9	12	18	24
5	4	8	8	12	12	16	24	32
6	5	10	10	15	15	20	30	40
7	6	12	12	18	18	24	36	48
8	7	14	14	21	21	28	42	56
9	8	16	16	24	24	32	48	64
10	9	18	18	27	27	36	54	72

Glossary

ORCA	Optimized Reconfigurable Cell Array
FPGA	Field-Programmable Gate Array
LSB	Least Significant Bit
LUT	Look-up Table
MSB	Most Significant Bit
PFU	Programmable Function Unit—the major logic block of an *ORCA* FPGA.
Multiplier	One of two factors, the multiplication of which generates a product. The multiplier is used as an index for the number of times the multiplicand is added to itself to effect the multiply.
Multiplicand	One of two factors, the multiplication of which generates a product. The multiplicand is added to itself the number of times specified by the multiplier to effect the multiply.
Partial Product	An interim result of a multiplication which is, itself, the product of the multiplicand and some part of the multiplier factor.

Implementing and Optimizing Multipliers in *ORCA* Field-Programmable Gate Arrays

Introduction

Multiplication is at the heart of the majority of digital signal processing (DSP) algorithms. Currently, digital multiplier functions are primarily the domain of DSP microprocessors and dedicated multiply or multiply-accumulate (MAC) devices. The DSP microprocessor solution provides flexibility due to programmability.

However, this flexibility is offset by the intrinsic performance limitations of the fetch/decode/execute/store software methodology. For highest-performance designs, engineers often turn to dedicated multiplier or MAC devices.

In the converse of the DSP micro solution, dedicated devices sacrifice flexibility (for example, in the width of data words) and add to board space, cost, power consumption, and overall circuit complexity. The **ideal** solution would provide algorithmic flexibility **and** high performance. The *ORCA* Series of FPGAs from Lucent Technologies offers just such a solution.

In the past, multiplication in FPGAs has been problematic due to the limited performance and density of the available devices. *ORCA* offers a very large number of usable logic cells and specialized architectural elements, such as fast-carry chains. These features bring the idea of a configurable, high-performance algorithm engine to reality.

In this application note, four basic multiplier circuits will be discussed. The first three circuits focus on an 8 x 8 multiplier and use various degrees of pipelining to trade off performance versus density. These circuits are based on the digital multiplication algorithm described below. The fourth example illustrates *ORCA*'s on-chip memory configured to implement a look-up table based 2 x 4 multiplier.

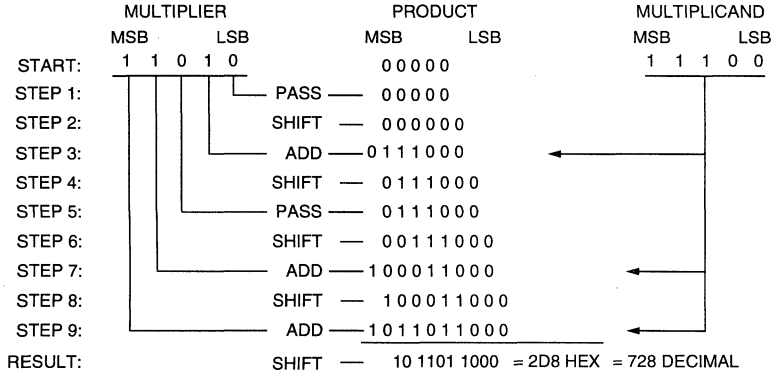
The Digital Multiplication Algorithm

A number of methods exist for performing binary multiplication. The simplest and most frequently used involves a bit-by-bit analysis of one operand controlling an add-or-pass decision on the other operand. Assuming two inputs called OPA and OPB, the method can be described as follows:

1. If bit 0 (the LSB) of OPA is a zero, pass a value of zero to the next step. If the LSB of OPA is a one, pass the value of OPB to the next step.
2. Shift the value from step 1 one bit to the right. The resultant LSB from this shift is the LSB (bit 0) of the final, overall product. The other bits are then passed on to the subsequent step.
3. Based on the next least significant bit of OPA (bit 1), either pass the value from the previous step (if OPA bit 1 is a zero) or ADD OPB to this value (if OPA bit 1 is a one).
4. Shift the result from step 3 one bit to the right. The resultant LSB from this shift is the next LSB (bit 1) of the final, overall product. The other bits are then passed on to the subsequent step.
5. Repeat steps 3 and 4 for all remaining bits in OPA, evaluating from the least significant to the most significant bit. The MSBs of the final product will appear at the output of the last stage's add/pass circuit, and the LSBs will be the shifted-out LSBs from the previous steps.

As an example, consider the multiplication of the two 5-bit numbers, $1A and $1C (decimal 26 and 28):

The Digital Multiplication Algorithm (continued)



Note: For any binary multiplication, the number of bits needed for the product will be the total of the bits found in both operands. In this case, with two 5-bit operands, the product can be up to 10 bits long. In the 8 x 8 multiplier examples to follow, the product can be as long as 16 bits.

Figure 1. Multiplication of Two 5-bit Numbers ($1A and $1C)

Reference Designs

The nibble-oriented *ORCA* architecture makes the popular 8-bit word width an ideal test case. Asynchronous, fully pipelined, and partially pipelined versions of an 8 x 8 multiplier are covered in the following discussions covering reference designs 1 through 4. While the asynchronous version provides the highest density (least programmable logic cells or PLCs), the pipelined version offers maximum performance (in this case over 66 MHz).

The partially pipelined version represents one possible compromise between the conflicting goals of speed and density. The algorithms and design techniques can be extrapolated to construct similar multipliers for any word width and to meet a wide range of performance/density constraints.

Table 1 shows the number of PLCs (density) and the corresponding performance of each of the 8 x 8 circuits.

Table 1. Relative Density/Performance

Sample Design	No. of PLCs	Speed
Asynchronous 8 x 8 Multiplier	18	75 ns (tpd)
4-stage Pipelined 8 x 8 Multiplier	27	54 MHz
8-stage Pipelined 8 x 8 Multiplier	46	66 MHz

Note: The performance shown is for the ATT2Cxx family of devices in the -3 speed grade.

Reference Design 1—Asynchronous 8 x 8 Multiplier

This design implements an 8 x 8 bit multiplier in the minimal amount of space (18 PFUs). However, this compactness exacts a price in terms of performance; the inputs to OPB[7:0] (operand B) must propagate through eight levels of logic before the output is realized. Figure 2 shows the top-level schematic for this multiplier.

Reference Designs (continued)

Reference Design 1—Asynchronous 8 x 8 Multiplier (continued)

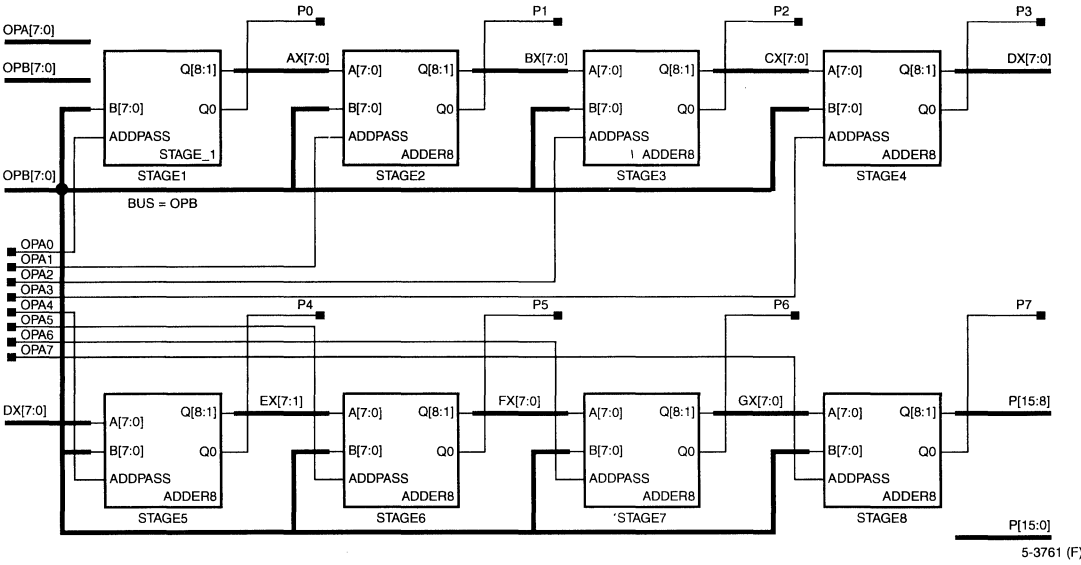


Figure 2. Asynchronous 8 x 8 Multiplier—Top Level Schematic

Two 8-bit operands (OPA[7:0] and OPB[7:0]) are multiplied to yield a 16-bit product P[15:0]. Note that OPB is connected to each stage of the multiplier; OPA is broken-up so that each individual bit controls the add-or-pass decision for each stage.

Each block in the design is identical except for the first stage (see Figure 3). Since this stage will only pass OPB or all zeros, there is no need to supply inputs other than OPB and the ADD/PASS control. The schematics for the first stage and all subsequent stages are shown in Figures 3 and 4.

Reference Designs (continued)

Reference Design 1—Asynchronous 8 x 8 Multiplier (continued)

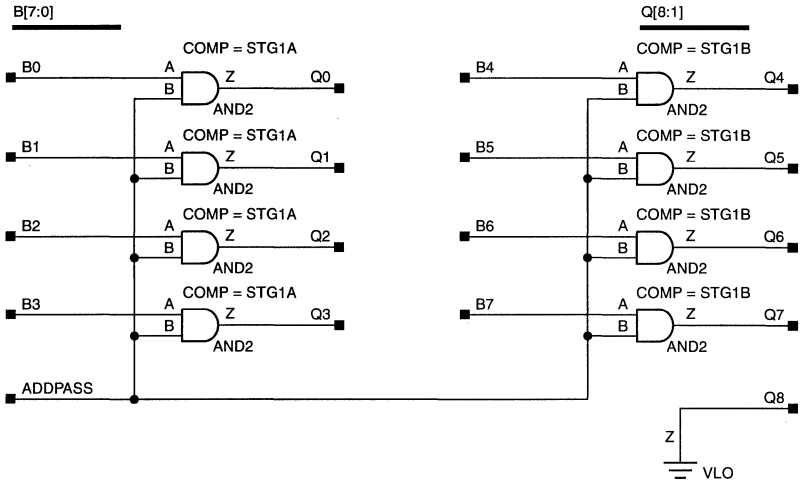
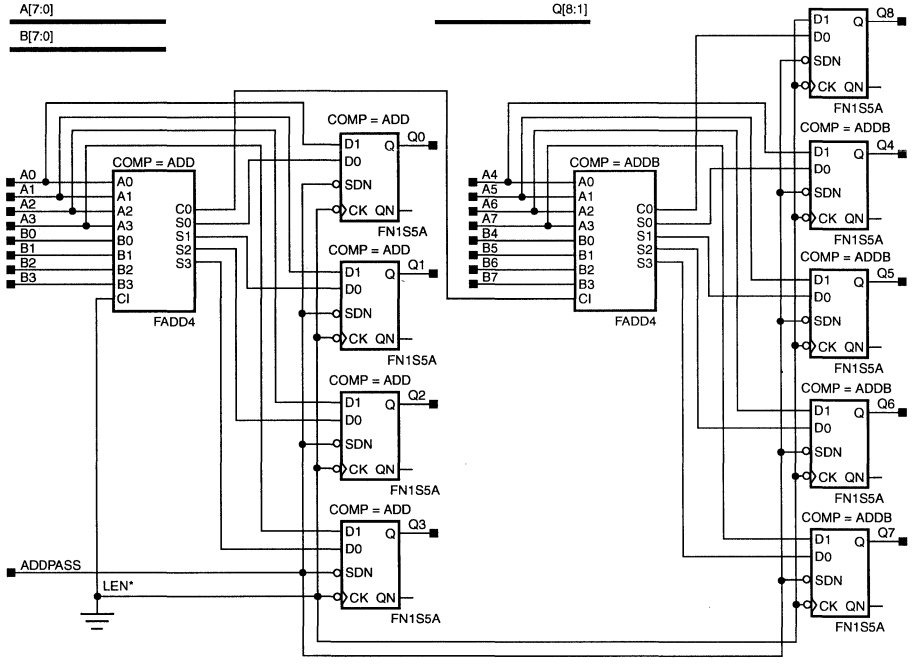


Figure 3. Stage 1 of the Asynchronous Multiplier



* LEN = 0 for asynchronous multiplier.

Figure 4. Stages 2—8 of the Asynchronous Multiplier

Reference Designs (continued)

Reference Design 1—Asynchronous 8 x 8 Multiplier (continued)

Note the use of the COMP = attribute. This attribute forces circuit elements into a common PLC. For example, in Figure 4, the COMP = ADD attribute forces both a 4-bit adder (FADD4) and four multiplexed-input latches into a single PLC. The use of the multiplexed-input latches allows the add-or-pass decision to be performed in the same programmable function unit (PFU) as the add function. By tying the latch enable signal (LEN) low, the latches are placed in a transparent, flow-through mode. This saves implementing the multiplexer function in a separate PFU's look-up table (LUT). Performance is improved by eliminating the logic and interconnect delay associated with a separate PFU. As a bonus, the multiplexed-input latches ease the modification of the circuit to a fully pipelined multiplier. The transparent latches are simply changed to multiplexed-input flip-flops.

Multiplexed-input latches and flip-flops are basic library elements for *ORCA*. These elements select either the output from a given LUT or the direct data input to the register/latch. The direct path bypasses the LUT for faster operation. For each PLC, the path can either be selected via a signal (as in this case) or fixed through configuration.

Reference Design 2—Pipelined, 66 MHz 8 x 8 Multiplier

Figure 5 shows the top-level schematic for the fully pipelined version of the multiplier. While similar in function to the asynchronous version, the inputs to and outputs from each stage must be pipelined to ensure the correct bits arrive at the correct times. There is a latency of eight clock cycles until the first result appears at the output. However, new operands can be supplied with each clock cycle, and a throughput of 66 MHz can be maintained.

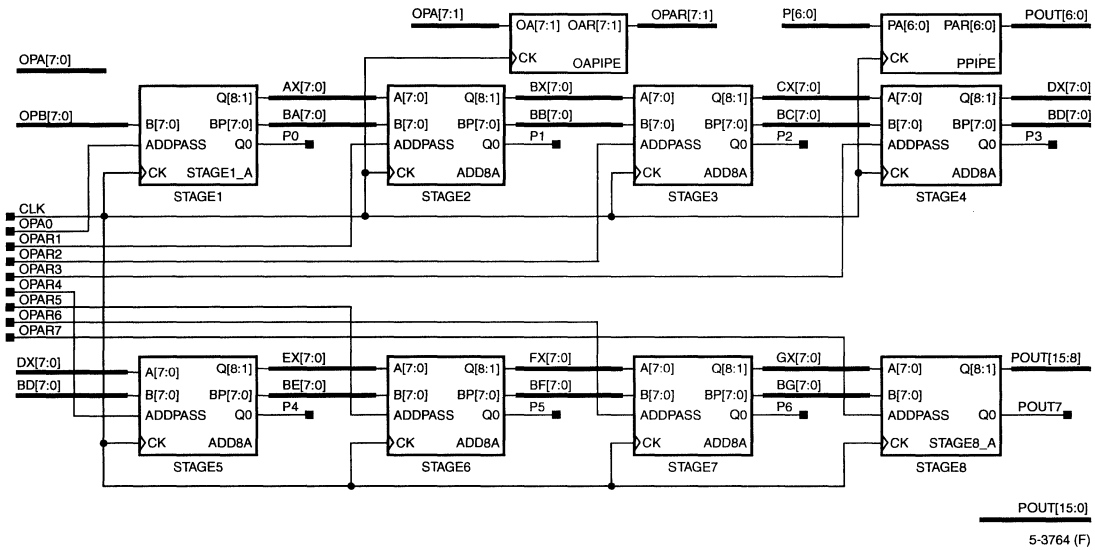


Figure 5. Pipelined 8 x 8 Multiplier (66 MHz)—Top-Level Schematic

Reference Designs (continued)

Reference Design 2—Pipelined, 66 MHz 8 x 8 Multiplier (continued)

Figure 6 shows the underlying schematic for stages 2 through 7 of the pipelined multiplier. Note that stage 1 is similar, but, as with the asynchronous version, it only needs to pass a value of zero or OPB[7:0]: no adding is necessary—any number added to zero yields the same number. There are two primary differences in the stages of the pipelined vs. asynchronous versions. First, the flow-through latches (library element FN1S5A) are replaced with registers (library element FN1S3AX). This provides the pipelining of the intermediate results. The second difference is the addition of eight registers to pipeline OPB[7:0] (operand B). These registers simply move operand B down the chain of add/pass stages with each clock cycle. In the final stage, these registers are not required.

To ensure that the appropriate bit of OPA[7:0] (operand A) arrives at each stage at the correct time, block OAPIPE (see Figure 7) inserts seven registers in the path of OPA bit 7, six registers in the path of OPA bit 6, etc. Block PPIPE performs a similar pipelining function for the partial results formed at each stage. PPIPE ensures that all the results for a particular set of operands arrive at the product outputs (P[15:0]) at the same time.

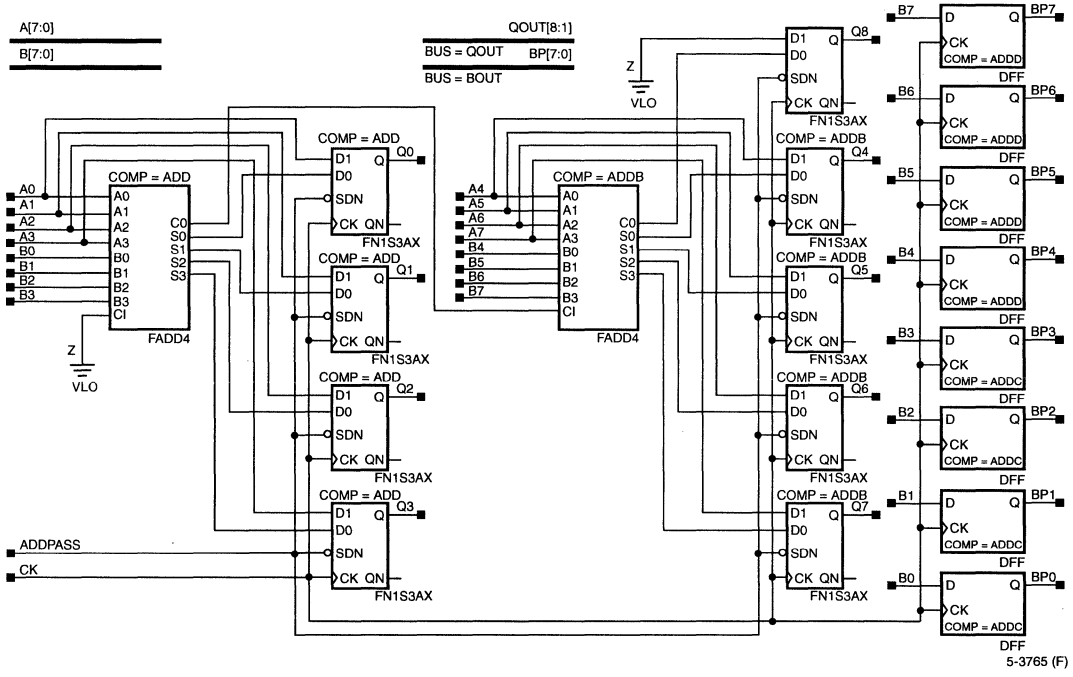


Figure 6. Stages 2—8 of the Pipelined Multiplier

Reference Designs (continued)

Reference Design 2—Pipelined, 66 MHz 8 x 8 Multiplier (continued)

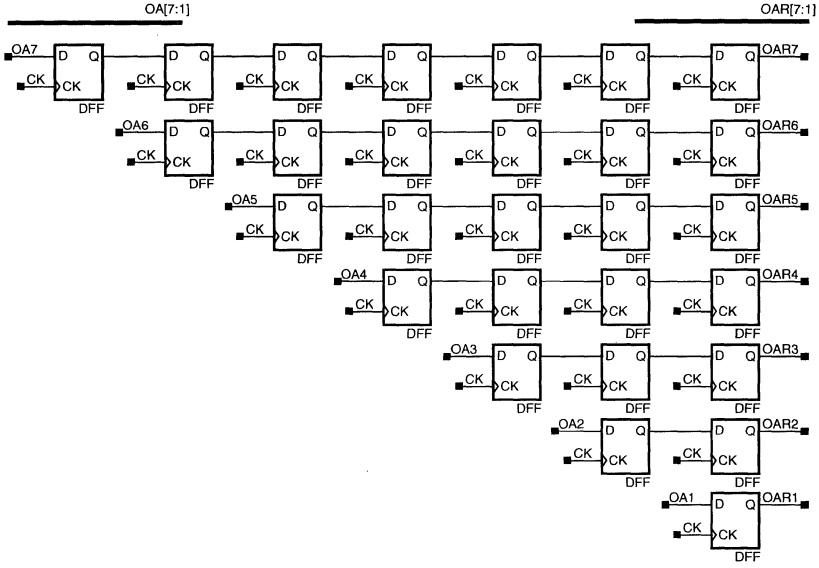


Figure 7. Pipeline Registers for OPA[7:1] (OAPIPE)

Reference Designs (continued)

Reference Design #3—Partially Pipelined, 54 MHz 8 x 8 Multiplier

Figure 8 shows the top-level schematic for an 8 x 8 multiplier that employs a pipeline stage after every two add/shift stages. This lowers the overall PLC count without severely impacting the performance of the fully pipelined version. The pipeline registers for operand A (OPA[7:0]) and the partial results (P[7:0]) are shown individually (rather than as a block in the fully pipelined version).

Each of the four stages actually operates on **2** bits from OPA[7:0]. Figure 8 shows the underlying schematic for add/pass stages 2 and 3. Note that the lower-order half of each stage uses FN1S5A multiplexed-input latches (transparent, combinatorial mode) while the upper half uses FN1S3AX flip-flops. These flip-flops register the partial results in groups of two. This reduces the number of pipeline registers needed (for the OPA[7:0] and partial products) by one half.

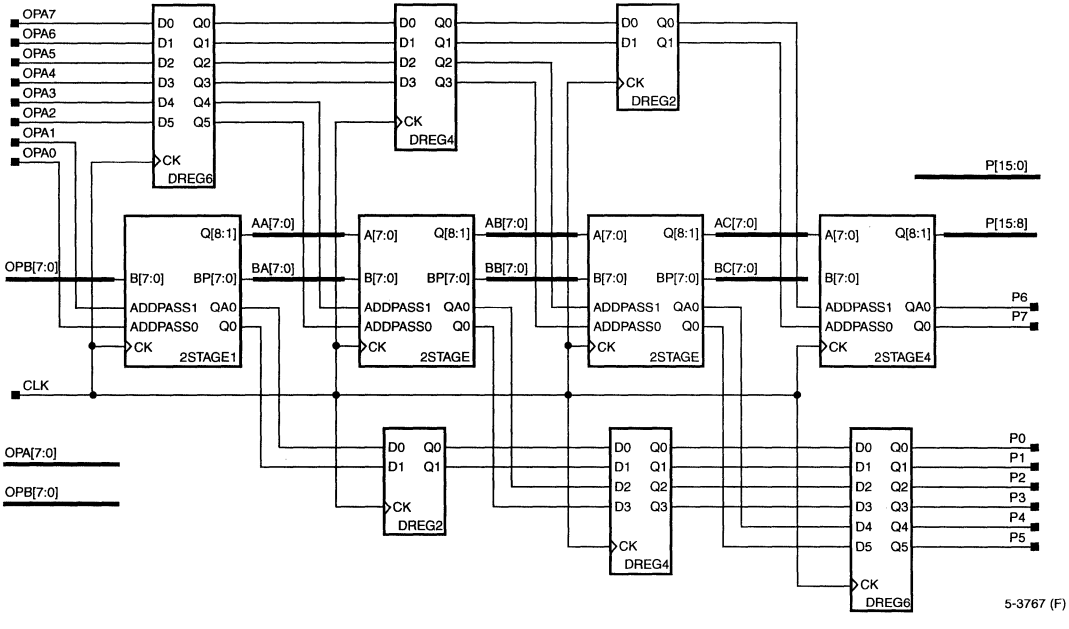


Figure 8. 8 x 8 Multiplier—Pipelined Every Two Stages

Reference Designs (continued)

Reference Design #3—Partially Pipelined, 54 MHz 8 x 8 Multiplier (continued)

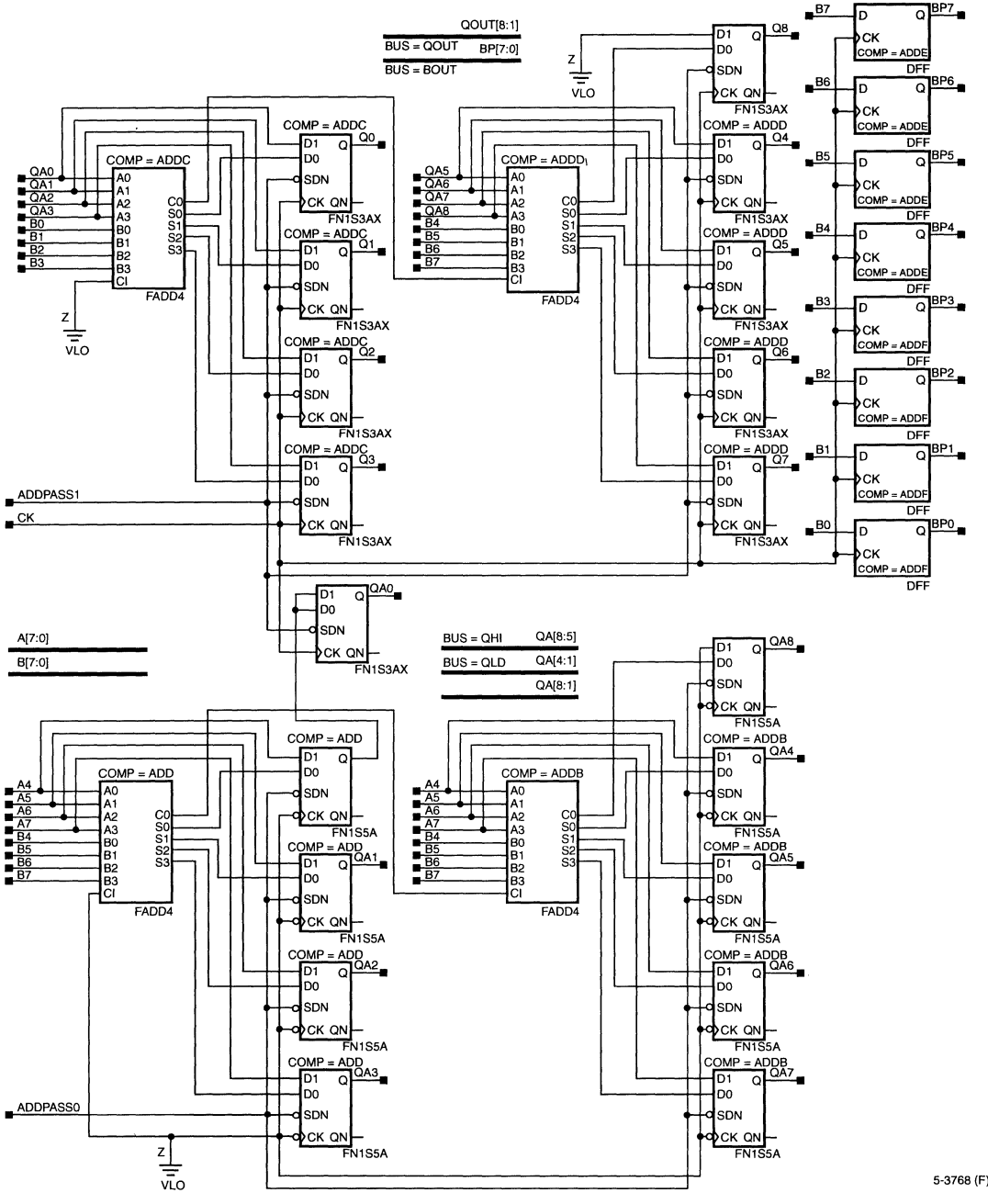
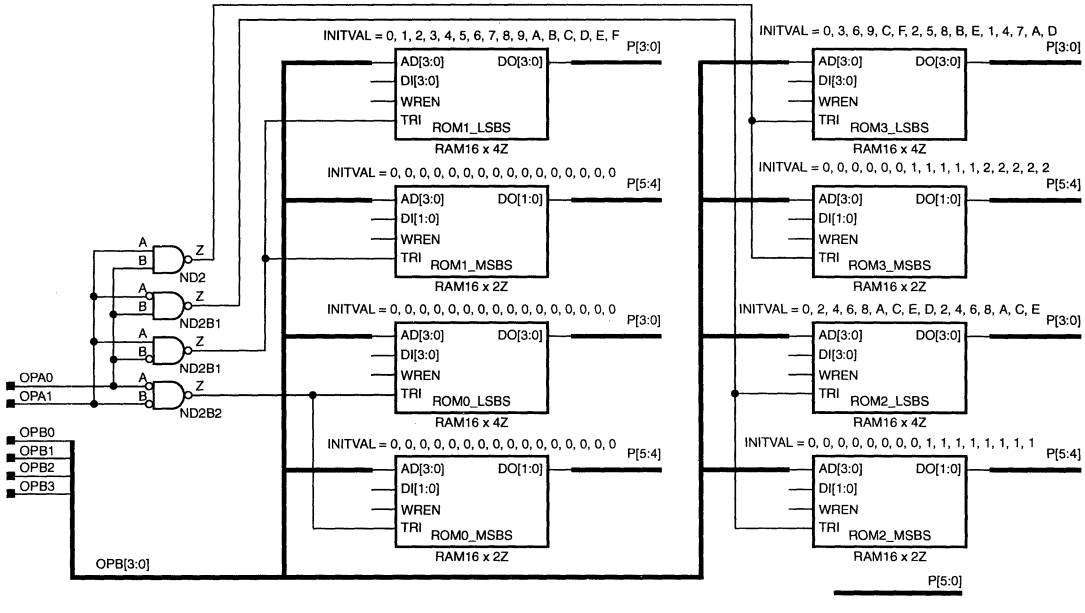


Figure 9. Stages 2—3 of the Partially Pipelined 8 x 8 Multiplier

Reference Designs (continued)

Reference Design #4—Fast Look-Up Table Based 2 x 4 Multiplier

Figure 10 illustrates an alternative approach to binary multiplication in FPGAs. For reasons that will be explained, only small multipliers are practical in this configuration. The circuit in Figure 10 multiplies a 2-bit number by a 4-bit number.



5-3769 (F)

Figure 10. 2 x 4 Multiplier Implemented with Look-Up Tables (ROM)

The PLCs in *ORCA* may be used as blocks of memory. Initial values may be loaded into these memory cells via the configuration bit stream. In this manner, the cells are configured as ROM. To create a multiplier in this ROM, both operands are applied to the address inputs, and the product is seen at the ROM data outputs. Obviously, the correct results must be precalculated and coded into the configuration bit stream for storage in the proper address. The INITVAL = attributes in Figure 10 provide the initial values for each ROM block. During configuration, the ROM is loaded with these values for each of the 64 locations (64 locations are needed due to the 6 operand bits that form the address).

The performance of such an implementation is limited only by the address decode time and the access time of the on-chip memory. With *ORCA* devices, the read cycle time is only 3.6 ns (ATT2Cxx-3 family). Adding output registers increases performance even further and costs nothing in terms of additional logic and routing; the registers already exist in the PLCs that are doing the ROM function. With this registered output scheme, the performance is limited by the address-to-clock setup time within a PLC. For the ATT2Cxx-3 family, this time is 1.8 ns.

Despite the extremely high performance predicted by these memory access times, the size of practical look-up table based multipliers is limited. The amount of ROM required increases exponentially as the width of the operands increases. Even the sizable on-chip memory of *ORCA* (each PLC can provide a 16 x 4 block of memory) can be quickly consumed. However, most other FPGAs offer no on-chip memory at **all**; look-up table based multipliers in these devices are impractical (if not impossible) to implement.

Reference Designs (continued)

Reference Design #4—Fast Look-Up Table Based 2 x 4 Multiplier (continued)

Table 2 lists the number of PLCs required just to do the ROM function for given size multiplier.

Table 2. Memory and *ORCA* PLCs Needed for Various Look-Up Table Based Multipliers

Size (Bits)	Output Width (Bits)	ROM Needed	*ORCA* PLCs Needed
2 x 2	4	16 x 4	1
2 x 4	6	64 x 6	6
4 x 4	8	256 x 8	32
4 x 8	12	4K x 12	768*
8 x 8	16	64K x 16	16276*

* Cannot be implemented in the current generation of devices.

In addition to the amount of available logic used for ROM, look-up table multipliers must also provide address decoding circuitry. This circuitry becomes more complex (and, therefore, slower) as the size of the array addressed increases.

Another interesting observation from Figure 10 is that three of the eight LUTs needed are preloaded with all zeros. Therefore, these LUTs could be eliminated and more simple circuitry substituted. For consistent prop delays and for clarity in this example, the all zeros LUTs have been retained.

Another situation that could warrant a LUT based multiplier is when one of the operands is a constant. For example, multiplying two 8-bit numbers requires 64K addresses (16 address lines); multiplying an 8-bit number times an 8-bit constant only requires 256 locations (eight address lines).

Conclusion

Multipliers are a critical element of digital signal processing and a host of other applications. Implementing multipliers in reconfigurable FPGAs offers a maximum of speed and flexibility. FPGAs with on-chip memory add the ability to store coefficients and implement look-up table based multipliers. The multiplication algorithms and the coefficients may be modified and/or adapted by simply reconfiguring the FPGA. Imagine the flexibility this provides in fine-tuning a DSP system!

Two common types of binary multiplier schemes are the iterative add/pass/shift algorithm and the look-up table. In all but the smallest multiplier functions, the add/pass/shift algorithm will yield a superior combination of speed and density.

The circuits shown in this application note are available 24 hours a day on the Lucent Technologies FPGA BBS (610-712-4314). For a copy of the *ORCA FPGAs as Configurable DSP Coprocessors* Application Note (AP94-041FPGA), contact your local Lucent sales professional.

The author would like to thank Rick Collins of Applied Signal Technology for his contributions to this application note.

Notes

ORCA FPGAs Excel in Multiplexing and On-Chip SRAM Applications

Article reprinted from *App Review*, April 11, 1994.

Introduction

One of the goals of any FPGA architecture is to pack as much usable logic as possible into the least amount of silicon area. To reach for this goal, considerable research has been conducted in the quest for the optimal programmable architecture. Areas of study have included programming elements, basic logic building blocks, and interconnect resources. Bell Labs' engineers contributed significantly to this body of research during the development of the *ORCA* architecture. This application note illustrates the versatility of the *ORCA* architecture in the important application areas of multiplexing signals and on-chip SRAM. A 16 x 4 SRAM buffer with triple-multiplexed inputs is given as an example.

Some applications of this include telecom/datacom standards such as ATM. One key to providing a powerful FPGA architecture lies in the flexibility of the on-chip resources. In the case of multiplexing circuits, *ORCA* offers not one, but **four** device elements that can be used to provide a multiplexing function. These components are 16-bit look-up tables (LUTs), the PFU_MUX function, Latch/FF input data selectors, and internal tristate buffers. Figure 1 shows each of these architectural features, which are contained in each programmable logic cell (PLC).

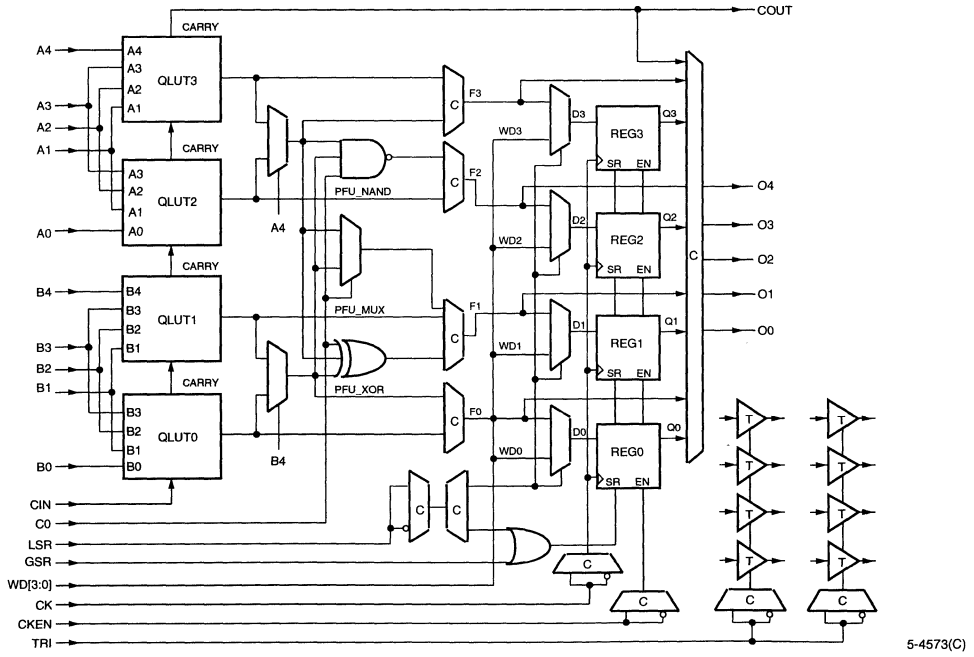


Figure 1. Four Multiplexing Elements in a Single *ORCA* PFU

Introduction (continued)

The design example in this article illustrates how these features can be used to create a 12-to-4 multiplexer that is 50% denser and 3x faster than a conventional LUT-only multiplexer implementation. The LUTs in *ORCA* can also be used as blocks of user SRAM or ROM. Here again, flexibility is the key. Figure 2 illustrates how one *ORCA* logic element (a PFU, or Programmable Function Unit) can be used to implement one 16 x 4 SRAM block, two 16 x 2 SRAM blocks, or one 16 x 2 SRAM block with the remaining half PFU used for random logic. Also shown is the total number of user-SRAM bits available in each member of the family. The nibble-wide organization of the logic and routing in the *ORCA* architecture enables easy width expansion for byte, word, or other width memories, and tristate buffers are available at each PFU to provide for expansion in depth. Memory blocks of any width and depth can be easily built up schematically or synthesized from textual descriptions.

CPLDs and earlier FPGA devices are extremely limited in providing SRAM or ROM. In CPLDs, one bit of storage consumes an entire macrocell. In earlier FPGAs, the cost is from two logic cells per bit (antifuse types) to two bits per logic cell (SRAM-based types). The problem is compounded by the decoding and MUX/deMUXing circuitry necessary to form the complete SRAM structure. The option of SRAM or EPROM external to the CPLD/FPGA adds not only to cost and board space, but requires a large number of I/Os on the programmable device for communicating the data, address, and control signals.

Obviously, only very small on-chip memory blocks are practical in such devices. Compare this to the complete 16 x 4 SRAM function that can be implemented in a single *ORCA* PFU.

Flexibility in implementing any logical function provides several benefits. First, the function can be optimized to satisfy a wide range of speed/density constraints. With *ORCA* Foundry, these optimizations can be performed automatically or under user direction. Third-party synthesis tools also benefit from this flexibility. Lucent Technologies provides an *ORCA*-specific synthesis tool, *SCUBA*, to compile large SRAM (as well as other memory and datapath functions).

Design Example—16 x 4 SRAM Buffer with Multiplexed Inputs and Outputs

Figure 3 shows the top-level schematic for a 16 x 4 SRAM buffer with three separate input nibbles. It was borrowed from the channel select/buffer circuitry in an ATM switcher/router application. The entire design fits into two *ORCA* PFUs—one for the 16 x 4 SRAM element and one for the 12-to-4 multiplexer.

The design exploits two of the four multiplexing components within *ORCA* to fit the entire 12-to-4 MUX in a single PFU. The first two nibbles are MUXed in the LUTs. The third is then MUXed in using the data selectors in the latches. By tying the LE control to the latches high, the latches operate in a flow-through mode.

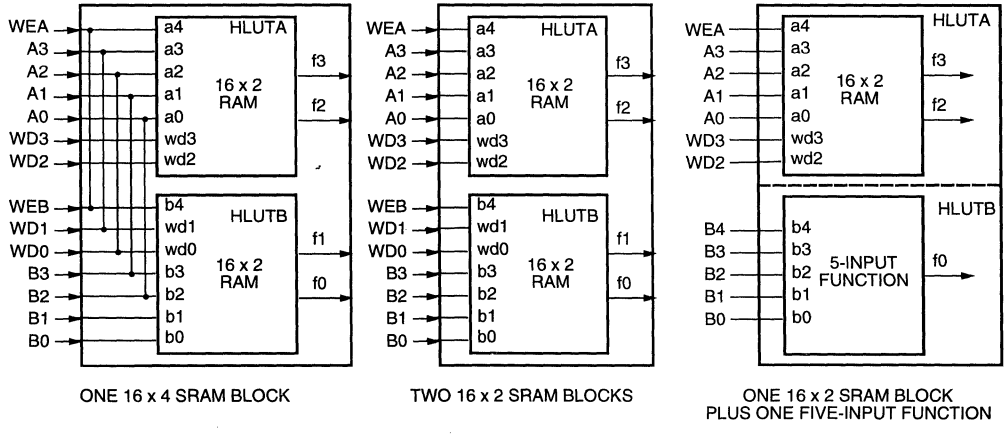


Figure 2. SRAM Elements in *ORCA*

Design Example—16 x 4 SRAM Buffer with Multiplexed Inputs and Outputs
(continued)

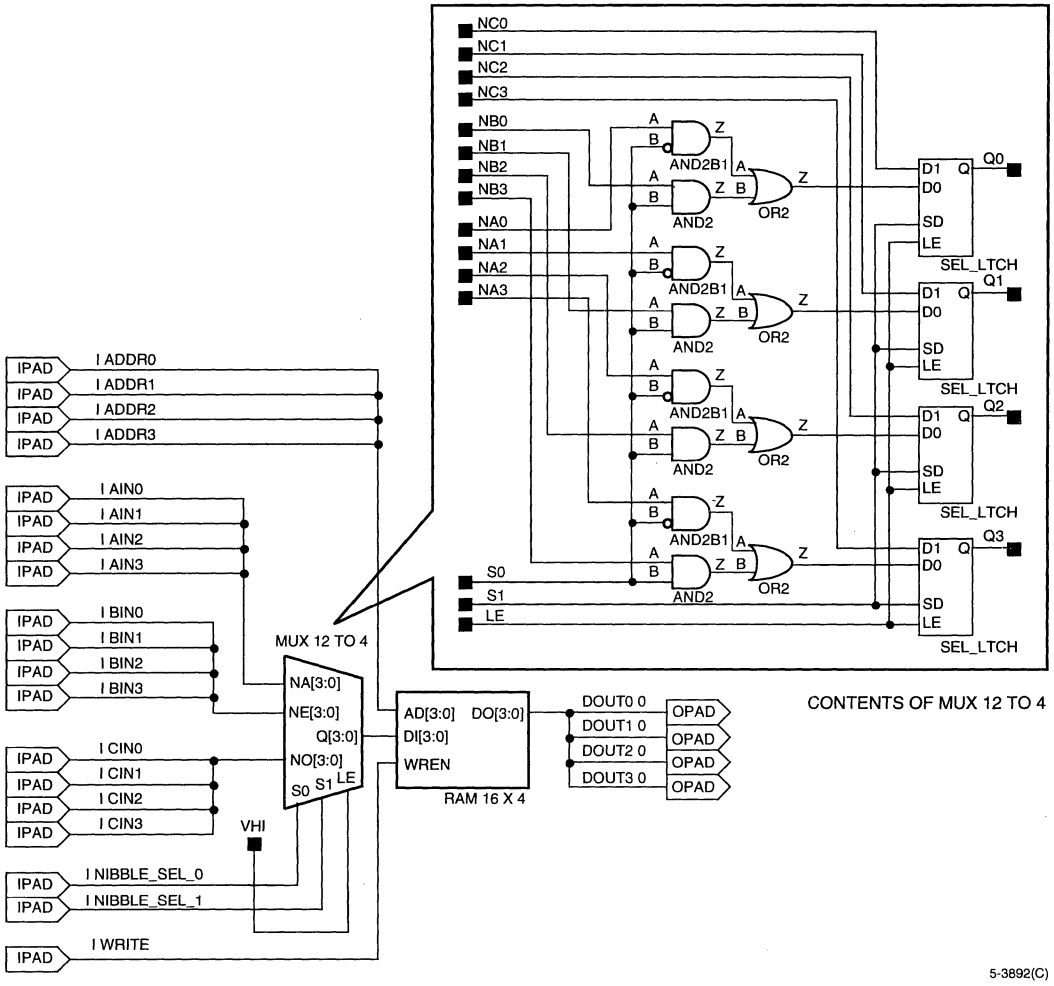


Figure 3. 16 x 4 SRAM Buffer with Triple-Multiplexed Inputs (2 PFUs)

Implementing the same 12-to-4 MUX in a conventional LUT-only method would use one PFU to MUX the two lower-order nibbles, and then MUX the third with the result of the other two in a separate PFU. This would require twice the PFUs (two vs. one) and would operate roughly one-third slower than the approach shown.

Figure 4 shows how the RAM outputs may be multiplexed with other signals using the tristate buffers. This allows wide tristate buses to be easily implemented in *ORCA*. It should also be noted that these tristate buffers can be used independent of the PFU, thus allowing even more flexibility.

Design Example—16 x 4 SRAM Buffer with Multiplexed Inputs and Outputs
(continued)

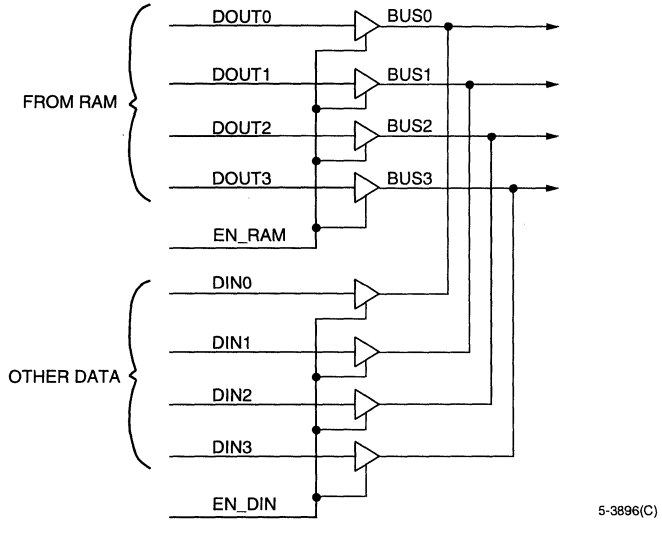


Figure 4. Multiplexing of RAM Outputs Using Tristate Buffers

Conclusion

The utility of an FPGA is directly affected by the flexibility of the logic and routing resources. *ORCA* FPGAs offer a number of ways to implement multiplexer circuits that can be fine-tuned for a given application. On-board SRAM blocks enable control and data stores, FIFOs, and memory buffers to be implemented without costly external SRAM or EPROM devices. (Lucent Technologies offers a number of memory-block, FIFO, and bidirectional FIFO designs in different sizes as examples on their BBS: 610-712-4314.) Applications that require both multiplexers and storage elements include ATM and other telecom/datacom switches, hubs, and routers. In these and many other applications, Lucent Technologies' *ORCA* FPGAs are an excellent solution.

Implementing First-In First-Out (FIFO) Memory Blocks in *ORCA* FPGAs

Overview

This application note provides specific details regarding the implementation of first-in, first-out (FIFO) memory blocks using elements from the *ORCA* library. Applications for this include data buffering between systems with either different data rates or with asynchronous interfaces that require handshaking between them. *ORCA* elements are particularly useful for this function because they contain configurable 16-bit look-up tables (LUTs) that may be used as static RAM blocks.

Each programmable function unit (PFU) within the device contains four LUTs. Each PFU may be used as a 16 x 4 memory element. *ORCA*'s nibble-wide architecture allows for easy width expansion for byte-wide and word-wide FIFO designs. For depth expansion, 3-state buffers are provided at each PFU.

Read and write cycle times for this design are under 40 ns (over 25 MHz). With some simple modifications, the design can be improved to yield speeds in excess of 33 MHz. The schematics for the design are included in this document.

The design discussed in this document applies to a 127 x 4 FIFO; however, by using *ORCA*'s depth and width expansion techniques, it is possible to construct virtually any size FIFO within the limits of the available resources.

Functional Description

Within the FIFO design, there are five major functional blocks:

- Input/output ports and top-level schematic (See Figure 1.)
- Read/Write address pointers (See Figure 2.)
- Address multiplexer (See Figure 3.)
- Memory array (127 x 4) (See Figure 4.)
- Flag circuitry (full, empty, busy) (See Figure 5.)

Functional Description (continued)

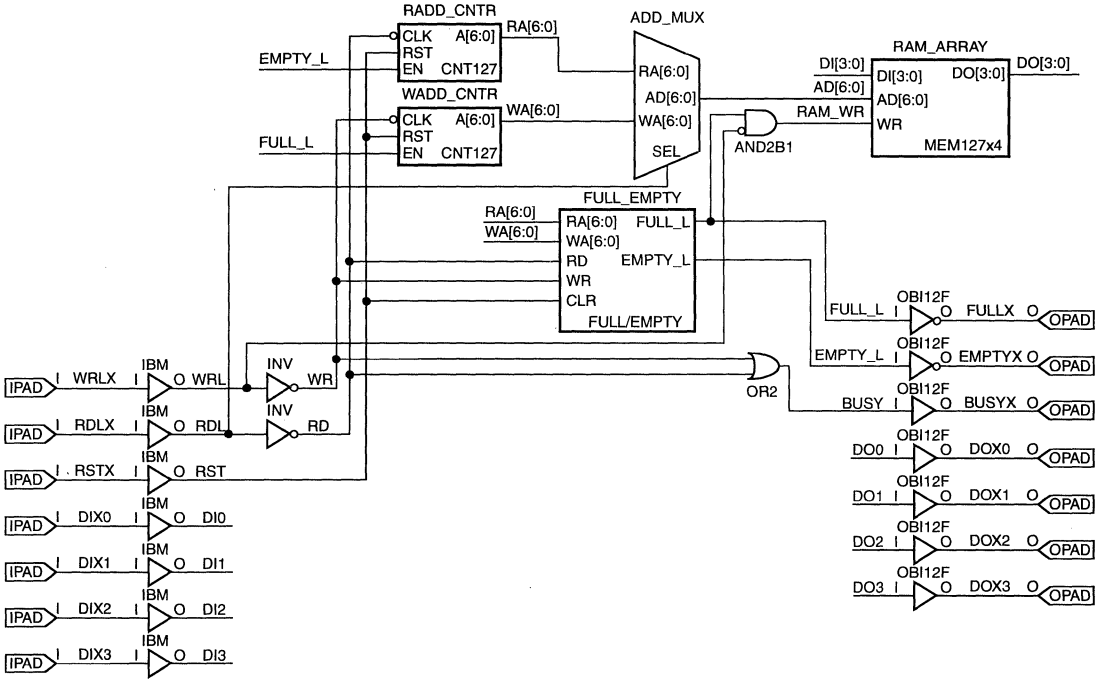


Figure 1. Top-Level Schematic

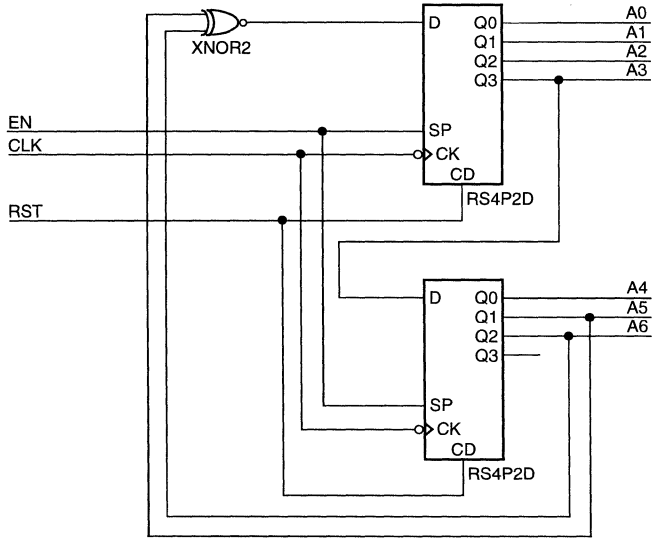


Figure 2. Read/Write Pointers

Functional Description (continued)

Input/Output Pads

All data, control, and flag signals are routed to I/O pins in the subject design. Usually, these signals are generated or used internally. In this design, the OB12F output buffers are used (see Figure 1). These buffers provide 12 mA of sink current and offer the fastest transition times. Versions of these buffers are also available that offer 3-state output enable (OE) signals for using the FIFO in a bus-oriented system.

Read/Write Address Pointers

The implementation of the read/write address pointers causes the design to be 127 locations deep, rather than 128 (see Figure 2). Conventional binary counters require significant amounts of routing and logic resources. In this design, a technique known as maximal length linear feedback shift registers (LFSR) is used for implementing the read and write address pointers.

LFSRs count in a pseudorandom sequence; however, this is of no consequence in a FIFO read or write address pointer. The only condition for proper operation is that the output sequences track each other. By using LFSRs, only 2 bits are needed in the feedback path. This requires less routing within the FPGA than conventional binary counter techniques. All other routing needed to implement the 7-bit address pointers is handled by the dedicated fast carry routing.

The operation of the read/write pointers is simple—for each RD pulse received, the read pointer advances one step; and for each WR pulse, the write pointer advances. Due to the pseudorandom nature of the output sequence, the pointers do not actually increment, but simply advance to the next state in the sequence. A reset pulse sets both counters to the all-zeros state and subsequently forces the empty flag active.

Address Multiplexer

The address multiplexer selects which pointer (read or write) is applied to the SRAM blocks' address inputs (see Figure 3). By using the RD signal directly as the SEL input, the read address is applied only during read operations; at all other times, the write address is applied.

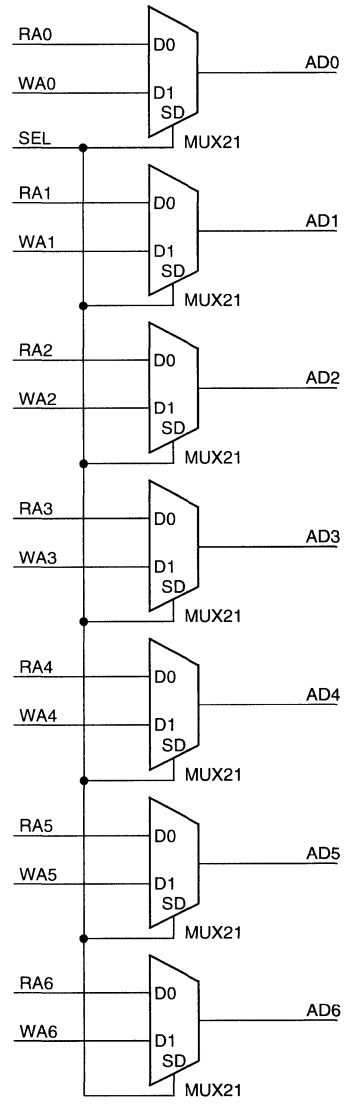


Figure 3. Address Multiplexer

Functional Description (continued)

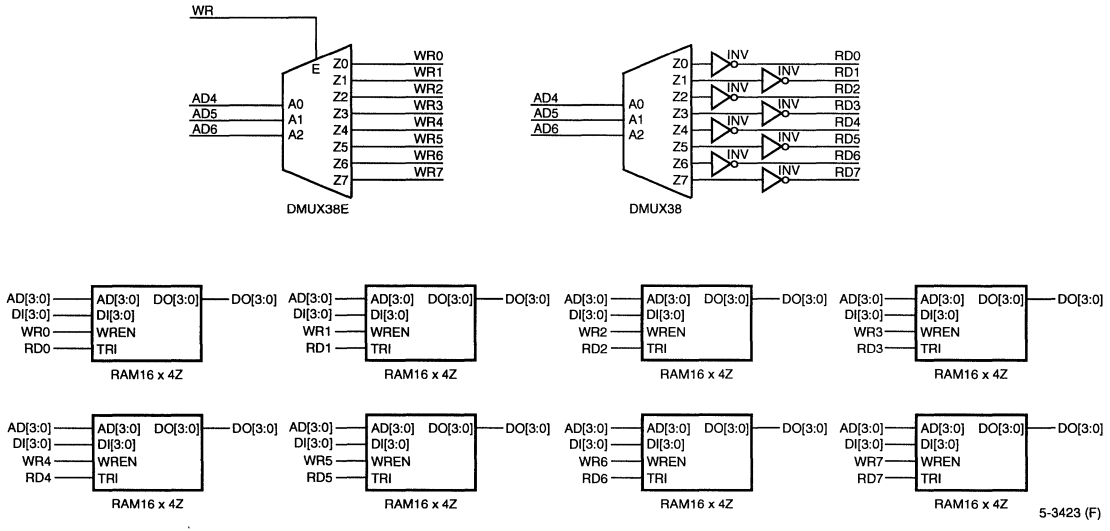


Figure 4. Memory Array

127 x 4 Memory Array

The memory array consists of eight RAM 16 x 4Z *ORCA* library elements. The Z in the library element's name indicates that the outputs are connected through 3-state buffers. This mechanism is used for cascading the elements in depth. The four low-order address lines from the address multiplexer (AD[3:0]) are routed to all eight of the elements. The upper three address lines (AD[6:4]) connect to two demultiplexers: one generates write pulses, and the other selects which element to turn on to the 3-stated output bus.

Note the inverters on the outputs of the read demultiplexer (see Figure 4); since the output enables on the RAM elements are active-low, these inverters are necessary. However, the optimizer in the ODS tools removes these inverters by changing the functionality of the demultiplexer outputs.

Flag Circuitry (Full, Empty, and Busy)

The flag circuitry in the FIFO design consists of a Read/Write address identity comparator and two flip-flops. The flip-flops are used to detect write and read/reset activity. In a FIFO, the read address equals the write address in only two cases: if the FIFO is full, or if the FIFO is empty.

To illustrate, treat the FIFO as empty. This implies that the last operation was either a read or a reset operation. As can be seen in the schematic (see Figure 5), the falling edge of either of these signals clocks a logical 1 into the read detect flip-flop. If the read and write addresses are equal (as checked by the WR_EQ_RD_ADDR comparator), the EMPTY_L signal goes active-low by NANDing together the outputs of the read detect flip-flop and the comparator.

Conversely, if the FIFO is full, the last operation must have been a write. In this case, the addresses are equal and the write detect flip-flop is active, generating the active-low FULL_L flag. Clearing either flag is accomplished by asynchronously clearing the respective flip-flop. For example, reading from a full FIFO makes it **not** full because the read signal resets the write detect flip-flop. This, in turn, clears the write flag and advances the read pointer.

The full flag is also ANDed with the WR signal prior to the write input on the SRAM block. This prevents any additional writes to the SRAM after a full condition is detected, preserving the last data written in.

The busy flag is simply the logical OR of the RD and WR signals. This signal indicates that the FIFO is currently busy. It is important to check this flag prior to any read or write operation to make sure the FIFO is available. Arbitration circuitry can also be built upon this signal.

Functional Description (continued)

Due to the pseudorandom nature of the address scheme used, half-full and/or half-empty flags would be difficult to implement. There is a 16 x 32 bidirectional FIFO design on the Lucent Technologies FPGA BBS that shows how such a flag can be implemented by subtracting the read pointer from the write pointer and checking for a value equal to exactly one-half of the total number of address locations. In this situation, traditional binary counters are used. This enables the subtraction mechanism to function in the half-full/half-empty flag generation logic.

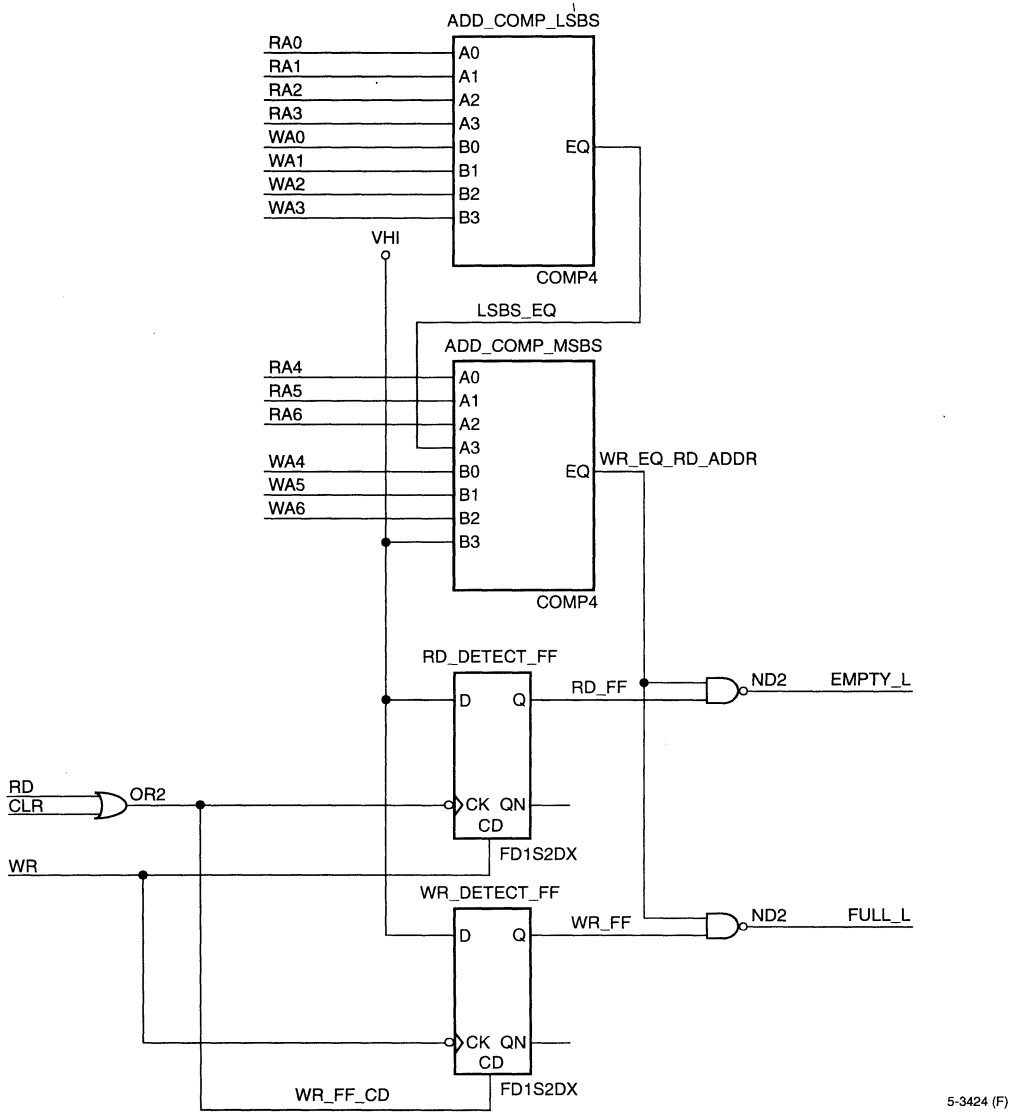


Figure 5. Flag Circuitry

Functional Description (continued)

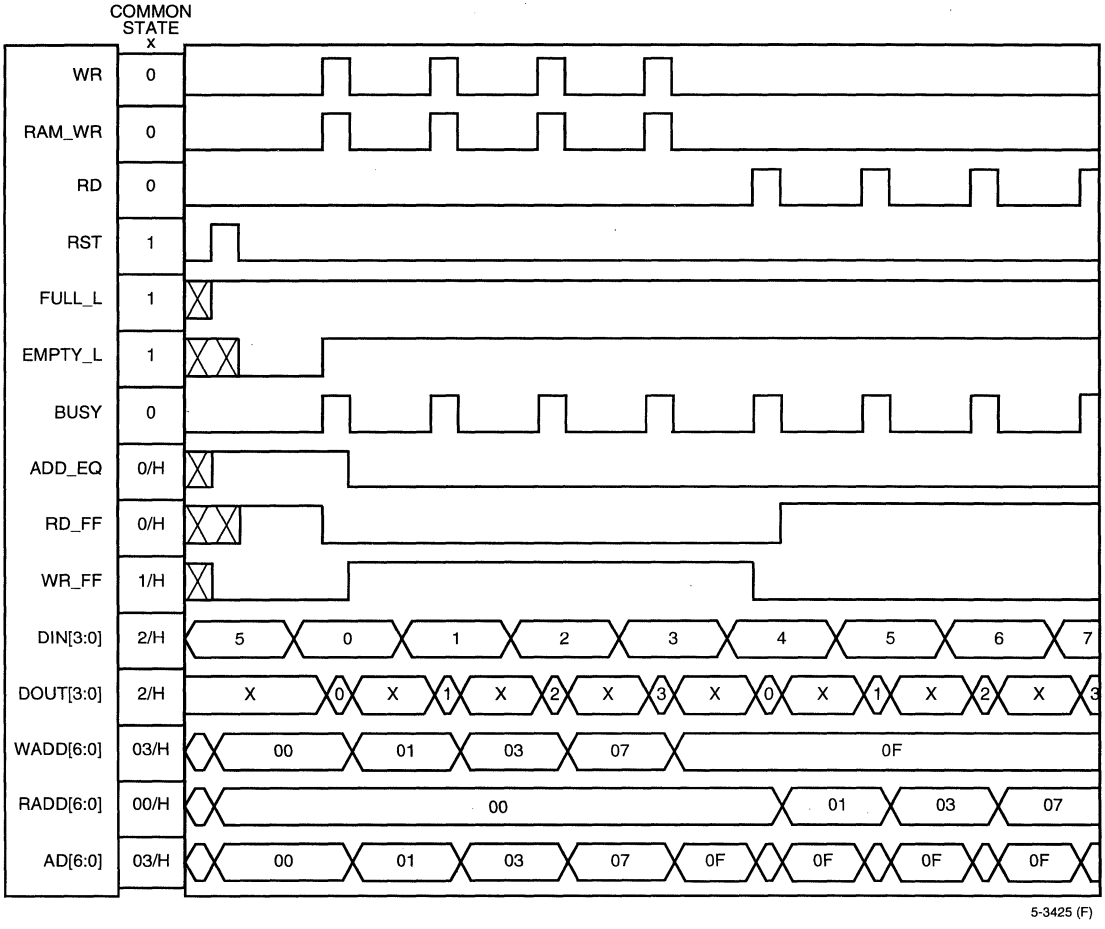


Figure 6. Functional Simulation Waveforms

Simulation

Included with the *ORCA* design is a simulation command file called FSIM (no extension). The file is written for use with the *ViewSim* simulator from Viewlogic Systems. It verifies the read and write cycle times and checks for proper operation of the full, empty, and busy flags. By using the back annotation feature in the *ORCA* Foundry Development System, the actual timing parameters for the routed design can be used to verify critical timing parameters in the design.

Designing a Data Path Circuit in *ORCA* Field-Programmable Gate Arrays

Introduction

Field-Programmable Gate Array (FPGA) densities have grown from a few thousand gates to 40,000 gates and beyond as FPGA applications, in turn, are becoming more data path oriented. Smaller FPGAs are used primarily in data interface functions consisting of address decoding and some combinatorial logic. However, with many thousands of available gates, designers can now implement advanced data path functions such as ALUs, FIFO, compression and encryption circuits, and other LSI functions. To effectively realize these functions in an FPGA, the designer must choose an architecture optimized for data path designs, such as the Optimized Reconfigurable Cell Array (*ORCA*) FPGAs from Lucent Technologies.

Data path logic is typically part of any modern digital-electronic system, along with control, interface, and glue-logic functions. Organized by n-bit-wide buses, data path logic is the subsystem that interfaces and processes data within a larger digital system. It usually holds the largest influence on overall FPGA performance, functionality, and density.

To illustrate why *ORCA* FPGAs are the optimal choice for data path design, a simple data path design will be used as an example which implements an *ORCA* FPGA using the *ORCA* Foundry Development System. The example design will highlight many of the *ORCA* FPGAs' features.

Features

- 3,500 to 40,000 usable gates
- Over 384 user-definable I/Os
- Nibble-wide logic and routing architecture
- Hierarchical and flexible routing resources
- Advanced Programmable Function Unit (PFU):
  — Combinatorial mode (up to eleven inputs)
  — SRAM mode (64 bits)
  — Ripple mode (arithmetic functions with fast carry)
- Large number of registers (four flip-flops per PFU)
- Up to eight global low-skew (<2.0 ns) clocks
- Internal 3-state drivers for buses
- High-speed direct routing from registers to and from I/O pins
- Flexible I/O modes:
  — TTL or CMOS selectable/pin
  — Active, passive, or 3-state
  — Programmable slew rates
- JTAG boundary-scan testability

Features (continued)

Each of these features is a necessary ingredient for efficient data path designs in FPGAs, and Lucent Technologies' *ORCA* family of FPGAs offers this combination of features at a wide range of gate densities.

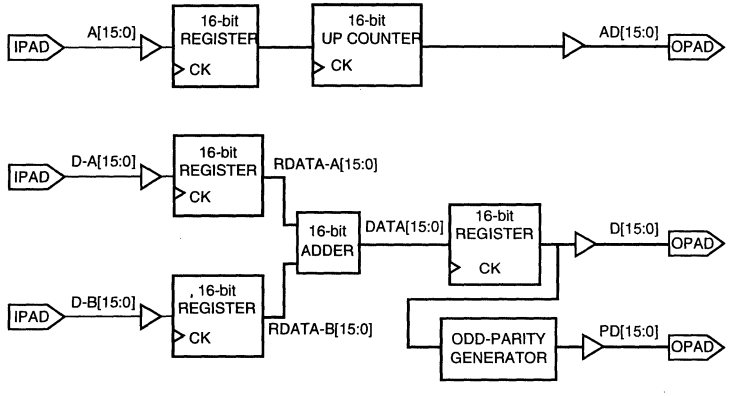


Figure 1. Example Data Path Design Block Diagram

ORCA Foundry Development System Design Flow

The first step in this example data path design is to capture the schematic of the block diagram shown in Figure 1. The *Viewlogic* design flow is shown in Figure 2. *Viewlogic*'s *ViewDraw* tool was used to capture the design using standard logic, interface, and *ORCA* architectural-specific elements from the Lucent Technologies *ORCA Viewlogic* library.

The *ORCA* library includes over 250 elements, approximately 50 of which are specific to the *ORCA* architecture and data path design. These elements include fast counters, fast adders, registers, accumulators, and multipliers of various bus widths. By using these elements, the design will more effectively utilize the *ORCA* architecture and achieve better performance than FPGAs without data path optimized architectures.

For this design example, a number of these elements were used to create a simple data path design. The bus structure is 16 bits wide, so 16-bit fast counters, 16-bit fast adders, and 16-bit registers were used. These data path elements use a unique ripple-mode feature of the *ORCA* programmable function unit (PFU) or logic cell shown in Figures 3 and 4.

The ripple mode enables the PFU to implement nibble-wide arithmetic functions with high-speed carry lines for cascading and building n-bit-wide functions. For example, a 16-bit adder only takes four PFUs configured in ripple mode and connected with high-speed carry lines. Note that a PFU is a subset of an *ORCA* programmable logic cell (PLC), which consists of a single PFU and its associated routing resources.

Also unique to *ORCA* architecture are carry lines that can be connected on any of the four sides of the PFUs. As a result, PFUs only need to be adjacent to implement wide functions. Other architectures require logic cells to be placed in a row or column to cascade fast-carry lines. This inhibits the software's place and routing options later in the design flow.

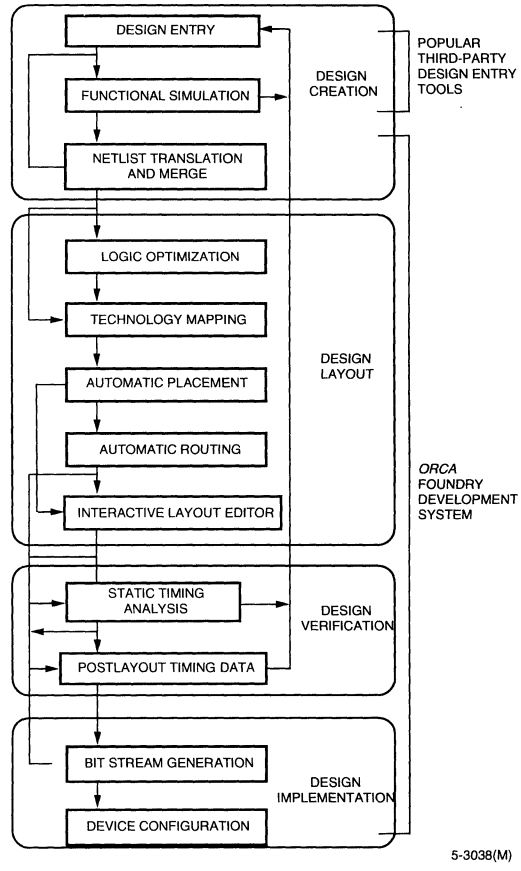


Figure 2. *ORCA* Foundry Development System *Viewlogic* Design Flow

ORCA Foundry Development System Design Flow (continued)

The nibble-wide architecture is prevalent in all aspects of the *ORCA* architecture. All routing lines, look-up tables, registers, and I/Os are in groups of four. As a result, designs with wide bus widths can be easily realized. Furthermore, the specialized elements within the *ORCA* library implement the data path functions in the most optimal manner with respect to mapping and placement. These elements ensure that the function is placed into an *ORCA* chip array with all the required PFUs next to each other for reduced delay, cascaded fast carries, and overall higher system performance.

After the logic of the design has been captured, the interface to the printed-circuit board must be designed. *ORCA*'s programmable interface cell (PIC) is designed to effectively route 4-bit-wide buses—yet another example of the nibble-wide architecture.

Table 1. Input/Output Cell Options

Input	Option
Input Levels	TTL/CMOS (selectable per pin)
Input Speed	Fast/Delayed
Float Value	Pull-up/Pull-down/None
Output	**Option**
Output Drive	12 mA/6 mA or 6 mA/3 mA
Output Speed	Fast/Slewlim/Sinklim
Output Source	FF-Direct Out/General Routing
Output Sense	Active-high/-low
3-State Sense	Active-high/-low (3-state)

Each PIC contains four programmable I/Os, as shown in Figure 5. These are highly versatile I/Os that can be configured as inputs, outputs, or bidirectional 3-state buffers. Inputs can be TTL, CMOS, or programmed for passive pull-up or pull-down. Another important feature for data path designs is high-speed clock-to-output and input-to-clock specifications. *ORCA*'s architecture provides for this capability with dedicated, special routing lines that connect PFU registers on the edge of a chip directly with PICs. Table 1 illustrates the various options for a PIC.

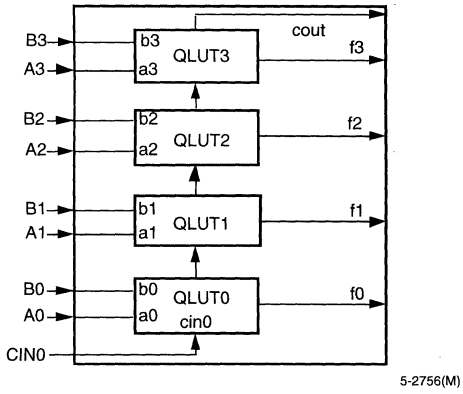


Figure 3. Ripple Mode

ORCA Foundry Development System Design Flow (continued)

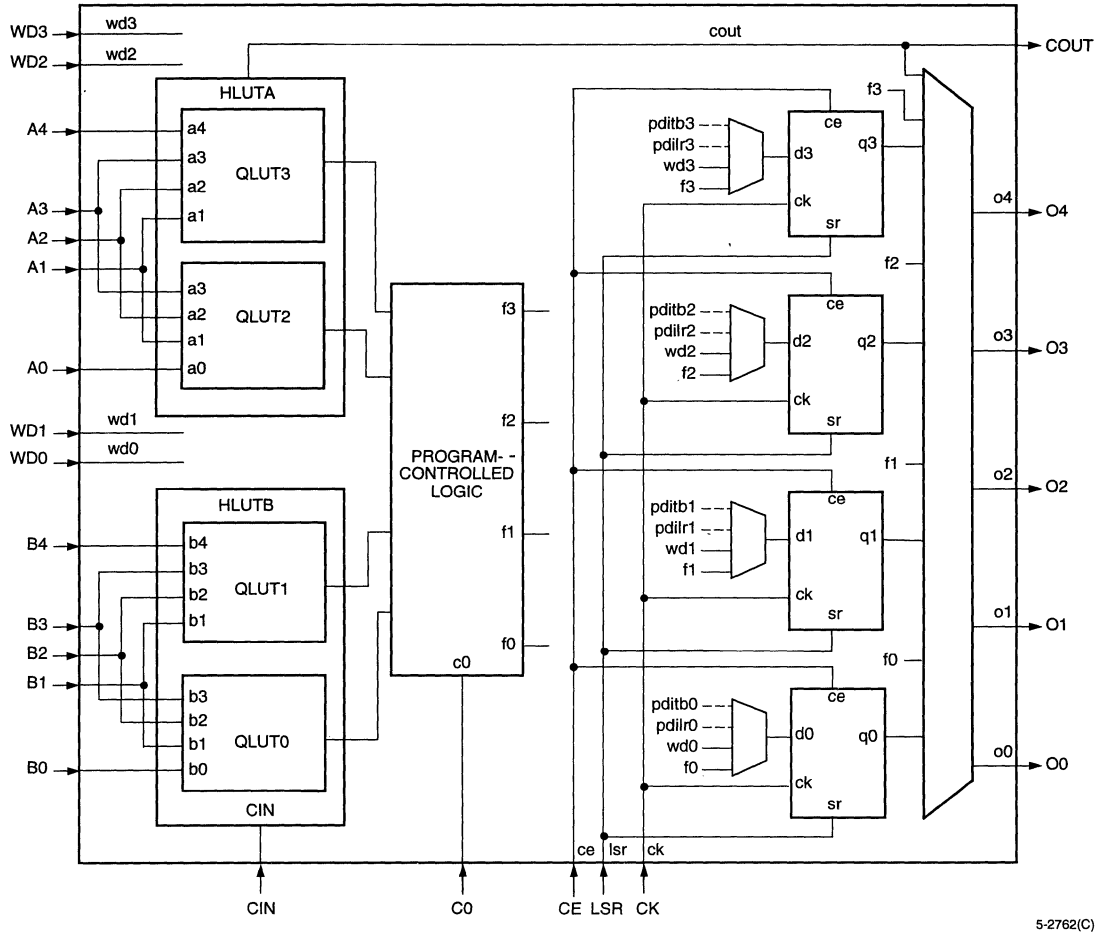


Figure 4. Simplified PFU Diagram

To use these various options, the designer must choose the library element that will implement the desired function. In addition, to ensure that the FPGA pinout matches the printed-circuit board pads, certain attributes (such as pin assignment or placement) are placed on the schematic.

After creating the schematic, the designer must verify the design's functionality. Lucent Technologies provides a unit-delay model for its library elements for prelayout simulation, in addition to a flow for back-annotating delay data after routing. By utilizing these models with *Viewlogic*'s *ViewSim* simulator and *ViewWave* waveform analyzer, a designer can guarantee that the design will function per the specification.

ORCA Foundry Development System Design Flow (continued)

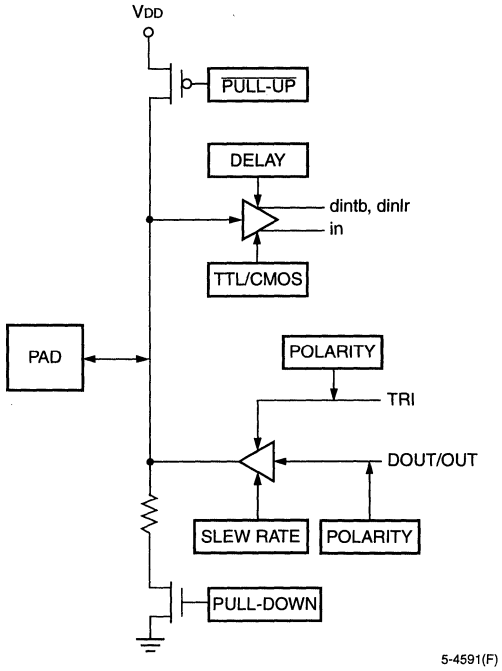


Figure 5. Simplified Diagram of Programmable I/O Cells

The next step in the design flow is to read the design netlist, which is in *Viewlogic*'s proprietary netlist WIR format, into the *ORCA* Foundry software. The software optimizes or reduces the random logic (optional), maps the library elements into PFUs and PICs, places the PFUs and PICs into a preselected *ORCA* device type, and finally routes the PFUs and PICs to realize the circuit design.

The router creates clock distribution trees for low-skew clocking. The *ORCA* architecture allows any signal to be a global clock signal, with up to eight global clock signals. The clock skew for *ORCA* FPGAs is typically less than 2 ns.

After mapping, placing, and routing are completed, a static-timing report is created. This report is based on worst-case conditions of the power supply voltage and temperature. The report is extremely informative, and allows the designer to determine if the design meets timing requirements.

The final step in the design process is to create the FPGA's configuration bit stream file, which defines the I/O functionality, logic, and interconnections. This file is loaded into the FPGA using one of the configuration modes detailed in the *ORCA* data sheet.

Example Data Path Design Results

The example data path design uses a total of 157 library elements, including 64 latches, 51 input buffers, and 34 output buffers, and it contains 246 signals. Table 2 summarizes some key design results.

An ATT1C05 *ORCA* FPGA with 5000 usable gates was used to verify the design. The design is fairly compact, using only 22 of the 144 possible PLCs.

Table 2. Example Data Path Design Results

Density Performance	
Number of PLCs Used	22
Number of I/Os	85
Speed Performance*	
Maximum Overall Clock Speed	46.2 MHz
Maximum Counter Speed	48.5 MHz

* Worst case.

Summary

Optimized for data path design applications, the *ORCA* FPGA architecture offers high-density logic, abundant I/O pins, and high speed in a single programmable chip. When using a schematic design entry flow like the one shown in this design example, a rich library of optimized elements is offered that takes advantage of such unique *ORCA* FPGA architectural features as the PFU ripple mode used for building very fast n-bit counters. With gate densities of 40,000 usable gates, *ORCA* FPGAs are an excellent alternative to masked gate arrays and discrete logic for today's advanced systems.

Designing High-Speed Counters in *ORCA* FPGAs Using the Linear Feedback Shift Register Technique

Introduction

This application note contains information on designing high-speed, FPGA-based counters using the maximal-length linear feedback shift register (LFSR) technique. The advantages of this technique over conventional binary counters are illustrated through a 15-bit LFSR reference design. When placed and routed in a Lucent Technologies *ORCA* 1C Series FPGA, the design achieves a 137.1 MHz count speed as reported by the Development System Static Timing Analyzer. For the 2C Series of FPGAs, even higher speeds are obtainable. In the laboratory, count speeds in excess of 250 MHz have been observed for the 1C Series.

Background on LFSR

LFSR counters are formed by feeding back certain bits, called taps, through an exclusive OR (XOR) or exclusive NOR (XNOR) gate to form the input to the least significant stage of a shift register chain. By selecting the correct taps, an N-bit counter circuit can be constructed that sequences through $2^N - 1$ unique states before the sequence repeats. In a traditional binary counter, 2^N unique states are visited before repeating. For example, an 8-bit binary up-counter sequences through 256 states from $00 to $FF (255) in increasing order. An 8-bit LFSR counter visits 255 states, and the order is pseudorandom.

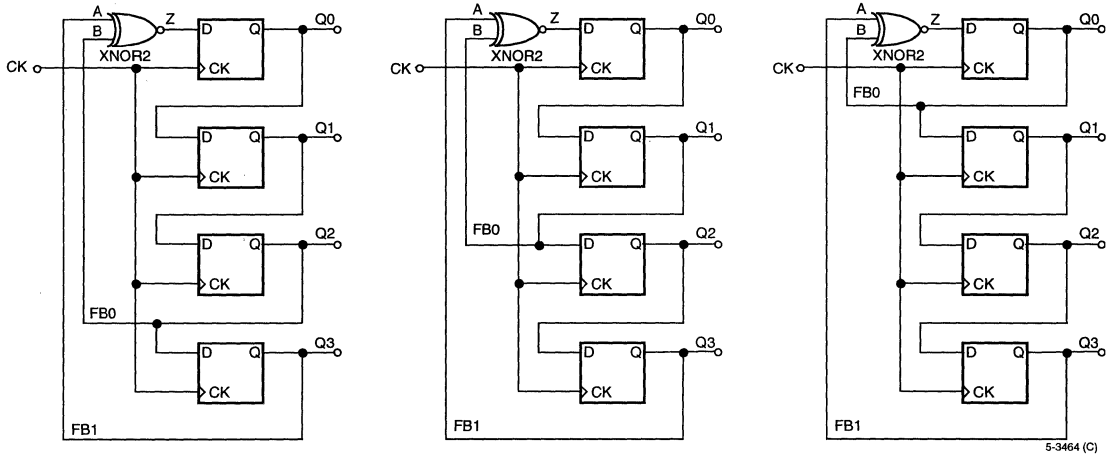
The pseudorandom nature of the output sequence precipitates the use of LFSRs in such applications as random number generators and encryption/decryption devices. When used as counters, LFSRs can be used in applications where the nonsequential nature of the output states is of no consequence. These include circular buffer address generators, FIFO read/write pointers, and programmable frequency dividers. Other counting circuits can use the technique if the LFSR's output may be decoded through a look-up table (ROM) or attached processor to its binary count equivalent.

The one missing state in an LFSR counter is called the lock-up state. Entering this state would lock the counter into an infinitely repeating cycle of all zeros or all ones, depending upon whether XOR or XNOR feedback was used. Since each register in an *ORCA* device can be **individually** configured to be either set or reset after powerup or a set/reset pulse, either method may be used. However, since the default powerup condition is all zeros, XNOR feedback is used in the counter design given here.

Understanding Maximal Length

As mentioned previously, only certain feedback taps will result in a maximal-length sequence. Consider the three 4-bit LFSR circuits of Figure 1. Figure 1a uses taps from outputs Q2 and Q3; 1b uses Q1 and Q3; and 1c uses Q0 and Q3. Only the circuits of 1a and 1c result in maximal-length sequences of 15 ($2^4 - 1$) states. Circuit 1b sequences through only six unique states before repeating. Another key point illustrated here is that there can be more than one feedback configuration that will result in a maximal-length sequence for a given length counter.
The 15-bit reference design detailed in this application note uses taps at bit 15 and bit 14 to achieve the maximal length of 32,767 states.

Understanding Maximal Length (continued)



State	Q3	Q2	Q1	Q0	XNOR	HEX
0	0	0	0	0	1	$0
1	0	0	0	1	1	$1
2	0	0	1	1	1	$3
3	0	1	1	1	0	$7
4	1	1	1	0	1	$E
5	1	1	0	1	1	$D
6	1	0	1	1	0	$B
7	0	1	1	0	0	$6
8	1	1	0	0	1	$C
9	1	0	0	1	0	$9
10	0	0	1	0	1	$2
11	0	1	0	1	0	$5
12	1	0	1	0	0	$A
13	0	1	0	0	0	$4
14	1	0	0	0	0	$8
Then sequence repeats . . .						

State	Q3	Q2	Q1	Q0	XNOR	HEX
0	0	0	0	0	1	$0
1	0	0	0	1	1	$1
2	0	0	1	1	0	$3
3	0	1	1	0	0	$6
4	1	1	0	0	0	$C
5	1	0	0	0	0	$8
Then sequence repeats . . .						

State	Q3	Q2	Q1	Q0	XNOR	HEX
0	0	0	0	0	1	$0
1	0	0	0	1	0	$1
2	0	0	1	0	1	$2
3	0	1	0	1	0	$5
4	1	0	1	0	0	$A
5	0	1	0	0	1	$4
6	1	0	0	1	1	$9
7	0	0	1	1	0	$3
8	0	1	1	0	1	$6
9	1	1	0	1	1	$D
10	1	0	1	1	1	$B
11	0	1	1	1	0	$7
12	1	1	1	0	0	$E
13	1	1	0	0	0	$C
14	1	0	0	0	0	$8
Then sequence repeats . . .						

Figure 1. Three 4-bit LFSR Circuits Using Different Taps

Advantages of the LFSR Technique over Conventional Binary Counters

In a conventional binary counter, each successive bit requires knowledge of all previous bits to determine when to toggle. For example, the eighth bit of an 8-bit counter needs inputs from the seven previous stages in order to define its toggle condition. This same counter requires 8 + 7 + 6 + 5 + 4 + 3 + 2 + 1 or a total of 36 terms to define the toggle states for all bits in the counter. In an FPGA implementation, this causes two problems. First, the number of inputs into the basic logic elements in FPGAs is limited compared to the wide sum of products that feed the macrocells of complex programmable logic devices (CPLDs). There are additionally a finite number of routing resources available in the FPGA to accommodate these terms. These routing resources have delays that vary with length and loading. Fast look-ahead carry schemes can mitigate these problems, but sacrifice additional registers or other logical resources to implement the look-ahead circuitry.

Conversely, LFSR counters need only to route the feedback taps and the bit-to-bit shift chain. For most counter lengths, only two or three feedback taps are required to achieve a maximal-length sequence. In an FPGA, this minimal requirement does not tax either the available routing or the width of the inputs to the logic cells. Within an *ORCA* FPGA, this advantage is further augmented by the presence of the fast carry routing resources. These resources take care of the bit-to-bit shift chain, leaving only the feedback taps to general-purpose routing.

Another advantage of LFSR counters over conventional binary counters is in the generation of simultaneous switching noise. While the noise generated by a conventional binary counter is coherent and periodic, the pseudorandom output sequence of an LFSR spreads the noise out over the entire spectrum from dc to the input clock frequency. For example, a 15-bit binary counter produces a significant noise spike when rolling over from all ones to all zeros ($7FFF→ $0000). Smaller, but still significant, spikes occur at transitions such as $3FFF → $0000, $1FFF → $0000, etc. On printed-circuit boards that contain sensitive analog circuitry, this noise can be an important consideration.

Motivation for the Reference Design

The 15-bit length of the reference design was selected for the following reasons:

- The feedback taps for a 15-bit LFSR counter are bits 14 and 15. The fact that they are adjacent bits allows for identical type and length of routing resources to be used for both taps. This minimizes skew and ensures maximum performance.
- 32K x 8 SRAMs are very common and cost effective. These devices have 15 address inputs, making them attractive as the storage elements in such ideal LFSR applications as FIFOs and circular buffers.
- A 16-bit LFSR counter requires four taps (either bits 16 + 12 + 3 + 1 or bits 16 + 15 + 13 + 4) to realize a maximal-length sequence. While a 16-bit LFSR counter can indeed be implemented in an *ORCA* device and achieve similar performance to its 15-bit counterpart, it lacks the simple elegance of the 15-bit solution. An advantage of a 16-bit design would be that it would consume four complete *ORCA* PFUs (four registers per PFU).

Reference Design Discussion

The schematic for the reference design is included in Figure 2. Note the use of the @comp attribute. The comp = SR1 attribute on the XNOR gate and the first RS4S3I 4-bit shift register forces the XNOR to be placed in the same PFU as the shift register. Therefore, no routing is required between the XNOR and the first register in the shift chain; the internal connection between the LUT output and the register's input is used. In general, all the RS4S3I shift register elements will be placed in close proximity. This enables the dedicated, high-speed carry routing resources to connect the 4-bit shift registers into the 15-bit chain.

As an interesting side note, the placer can actually bend a corner with the placement if it needs to. It is able to do this because of the symmetrical and homogeneous nature of *ORCA's* logic cells and routing resources. Performance is exactly the same as if the placer had put all the PFUs on a single row or column.

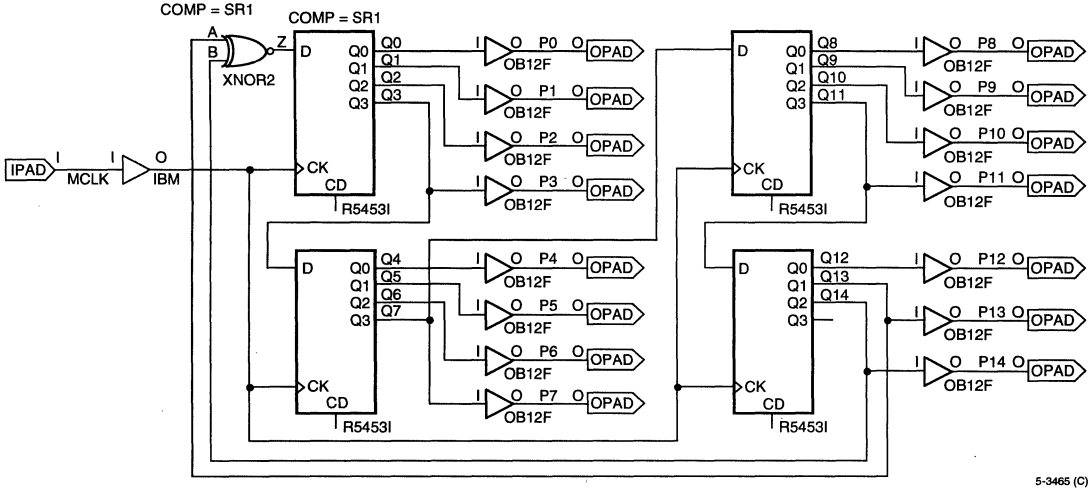


Figure 2. Schematic of 135 MHz 15-Bit LFSR Counter

ORCA FPGAs Integrate Datacom's Paths

Introduction

Traditional datacom applications have largely relied on discrete logic components for such supporting circuits as data registers; first-in, first-outs (FIFOs); last-in, first-outs (LIFOs); control; and data paths. Advanced field-programmable gate array (FPGA) architectures, however, offer the datacom system designer on-chip SRAM functions plus control logic to allow cost-effective integration of the necessary functions. Equally important, circuit designs can be custom tailored to meet a datacom application's special requirements.

High System Performance with Low Power Consumption

FPGAs with on-chip SRAM and supporting logic are especially valuable because they hand datacom system designers the necessary integrated silicon to implement the storing, checking, and control functions used for processing small packets of data. Because all the required functions can be performed on-chip, higher system frequencies can be attained at a lower power consumption.

Unfortunately, to create very wide RAM functions, the designer must use combinatorial logic to link the SRAM blocks. As the width of the RAM increases, the combinatorial logic eventually becomes so wide that it makes more sense to use discrete RAM. A rule of thumb is that a RAM with a depth greater than 256 words should be implemented in discrete RAM. Such a rule should be used only as a guide. The final decision should be based on a number of constraints, including board space, power consumption, and system timing.

Off-the-shelf discrete ICs, such as FIFOs, are also available for datacom designs. However, these standard parts may not have the required timing or the necessary control. By using an FPGA, the designer can customize the function to meet the system's timing requirements, allow special controls, and access system-specific flags. The designer can also pare power and pc board real estate requirements.

ORCA Series devices are FPGAs that are currently available on the market with on-chip SRAM capability. The following table shows each device's SRAM capability.

Table 1. *ORCA* FPGAs with On-Chip SRAM

Device	Usable Gates	Latches/ Flip-Flops	Max User RAM Bits	User I/Os	Array Size
2C04	3,500—4,300	400	6,400	160	10 x 10
2C06	5,000—6,200	576	9,216	192	12 x 12
2C08	7,000—8,800	784	12,544	224	14 x 14
2C10	9,000—11,400	1024	16,384	256	16 x 16
2C12	12,000—14,600	1296	20,736	288	18 x 18
2C15	15,000—18,000	1600	25,600	320	20 x 20
2C26	22,000—26,000	2304	36,864	384	24 x 24
2C40	35,000—40,000	3600	57,600	480	30 x 30

ORCA FPGAs Dramatically Increase Memory Depth

ORCA FPGAs are programmable, with their configurability based on a look-up table (LUT) architecture. LUTs can be configured as a complex logic function, SRAM, or ROM. One logic element or programmable function unit (PFU) can be used to implement one 16 x 4, two 16 x 2, or one 16 x 2 SRAM block(s) with the remaining half of the PFU used for random logic consolidation.

In memory mode, each half-LUT (HLUT) is configured independently so that logic designers can opt for the 16 x 2 in one HLUT, a 16 x 4 using both HLUTs, or the 16 x 2 in one HLUT and a logic function of five input variables or less in the other HLUT. Two or more programmable logic cells (PLCs) in *ORCA* are used to increase memory depth to a value of greater than 16.

Deeper memories are more complex and require additional logic. To increase the memory depth, data outputs from each PLC are multiplexed. Combinatorial logic is required to access the write enables of the individuals PLCs. An example of a 32 x 4 RAM block would be created by using two 16 x 4 blocks in the *ORCA* FPGA. The 32 x 4 RAM module would require two *ORCA* PLCs in the 16 x 4 mode. The two blocks would share the same address and data lines, with each block having different write enables that have been decoded from a separate address signal.

The *ORCA* FPGA also has an advantage because within each PLC are four 3-statable buffers, called bidis, that can be used to create a 2:1 multiplexer. The outputs of the bidis would be tied together to create the 4-bit output bus, with the inputs to one set of bidis from one PLC RAM outputs and the inputs to the other set of bidis from the other PLC RAM outputs. The same address signal used to decode the write enables is also used to decode the bidi enable signals, thus enabling the appropriate RAM contents onto the 4-bit output bus.

Two or more PLCs are used again to increase an SRAM's word size, such as 16 x 8. The same address and write enable are connected to all PLCs, and data is different for each PLC. The number of address locations and the word size are increased by using a combination of the two techniques.

On-Chip SRAM FIFOs Synchronize Datacom Applications

On-chip SRAM-based FIFOs are particularly useful in network system designs, whether the network type is asynchronous transfer mode (ATM), fiber channel, token ring, fiber distributed data interface (FDDI), or Ethernet.

FIFOs synchronize data between two asynchronous systems and usually need dual-ported SRAM. In that case, separate read and write address lines simultaneously access the SRAM. On-chip SRAM in today's FPGAs is single-ported, however, so that only one set of address inputs goes to the SRAM. But converting a single-port SRAM into a dual-port type is easy. Read and write address lines are time-division multiplexed to form a single address input. The read count has the current read address, and the write count points to the current write location. Both counter outputs are multiplexed together to form the single address SRAM input.

A read accesses the SRAM and is then followed by a write, and so on. The FIFO processes each request in two clock cycles. The control block arbitrates between read and write. When a read and a write request occur simultaneously, the arbiter performs either a read before a write (if the FIFO isn't empty), or a write before a read (if the FIFO is empty). A comparator circuit checks the read and write count pointer outputs to detect either full or empty conditions for the FIFO. Control circuitry doesn't acknowledge a read request from an empty FIFO or a write request from a full FIFO. Though it is synchronous internally, the FIFO supports asynchronous handshakes to the request and data ports.

On-Chip SRAM FIFOs Synchronize Datacom Applications (continued)

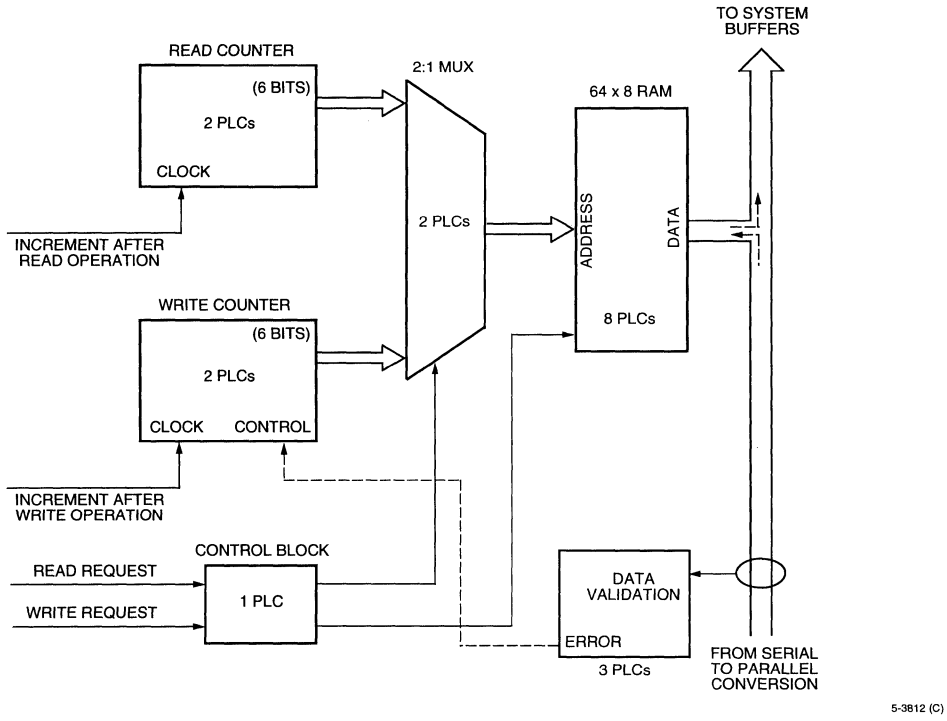


Figure 1. On-Chip SRAM FIFO (ATM Application)

This figure shows a more datacom-specific, on-chip SRAM FIFO used to hold an ATM data packet or cell. An ATM cell is constructed of 53 bytes. The first 5 bytes contain the header; the remaining 48 bytes hold the information field. At the user-network interface, the header is broken into the generic flow control (GFC), the virtual path identifier (VPI), the virtual channel identifier (VCI), the payload type identifier (PTI), the cell lost priority (CLP), and the header error control (HEC).

The first 5 bytes are broken up as follows: the GFC uses the 4 most significant bits of the first byte; the VPI uses the 4 least significant bits of the first byte and the 4 most significant bits of the second byte; the VCI uses the 4 least significant bits of the second byte, the entire third byte, and the 4 most significant bits of the fourth byte; the PTI uses the next 3 bits of the fourth byte; the CLP uses the least significant bit of the fourth byte; and the HEC uses the entire fifth byte.

The FIFO design in the ATM application holds the 53-byte cell and performs a cycle redundance check (CRC) on the 5-byte header. If the header is found to be error-free, the complete cell is forwarded to system buffering. But if an error is detected when the 5-byte header is verified, the complete 53-byte cell is immediately discarded when the write counter is cleared.

Based on a 6,000-gate ATT2C06 device, the 64-byte FIFO uses a total of 18 PLCs. The read and write counters each use two PLCs. Eight more are used for storage—one for the two-way control arbiter, two for the 2:1 multiplexer, and three for data validation.

The purpose of the single-cell FIFO implemented on-chip in the ATM application is to validate and discard any "corrupt" ATM cells before they are allowed to enter the network. The on-chip FIFO is used during that validation and is not intended to serve any large buffering or rate-decoupling purpose. Those functions are instead left to external discrete FIFO devices. The intent of the application is to screen out corrupt cells and pass only valid cells to the external FIFO(s) that may serve as system buffers.

On-Chip SRAM as a Register File

Another application for on-chip RAM is as a register file. A register file can be thought of as a set of registers that share a common multiplexed output. To be able to use on-chip RAM for register files, the outputs from different registers cannot be accessed simultaneously. Before on-chip RAM was available, designers would create a register file using individual flip-flops—an approach that clearly requires an inordinate amount of resources.

Some data networking products use proprietary system addressing schemes to move data from one user port to another within the unit. Such addressing schemes are likely to be tailored to the specific hardware architecture of the unit. Therefore, data conforming to certain standards must be translated to meet those addressing schemes so it can be used within the system. Likewise, when the data leaves the system, it must be translated back to meet the original standard. Register files can be used to translate the data.

This application explores one possible (and simple) translation, where the 4 most significant bits of the VPI and the entire GFC are replaced by a byte of routing information that is extracted from the on-chip RAM. The VPI is thus effectively expanded from 8 bits to 12 bits. The expansion is beneficial and, in most cases, necessary. If one user port (or end point) is allowed 256 possible VPIs, then a switch capable of supporting multiple end points should be able to identify and support a larger set of connections. In the example cited here, the additionally expanded 4 bits could contain a destination port number.

In the register file application, the translator uses the 4 most significant bits of the VPI as the address into a 16 x 8 on-chip RAM. They access a byte of data that, when used in conjunction with the 4 least significant bits of the VPI, creates a 12-bit VPI.

The circuit comprises four 2:1 multiplexers, a small control block, an 8-bit register, and a 16 x 8 RAM. When a cell of information is transmitted, the control block uses 4 bits of the header (the 4 most significant bits of the VPI) as the address to the RAM.

That address accesses 8 data bits, which are then used to replace the first byte of the cell. All 52 remaining cells are passed through an 8-bit register and are not altered. Inversely, when the cell leaves the switch, the data must be transformed back to its original state.

The above application requires seven PLCs within the *ORCA* device. A similar implementation using discrete logic clearly would not fully utilize the off-chip RAM and would require a large amount of board real estate. If the design was implemented in an FPGA that did not have the added advantage of on-chip RAM, the register file would require 128 flip-flops or 32 PLCs within the *ORCA* family of FPGAs.

ORCA FPGAs As Configurable DSP Coprocessors

Article reprinted from *App Review*, September 12, 1994.

Introduction

The advent of fast, high-density FPGAs has opened up an exciting new area of applications: configurable digital signal processing (DSP) coprocessors. Many simple DSP functions can run much faster in dedicated hardware than through the traditional fetch-decode-execute-store software approach of conventional DSP microprocessors. Infinite, in-circuit reprogrammability and on-chip SRAM make Lucent Technologies' optimized reconfigurable cell array (*ORCA*) FPGAs ideal for such applications.

FPGA Advantages in DSP Functions

The advantage of special-purpose hardware for DSPs is evidenced by the proliferation of dedicated, fixed-function devices. Chips now available include digital filters, binary correlators, numerically controlled oscillators, etc. These fixed-function devices offer levels of performance that far outstrip the capabilities of today's general-purpose DSPs. SRAM-based FPGAs used as DSP coprocessors extend this concept; infinite reprogrammability allows them to be changed in-circuit to perform any number of different tasks.

ORCA's on-chip memory allows data and/or coefficients (such as data-windowing functions) to be stored or modified. By connecting the FPGA directly to a general-purpose DSP (such as Lucent Technologies' DSP32C), specific functions can be off-loaded to the FPGA for high-performance, hardware-based processing. In this case, the FPGA may be configured directly by the host processor in the asynchronous peripheral mode. The configuration data-bus pins on the FPGA can double as the DSP-to-FPGA data pins after configuration is complete. The extremely flexible routing resources of *ORCA* allow the data bus pins to be locked in place prior to placing and routing the FPGA. Unlike other FPGAs, this preplacement of pins will not have a significant effect on performance or on the ability of the tools to fully route the design.

Limitations

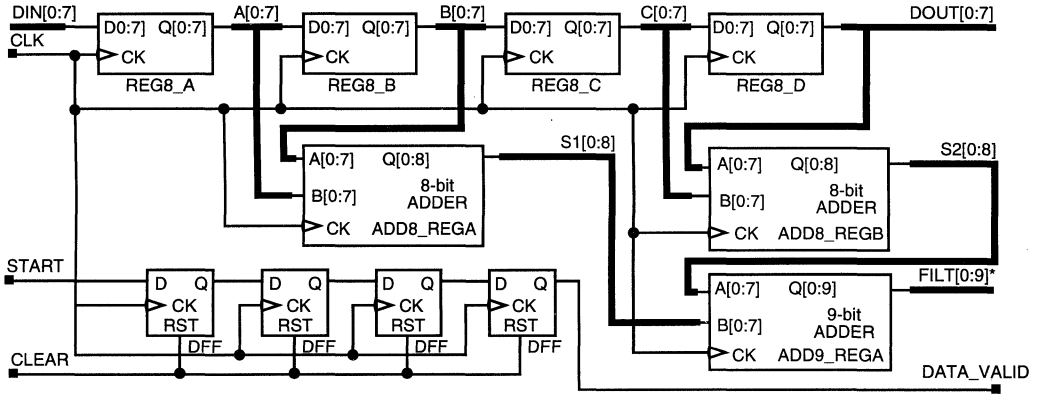
Many DSP algorithms require a multiply-and-accumulate (MAC) function. Multiply and divide functions are easiest to implement in FPGAs if one of the operands is a power of two. By limiting multiplies and divides to powers of two, the functions can be implemented with simple shift-left and shift-right circuits. An example of this technique is shown in the four-stage moving average filter shown in Figure 1.

Before *ORCA*, multiply and divide functions were difficult (if not impossible) to implement in FPGAs due to speed and/or density considerations. Lucent Technologies offers a detailed application note, *Implementing and Optimizing Multipliers in* ORCA *FPGAs* (AP94-035FPGA) that discusses the speed and density trade-offs encountered. The application note shows two 8 x 8 integer multipliers; one is optimized for density (18 PLCs), the other for speed (over 50 MHz). A copy of this application note can be obtained from your local Lucent Technologies sales professional.

Another problem with the FPGA-based DSP coprocessor solution is floating-point calculations. While the nibble-wide *ORCA* architecture easily handles byte, word, and even double-word integers, floating-point calculations are significantly more complex. Fortunately, the number of usable gates in FPGAs and the size and complexity of library macros continue to grow. These factors will allow more and more logic—including multiply, divide, and floating point circuitry—to be packed into FPGAs.

Reference Design—Moving Average Filter

Figure 1 shows the schematic for a four-stage moving average filter. The moving average filter (or tapped delay line) represents one of the simplest yet most common DSP applications. It is a nonrecursive or finite-impulse-response (FIR) filter. It is finite because the output will become constant N steps or taps after the input becomes constant, and there is no recursion (i.e., feedback). The output depends only on the input and the coefficients; no output terms are fed back to formulate the output.



* Only the MSBs (bits 2—9) are used. The divide-by-4 function is performed in this manner.

Figure 1. Four-Stage Moving Average Filter

Moving average filters are typically used for data smoothing. Nearby points (in this case four) are averaged. In the reference design, this is represented as (A + B + C + Dout)/4. The general form of this equation is:

$$y(n) = \sum_{m=0}^{N-1} h(n)x(n-m)$$

where y(n) is the filtered output sequence, N is the number of data points, and m is an index number assigned to a data point (starting with zero) for a point associated with n. More sophisticated filters use a function in place of the coefficient or weighting factor h(n). However, in this example, h(n) is constant (1/4) for all n and may be moved to the left of the summation sign in the generalized equation:

$$y(n) = 1/4 \sum_{m=0}^{3} x(n-m)$$

The reference design consists of four 8-bit pipeline registers, two registered 8-bit adders, and one registered 9-bit adder. The four D flip-flops at the bottom left of the schematic are used to generate the DATA_VALID output that signals when the first data word has reached the last stage of the pipeline. After a latency of six clock cycles (four to fill the pipeline and two to clock real data through the registered adders), the original input data sequence appears at the DOUT outputs and the filtered version appears at the FILT outputs.

The FILT outputs consist of the eight MSBs of the 9-bit adder output; by left-justifying this data and only considering the 8 MSBs, the divide-by-four function is realized. This takes advantage of the fact that a logical shift-right is equivalent to dividing by two in binary arithmetic.

The response of the filter to a noisy sinusoidal input is shown in Figure 2. The data-smoothing (low-pass) properties of the filter are clearly evident.

This entire filter circuit requires only 16 programmable logic cells (PLCs) in an *ORCA* device. Therefore, the design consumes only 16% of the smallest *ORCA* device, the 100 PLC ATT1C03. The design takes advantage of the dedicated fast-carry resources in *ORCA* to implement the adder functions. This allows the design to operate at speeds in excess of 60 MHz. Adding additional bits to the width of the data words or increasing the number of taps will only add approximately 1 ns of delay per bit/tap.

Reference Design—Moving Average Filter (continued)

For buffering data coming into or going out of the filter, each PLC in *ORCA* can be used as one 16 x 4 memory cell, two 16 x 2 memory cells, or one 16 x 2 memory cell with the other half PLC used as a five-input logical function. These memory cells may be easily organized into larger blocks, or FIFOs, to equalize the data flows to and from the filter's data sources and destinations. Lucent Technologies offers a detailed application note on implementing FIFOs in *ORCA*, and several uni/bidirectional FIFO designs can be downloaded from the Lucent Technologies FPGA Bulletin Board System: 610-712-4314.

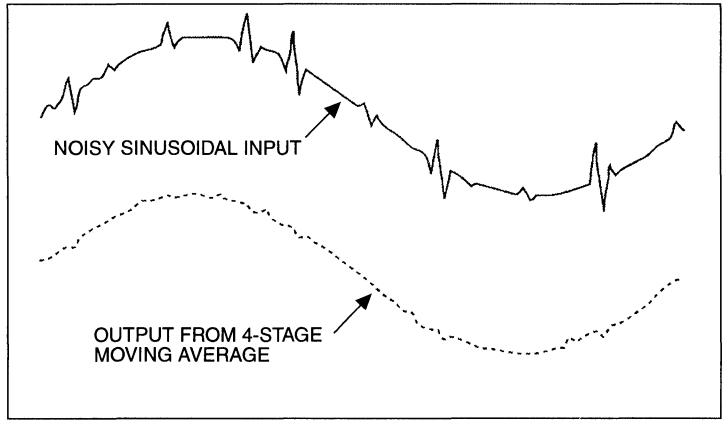


Figure 2. Response of Four-Stage Moving Average

Conclusion

Many DSP functions can see order-of-magnitude performance enhancements by executing them in dedicated hardware rather than through the traditional software approach. SRAM-based FPGAs may be configured directly by attached DSP chips to perform a wide range of such tasks. Examples include moving average and other FIR filters, data pre/postconditioning, pipeline registers, correlators, and data extractors (a complete application note that includes examples of several of these functions is currently being developed). In addition, *ORCA*'s on-chip memory capability allows for FIFO/LIFO buffers, control stores, and modifiable high-speed coefficient storage. The speed, density, in-circuit reconfigurability, and on-chip memory capabilities of Lucent Technologies' *ORCA* FPGAs make them ideal devices in this type of application.

ISA Bus Plug and Play in an FPGA

Article reprinted from *App Review*, December 12, 1994.

Introduction

With Lucent Technologies' inexpensive *ORCA* FPGAs, designers can add custom, fully compliant Plug and Play functionality to their PC add-in card designs.

Background on Plug and Play

The Plug and Play standard for ISA bus-based computers was developed to ease the installation and configuration of PC add-in boards. Before Plug and Play, users were required to set dip switches to resolve memory, I/O, DMA, and IRQ resource conflicts. Cards that implement Plug and Play are automatically isolated, assigned a unique handle, queried about what resources they require, and then assigned nonconflicting system resources.

This process is accomplished in one of two ways. In newer PCs, manufacturers are adding Plug and Play support to the system BIOS. In existing non Plug and Play BIOS machines, the add-in card vendor supplies specialized device-driver software. Virtually all major PC vendors are offering, or are planning to offer, systems with Plug and Play in the BIOS. In addition, *Microsoft's Windows* '95 operating system provides full Plug and Play support.

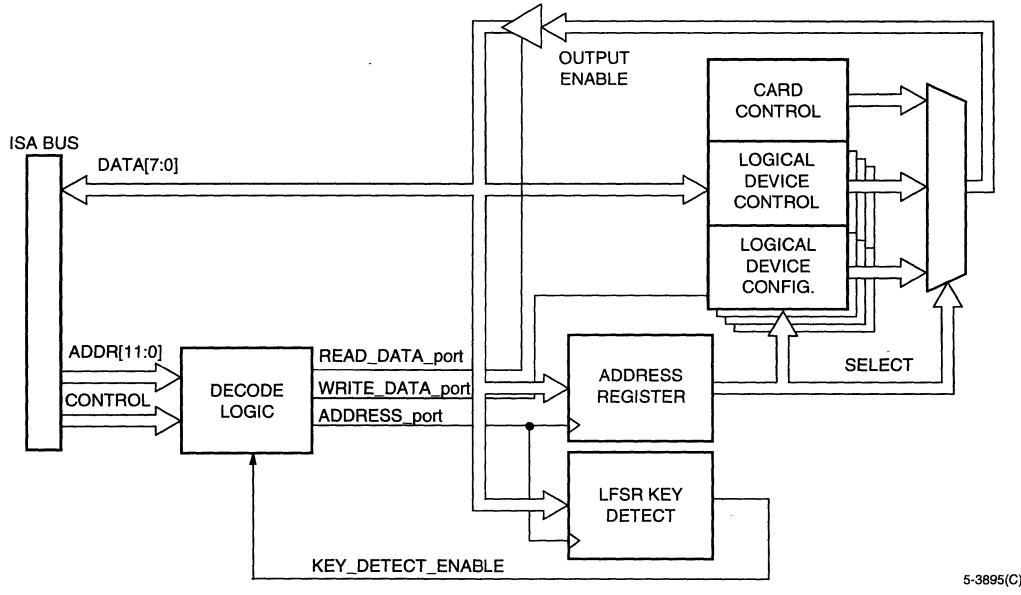


Figure 1. Logic Flow for Plug and Play Autoconfiguration

Background on Plug and Play

(continued)

These factors, and the ease of installation that Plug and Play brings to add-in board customers, is fueling a flood of new Plug and Play products. Figure 1 shows the logic flow for the Plug and Play autoconfiguration circuitry.

Selecting a Silicon Solution

The fiercely competitive market for PC add-in cards dictates that any Plug and Play solution be low cost and easy to implement. While a discreet 74xx TTL logic solution is feasible, even the simplest implementation would require over 30 ICs. The component, board space, and assembly costs for this solution eliminate it from consideration. Several devices are currently available that implement a subset of the Plug and Play standard. Unfortunately, these devices provide little flexibility; unless the card's resource requirements exactly match those provided by the device, there will either be wasted logic or additional support chips needed.

Alternatively, some semiconductor vendors have begun to integrate Plug and Play directly into the chips or chip sets used to implement a card's primary function. Examples include modem chip set and graphics controllers. Currently, few such devices are available, and there is a healthy premium exacted for the Plug and Play support logic in silicon area and unit cost.

The final option is some form of custom logic—either a gate array, standard cell, or programmable logic device. The very high volumes associated with PC add-in cards tend to draw designers toward the gate-array or standard-cell solution. However, several factors have combined to make the programmable logic alternative an increasingly attractive choice.

ORCA—The Quick-Turnaround, Flexible Design Alternative

Programmable devices offer a distinct time-to-market advantage over the long turn-around time of gate arrays and standard cell devices. In the fast-paced, highly competitive world of PC add-in cards, weeks shaved in the development cycle translate directly into market share. Some FPGA vendors now offer FPGA-to-gate array or standard cell migration paths. An example of this is MACO from Lucent Technologies.

This process combines the time-to-market advantage of an FPGA solution with a fully-proven, low-cost, full-custom device for the long term. The production ASIC will be functionally equivalent to the FPGA used for development and initial production. This eliminates the real risk that the ASIC may require multiple design iterations or "spins". The time and nonrecurring engineering costs associated with each spin are also eliminated.

A second factor that is making programmable logic devices more attractive as a Plug and Play solution is their ever-increasing capacity and their ever-shrinking cost. Devices now available encroach upon densities that, in the past, were the sole domain of gate arrays and standard cell devices. For example, consider Lucent's *ORCA* family of FPGAs. The ATT2C40 offers up to 40,000 usable logic gates (more if RAM is used) and 480 user I/O pins. The ATT2C04—the FPGA used in this reference design and the smallest member of the *ORCA* family—offers 400 registers, 400 look-up tables (LUTs), and up to 160 user I/Os.

FPGAs or CPLDs?

Within the realm of programmable devices, there are several options. First, the designer must decide between the sum-of-products/macrocell architecture of complex programmable logic devices (CPLDs) or the logic cell/channeled routing of FPGAs. The CPLD option suffers from the same logic-limited malady as the discreet TTL 74xx solution: the large number of registers and logic functions required for even the simplest Plug and Play implementation will require either multiple devices or a very large (and, therefore, expensive) CPLD.

Compare the 400 registers/LUTs of the ATT2C04 to even the largest available CPLD devices. Such CPLDs top out at 128 to 256 registers, depending on the family. The *ORCA* architecture and Lucent's advanced process technologies yield a device that is much smaller (in terms of silicon area) and far more cost-effective than these large CPLDs.

Device Selection Criteria for a Plug and Play FPGA

An FPGA selected to implement Plug and Play will ideally have certain features and capabilities that will suit it to the application. The Plug and Play specification dictates the actual functionality of any Plug and Play implementation; therefore, some of these features and capabilities are implicit in this specification. Others are simply a function of the Plug and Play application itself. Device capabilities that are desirable in an FPGA used to implement Plug and Play include the following:

- Wide-input functions in a single logic level
- On-chip memory for isolation/configuration data storage
- I/O buffers that meet ISA bus drive/loading requirements
- On-chip tristatable buses
- Flexible registers/latches
- Small size and low power
- Low cost
- Easy migration to ASIC (gate array/standard cell)

The following section will address each of these bullets individually. The relevance to Plug and Play and the features/capabilities of the ATT2C04 that address these bullets will also be discussed.

Wide Input Functions in a Single Level of Logic

It is important that the device selected be able to decode the ISA bus address in as few logic levels as possible. The obvious reasons for this are speed and the amount of logic and routing necessary. In an *ORCA* device, functions of up to 11 inputs can be realized in a single logic level. Compare this with some earlier FPGAs and even some current, fine-grained architectures; narrower input functions mean more logic cells and more routing resources for these wide decode functions. This effects the design's overall performance and routability.

On-Chip Memory

The required serial isolation and logical device configuration data may be stored in the ATT2C04's available on-chip memory. This memory can be configured to be RAM or ROM. The reference design in Lucent's Plug and Play Design Kit requires 32 bytes of storage. The entire 32 x 8 structure fits in just four *ORCA* programmable logic cells (PLCs) or 4% of the logic in the 100 PLC ATT2C04. Other solutions have the isolation and configuration data stored in a separate ROM or EEPROM. This adds significantly to cost and board space. In addition, the slow off-chip memory requires that wait-states be inserted, slowing the isolation and configuration data read processes.

I/O Buffer Drive

The I/Os in *ORCA* are selectable on a pin-by-pin basis to provide 12 mA sink/6 mA source or 6 mA sink/3 mA source current. Inputs can be configured on a pin-by-pin basis to be either TTL or CMOS compatible. Each I/O pin can optionally be configured with a pull-up or pull-down resistor. Each output can be true or inverted and the slew rate of each is selectable to trade off speed versus system noise. Each I/O has a programmable delay in the input path to adjust setup and hold windows.

On-Chip Tristate Buses

Only three ports are defined in the Plug and Play specification: the fixed-location WRITE ADDRESS and WRITE DATA ports, and the relocatable READ DATA port. All transactions between the ISA bus and the Plug and Play hardware take place through these three ports. With 11 defined control registers and an application-dependent number of configuration registers sharing the READ and WRITE DATA ports, it is obvious that on-chip busing is desirable. Without on-chip tristate capability, these signals would need to be multiplexed and demultiplexed. This MUX/deMUX methodology would require copious amounts of on-chip logic and routing. The *ORCA* architecture is nibble oriented; each PLC contains four LUTs and four registers/latches. In addition, there are four tristate buffers adjacent to each PLC. This combination of a nibble-oriented architecture and abundant tristate buffers eliminates the need for logic-and-routing-consuming multiplexers and demultiplexers.

Device Selection Criteria for a Plug and Play FPGA (continued)

Flexible Registers/Latches

Certain aspects of the Plug and Play specification require flexible registers. For example, the initialization key circuitry (see Figure 2) must be loaded to a value of $6A (hex) upon system reset and any time a non-matching key is detected. In most FPGAs, only one type of set/reset is available (for example, active-high asynchronous reset). While a device like this could be made to emulate an asynchronous preset by inserting inverters in front of the D input and behind the Q output, this is very slow and wasteful in terms of logic and routing. Conversely, in *ORCA*, each flip-flop can be individually configured to be set or cleared upon either the global reset or a local signal. This is ideal for initializing state machines and counters. In addition, on a PLC-by-PLC basis (four registers), the registers in *ORCA* can be positive or negative edge-triggered flip-flops or positive or negative level-sensitive latches. D, T, JK, and SR configurations are supported. Each PLC can additionally have an active-high or active-low clock enable input, a synchronous or asynchronous preset or clear (either edge or level), and data multiplexers to dynamically switch between the output from the LUTs or the direct inputs to the registers.

Figure 2 is extracted from the Plug and Play Reference Design. Note the variety of flip-flop types employed. The *ORCA* library offers many permutations of flip-flops and latches.

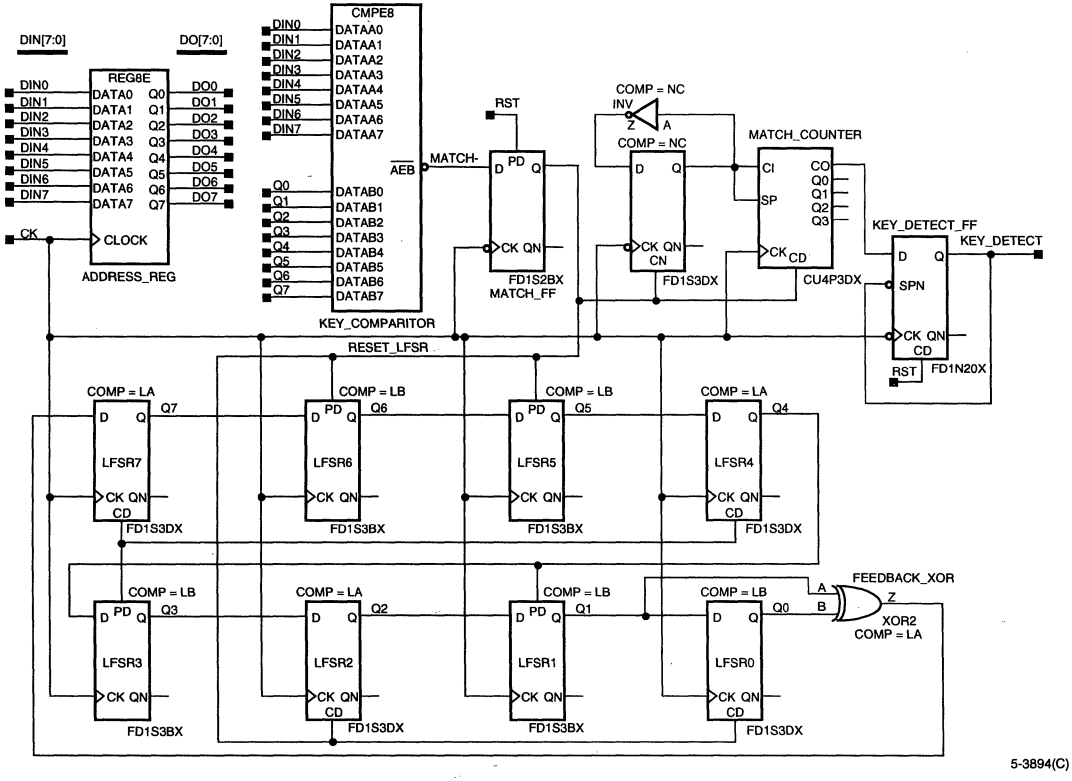


Figure 2. LFSR Initialization Key Circuitry from Lucent's *ORCA* Plug and Play Reference Design—Illustrates the Need for Flexible On-Chip Registers/Latches

Device Selection Criteria for a Plug and Play FPGA (continued)

Small Size and Low Power

In the reference design, the ATT2C04 is housed in the extremely small, low-profile, 144-pin thin-quad flat pack (TQFP). This package is thin enough to fit in Type I PCMCIA cards. Since all *ORCA* devices are SRAM-based and implemented in true CMOS technology, they are very low power. Since power consumption is a function of operating frequency in CMOS devices, the relatively slow speed of the ISA bus makes *ORCA* an excellent choice in terms of power.

Low Cost

Even with 400 registers and 400 LUTs, the ATT2C04 has a very small die size. This fact, and wide industry acceptance, has allowed Lucent to move the device rapidly down the semiconductor pricing curve. In addition, Lucent owns and runs its own semiconductor fabrication facilities. This gives Lucent a built-in cost advantage over its fabless competitors. For absolute lowest cost, Lucent offers the MACO migration program.

Easy Migration to ASIC (Gate Array/ Standard Cell)

Lucent Technologies is the world's largest supplier of standard-cell ASICs. This experience in the design, test, and manufacture of such devices makes the transition from FPGA to gate array or standard cell a fast and easy process. Migration paths offered by other FPGA vendors do not yield true gate arrays or standard-cell devices. Instead, the devices they produce are simple metal-mask derivatives or FPGAs with the programming elements removed. These techniques yield some cost reduction and, in some cases, smaller die. History shows that for absolute smallest die size and lowest cost, a full-custom approach is the only answer. Lucent is the only FPGA vendor that offers a complete solution, from FPGA to standard cell.

Conclusion

The race to bring Plug and Play PC add-in cards to market is on. With the ATT2C04 FPGA, engineers can custom-tailor a solution to their specific requirements. In addition to the Plug and Play circuitry, additional logic functions can be integrated into the device. The size, speed, and routability of the *ORCA* family of FPGAs brings the concept of a digital system-on-a-chip to reality for programmable devices.

Notes

Using a Global Set/Reset Signal in *ORCA* Designs with *Synopsys*

Overview

The Lucent Technologies Optimized Reconfigurable Cell Array (*ORCA*) Series Field-Programmable Gate Arrays (FPGAs) have a dedicated routing resource used to implement a global set/reset (GSR). Using the GSR can significantly reduce routing congestion and improve design performance.

This application note explains how to instantiate the GSR in your VHDL or *Verilog* HDL design. The note assumes you are using the *Synopsys Design Compiler* or *FPGA Compiler* to synthesize an *ORCA* design and *ORCA* Foundry to map and place & route the design.

The *ORCA* Global Set/Reset Signal

The GSR net can be either disabled (default), directly connected to a dedicated input pad named $\overline{\text{RESET}}$, or sourced by a lower-right programmable corner cell signal. If the $\overline{\text{RESET}}$ input pad is not used as a GSR after configuration, the pad can be used as a normal input pad.

Use of the dedicated $\overline{\text{RESET}}$ input pad and its effect on the device during initialization, configuration, and start-up is discussed in the March 1995 *ORCA 2C Series Field-Programmable Gate Arrays* Data Sheet (DS95-031FPGA). This application note, however, assumes the use of the GSR signal during user operation.

Flip-Flops and Latches—Preset or Clear

Each flip-flop or latch can be either preset or cleared—but not both. The flip-flops and latches can be preset or cleared during user operation by asserting the GSR signal or the individual preset (PD) or clear (CD) pins on a flip-flop or latch. If the GSR signal is asserted low (default) during user operation, all of the device's registers are forced to the same state they had at the end of configuration as illustrated in Table 1.

The start-up state of each flip-flop and latch is determined by whether it has a PD pin, CD pin, or neither. If the cell has a PD pin, the cell starts up in a preset state. All other cells will start up in a clear state, regardless of whether they have a CD pin or not.

Using the GSR to Improve Performance and Routability

Many designs have a common signal that sets or resets all registers. This type of design can benefit from the use of the GSR. Because the *Design Compiler* cannot infer the use of a GSR signal from HDL code, a GSR must be instantiated. The NAORGSR component from the *ORCA* Foundry Library is used for *ORCA* designs.

Table 1. Initialization State of Flip-Flops and Latches After Configuration

Initialized to 0	FD1S1A, FD1S1D, FD1S1I, FL1S1A, FS1S1A, FD1P3AX, FD1P3DX, FD1P3IZ, FD1S3AX, FD1S3DX, FD1S3IX, FJ1S3AX, FJ1S3DX, FL1P3AZ, FL1S3AX, FT1S3DX
Initialized to 1	FD1S1AY, FD1S1J, FL1S1AY, FD1P3AY, FD1P3BX, FD1P3JZ, FD1S3AY, FD1S3BX, FD1S3JX, FL1P3AY, FL1S3AY, FT1S3BX, FD1S1B, FJ1S3BX

The *ORCA* Global Set/Reset Signal
(continued)

To implement the GSR in HDL code, you need to create a level of hierarchy that instantiates the NAORGSR and the core design. This is typically the top level. Figure 1 illustrates the top level of hierarchy of an example design implemented with HDL. Figure 2 illustrates the submodule code.

The example used in the following figures implements the NAORGSR at the top level. It is driven by a primary input signal named **gsr**. A local set/reset signal, **lsr**, is used to infer the appropriate flip-flops. In this case, two preset and two clear flip-flops will be inferred. Once the design is compiled, the lsr signal can be disconnected, allowing the gsr signal to control the set/reset function.

```
module gsr_example (ck, lsr, gsr, s);
input ck, lsr, gsr;
output [3:0] s;

NAORGSR i0 (.GSR(gsr));

submod i1 (.LSR(lsr),.CK(ck),.S(s));

endmodule

module NAORGSR (GSR);
input GSR;
endmodule
```

Figure 1. *Verilog* HDL Code for gsr_example.v

```
module submod (LSR, CK, S);
input LSR, CK;
output [3:0] S;
reg [3:0] S;

always @(posedge CK or posedge LSR)
begin
  if (LSR == 1'b1)
     S = 4b'1010;
  else
     S = S + 1;
end
endmodule
```

Figure 2. *Verilog* HDL Code for submod.v

Figure 3 illustrates the top level of the GSR example in VHDL code. Figure 4 illustrates the submodule code in VHDL.

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_unsigned.all;

entity gsr_example is
port(ck, lsr, gsr : in STD_LOGIC;
 s : buffer STD_LOGIC_VECTOR (3 downto 0));

end gsr_example;

architecture EXAMPLE of gsr_example is

component NAORGSR
port(GSR : in STD_LOGIC);
end component;

component submod
port(CK, LSR : in STD_LOGIC;
 S : buffer STD_LOGIC_VECTOR (3 downto 0));
end component;

begin

   i0 : NAORGSR port map(GSR => gsr);
   i1 : submod port map(ck, lsr, s);

end EXAMPLE;
```

Figure 3. VHDL Code for gsr_example.vhd

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_unsigned.all;

entity submod is
 port(CK, LSR : in STD_LOGIC;
  S:buffer STD_LOGIC_VECTOR(3 downto 0));
end submod;

architecture EXAMPLE of submod is

begin
  process (CK, LSR)
  begin
    if LSR = '1' then S <= "1010";
    elsif (CK'event and CK = '1') then
      S <= S + 1;
    end if;
  end process;
end EXAMPLE;
```

Figure 4. VHDL Code for submod.vhd

Synopsys Script

Figure 5 illustrates a sample script for use with *Design Compiler*. The operations to implement the GSR in *Verilog* or VHDL are the following:

1. Instantiate NAORGSR component in HDL code.
2. Analyze and elaborate design.
3. Place a dont_touch attribute on the NAORGSR cell with the **set_dont_touch** command.
4. Insert I/O buffers on all ports except LSR with **insert_pads** commands.
5. Compile the HDL code.
6. Save the design.
7. Disconnect the LSR signal with **disconnect_net** command.
8. Write out EDIF netlist.

The resultant EDIF netlist can be mapped, placed, and routed with *ORCA* Foundry. The lsr net will not appear in the design, thereby saving valuable routing resources and improving design performance.

```
/* Sample Synopsys script for GSR example */
/* For use with Design Compiler v3.2      */

/* Analyze & elaborate design files       */

   read -format vhdl gsr_example.vhd
   read -format vhdl submod.vhd

/* Set current design to top level        */

   current_design = gsr_example

/* Place dont_touch attribute on NAORGSR
   instance. NAORGSR instance has no
   outputs. It will be removed by Design
   Compiler unless this attribute is
   applied.                               */

   set_dont_touch i0

/* Insert I/O buffers on all ports except
   the LSR. The LSR will be disconnected  */

   set_port_is_pad { ck gsr s }
   insert_pads

/* Remove all design constraints          */

   remove_constraint -all

/* Compile the design                     */

   compile -map_effort low

/* Save the design                        */

   write -hier -f db -o gsr_example.db

/* Remove the LSR from the design         */

   disconnect_net lsr -all

/* Write out an EDIF netlist              */

   write -hier -f edif -o gsr_example.edn

/* Exit the Design Compiler               */

   exit
```

Figure 5. Sample *Design Compiler* Script

Notes

ORCA Series Boundary Scan

Introduction

The increasing complexity of integrated circuits and packages has increased the difficulty of testing printed-circuit boards. As integrated circuits become more complex, testing of the loaded board is one of the most difficult tasks in the design cycle. Advanced package technology has led to the need to develop new ways to test the printed-circuit boards instead of conventional probing techniques.

To cope with these testing issues, the *IEEE* Std. 1149.1-1990 (*IEEE* Standard Test Access Port and Boundary-Scan Architecture) was developed to provide a solution to the problem of testing the printed-circuit board. *IEEE* Std. 1149.1-1990 defines a test access port and boundary-scan architecture so that circuitry can test interconnections between integrated circuits on the printed-circuit board, test the integrated circuit itself, and analyze functionality of the device on the printed-circuit board.

To address these issues, the *IEEE* Std. 1149.1-1990 compatible boundary-scan architecture and test access port are implemented in the *ORCA* series. The boundary-scan logic, including a boundary-scan test access port (BSTAP) and other associated circuitry, is placed at the upper left corner of the device, while boundary-scan shift registers are located near each I/O pad.

The boundary-scan test access port controller conforms to the standard for the three mandatory instructions (bypass, extest, and sample/preload). This conformance was tested with vectors generated by the internal Lucent Technologies program *Tapdance* (boundary-scan conformance vector generator). In addition to the three mandatory instructions, four user-defined instructions are provided to support additional testability of the *ORCA* series.

Overview of Boundary-Scan Architecture

Figure 1 shows a block diagram of the boundary-scan architecture that is implemented in the *ORCA* series. There are three input pins (TDI, TMS, and TCK) and one output pin (TDO). The built-in power-on reset circuitry resets the boundary-scan logic during powerup. This is essential to prevent all I/O buffers from contention during power-on. In addition, an external pin ($\overline{\text{PRGM}}$) can be used to reset boundary-scan logic at any time when it is necessary, but this pin is not a standard reset pin as defined in *IEEE* Std. 1149.1-1990. The three input pins to the boundary-scan logic are user-accessible PIC I/O pins whose location varies from part to part, and the output pin is the dedicated TDO/RD_DATA output pin. The functionality of these pins is as follows:

- Test data input (TDI): Serial input data.
- Test mode select (TMS): Controls the BSTAP controller.
- Test clock (TCK): Test clock.
- Test data output (TDO): Serial data output.

Overview of Boundary-Scan Architecture (continued)

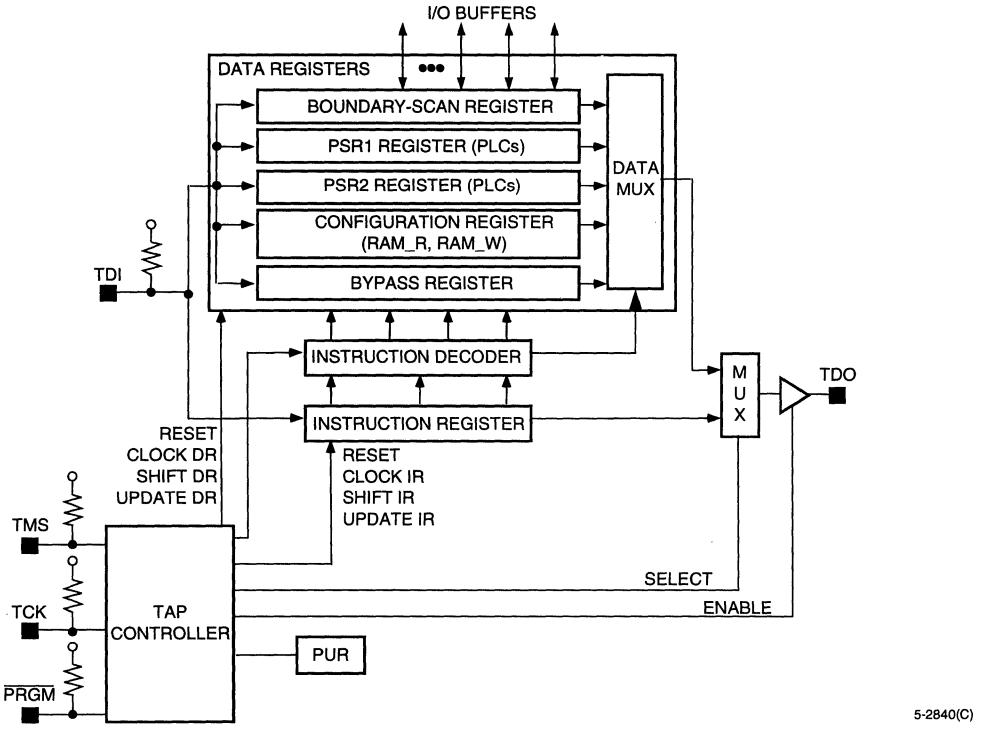


Figure 1. Diagram of Boundary-Scan Function

There are five data registers and one instruction register in the boundary-scan architecture, as shown in Figure 1.

- The primary data register is a boundary-scan register, comprising a shift register around the peripheral of the chip. This shift register is used for controlling the functionality of the I/O pins of the device. CCLK, DONE, and the four boundary-scan interface pins (TCK, TDI, TMS, and TDO) are not included in the boundary-scan chain.
- The bypass register, a single flip-flop, allows the serial input data from TDI to be shifted out of the TDO output without interfering with the FPGAs normal functionality.
- A pair of user-defined internal scan data registers can be optionally configured by using the registers in the PLCs. This allows user scan data to be shifted out of the TDO output.
- The existing configuration data shift register can be controlled by the BSTAP controller to either write the configuration memory or read back its contents.
- A 3-bit instruction register is implemented for the three mandatory instructions and the four user-defined instructions.

The boundary-scan logic is always enabled before configuration so that the user can activate the boundary-scan instructions. After configuration, a configuration RAM bit is used to determine whether the boundary-scan logic is to be used or not. If boundary scan is not used, the I/O pins TCK, TDI, and TMS can be used as normal user-defined I/O pins, and the TDO/RD_DATA output pin can be used to output configuration RAM read back data.

Boundary-Scan Circuitry

The boundary-scan circuitry includes a test access port controller, a 3-bit instruction register, a boundary-scan register, and a bypass register. It also includes other circuitry to support the four user-defined instructions.

Boundary-Scan Circuitry (continued)

The BSTAP controller in the *ORCA* series is an *IEEE* Std. 1149.1-1990 compatible test access port controller. The 16 state assignments, from the *IEEE* Std. 1149.1-1990 specification, are implemented in the BSTAP, which is controlled by TCK and TMS. All control signals are issued on the rising edge of TCK, except the ENABLE signal, which switches on the falling edge of the TCK.

The BSTAP controller state diagram is shown in Figure 2, and each state is described below.

Test-Logic-Reset. The boundary-scan logic is in reset mode, which allows the device to be in normal operation in this state. This state is entered during power-on and can be reached two different ways: holding the TMS at a logic 1 and applying five TCK clock cycles, or holding $\overline{\text{PRGM}}$ pin low.

Run-Test/Idle. The operation of the boundary-scan logic depends on the instruction held in the instruction register.

Select-DR/IR. Either the data register or the instruction register is selected. Thus, TDI is the input and TDO the output of the selected path.

Capture-DR/IR. Data or instructions (whichever is selected) are loaded from the parallel inputs of the selected data or instructions registers into their shift-register paths.

Shift-DR/IR. The captured data or instructions (whichever is selected) are shifted out while new data or instruction is shifted in the selected registers.

Update-DR/IR. This state marks the completion of the shifting process, and either the instruction register is loaded for instruction decode or the boundary-scan register is updated for I/O pin control.

Other States. Pause and exit states are provided to allow the shifting process to be temporarily halted for any test reason.

The 3-bit instruction register and the instruction decoder are provided for the three mandatory instructions and the four user-defined instructions, as shown in Table 1. The instruction register provides one of the serial paths between TDI and TDO, as shown in Figure 1.

The instruction register is a two-stage register in which the instruction data is shifted serially into the first stage and then updated in parallel into the second stage.

After the assertion of Update-IR, a new instruction can be shifted into the first stage of the instruction register without altering the previous instruction.

The output of the second stage makes up a 3-bit parallel instruction word that is sent to the instruction decoder. The instruction decoder will decode the signals (mode, capture, shiftn, update, etc.) to control the boundary-scan shift register and other test data registers.

The bypass register is a single-bit shift register so that the serial scan data (TDI) can be shifted out to TDO with a delay of one TCK clock period.

Table 1. *ORCA* Boundary-Scan Instructions

Code	Instruction
000	Extest.
001	PLC Scan Ring 1.
010	RAM Write (RAM_W).
011	Reserved.
100	Sample.
101	PLC Scan Ring 2.
110	RAM Read (RAM_R).
111	Bypass.

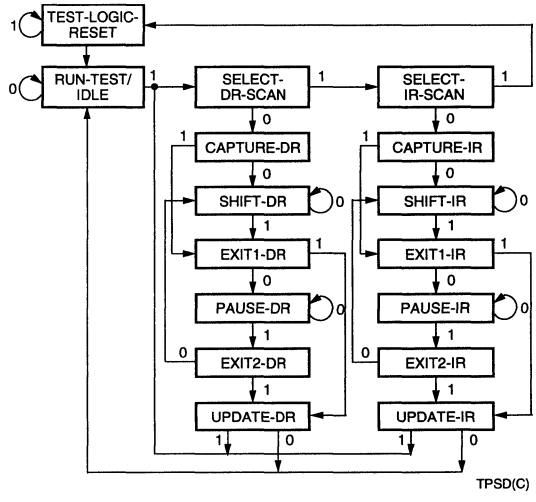


Figure 2. TAP Controller State Diagram

Boundary-Scan Circuitry (continued)

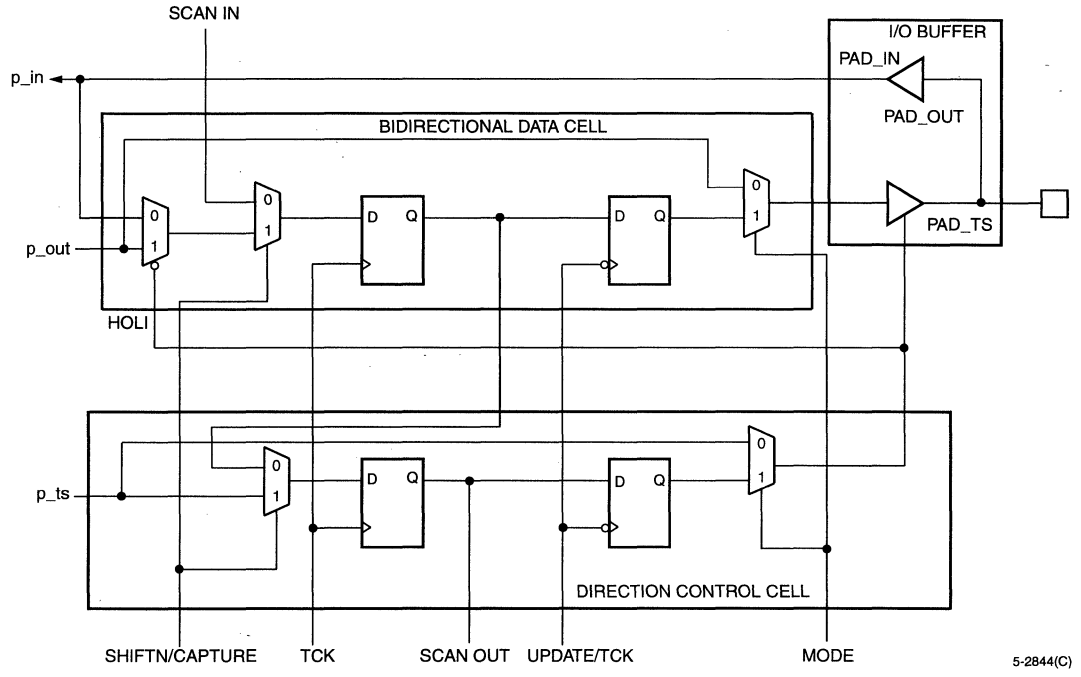


Figure 3. Boundary-Scan Cell (BSC)

The user-accessible I/O pins in the PICs are configured as bidirectional buffers during the device-level boundary-scan test. Therefore, two registers are needed to access the input, output, and 3-state values for each programmable I/O pin. The first is the bidirectional data cell which is used to access the input or output data. The second is the direction control cell which is used to access the 3-state value. The boundary-scan shift register is a series connection of a bidirectional data cell and a direction control cell for each I/O in the boundary-scan chain, as shown in Figure 3.

The bidirectional data cell and direction control cell each include a flip-flop used to shift scan data and an update flip-flop to control the I/O buffer. The bidirectional data cell is controlled by the high out, low in (HOLI) signal generated by the direction control cell.

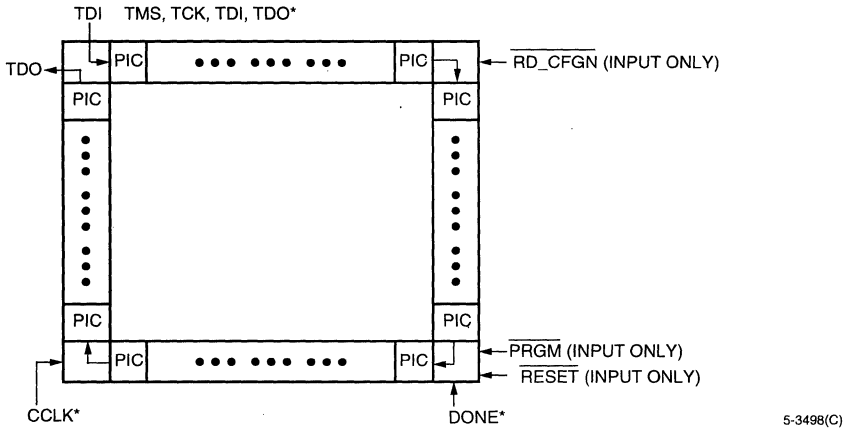
When HOLI is low, the bidirectional data cell takes input buffer data into the boundary-scan register. When HOLI is high, the boundary-scan register is loaded with functional data from the internal core logic.

When the MODE signal that is generated by the instruction decoder is high (extest mode) and the buffer is configured as an output, the scan data is propagated to the output buffer. When the MODE signal is low (bypass or sample modes) and the buffer is configured as an output, functional data from the internal core is propagated to the output buffer.

Four other global signals (capture, update, shiftn, and treset) that are generated by BSTAP controller, as well as the boundary-scan clock (TCK), are used to control the shift register containing the bidirectional data cells and the direction control cells.

The first flip-flop in the boundary-scan shift register is for the first PIC I/O pin on the left of the top side of the *ORCA* device. The shift register then proceeds in a clockwise motion until reaching the first PIC I/O pin on the top of the left side of the *ORCA* device. Figure 4 shows the full chip arrangement of this boundary-scan chain for the *ORCA* series.

Boundary-Scan Circuitry (continued)



* TMS, TCK, TDI, TDO, CCLK, and DONE are not included in the boundary-scan register.

Figure 4. *ORCA* Series Boundary-Scan Chain

Description of the Three Mandatory and the Four User-Defined Instructions

ORCA boundary-scan logic supports three mandatory instructions in *IEEE* Std. 1149.1-1990 and four user-defined instructions as shown in Table 1.

When the EXTEST instruction is activated in an *ORCA* device, either the scan data at input pins is loaded into the boundary-scan shift register with the rising edge of TCK in Capture-DR controller state, or the scan data at output pins is updated from the boundary-scan shift register with the falling edge of TCK in Update-DR controller state (depending on the value of the 3-state signal).

The SAMPLE/PRELOAD instruction allows a snapshot of the functional data present at the I/O pins. When the sample/preload instruction is selected in a device, the functional data at the pins is loaded into the boundary-scan shift register with the rising edge of TCK in the Capture-DR controller state and the captured functional data is shifted out to the TDO output pin with the Shift-DR controller state. The selection between input data and output data at each pin is determined by whether the pin is currently an input or an output.

The BYPASS instruction allows the serial scan data (through the TDI input pin) to be shifted out of the TDO output with a delay of one TCK clock period without interfering with the FPGA's normal functionality.

The PSR1 and PSR2 user instructions are provided to allow the implementation of a pair of user-defined internal scan paths using the PLC registers. The data is shifted in on TDI and shifted out on TDO by the test clock (TCK). Connectivity to the general FPGA routing is provided for TCK, TDI, a scan path enable signal, and a shift data out signal for each instruction. An example of these scan paths is shown in Figure 5.

The RAM_W (configuration RAM write) allows the user to program the configuration memory by shifting serial configuration data in on TDI with the test clock (TCK). This instruction is available both before and after configuration.

The RAM_R (readback RAM read) allows the user to read back the configuration memory by shifting the data out on the TDO output pin. This instruction is only available after configuration.

Description of the Three Mandatory and the Four User-Defined Instructions
(continued)

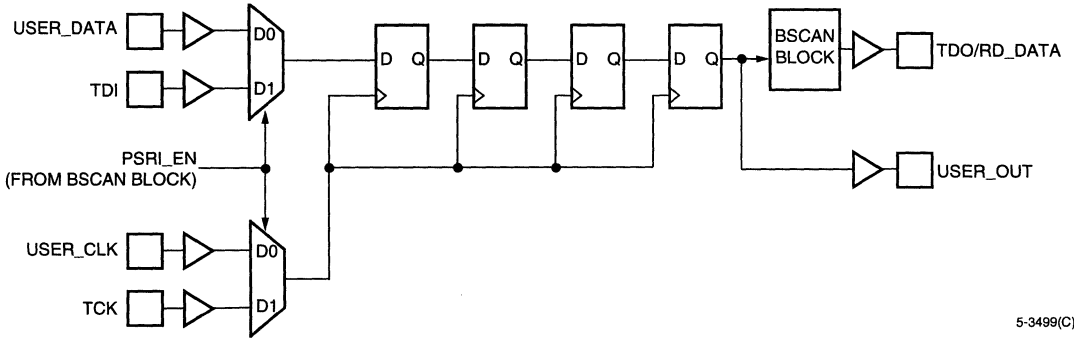


Figure 5. Example of Boundary-Scan User-Defined Scan Path Logic

Usage of Boundary-Scan Logic

As mentioned earlier, the power-on reset circuitry resets the boundary-scan logic during power-on to prevent all I/O pins from causing contention. In addition, an external $\overline{\text{PRGM}}$ pin is provided not only to reset the boundary-scan logic but also to reprogram the *ORCA* device when the $\overline{\text{PRGM}}$ pin is low.

The $\overline{\text{PRGM}}$ pin is a part of the boundary-scan chain. However, this pin must be high during boundary-scan operation, configuration, and normal operation unless it is necessary to reset the boundary-scan logic or to reconfigure the *ORCA* device. This pin is not considered the optional reset pin as defined in *IEEE* Std. 1149.1-1990.

The boundary-scan function is always enabled during initialization and configuration. Therefore, the bypass instruction is always available and can be executed at any time without any restriction. Although the boundary-scan logic is enabled during initialization, other instructions (extest, sample/preload) are not allowed to be used during initialization. This requires that the user keep the boundary-scan logic in either reset mode or bypass mode during initialization. This is also true during configuration when boundary scan is not being used.

It is recommended that either the TMS or the TCK pin be tied high so that the boundary-scan logic will be in the test-logic reset state when the boundary-scan logic is not to be used during initialization and configuration. This recommendation also applies to the user who is not using boundary-scan as well to avoid the accidental activation of the boundary-scan logic.

The three mandatory instructions (bypass, extest and sample/preload) and the ram_w instruction are available as soon as the $\overline{\text{INIT}}$ pin goes high (indicating the completion of initialization). Before configuration, the three mandatory instructions are fully supported and can be exercised to test interconnections between the integrated circuits on the board.

Because all the I/O buffers can be bidirectional buffers, it is necessary to avoid the simultaneous switching of too many output buffers when the Update-DR controller state is asserted. The normal rules in the *ORCA* data sheet for simultaneous switching outputs should be followed at all times.

The *ORCA* device can be configured by using the ram_w instruction. If this instruction is to be applied before configuration, the $\overline{\text{INIT}}$ pin must be high, signaling the end of initialization. Serial configuration data is then supplied on TDI and is clocked in with TCK. The special configuration pins $\overline{\text{LDC}}$, HDC, $\overline{\text{INIT}}$, and DONE function the same as during any other configuration process.

If the ram_w instruction is asserted after configuration is done, the FPGA is first reinitialized. Thus, the user must wait until the $\overline{\text{INIT}}$ pin is high before applying the serial configuration data. In order to guarantee that the ram_w instruction works properly after configuration, the user should apply at least three TCK cycles after applying the ram_w instruction. Timing diagrams of the ram_w instruction both before and after configuration are shown in Figures 6 and 7.

Usage of Boundary-Scan Logic (continued)

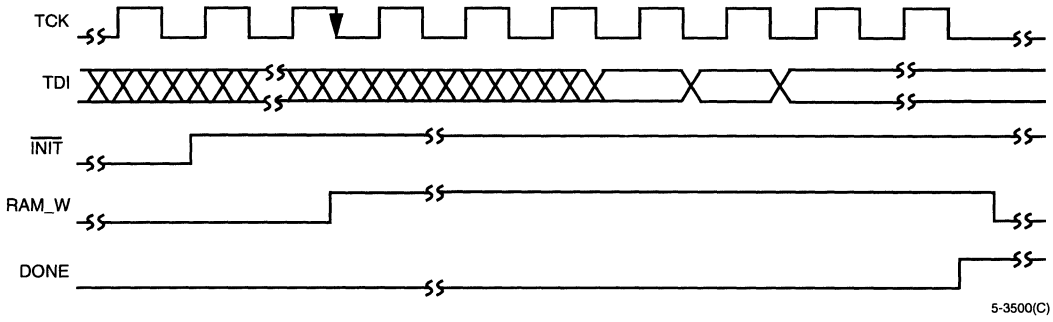


Figure 6. ram_w Timing Diagram Before Configuration

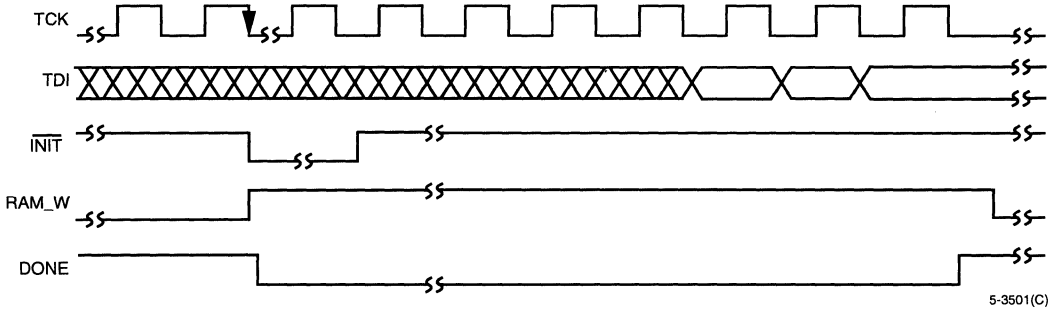


Figure 7. ram_w Timing Diagram After Configuration

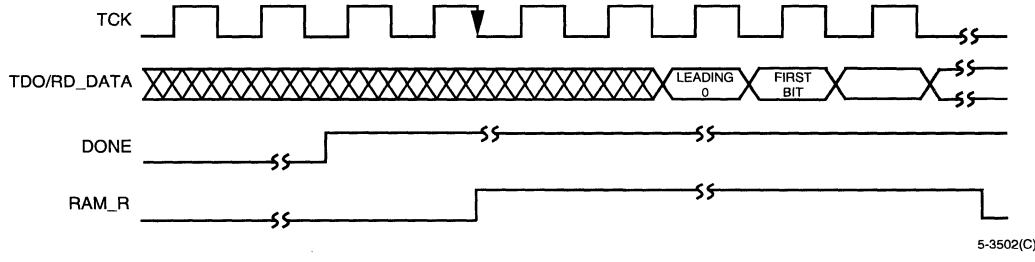


Figure 8. ram_r Timing Diagram After Configuration

The three mandatory and the four user-defined instructions are available after configuration when the boundary-scan logic is selected by setting the jtag_en ram bit high. If this RAM bit is not set, the boundary-scan logic is disabled and the TMS, TCK, and TDI pins can be used as normal I/O pins. When a ram_r instruction is asserted after configuration, the configuration memory contents is shifted out of the TDO output buffer. It should be noted that this readback data is clocked out of TDO on the falling edge of TCK and is valid two TCK clock cycles after the ram_r instruction is asserted. The maximum frequency of TCK is 10 MHz. Figure 8 is a timing diagram of the ram_r instruction.

Boundary-Scan Description Language

The boundary-scan description language (BSDL) files are generated by software and follow the standard format. These BSDL files provide the pinout for all of the different package types supported by *ORCA* as well as additional boundary-scan information. The BSDL files can be obtained from your Lucent Technologies FAE.

References

The following list contains the names of publications that can provide more detailed information about the *IEEE* Std. 1149.1-1990 boundary-scan specification and application.

C. M. Maunder & R. E. Tulloss. "An Introduction to the Boundary Scan Standard," Journal of Electronic Testing and Applications, 1991, pp. 27-42.

IEEE Std. 1149.1-1990 Test Access Port and Boundary Scan Architecture, *IEEE* Computer Society, May 1990.

Parker, Kenneth P., The Boundary Scan Handbook, Kluwer Academic Publishers.

Additional ATT3000 Data

Introduction

This application note is designed to serve as a supplement to the March 1995 *ATT3000 Series* Data Sheet (DS94-177FPGA). It covers a wide range of topics, including a number of electrical parameters not specified in the data sheet. These additional parameters are sufficiently accurate for most design purposes. Unlike the parameters specified in the data sheet, however, they are not worst-case values over temperature and voltage, and they are not 100% production tested. Therefore, they cannot be guaranteed.

Configurable Logic Blocks

The ATT3000 configurable logic block (CLB) shown in Figure 1 is comprised of a combinatorial function generator and two D-type flip-flops. Two output pins may be driven by either the function generators or the flip-flops. The flip-flop outputs may be routed directly back to the function generator inputs without going outside of the CLB.

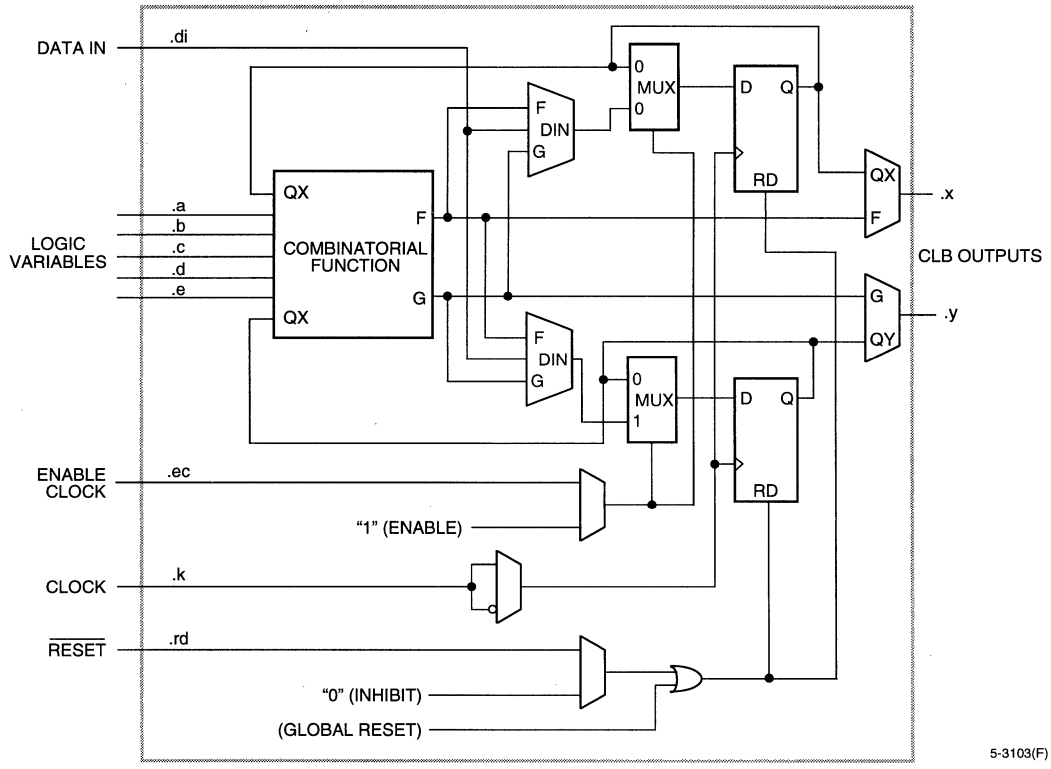


Figure 1. Configurable Logic Block (CLB)

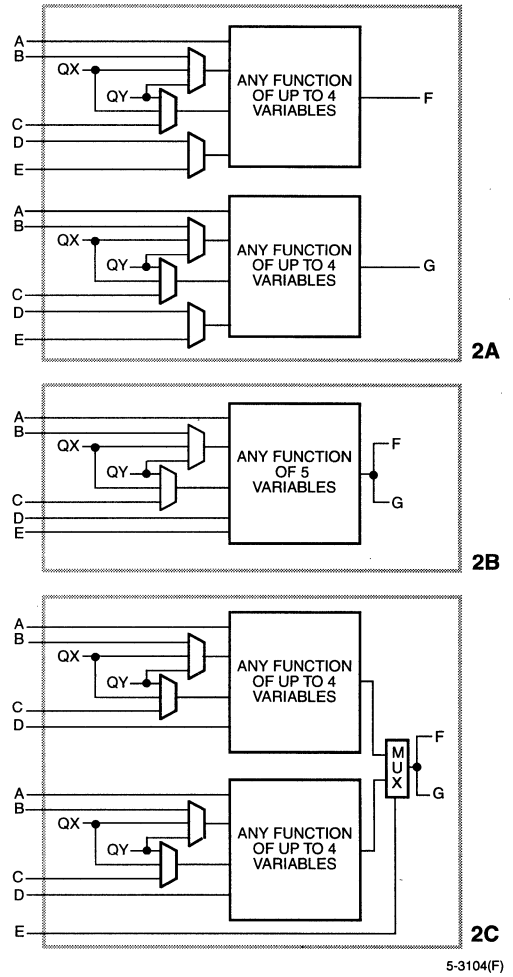
Configurable Logic Blocks (continued)

The function generator consists of two four-input look-up tables that may be used separately or combined into a single function. Figure 2 shows the three available options. Since the CLB only has five inputs to the function generator, inputs must be shared between the two look-up tables.

In the FG mode, the function generator provides any two four-input functions of A, B, and C plus D or E; the choice between D and E is made separately for each function. In the F mode, all five inputs are combined into a single five-input function of A, B, C, D and E. Any five-input function may be emulated. The FGM mode is a superset of the F mode, where two four-input functions of A, B, C, and D are multiplexed together according to the fifth variable, E.

In all modes, either of the B and C inputs may be selectively replaced by QX and QY, the flip-flop outputs. In the FG mode, this selection is made separately for the two look-up tables, extending the functionality to any two functions of four variables chosen from seven, provided two of the variables are stored in the flip-flops. This is particularly useful in state machine-like applications.

In the F mode, the function generators implement a single function of five variables that may be chosen from seven, as described above. The selection of QX and OY is constrained to be the same for both look-up tables. The FGM mode differs from the F mode in that QX and QY may be selected separately for the two look-up tables, as in the FG mode. This added flexibility permits the emulation of selected functions that can include all seven possible inputs.



2A. Combinatorial Logic Option 1 generates two functions of four variables each. One variable, A, must be common to both functions. The second and third variables can be any choice among B, C, Qx, and Qy. The fourth variable can be either D or E.

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

2C. 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 among B, C, Qx, and Qy for the remaining two variables. Option 3 can then implement some functions of six or seven variables.

Figure 2. CLB Logic Options

Configurable Logic Blocks (continued)

The two D-type flip-flops share a common clock, a common clock enable, and a common asynchronous reset signal. An asynchronous preset can be achieved using the asynchronous reset if data is stored in active-low form; the low created by RESET corresponds to the bit being asserted. The flip-flops cannot be used as latches.

If input data to a CLB flip-flop is derived directly from an input pad, without an intervening flip-flop, the data pad to clock pad hold time will typically be nonzero. This hold time is equal to the delay from the clock pad to the CLB but may be reduced according to the 70% rule, (described later in the IOB Input section of this application note). Under this rule, the hold time is reduced by 70% of the delay from the data pad to the CLB, excluding the CLB setup time. The minimum hold time is zero, even when applying the 70% rule results in a negative number.

The CLB pins to which long lines have direct access are shown in Table 1. Note that the clock enable pin ($\overline{CE}$) and the TBUF control pin are both driven from/to the same vertical long line. Consequently, $\overline{CE}$ cannot easily be used to enable a register that must be 3-stated onto a bus. Similarly, $\overline{CE}$ cannot easily be used in a register that uses the reset direct pin (RD).

Table 1. Long Line to CLB Direct Access

Long Line	CLB								TBUF
	A	B	C	D	E	K	$\overline{CE}$	RD	T
Left-most Vertical (GCLK)						X			
Left Middle Vertical		X					X	X	X
Right Middle Vertical			X		X				
Right-most Vertical (ACLK)						X			
Upper Horizontal				X					
Lower Horizontal	X							X	

Input/Output Blocks

The ATT3000 IOB, shown in Figure 3, includes a 3-state output driver that may be driven directly or registered. The polarities of both the output data and the 3-state control are determined by configuration bits. Each output buffer may be configured to have either a fast or a slow slew rate.

The IOB input may also be direct or registered. Additionally, the input flip-flop may be configured as a latch. When an IOB is used exclusively as an input, an optional pull-up resistor is available, the value of which is 40 kΩ—150 kΩ. This resistor cannot be used when the IOB is configured as an output or as a bidirectional pin. Unused IOBs should be left unconfigured. They default to inputs pulled high with the internal resistor.

Inputs

All inputs have limited hysteresis, typically in excess of 200 mV for TTL input thresholds and in excess of 100 mV for CMOS thresholds. Exceptions to this are the $\overline{PWRDWN}$ pin and the XTL2 pin, when it is configured as the crystal oscillator input.

Experiments show that the input rise and fall times should not exceed 250 ns. This value was established through a worst-case test using internal ring oscillators to drive all I/O pins except two, therefore generating a maximum of on-chip noise. One of the remaining I/O pins was configured as an input and tested for single-edge response—the other I/O was used as an output to monitor the response.

These test conditions are possibly overly demanding, although it was assumed that the PC board had negligible ground noise and good power supply decoupling. While conservative, the resulting specification is, in most instances, easily satisfied.

IOB input flip-flops are guaranteed to operate correctly without data hold times (with respect to the device clock-input pad), provided the dedicated CMOS clock-in pad and the GCLK buffer are used. The use of a TTL clock or a different clock pad will result in a data hold-time requirement. The length of this hold time is equal to the delay from the actual clock pad to the GCLK buffer minus the delay from the dedicated CMOS clock pad to the GCLK buffer.

Input/Output Blocks (continued)

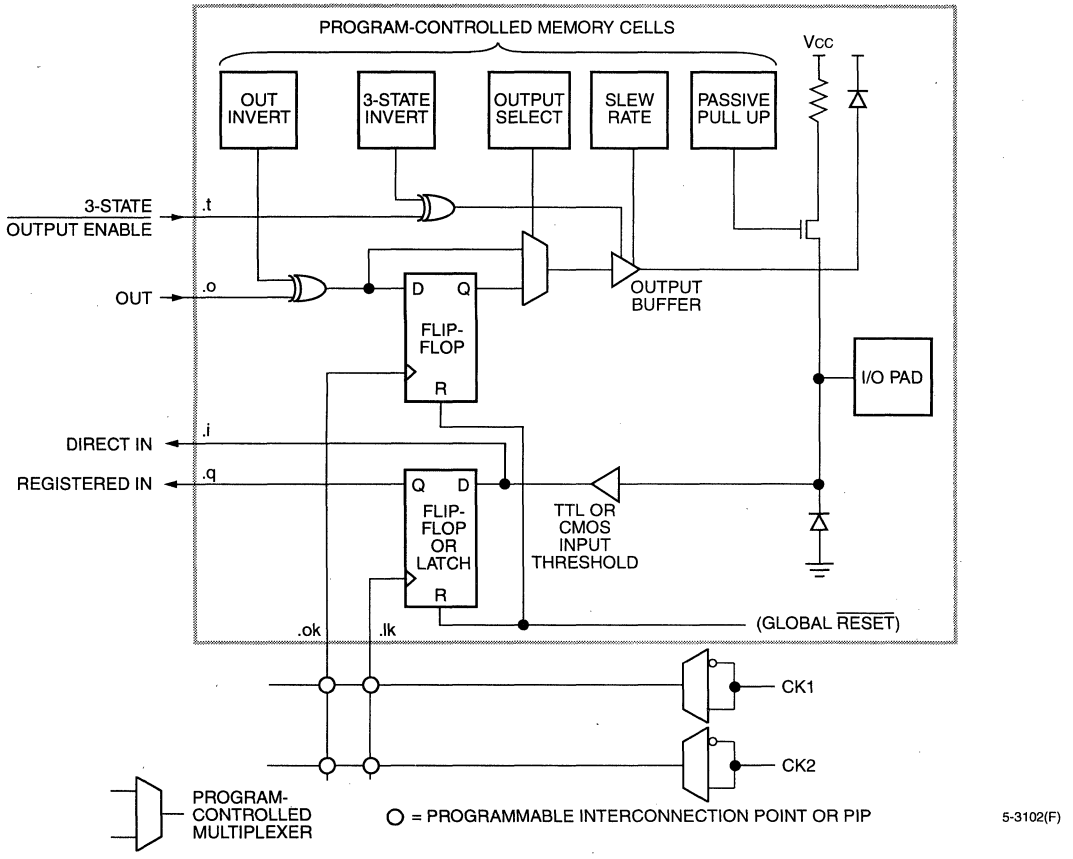


Figure 3. Input/Output Block (IOB)

To ensure that the input flip-flop has a zero hold time, delay is incorporated in the D input of the flip-flop, causing it to have a relatively long setup time. However, the setup time specified in the data sheet is with respect to the clock reaching the IOB. Since there is an unavoidable delay between the clock pad and the IOB, the input pad to clock pad setup time is actually less than the data sheet number.

Part of the clock delay can be subtracted from the internal setup time. Ideally, all of the clock delay could be subtracted, but it is possible for the clock delay to be less than its maximum while the internal setup time is at its maximum value. Consequently, it is recommended that, in a worst-case design, only 70% of the clock delay be subtracted.

The clock delay can only be less than 70% of its maximum if the internal setup time requirement is also less than its maximum. In this case, the data pad to clock pad setup time actually required will be less than that calculated.

For example, in the ATT3000-125, the input setup time with respect to the clock reaching the IOB is 16 ns. If the delay from the clock pad to the IOB is 6 ns, then 70% of this delay, or 4.2 ns, can be subtracted to arrive at a maximum pad-to-pad setup time of ~12 ns.

Input/Output Blocks (continued)

The 70% rule must be applied whenever one delay is subtracted from another. However, it is recommended that delay compensation only be used in connection with input hold times. Delay compensation in asynchronous circuits is not recommended. In any case, the compensated delay must not become negative. If 70% of the compensating delay is greater than the delay it is deducted from, the resulting delay is zero.

The 70% rule in no way defines the absolute minimum value delays that might be encountered from chip to chip and with temperature and power supply variations. It simply indicates the relative variations that might be found within a specific chip over the range of operating conditions.

Typically, all delays will be less than their maximum, with some delays being disproportionately faster than others. The 70% rule describes the spread in the scaling factors: the delay that decreases the most will be no less than 70% of what it would have been if it had scaled in proportion to the delay that decreased the least. In particular, in a worst-case design where it is assumed that any delay might not have scaled at all and remains at its maximum value, other delays will be no less than 70% of their maximum.

Outputs

All ATT3000 outputs are true CMOS, with n-channel transistors pulling down and p-channel transistors pulling up. Unloaded, these outputs pull rail-to-rail. Some additional ac characteristics of the output are listed in Table 2. Figures 4 and 5 show output current/voltage curves for typical ATT3000 devices.

Output short-circuit current values are given only to indicate the capability to charge and discharge capacitive loads. In accordance with common industry practice for other logic devices, only one output at a time may be short-circuited, and the duration of this short circuit to V_{CC} or ground may not exceed one second. A continuous output or clamp current in excess of 20 mA on any one output pin is not recommended. The data sheet guarantees the outputs for no more than 4 mA at 320 mV to avoid problems when many outputs are sinking current simultaneously.

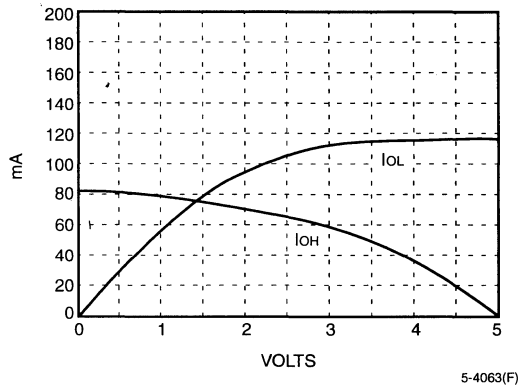


Figure 4. Output Current/Voltage Characteristics for -70, -100, -125, -150 Speed Grades

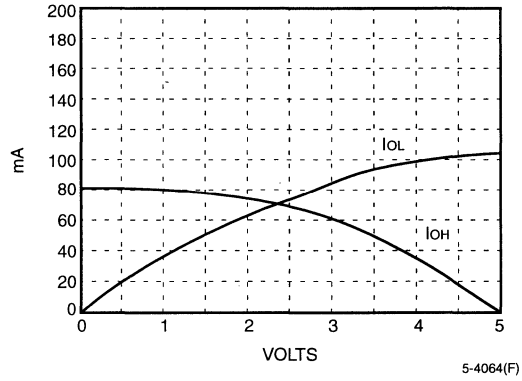


Figure 5. Output Current/Voltage Characteristics for -3, -4, -5 Speed Grades

The active-high 3-state control (T) is the same as an active-low output enable ($\overline{OE}$). In other words, a high on the T-pin of an OBUFZ places the output in a high-impedance state, and a low enables the output. The same naming convention is used for TBUFs within the FPGA device.

Input/Output Blocks (continued)

I/O Clocks

Internally, up to eight distinct I/O clocks can be used—two on each of the four edges of the die. While the IOB does not provide programmable clock polarity, the two clock lines serving an IOB can be used for true and inverted clock. The appropriate polarity can then be connected to the IOB. This does, however, limit all IOBs on that edge of the die to using only the two edges of the one clock.

IOB latches have active-low latch enables; they are transparent when the clock input is low, and they are closed when it is high. The latch captures data on what would otherwise be the active clock edge, and is transparent in the half-clock period before the active clock edge.

Table 2. Additional ac Output Characteristics

ac Parameters	Fast*	Slow*
Unloaded Output Slew Rate	2.8 V/ns	0.5 V/ns
Unloaded Transition Time	1.45 ns	7.9 ns
Additional Rise Time for 812 pF Normalized	100 ns 0.12 ns/pF	100 ns 0.12 ns/pF
Additional Fall Time for 812 pF Normalized	50 ns 0.06 ns/pF	64 ns 0.08 ns/pF

* Fast and slow refer to the output programming option.

Routing

Horizontal Long Lines

As shown in Table 3, there are two horizontal long lines (HLLs) per row of CLBs. Each HLL is driven by one TBUF for each column of CLBs, plus an additional TBUF at the left end of the long line. This additional TBUF is convenient for driving IOB data onto the long line. In general, the routing resources to the T and I pins of TBUFs are somewhat limited.

Table 3. Number of Horizontal Long Lines

Part Name	Rows x Columns	CLBs	HLLs	TBUFs per HLL
ATT3020	8 x 8	64	16	9
ATT3030	10 x 10	100	20	11
ATT3042	12 x 12	144	24	13
ATT3064	16 x 14	224	32	15
ATT3090	20 x 16	320	40	17

Optionally, HLLs can be pulled up at either end or at both ends. The value of each pull-up resistor is 3 kΩ—10 kΩ.

In addition, HLLs are permanently driven by low-powered latches (bus hold circuits) that are easily overridden by active outputs or pull-up resistors. These latches maintain the logic levels on HLLs that are not pulled up and that are temporarily not driven. The logic level maintained is the last level actively driven onto the line.

When using 3-state HLLs for multiplexing, the use of fewer than four TBUFs can waste resources. Multiplexers with four or fewer inputs can be implemented more efficiently using CLBs.

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.

Routing (continued)

Clock Buffers

ATT3000 devices contain two high fan-out, low-skew, clock distribution networks. The global clock net originates from the GCLK buffer in the upper-left corner of the die, while the alternate clock net originates from the ACLK buffer in the lower-right corner of the die.

The global and alternate clock networks each have optional fast CMOS inputs called TCLKIN and BCLKIN, respectively. Using these inputs provides the fastest path from the PC board to the internal flip-flops and latches. Since the signal bypasses the input buffer, well-defined CMOS levels must be guaranteed on these clock pins.

To specify the use of TCLKIN or BCLKIN in a schematic, connect an IPAD symbol directly to the GCLK or ACLK symbol. Placing an IBUF between the IPAD and the clock buffer will prevent TCLKIN or BCLKIN from being used.

The clock buffer output nets only drive CLB and IOB clock pins. They do not drive any other CLB inputs. In rare cases where a clock needs to be connected to a logic input or a device output, a signal should be tapped off the clock buffer input and routed to the logic input. This is not possible with clocks using TCLKIN or BCLKIN.

The clock skew created by routing clocks through local interconnect makes safe designs very difficult to achieve, and this practice is not recommended. In general, the fewer clocks used, the safer the design. High fan-out clocks should always use GCLK or ACLK. If more than two clocks are required, the ACLK net can be segmented into individual vertical lines that can be driven by PIPs at the top and bottom of each column. Clock signals routed through local interconnect should only be considered for individual flip-flops.

General Information

Recovery from Reset

Recovery from reset is not specified in the ATT3000 data sheet because it is very difficult to measure in a production environment. The following values may be assumed for all ATT3000 devices and speed grades:

- The CLB can be clocked immediately (<0.2 ns) after the end of the internal reset direct signal (RD).
- The CLB can be clocked no earlier than 25 ns (worst-case) after the release of an externally applied global reset signal, i.e., after the rising edge of the active-low signal.

Configuration and Start-Up

Until the chip goes active after configuration, all I/O pins not involved in the configuration process remain in a high-impedance state with weak pull-up resistors, and all internal flip-flops and latches are held at reset. Multiple FPGA devices hooked up in a daisy chain will all go active simultaneously on the same CCLK edge. This is documented in the ATT3000 data sheet.

Not documented, however, 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 IOB inputs. Even the crystal oscillator starts operating as soon as it receives its configuration data. Since all flip-flops and latches are being held at reset and all outputs are being held in their high-impedance state, there is no danger in this staggered awakening of the internal logic. The operation of the logic prior to the end of configuration is even useful—it ensures that clock enables and output enables are correctly defined before the elements they control become active.

General Information (continued)

Once configuration is complete, the device is activated. This occurs on a rising edge of CCLK, when all outputs and clocks that are enabled become active simultaneously. Since the activation is triggered by CCLK, it is an asynchronous event with respect to the system clock. To avoid start-up problems caused by this asynchronism, some designs might require a reset pulse that is synchronized to the system clock.

The circuit shown in Figure 6 generates a short global reset pulse in response to the first system clock after the end of configuration. It uses one CLB and one IOB, and also precludes the use of the LDC pin as I/O.

During Configuration, $\overline{\text{LDC}}$ is asserted low and holds the D-input of the flip-flop high, while Q is held low by the internal reset, and RESET is kept high by internal and external pull-up resistors. At the end of configuration, the LDC pin is unasserted, but D remains high since the function generator acts as an R-S latch; Q stays low, and RESET is still pulled high by the external resistor. On the first system clock after configuration ends, Q is clocked high, resetting the latch and enabling the output driver, which forces RESET low. This resets the whole chip until the low on Q permits RESET to be pulled high again.

The whole chip has thus been reset by a short pulse instigated by the system clock. No further pulses are generated, since the high on LDC prevents the R-S latch from becoming set.

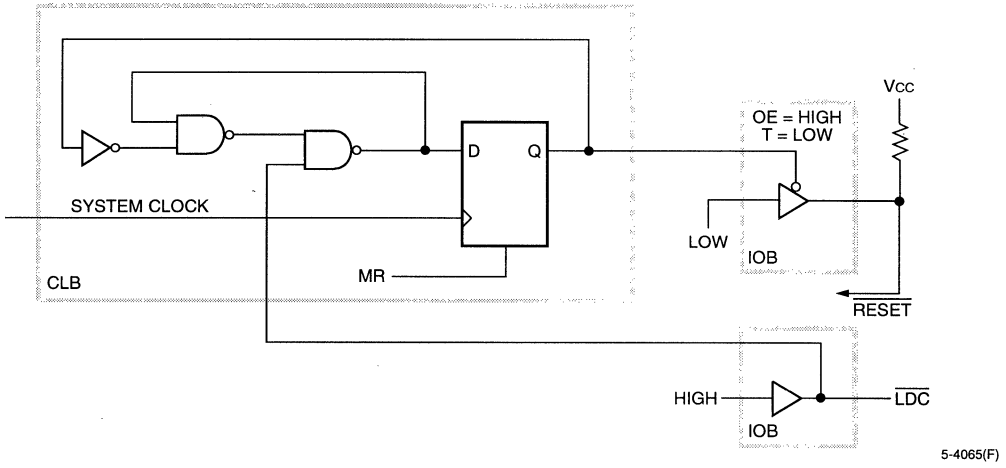


Figure 6. Synchronous Reset

General Information (continued)

Power Dissipation

As in most CMOS ICs, almost all FPGA power dissipation is dynamic and is caused by the charging and discharging of internal capacitances. Each node in the device dissipates power according to the capacitance in the node (which is fixed for each type of node) and the frequency at which the particular node is switching (which can be different from the clock frequency). The total dynamic power is the sum of the power dissipated in the individual nodes.

While the clock line frequency is easy to specify, it is usually more difficult to estimate the average frequency of other nodes. Two extreme cases are binary counters, where half the total power is dissipated in the first flip-flop, and shift registers with alternating zeros and ones, where the whole circuit is exercised at the clocking speed.

A popular assumption is that, on average, each node is exercised at 20% of the clock rate; a major EPLD vendor uses a 16-bit counter as a model, where the effective percentage is only 12%. Undoubtedly, there are extreme cases where the ratio is much lower or much higher, but 15% to 20% may be a valid approximation for most normal designs. Note that global clock lines must always be entered with their real, and well-known, frequency.

Consequently, most power consumption estimates only serve as guidelines based on gross approximations. Table 4 shows the dynamic power dissipation, in mW per MHz, for different types of ATT3000 nodes. While not precise, these numbers are sufficiently accurate for the calculations in which they are used, and may be used for any ATT3000 device. Table 5 shows a sample power calculation.

Table 4. Dynamic Power Dissipation

Example	ATT3020 (mW/MHz)	ATT3090 (mW/MHz)
One CLB driving three local interconnects	0.25	0.25
One device output with a 50 pF load	1.25	1.25
One global clock buffer and line	2.00	3.50
One long line without driver	0.10	0.15

Table 5. Sample Power Calculation for ATT3020

Node	Quantity	MHz	mW/ MHz	mW
Clock Buffer	1	40	2.00	80
CLBs	5	40	0.25	50
CLBs	10	20	0.25	50
CLBs	40	10	0.25	100
Long Lines	8	20	0.10	16
Outputs	20	20	1.25	500
Total Power =				**~800**

General Information (continued)

Crystal Oscillator

ATT3000 devices contain an on-chip crystal oscillator circuit that connects to the ACLK buffer. This circuit (see Figure 6) comprises a high-speed, high-gain, inverting amplifier, with its input connected to the dedicated XTL2 pin, and its output connected to the XTL1 pin. An external biasing resistor, R1, with a value of 0.5 MΩ to 1 MΩ is required.

A crystal, Y1, and additional phase-shifting components, R2, C1, and C2, complete the circuit. The capacitors, C1 and C2, in parallel, form the load on the crystal. This load is specified by the crystal manufacturer and is typically 40 pF. The capacitors should approximately equal 20 pF each for a 40 pF crystal.

Either series- or parallel-resonant crystals may be used, since they differ only in their specification. Crystals constrain oscillation to a narrow band of frequencies, the width of which is <1% of the oscillating frequency. The exact frequency of oscillation within this band depends on the components surrounding the crystal. Series-resonant crystals are specified by their manufacturers according to the lower edge of the frequency band; parallel-resonant crystals are specified according to the upper edge.

The resistor R2 controls the loop gain, and its value must be established by experimentation. If it is too small, the oscillation will be distorted; if it is too large, the oscillation will fail to start or will only start slowly. In most cases, the value of R2 is noncritical, and typically is 0 kΩ to 1 kΩ.

Once the component values have been chosen, it is good practice to test the oscillator with a resistor (~1 kΩ) in series with the crystal. If the oscillator still starts reliably, independent of whether the power supply turns on quickly or slowly, it will always work without the resistor.

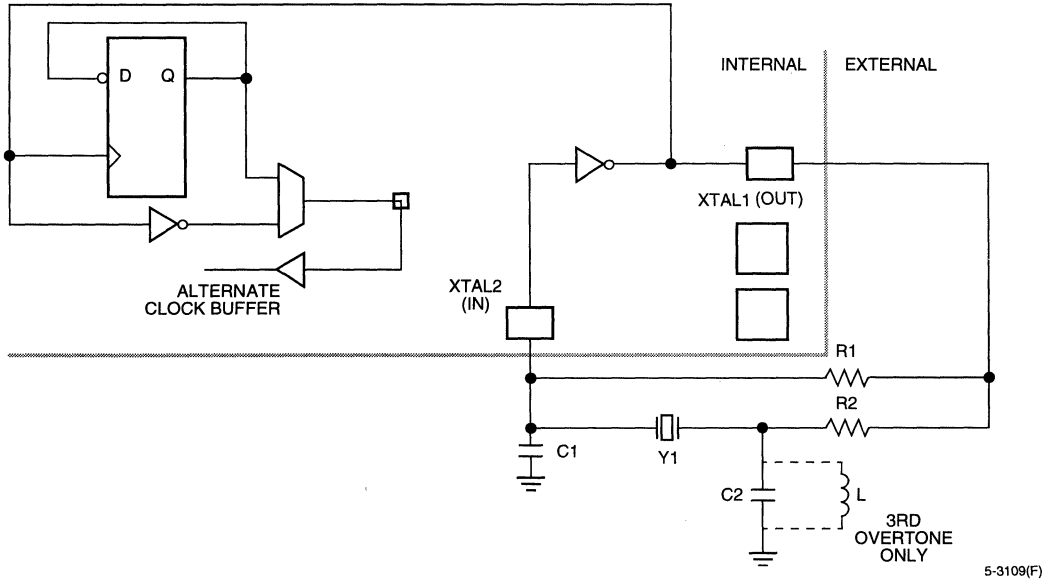


Figure 7. Crystal Oscillator Inverter

General Information (continued)

For operation above 20 MHz to 25 MHz, the crystal must operate at its third harmonic. The capacitor C2 is replaced by a parallel-resonant LC tank circuit tuned to ~2/3 of the desired frequency; i.e., twice the fundamental frequency of the crystal. Table 6 shows typical component values for the tank circuit.

Table 6. Third Harmonic Crystal Oscillator Tank Circuit

Frequency (MHz)	LC Tank				
	L (μH)	C2 (pF)	Freq. (MHz)	R2 (Ω)	C1 (pF)
32	1	60	20.6	430	23
35	1	44	24.0	310	23
49	1	31	28.6	190	23
72	1	18	37.5	150	12

CCLK Frequency Variation

The on-chip R-C oscillator that is brought out as CCLK also performs several other internal functions. It generates the power-on delay: 2^{16} = 65,536 periods for a master and/or 2^{14} = 16,384 periods for a slave or peripheral device. It generates the shift pulses for clearing the configuration array, using one clock period per frame, and it is the clock source for several small shift registers acting as low-pass filters for a variety of input signals. The nominal frequency of this oscillator is 1 MHz with a maximum deviation of +25% to –50%. The clock frequency, therefore, is between 1.25 MHz and 0.5 MHz.

Lucent Technologies' circuit designers make sure that the internal clock frequency does not get faster as devices are migrated to smaller geometries and faster processes. Even the newest and fastest Lucent Technologies FPGA is compatible with the oldest and slowest device ever manufactured. The CCLK frequency is fairly insensitive to changes in VCC, varying only 0.6% for a 10% change in VCC. It is, however, very temperature dependent, increasing 40% as the temperature drops from +25 °C to –30 °C (see Table 7).

Table 7. CCLK Frequency Variation

Vcc (V)	Temperature (°C)	Frequency (kHz)
4.5	+25	687
5.0	+25	691
5.5	+25	695
4.5	–30	966
4.5	+130	457

Metastable Recovery

Whenever a clocked flip-flop synchronizes an asynchronous input, there is a small 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 specifications, but actually occurs within the tiny timing window where the flip-flop accepts the new input. Under these circumstances, the flip-flop can enter a symmetrically balanced transitory state, called metastable (meta = between).

While the slightest deviation from perfect balance will cause the output to revert to one of its two stable states, the delay in doing so depends not only on the gain bandwidth product of the circuit, but also on how perfect the balance is and on the noise level within the circuit; the delay 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 to either a 0 or a 1) but the unpredictable timing of the final change to a valid logic state. If the metastable flip-flop drives two destinations with differing path delays, one destination might reflect the final data state, while the other does not.

With the help of a mostly self-contained circuit on the demonstration board, the ATT3020-70 CLB flip-flop was evaluated. The result of this evaluation shows the ATT3000 CLB flip-flop to be superior in metastable performance to many popular MSI and PLD devices.

Statistically, when an asynchronous event with a frequency approximately 1 MHz is being synchronized by a 10 MHz clock, the CLB flip-flop suffers an additional delay of 4.2 ns once per hour and 8.4 ns once per 1000 years.

General Information (continued)

The frequency of occurrence of these metastable delays is proportional to the product of the asynchronous event frequency and the clock frequency. If, as an example, a 100 kHz event is synchronized by a 2 MHz clock, the above delays (besides being far more tolerable) will occur 50 times less often.

Since metastability can only be measured statistically, this data was obtained by configuring an ATT3020 with eight concurrent detectors. Eight D-type flip-flops were clocked from a common high-speed source, and their D inputs driven from a common, lower frequency asynchronous signal, Figure 8. The output of each flip-flop fed the D inputs of two more flip-flops, one clocked one-half clock period later and the second a full clock period later.

If a metastable event in the first flip-flop increased the output settling time to more than one-half clock period, the second two flip-flops would capture differing data. Therefore, the occurrence of a long metastable delay could be detected using a simple comparator.

Deliberate skew in the input data to the eight metastable circuits ensured that, at most, one metastable event could occur each clock. This permitted the eight detectors to be ORed into a single metastable event counter.

As expected, no metastable events were observed at clock rates below 25 MHz, since one-half clock period of 20 ns is adequate for almost any metastability resolution delay, plus the flip-flop setup time. Increasing the clock rate to around 27 MHz brought a sudden burst of metastable events. Careful adjustment of the clock frequency gave repeatable, reliable measurements, showing that a 500 ps decrease in the half clock period increased the frequency of metastable occurrences by a factor of 41.

To be conservative, to compensate for favorable conditions at room temperature, and to avoid any possibility of overstating a good case, the measurements were interpreted as follows: when capturing asynchronous data, the error rate decreases by a factor of 40 for every additional nanosecond of metastability resolution delay that the system can tolerate.

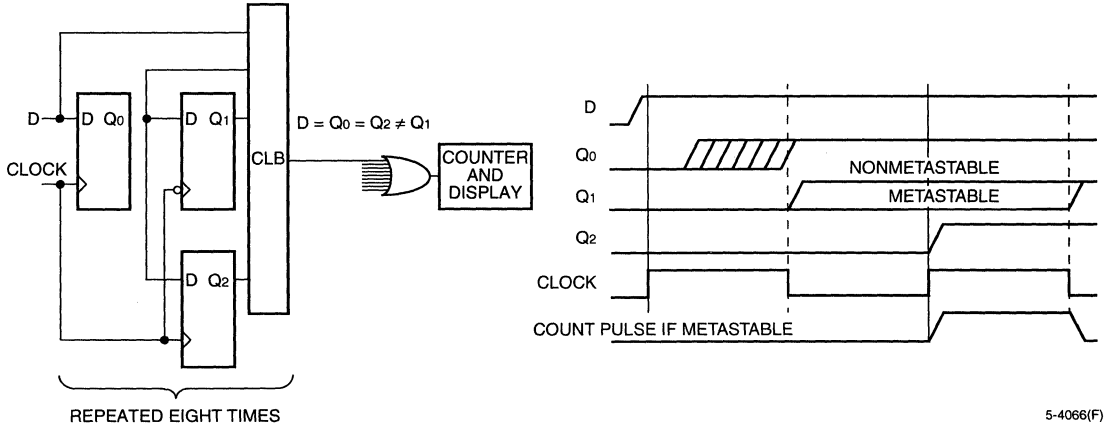


Figure 8. Metastable Measuring Circuit

General Information (continued)

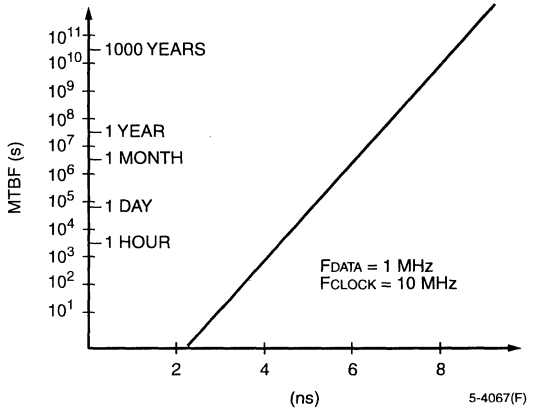
Metastability Calculations

The mean time between failures (MTBF) can only be defined statistically. It is inversely proportional to the product of the two frequencies involved: the clock frequency (F_1) and the average frequency of data changes (F_2), provided that these two frequencies are independent and have no correlation. K_1 is a factor that has the dimension of time, and describes the likelihood of going metastable. K_2 is an exponent that describes the speed with which the metastable condition is being resolved.

$1/MTBF = F_1 \bullet F_2 \bullet K_1 \bullet e(-K_2 \bullet t)$

MTBF in seconds
F_1 and F_2 in Hz
$K_1 = 1.5 \bullet 10^{10}$ seconds (measured for ATT3020-70)
$K_2 = \ell n(40)$ per ns $= 3.69 \bullet 10^9$ per second (ATT3020-70)

For a 10 MHz clock and approximately a 1 MHz data rate, the table below gives the expected MTBF as a function of the acceptable extra delay at the output of the metastable going flip-flop.



Extra Delay (ns)	MTBF
1.0	27 ms
4.2	1 hr
6.7	423 days
8.5	890 years
10.0	225,000 years
11.0	9 million years
12.0	360 million years

Figure 9. Metastable MTBF as a Function of Additional Acceptable Delay

General Information (continued)

Battery Backup

Since logic cell arrays are manufactured using a high-performance, low-power CMOS process, they can preserve the configuration data 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 the minimal current requirement of 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 the primary VCC supply and battery

Important considerations include the following:

- Ensure that $\overline{\text{PWRDWN}}$ is asserted logic low prior to VCC falling, that it is held low while the primary VCC is absent, and that it is returned high after VCC has returned to a normal level. $\overline{\text{PWRDWN}}$ edges must not rise or fall slowly.
- Ensure glitch-free switching of the power connections to the FPGA device from the primary VCC to the battery and back.
- Ensure that during normal operation the FPGA VCC is maintained at an acceptable level, 5.0 V + 5% (+10% for industrial).

Figure 10 shows a powerdown circuit developed by Shel Epstein of Epstein Associates, Wilmette, IL. Two Schottky diodes power the FPGA from either the 5.2 V primary supply or a 3 V Lithium battery. A Seiko S8054 three-terminal power monitor circuit monitors VCC and pulls $\overline{\text{PWRDWN}}$ low whenever VCC falls below 4 V.

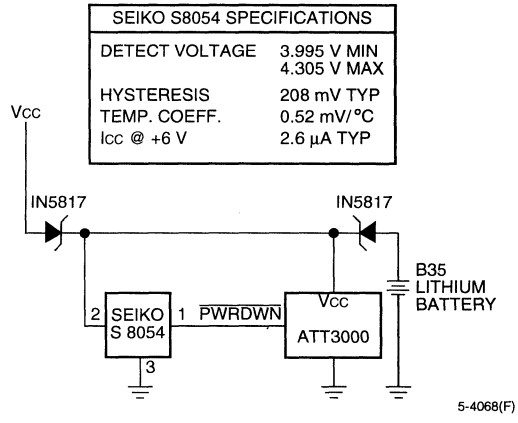


Figure 10. Battery Backup Circuit

ATT3000 Series FPGAs Test Methodology

Introduction

Lucent Technologies is committed to building in the highest level of quality and reliability into its Field-Programmable Gate Arrays (FPGAs). 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.

Component Test Methodology

As quality consciousness has grown among semiconductor users, awareness of the importance of testability has also increased. Testing of 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 production testing costs, it can add about 30% more circuitry, decrease performance by up to 20%, and increase design time.

Component Test Methodology
(continued)

Field-Programmable Gate Arrays (FPGAs): The testability of the FPGA device is similar to other standard products, including microprocessors and memories. The following outlines the FPGA 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 in such a way 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 Lucent 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 it 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.

FPGA Test Strategies

Every Lucent FPGA 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 V_{CC} minimum and maximum.

FPGA Test Strategies (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.

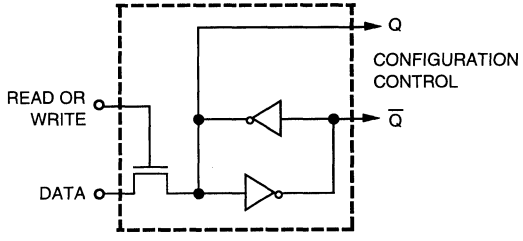


Figure 1. Configuration Memory Cell

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 checking for continuity of the line.

FPGA Test Strategies (continued)

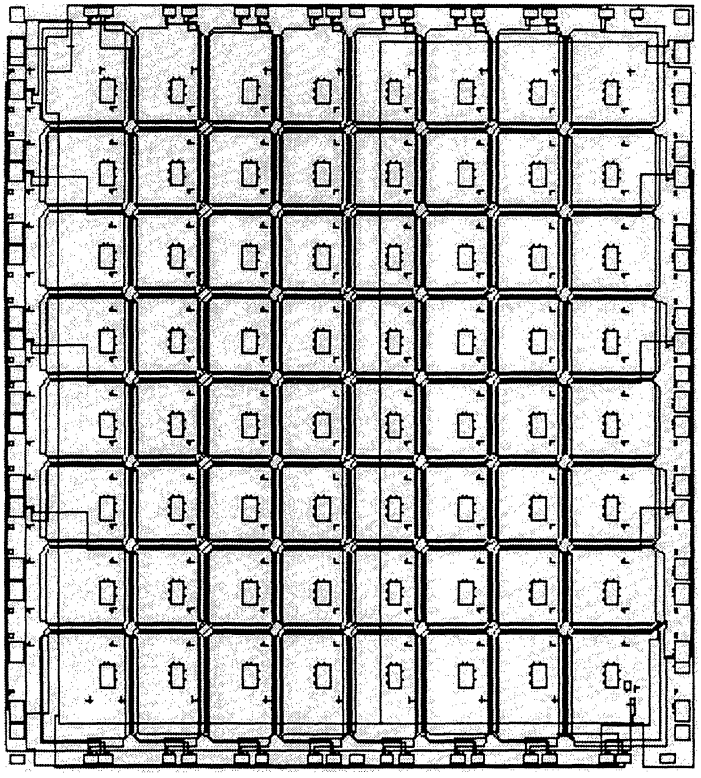


Figure 2. Interconnect Test Pattern

FPGA Test Strategies (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.

Speed Testing

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 T_{CKO} + T_{ILO} + INTERCONNECT + T_{ICK} 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 Configurations

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

Hardware Testing Configurations (continued)

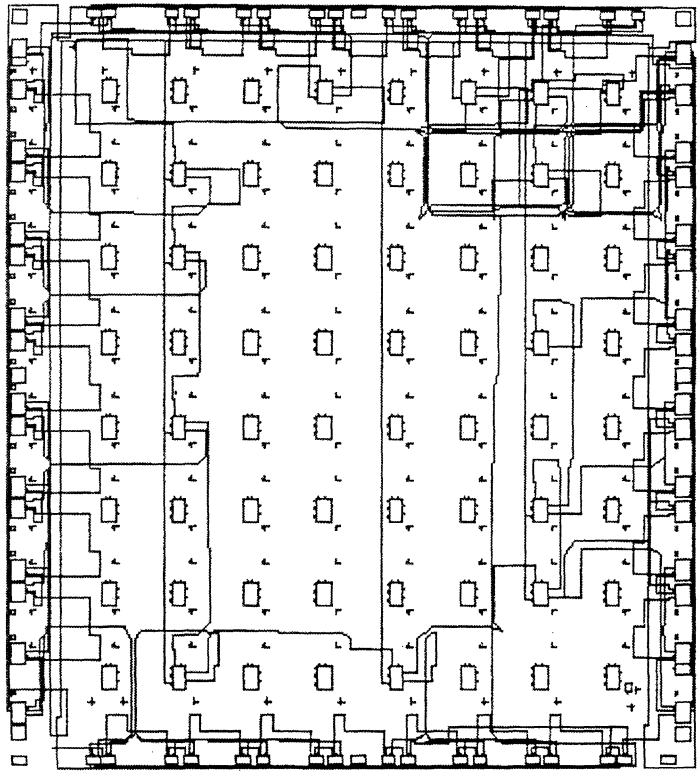


Figure 3. IOB Test Pattern

Hardware Testing Configurations (continued)

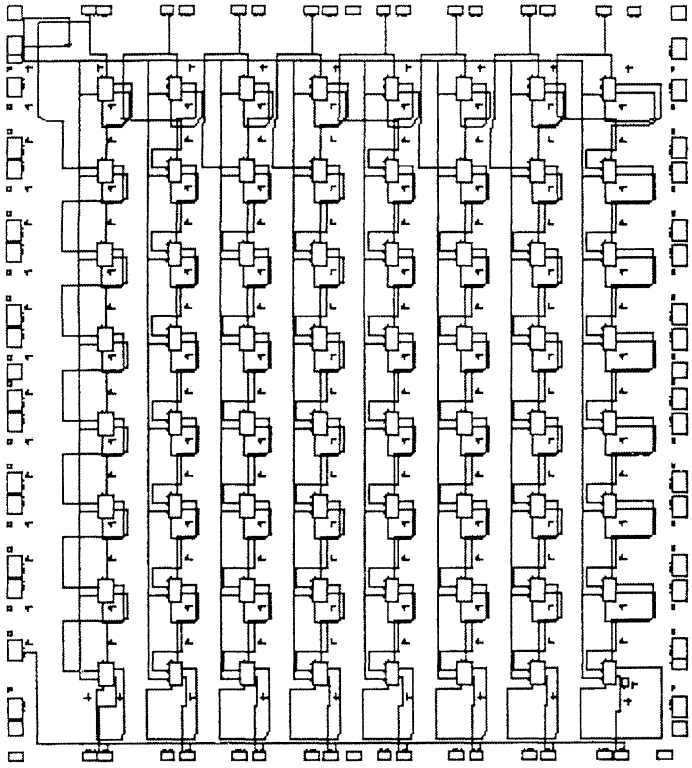


Figure 4. Speed Test Pattern

Hardware Testing Configurations (continued)

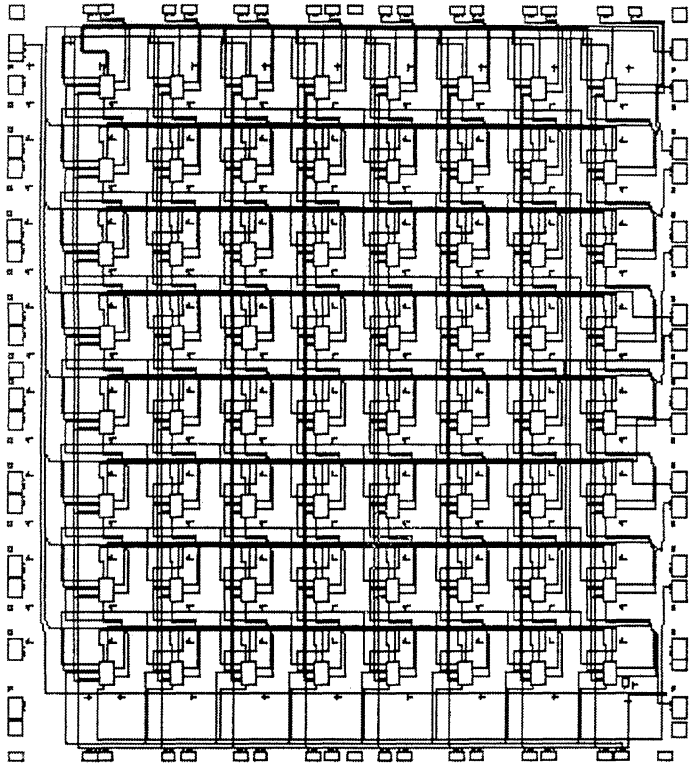


Figure 5. CLB Test Pattern

Lucent Technologies
Bell Labs Innovations

Technical Support

Introduction

The Lucent Technologies FPGA product line is committed to providing first-class technical support. We recognize that a successful support system employs multiple methods of distributing technical information. Consequently, we have structured our technical support team to provide a number of support options to our customers.

If you have comments or suggestions regarding our technical support program, please share them with us. We consider your feedback our guarantee that we will be able to continually improve our support efforts.

How to Reach Us

Type of Request	Method	Phone/FAX/Address
Technical Support	Help Line (9 a.m. to 7 p.m. EST) FAX E-Mail FTP Site BBS Field Engineers	1-800-EASY-FPGA or 1-800-327-9374 or 1-610-712-4331 1-610-712-4754 orcafpga@aloft.lucent.com orca.fast.net 1-610-712-4314 See following list
Product Information	Worldwide Web	http://www.lucent.com/micro/fpga
Technical Documentation	Marketing Communications	United States: 1-800-372-2447 (Phone) 1-610-712-4106 (FAX) Canada: 1-800-553-2448 (Phone) 1-610-712-4106 (FAX) Asia Pacific: +65-778-8833 (Phone) +65-777-7495 (FAX) Europe: +44-1734-324-299 (Phone) +44-1734-328-148 (FAX) Japan: +81-3-5421-1600 (Phone) +81-3-5421-1700 (FAX)
File Transfers	E-Mail FTP Site BBS	orcafpga@aloft.lucent.com orca.fast.net 1-610-712-4314

Field Application Engineers

We encourage you to make full use of our FPGA field application engineers (FAEs). Each engineer has been fully trained and is available to provide front-line support over the telephone or on-site (please see the list below). A complete list of all Lucent Technologies sales offices appears in Chapter 10.

Location	Phone
Domestic	
Boston	508-626-3720 or 508-626-3756
Orlando	407-667-4716
Maryland/ North Carolina	410-583-6060
Chicago	708-979-1646
Colorado	303-799-8100
Dallas	214-556-8833
Minneapolis	612-885-4325
Northern California	408-980-3804 or 408-980-3848
Southern California	818-587-9600 or 714-890-6204
Allentown	610-712-5431
Northern NJ	908-949-9724
Foreign	
Canada	514-747-1211 or 905-672-2030
London	+44-1344-487-111
Munich	+49-89-95086-0
Singapore	+65-778-8833
Tokyo	+81-03-5421-1790

Bulletin Board Service

Lucent Technologies has established a 24-hour bulletin board service (BBS) that permits instant access to the latest software patches, software utility programs, application notes, and reference designs. The BBS can be used to transfer files to and from the technical support organization for review. The BBS software and hardware requirements are listed below. For more detailed information on its access, please see the September 1996 *FPGA Bulletin Board Service* product brief (PN96-070FPGA), which appears after this Introduction.

FTP Site

The FTP site can be used to transfer files to and from the technical support organization for review. For more detailed information on accessing the FTP site, please see the September 1996 ORCA *FPGA FTP Site* product brief (AP96-050FPGA), which appears after this Introduction.

E-Mail

Customers can use E-Mail to send technical question about devices and software. Requests made to **orcafpga@aloft.lucent.com** are checked regularly throughout the day and are given the same priority as telephone calls. Please note, however, that urgent issues should be directed to the Technical Support Helpline or FAX in order to eliminate Internet-related delays.

Worldwide Web

Lucent Technologies has established a 24-hour worldwide web site at **http://www.lucent.com/micro/fpga** for instant access to FPGA product information, newsworthy information, upcoming events, employment opportunities, application notes, product briefs, data sheets, and sales information. The Web site will be updated regularly to provide customers with the latest news and information.

Technical Documentation

The Lucent Technologies Marketing Communications Group offers a variety of FPGA technical literature; including application notes, data sheets, data books, product briefs, and marketing briefs. Documentation requests can be referred to them worldwide, as noted previously.

Lucent Technologies
Bell Labs Innovations

FPGA Bulletin Board Service

Introduction

In order to provide customers with the latest technical information on its 3000 and *ORCA* series field-programmable gate arrays, Lucent Technologies Microelectronics Group has established an electronic bulletin board service (BBS). The BBS can be reached at 1-610-712-4314.

This application note will define the software, hardware, and general user information required for accessing the Lucent FPGA bulletin board service.

Description

Software and Hardware Requirements

Table 1. Software and Hardware Requirements

Baud Rates	28,800 baud or slower
Character Format	8 data, no parity, 1 stop bit
Transfer Protocols	Xmodem, Ymodem, Zmodem, Kermit
Phone Number	1-610-712-4314
Hours	24 hours, 7 days per week

Logging On

After the BBS connection is established, new users will be prompted to supply a user-ID and password and also to answer several questions about themselves. Following the questions, users will be asked to download a document named BBSUSER.DOC. This file provides general help about BBS operations. It is strongly recommended that new users download this file for reference.

The Main Menu, from which the most commonly used functions are accessed, appears next. The functions available are Application Notes and Miscellaneous Utilities, Upload/Download Files to Applications, Patches to *ORCA* Foundry, Account Display Edit, and Information Center. Each function is accessed by entering the letter next to the description.

BBS Functions

- **Information Center** provides information and background about the BBS and Lucent's FPGA Applications Group.
- **Application Notes and Miscellaneous File Utilities** accesses libraries that contain various application notes and miscellaneous file utilities for *ORCA* Foundry.
- **Upload/Download Files** accesses a submenu from which the user can upload or download files to/from the Lucent FPGA Applications Group. Please do not place any files here before speaking with a Factory Applications or Technical Support Engineer.
- **Patches** contains patches to the latest version of *ORCA* Foundry software. A copy of the *PKUNZIP* program for *SunOS* is also located in this area and is available for download.
- **Account Edit/Display** allows users to change their personal information, including their password.

Description (continued)

File Uploads

The UPLOADS area is the only area that a user is permitted to upload files to. The UPLOADS Library is intended for customer designs and files that require attention by the FPGA Applications Group. The UPLOADS area is secure since no user is permitted to download or examine files in this area. To gain access to the UPLOADS area, execute the following steps:

1. Select (**F**) Upload/Download Files to Applications from the Main Menu.
2. At the next menu, select the (**U**) UPLOADS Function.
3. Select (**U**) UPLOADS again.
4. Supply the name of the file to be uploaded.
5. When asked for a description, please specify the applications engineer who should receive the file(s).
6. Select a transfer protocol compatible with your system and communications software.
7. Initiate the transfer by uploading the file from your remote PC. For example, if you are using *Windows* terminal software, click on the Transfer menu and select the Send Binary File option. Please compress all files with compression software (e.g., *PKZIP*) before attempting an upload.

File Downloads

There are several libraries that contain information, data, programs, release notes, and software patches. These are available for free download from the BBS to the user's system. Most files are compressed with *PKZIP* to conserve space and transfer time.

Customer Support

If you have any questions about the BBS, please call our toll-free hotline at 1-800-EASY-FPG(A) in the U.S. and Canada, or 1-610-712-4331 outside the U.S.A. and Canada.

Lucent Technologies
Bell Labs Innovations

ORCA FPGA FTP Site

Overview

In order to supply its *ORCA* (optimized reconfigurable cell array) Field-Programmable Gate Array (FPGA) customers with the latest technical information, Lucent provides an anonymous file transfer protocol (FTP) site. The FTP site is an account that any Lucent Technologies Microelectronics Group customer can access to get software patches, libraries, and technical information, and to upload data. The site makes use of ARPANET—standard FTP facilities to send and receive data. The account can be accessed by any PC or workstation supporting the ftp command.

Logging Into the Account

The following is a brief primer in the form of a sample session, with comments. Text in **boldface** is what you should type, comments are in *italics*.

```
% ftp orca.fast.net
Connected to tcp!204.179.186.34!1735.
220-
220-  If your FTP client crashes or hangs shortly after login, try using a
220-  dash (-) as the first character of your password.  This will turn off
220-  the informational messages which may be confusing your ftp client.
220-
220-                    ***NOTICE***
220-  All ftp connections are tracked and logged.  If this upsets you,
220-  please disconnect now.  Problems to ftp-bugs@fast.net -- Emergency
220-  (610) 954-5910.  You are logged in from site gw4.att.com
220-
220   ws.fast.net FTP server (Version wu-2.4(1) Fri Jul 21 00:38:02 EDT 1995) ready.
Name (orca.fast.net:mylogin): anonymous
331 Guest login ok, send your complete e-mail address as password.
Password: mylogin@
230   Guest login ok, access restrictions apply.
Remote system type is UNIX.
Using binary mode to transfer binary files.
200 Type set to I.
ftp> cd orca
250-_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/
```

Logging into the Account (continued)

```
250-
250-            Welcome to the
250-
250-           * ORCA FTP Site *
250-
250-     This site is maintained by the
250-      Lucent FPGA Applications Group
250-   Problems with this site can be re-
250-   ported to the Lucent FPGA Hotline at
250-
250-            1-800-EASY-FPG(A)
250-                    or
250-             1-800-327-9374
250-
250-   International customers should call
250-
250-             1-610-712-4331
250-
250-   or you may E-mail the postmaster at
250-
250-           orca@orca.fast.net
250-
250-   Please use BINARY mode for all file
250-   transfers. Files may be uploaded to
250-   the following directory:
250-
250-             /orca/incoming
250-
250-_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/
250-
250 CWD command successful.
```

Uploading Data to the Account

A directory named **incoming** exists for depositing design data. To put data in the account, issue the following commands:

```
ftp> cd incoming (change to this area to deposit data)
250-_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/
250-
250-         Current directory is
250
250-            /orca/incoming
250-
250-  *** This is a write-only directory ***
250-
250-  Please insure that all files for upload
250-  have first been compressed with one of
250-  the following utilities:
250-
250-  UNIX Systems: tar, compress, gzip, zip
250-    PC Systems: PKZIP
250-
250-  Also, please notify the FPGA Applica-
250-  tions Engineering group that you are
250-  placing files at this site.
250-
250-  The Lucent Applications Engineering Group
250-  can be contacted at:
250-
250-            1-800-EASY-FPG(A)
250-
250-                   or
250-
250-             1-800-327-9374
250-
250-   International customers should call
250-
250-             1-610-712-4331
250-
250-_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/
250-
250 CWD command successful.
ftp> put design.tar.Z (deposit the data)
```

Please contact an Applications Engineer after depositing the data. There is no automatic notification that data has been received.

Downloading Data from the Account

The account contains a number of directories structured according to subject:

orca/foundry7.1	Patches for bugs/enhancements to *ORCA* Foundry v7.1
orca/foundry9.0	Patches for bugs/enhancements to *ORCA* Foundry v9.0
orca/exemplar	Files related to Exemplar's *CORE* and ORCA'x'press products
orca/viewlogic	Files related to *Viewlogic* products
orca/mentor	Files related to *Mentor Graphics* products
orca/synopsys	Files related to *Synopsys* products
orca/incoming	Upload area
orca/misc	Miscellaneous data/information/programs
orca/docs	Documentation and application notes
orca/outgoing	Download area

To download a file from one of these areas, use the following commands:

```
ftp> cd synopsys (changes directory to appropriate area—assumes you are already in ORCA directory tree)
...
ftp> get synopsys.tar.Z (retrieves the data)
200 PORT command successful.
```

Using a Web Browser to Access the Account

You can also access the FTP site with a WWW browser, such as Netscape *Navigator* or *Mosaic*. The URL (universal resource locator) for accessing the account is **ftp://orca.fast.net/orca**. Figure 1 illustrates the page as displayed with the Netscape browser.

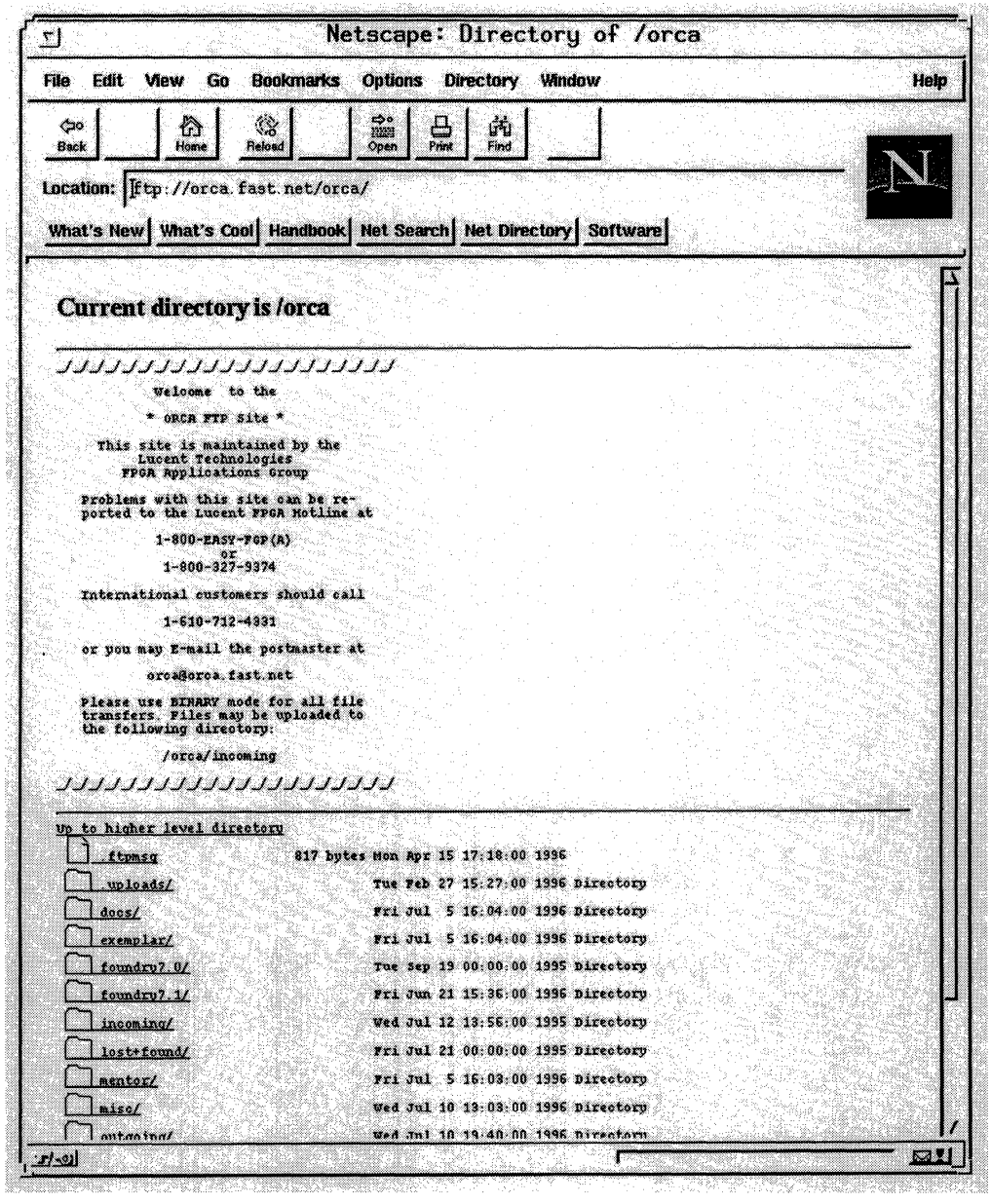


Figure 1. Accessing the FTP Site with the Netscape Browser

Notes

- If there is no activity for 15 minutes, you will be disconnected and given the following message:

```
421 Timeout (900 seconds): closing control connection.
```

- You should always specify binary mode for file transfers since most files will be compressed. Use the binary command to change modes, as in the following example:

```
ftp> binary
200 Type set to I.
```

- Files for *UNIX* workstations will have a *ZIP*, *TAR*, and/or *Z* file extension. PC files will be compressed with *PKUNZIP* v2.04g and have a *ZIP* file extension.
- Up to 40 users can be logged into the FTP site at one time. Peak usage is between 10 a.m. and 5 p.m. E.S.T. If you can't connect, try again at one-half hour intervals.
- For questions or comments, contact the Lucent FPGA Hotline at 1-800-EASY-FPG(A), or 1-610-712-4331 outside the U.S. and Canada.

Lucent Technologies Microelectronics Group World Headquarters

Two Oak Way
Berkeley Heights, NJ 07922
(1) 908-771-2000 (phone)
(1) 908-771-3551 (FAX)

Sales Offices

Lucent Technologies' sales offices are located worldwide. For the nearest location, please call 1-800-372-2447.

U.S. Manufacturing Locations

Allentown
Lucent Technologies ME
Allentown Works
555 Union Boulevard
Allentown, PA 18103
(1) 610-712-6011 (phone)

Dallas
Lucent Technologies ME

Dallas Works
3000 Skyline Drive
Mesquite, TX 75149
(1) 972-284-2000 (phone)

Reading
Lucent Technologies ME
Reading Works
2525 North 12th Street
Reading, PA 19612
(1) 610-939-7011 (phone)

Merrimack Valley
Lucent Technologies ME
Merrimack Valley Works
1600 Osgood Street
North Andover, MA 01845
(1) 508-960-2000 (phone)

Orlando
Lucent Technologies ME

Orlando
9333 S. John Young Parkway
Orlando, FL 32819-8698
(1) 407-345-6000 (phone)

Richmond
Lucent Technologies ME
Richmond Works
4500 S. Laburnum Avenue
Richmond, VA 23231
(1) 804-226-5000

SSTC
Lucent Technologies ME - Bell Labs
Solid State Tech. Center
9999 Hamilton Boulevard
Breinigsville, PA 18031-9359
(1) 610-391-2000 (phone)

Design Centers

Bracknell
Lucent Technologies ME

Microelectronics House
Broad Lane
Bracknell
Berkshire RG 129GX
United Kingdom
(44) 1344-865-935 (phone)
(44) 1344-865-923 (FAX)

Madrid
Lucent Technologies
Microelectronica, S.A.
Poligono Industrial
Tres Cantos, Zona Oeste
28760 Tres Cantos
Madrid, Spain
(341) 807-1700 (phone)
(341) 807-1699 (FAX)

Munich
Lucent Technologies
Deutschland GmbH
Bahnhofstrasse 27A
D-85774 Unterfoehring
F.R.G.
(49) 89-95086-211 (phone)
(49) 89-95086-222 (FAX)

Japan
Lucent Technologies Japan
7-18, Higashi-Gotanda 2-chome
Shinagawa-ku, Tokyo 141,
Japan
(81) 3-5421-1692 (phone)
(81) 3-5421-1702 (FAX)

Singapore
Lucent Technologies Bell Labs-
Microelectronics Asia/Pacific
Regional I.C. Design Center
77 Science Park Drive
#03-18 Cintech III
Singapore Science Park
Singapore 118256
(65) 770-7600 (phone)
(65) 777-5464 (FAX)

Santa Clara
Lucent Technologies ME
4995 Patrick Henry Drive
Santa Clara, CA 95054
(1) 408-980-3710 (phone)
(1) 408-980-3733 (FAX)

Taiwan
Lucent Technologies Taiwan, Inc.
7/12/14th Floor,
Overseas Trust Building
249, Section 1, Tun Hwa South Rd.
Taipei, Taiwan 106, R. O. C.
(886) 2-775-6374 (phone)
(886) 2-781-5690 (FAX)

OEM Sales Offices

United States Regional OEM Sales Offices

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
450 East Devon Avenue
Suite 225
Itasca, IL 60143-1262
(1) 708-773-8858 (phone)
(1) 708-773-8854 (FAX)

Dallas
222 West Las Colinas Blvd.
Suite 950
Irving, TX 75039
(1) 214-556-8800 (phone)
(1) 214-401-0390 (FAX)

Denver
7979 East Tufts Avenue
2nd Floor
Denver, CO 80237
(1) 303-290-2336 (phone)
(1) 303-290-4241 (FAX)

Los Angeles
7711 Center Avenue
Suite 100
Huntington Beach, CA 92647
(1) 714-890-4198 (phone)
(1) 714-890-6229 (FAX)

Minneapolis
1650 West 82nd Street
Suite 550
Bloomington, MN 55431
(1) 612-885-4300 (phone)
(1) 612-885-4331 (FAX)

Portland
10220 S.W. Greenburg Road
Suite 520
Portland, OR 97223
(1) 503-244-3883 (phone)
(1) 503-244-2315 (FAX)

San Francisco
4995 Patrick Henry Drive
Santa Clara, CA 95054
(1) 408-980-3700 (phone)
(1) 408-980-3805(FAX)

7711 Center Avenue
Suite 100
Huntington Beach, CA 92647
(1) 714-890-4101 (phone)
(1) 714-890-4133 (FAX)

OEM Sales Offices (continued)

International Regional OEM Sales Offices

Berlin
Lucent Technologies ME
Rudower Chaussee 5
D-12484 Berlin, Germany
(49) 30-639-25370 (phone)
(49) 30-664-3225 (FAX)

Bracknell
Lucent Technologies ME
Microelectronics House
Broad Lane
Bracknell, Berkshire
RG129GS, United Kingdom
(44) 1344-865-900 (phone)
(44) 1344-865-990 (FAX)

Helsinki
Lucent Technologies ME
Tekniikantie 12
SF-02150 ESPOO
Helsinki, Finland
(358) 0-4354-2800 (phone)
(358) 0-463-263 (FAX)

Madrid
Lucent Technologies ME
Poligono Industrial de Tres Cantos
s/n (Zona Oeste)
E-28770 Colmenar Viejo
Madrid, Spain
(34) 1-807-1700 (phone)
(34) 1-807-1420 (FAX)

Milan
Lucent Technologies ME
Viale Fulvio Testi 117
I-20092 Cinisello Balsamo
Milan, Italy
(39) 2-6601-1800 (phone)
(39) 2-6127-005 (FAX)

Munich
Lucent Technologies ME
Bahnhofstrasse 27a
D-85774 Unterfoehring
Munich, Germany
(49) 89-95086-0 (phone)
(49) 89-95086-333 (FAX)

Paris
Lucent Technologies ME
Tour Horizon
52 Quai de Dion Bouton
Paris, France
F-92800 Puteaux Cedex
(33) 1-4767-4767 (phone)
(33) 1-4767-4724 (FAX)

Stockholm
Lucent Technologies ME
Lucent Nordics AB
PO Box 10703
121 29 Stockholm-Globen
Sweden
(46) 8600-7000 (phone)
(46) 8600-7010 (FAX)

Korea-Seoul
Lucent Technologies ME
17th Floor Securities Supervisory Board Building
#27 Yoido-Dong Youngdeungpo-ku
Yoido P.O. Box 341
Seoul, Korea
(82) 2-767-4000 (phone)
(82) 2-767-4001 (FAX)

Japan-Osaka
Lucent Technologies ME
2-10 Minami Semba, 4-chome
Chuo-ku, Osaka 542
Japan
(81) 6-258-6222 (phone)
(81) 6-258-6448 (FAX)

Japan-Tokyo
Lucent Technologies ME
7-18, Higashi-Gotanda 2-chome
Shinagawa-ku, Tokyo 141
Japan
(81) 3-5421-1615 (phone)
(81) 3-5421-1700 (FAX)

Singapore
Lucent Technologies ME

14 Science Park Drive
#03-02A/04 The Maxwell
Singapore 0511
(65) 778-8833 (phone)
(65) 777-7495 (FAX)

Taiwan-Taipei
Lucent Technologies ME Taiwan, Inc
16F, Hung Tai Century Tower
156, Min Sheng E. Rd., Sec. 3
Taipei 105
Taiwan
R. O. C.
(886) 2-5473131 (phone)
(886) 2-7133246 (FAX)

China
Lucent Technologies China Inc.

17/F Shartex Plaza
No. 88 Zun Yi Nan Road
Shanghai 200335
People's Republic of China
(86) 21-219-0468 (phone)
(86) 21-219-0467 (FAX)

National Accounts Sales Offices and Location Codes

Address	Loc. Code
Arkansas **Lucent Technologies ME** Department 433 P.O. Box 8912 7600 Interstate 30 Little Rock, AR 72219-8912 (1) 501-569-4483	**TY**
Colorado **Lucent Technologies ME** Denver Works Room 2H10, Bldg. 22 11900 North Pecos Street Denver, CO 80234 (1) 303-538-2597	**DR**
Florida **Lucent Technologies ME** 8545 126th Avenue North P.O. Box 2826 Room LG140 Largo, FL 34649-2826 (1) 813-530-2336	
Illinois **Lucent Technologies ME** Indian Hill West Room 2A107 1100 East Warrenville Road Naperville, IL 60566 (1) 708-713-1878	**IW**
Lucent Technologies Ntwk Sys Indian Hill South Room 1G129 1200 East Warrenville Road Naperville, IL 60566 (1) 708-979-1763	**IX**
Indiana **Lucent Technologies ME** Heritage Park Room 3F125 6612 East 75th Street P.O. Box 1008 Indianapolis, IN 46206 (1) 317-845-3649	**INH**
Louisiana **Lucent Technologies ME** Department 931 2929 Baird Road Shreveport, LA 71118-3823 (1) 318-459-7524	**SP**

Address	Loc. Code
Massachusetts **Lucent Technologies Bell Labs** Merrimack Valley Works Room 2B-05BL 1600 Osgood Street North Andover, MA 01845 (1) 508-960-6063	**MV**
New Jersey **Lucent Technologies Bell Labs** Room 1B307 Crawfords Corner Road Holmdel, NJ 07733 (1) 908-949-1471	**HO**
Lucent Technologies Bell Labs Room 3K275 200 Laurel Avenue Middletown, NJ 07748 (1) 908-957-3870	**MT**
Lucent Technologies Bell Labs Room 4A102 67 Whippany Road Whippany, NJ 07981 (1) 201-386-6649	**WH**
North Carolina **Lucent Technologies ME** I-85 and Mt. Hope Church Road Room E1G26 McLeansville, NC 27301 (1) 919-279-7280	**GC**
Ohio **Lucent Technologies Ntwk Sys** Columbus Works Room 4G05-2S 6200 East Broad Street Columbus, OH 43212 (1) 614-860-7466	**CB**
Oklahoma **Lucent Technologies Net Sys.** Oklahoma City Works 7725 W. Reno Avenue Third Floor, Col. M2 Oklahoma City, OK 73126-0060 (1) 405-491-4239	**OC**

Manufacturers' Representatives

North America

Alabama

Prime Components, Inc.
4801 University Square Suite 31
Huntsville, AL 35816
(1) 205-837-9668 (phone)
(1) 205-837-9687 (FAX)
attme!attmail!primecomp3(E-mail)
Also represents Mississippi

Arizona

Aztech Component Sales, Inc.
15230 North 75th St., Suite 1031
Scottsdale, AZ 85260
(1) 602-991-6300 (phone)
(1) 602-991-0563 (FAX)
attme!attmail!aztechcomp (E-mail)

Arkansas

Southern States Marketing
23 Inca Lane
Hot Springs Village, AR 71909
(1) 501-922-0807 (phone/FAX)
attme!attmail!ssmark (E-mail)
Also represents N. Louisiana

California

Electec Sales, Inc.
4701 Patrick Henry Drive
Suite 2101
Santa Clara, CA 95054
(1) 408-496-0706 (phone)
(1) 408-727-9617 (FAX)
attme!attmail!esales (E-mail)

Electec SoCal, Inc.
20301 Ventura Blvd Suite 305
Woodland Hills, CA 91364
(1) 818-226-6010 (phone)
(1) 818-226-6019 (FAX)
attme!attmail!esalesla
Los Angeles Office

Electec SoCal, Inc.
15520 Rockfield Blvd., Suite "D"
Irvine, CA 92718
(1) 714-580-0270 (phone)
(1) 714-581-1091 (FAX)
attme!attmail!esalesoc
Orange County Office

Electec Sales San Diego
9666 Business Park Ave
Suite 207
San Diego, CA 92131
(1) 619-635-5959 (phone)
(1) 619-635-5964 (FAX)
attme!attmail!esalessd2

Colorado

Quorum 3
23 Inverness Way East, Suite 120
Englewood, CO 80112
(1) 303-799-8100 (phone)
(1) 303-790-8686 (FAX)
attme!attmail!quorum3 (E-mail)

Florida

Southern States Marketing
600 W. Hillsboro Blvd, Suite 300
Deerfield Beach, FL 33441
(1) 954-725-8679 (phone)
(1) 954-725-8681 (FAX)

Southern States Marketing
101 Southhall Lane, Suite 400
Maitland, FL 32751
(1) 407-667-4858 (phone)
(1) 407-664-4859 (FAX)

Manufacturers' Representatives (continued)

Georgia
Prime Components, Inc.
6825 Jimmy Carter Blvd.
Suite 1890
Norcross, GA 30071
(1) 770-441-3343 (phone)
(1) 770-441-9750 (FAX)
attme!attmail!primecomp2 (E-mail)
Also represents TN

Illinois
Victory Sales Inc.
1030 West Higgins Road, Suite 101
Hoffman Estates, IL 60195
(1) 847-490-0300 (phone)
(1) 847-490-1499 (FAX)
attme!attmail!victorysales (E-mail)

Indiana
Victory Sales Inc.
3077 E. 98th St., Suite 135
Indianapolis, IN 46280
(1) 317-581-0880 (phone)
(1) 317-581-0882 (FAX)
attme!attmail!victorysls (E-mail)

Kansas
MIDTEC Associates, Inc.
11900 W. 87th Street Pkwy.
Suite 220
Lenexa, KS 66215
(1) 913-541-0505 (phone)
(1) 913-541-1729 (FAX)
Also represents IA & NE

Maryland
Advanced Technology Sales
201 International Circle, Suite 570
Hunt Valley, MD 21031
(1) 410-771-0880 (phone)
(1) 410-771-0888 (FAX)
attme!attmail!advancedtech (E-mail)
Also represents E. PA & S. NJ, DE, & VA

Massachusetts
Universal Technology, Inc.
460 Totten Pond Road, 2nd Floor
Waltham, MA 02154
(1) 617-890-8523 (phone)
(1) 617-890-8589 (FAX)
attme!attmail!univtech (E-mail)
Also represents CT, ME, NH, RI, & VT
(Previously Paramount Technology, Inc.)

Minnesota
D.A. Case Associates
4620 West 77th Street, Suite 250
Minneapolis, MN 55435
(1) 612-831-6777 (phone)
(1) 612-831-7076 (FAX)
attme!attmail!dacase (E-mail)
Also represents North and South Dakota

Missouri
MIDTEC Associates, Inc.
55 Westport Plaza
Suite 614
St. Louis, MO 63146
(1) 314-275-8666 (phone)
(1) 314-275-8859 (FAX)
Also represents Central & Southern Illinois

New Jersey
Technical Marketing Group
175-3C Fairfield Avenue
West Caldwell, NJ 07006
(1) 201-226-9408 (phone)
(1) 201-226-9518 (FAX)
attme!attmail!techmarkgrp (E-mail)

New York
Entec Group, Inc.
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
150 Broad Hollow Road, Suite 310
Melville, NY 11747
(1) 516-351-8833 (phone)
(1) 516-351-8667 (FAX)
attme!attmail!techmarkgrp1 (E-mail)

North Carolina
Prime Components, Inc.
805 Spring Forest Road, Suite 800
Raleigh, NC 27609
(1) 919-850-9866 (phone)
(1) 919-850-9867 (FAX)
attme!attmail!primecomp (E-mail)
Represents NC

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)
Represents SC

Prime Components, Inc.
1605 Maple Ridge Court
Greensboro, NC 27455
(1) 910-282-7855 (phone)
(1) 910-282-7853 (FAX)

Manufacturers' Representatives (continued)

Ohio

Victory Sales Inc.
7333 Paragon Rd., Suite 210
Centerville, OH 45458
(1) 513-436-1222 (phone)
(1) 513-436-1224 (FAX)
attme!attmail!victoryohio (E-mail)
Represents S. OH & KY

Victory Sales Inc.
32901 Station St., Suite 104
Solon, OH 44139
(1) 216-498-7570 (phone)
(1) 216-498-7574 (FAX)
attme!attmail!victorysolon
Represents W. PA & N. OH

Oregon

Advanced Technical Marketing Inc.
4900 S.W. Griffith Drive, Suite 155
Beaverton, OR 97005
(1) 503-643-8307 (phone)
(1) 503-643-4364 (FAX)
attme!attmail!advancedtm1
(E-mail)

Pennsylvania

HOTTEC, Inc.
146 Surrey Lane
Harleysville, PA 19438
(1) 215-256-6003 (phone)
(1) 215-256-6444 (FAX)

Texas

Southern States Mktg.
400 E. Anderson Lane, Suite 118
Austin, TX 78752
(1) 512-835-5822 (phone)
(1) 512-835-1404 (FAX)
attme!attmail!ssmaus (E-mail)

Southern States Mktg.
13831 Northwest Freeway,Suite151
Houston, TX 77040
(1) 713-895-8533 (phone)
(1) 713-895-8567 (FAX)
attme!attmail!ssmhou (E-mail)
Also represents S. Louisiana

Southern States Mktg.
1702 North Collins Blvd., Suite 250
Richardson, TX 75080
(1) 214-238-7500 (phone)
(1) 214-231-7662 (FAX)
attme!attmail!ssmdal (E-mail)

Utah

Quorum 3
2981 Holdershill Lane
Salt Lake City, UT 84118-2276
(1) 801-955-7700 (phone)
(1) 801-955-7711 (FAX)
attme!attmail!q3utah (E-mail)

VISCON Corp.
307 W. 200 St., Suite 2001
Salt Lake City, UT 84101
(1) 801-364-9400 (phone)
(1) 801-364-9010 (FAX)
attme!attmail!tdebonis (E-mail)

Virginia

Advanced Technology Sales
13615 Knobhill Ct.
Midlothian, VA 23113
(1) 804-378-5275 (phone)
(1) 804-378-4689 (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)

Wisconsin

Victory Sales Inc.
405 N. Calhoun Road, Suite 208
Brookfield, WI 53005
(1) 414-789-5770 (phone)
(1) 414-789-5760 (FAX)

Manufacturers' Representatives (continued)

Canada

Ontario

J-Squared Technologies, Inc.
300 March Road, Suite 501
Kanata, Ontario, Canada
K2K 2E2
(1) 613-592-9540 (phone)
(1) 613-592-7051 (FAX)
attmail!jtech (E-mail)

J-Squared Technologies, Inc.
3405 American Drive
Building 307, Unit #11
Mississauga, Ontario, Canada
L4V 1T6
(1) 905-672-2030 (phone)
(1) 905-672-2047 (FAX)
attmail!jtechtor (E-mail)

Quebec

J-Squared Technologies, Inc.
1405 Transcanada Highway
Suite 350
Dorval, Quebec, Canada
H9P 2V9
(1) 514-421-7800 (phone)
(1) 514-421-0630 (FAX)
attmail!jtechmont (E-mail)

British Columbia

J-Squared Technologies, Inc.
4170 Still Creek Drive, Suite 200
Burnaby, B.C. Canada
V5C 6C6
(1) 604-473-4666 (phone)
(1) 604-473-4699 (FAX)
attmail!jtechvcr (E-mail)

Europe

Darkhorse Technologies Ltd.
Ellenborough House
Dublin Rd.
NAAS, Co. Kildare
Ireland
(353) 45-871316 (phone)
(353) 45-871315 (FAX)
attme!attmail!micromarket
Pat O'Leary

Peter Sibities Ingenieur Buero
Heckenstallerstrasse 153
D-81369 Munich
Germany
(49) 89-74141166(phone)
(49) 89-716437 (FAX)
attme!attmail!sibimuc
Peter Sibities
FPGA Products Only

HTS di Tiziano Aresi & C
Via Garibaldi, 8/C
I-20090 Buccinasco (MI)
Italy
(39) 2-4883179 (phone)
(39) 2-48843683 (FAX)

Stuart Electronics Ltd.
Phoenix House
Bothwell Road
Castlehill
Carluke ML8 5UF
Scotland
(44) 555-751572 (phone)
(44) 555-75028 (FAX)
attme!attmail!wadd
Archie Waddell

Ultimate Marketing
Highlands House
Grosvenor Rd.
Soldridge, Alton
Hants GU34 5JE
England
(44) 252-781212 (phone)
(44) 252-781313 (FAX)
attme!attmail!ppittman
Phil Pittman

Distributors/Trading Companies

United States

Alabama

Anthem Electronics, Inc.
4920 H. Corporate Drive
Huntsville, AL 35805
(1) 205-890-0302 (phone)
(1) 205-890-0130 (FAX)
Joe Dombrowski

Arrow/Schweber Electronics
1015 Henderson Road
Huntsville, AL 35816
(1) 205-837-6955 (phone)
(1) 205-895-0126 (FAX)
Timothy Carr

Pioneer Standard
4835 University Square, Suite 5
Huntsville, AL 35816
(1) 205-837-9300 (phone)
(1) 205-837-9358 (FAX)
Charlie Hatch

Arizona

Anthem Electronics, Inc.
1553 West 10th Place, Suite 101
Tempe, AZ 85281
(1) 602-966-6600 (phone)
(1) 602-966-4826 (FAX)
Steve Call, Ken Chumley

Arrow/Schweber Electronics
2415 West Erie Drive
Tempe, AZ 85282
(1) 602-431-0030 (phone)
(1) 602-431-9555 (FAX)
Rollie Quissel

California

Anthem Electronics, Inc (Corporate Office)
1160 Ridder Park Drive
San Jose, CA 95131
(1) 408-452-2219 (phone)
(1) 408-441-4504 (FAX)
Karen Garcia, Sue McDonald

Anthem Electronics, Inc.
1 Oldfield Drive
Irvine, CA 92718-2809
(1) 714-768-4444 (phone)
(1) 714-768-6456 (FAX)
Daphne Collins

Anthem Electronics, Inc.
9369 Carroll Park Drive
San Diego, CA 92121
(1) 619-453-9005 (phone)
(1) 619-546-7893 (FAX)
Cliff McCann

Anthem Electronics, Inc.
9131 Oakdale Avenue
Chatsworth, CA 91311
(1) 818-775-1333 (phone)
(1) 818-775-1302 (FAX)
Jill Dimmick, Jeff Polig

Anthem Electronics, Inc.
580 Menlo Drive, Suite 8
Rocklin, CA 95677
(1) 916-624-9744 (phone)
(1) 916-624-9750 (FAX)
Lou Rapa

Arrow/Schweber Electronics
1180 Murphy Avenue
San Jose, CA 95131
(1) 408-441-9700 (phone)
(1) 408-453-4810 (FAX)
Michelle Jacobini/Mark Walker

Arrow/Schweber Electronics
9511 Ridgehaven Court
San Diego, CA 92123
(1) 619-565-4800 (phone)
(1) 619-279-0862 (FAX)
Scott Walker, Lisa Livingston

Arrow/Schweber Electronics
6 Cromwell Street, Suite 100
Irvine, CA 92718
(1) 714-454-4303 (phone)
(1) 714-454-4214 (FAX)
Ellen Vukovich

Arrow/Schweber Electronics
27607 West Agoura Rd.,
Malibu Canyon Bus. Pk.
Calabasas, CA 91302
(1) 818-880-9686 (phone)
(1) 818-880-4687 (FAX)
Eric Skager, Dan Lubick

Zeus Electronics, Inc.
22700 Savi Ranch Parkway
Yorba Linda, CA 92687
(1) 714-581-4622 (phone)
(1) 714-454-4355 (FAX)
Jose Vargas

Zeus Electronics, Inc.
6276 San Ignacio Ave., Suite E
San Jose, CA 95119
(1) 408-629-4789 (phone)
(1) 408-629-4792 (FAX)
Scott Hodges

Pioneer Standard
333 River Oaks Parkway
San Jose, CA 95134
(1) 408-954-9100 (phone)
(1) 408-954-9113 (FAX)
Gregg Jordan

Distributors/Trading Companies (continued)

Colorado

Anthem Electronics, Inc.
373 Inverness Drive
South Englewood, CO 80112
(1) 303-790-4500 (phone)
(1) 303-790-4532 (FAX)
Walter Jurkiewicz

Arrow/Schweber Electronics
101 Inverness Drive East
Englewood, CO 80112
(1) 303-799-0258 (phone)
(1) 303-799-0730 (FAX)
Diane Patrick

Connecticut

Anthem Electronics, Inc.
61 Mattatuck Heights Road
Waterbury, CT 06705
(1) 203-575-1575 (phone)
(1) 203-596-3232 (FAX)
Debbie Boudreau

Arrow/Schweber Electronics
860 N. Main St. Ext.
Wallingford, CT 06492
(1) 203-265-7741 (phone)
(1) 203-265-7988 (FAX)
Linda Adams

Florida

Arrow/Schweber Electronics
400 Fairway Drive
Deerfield Beach, FL 33441
(1) 305-429-8200 (phone)
(1) 305-428-3991 (FAX)
Michelle Alt

Arrow/Schweber Electronics
37 Skyline Drive
Building D, Suites 3101-3
Lake Mary, FL 32746
(1) 407-333-9300 (phone)
(1) 407-333-9320 (FAX)
Libby Bates

Anthem Electronics, Inc.
598 South Northlake Blvd.
Suite #1024
Altamonte Springs, FL 32701
(1) 407-831-0007 (phone)
(1) 407-831-6990 (FAX)
Pat Thompson

Anthem Electronics, Inc.
5200 Northwest 33rd Ave.
Suite 206
Ft. Lauderdale, FL 33309
(1) 305-484-0990 (phone)
(1) 305-484-0951 (FAX)
Jackie Contello

Zeus Electronics, Inc.
37 Skyline Dr.
Building D, Suites 1301-3
Lake Mary, FL 32746
(1) 407-333-3055 (phone)
(1) 407-333-9681 (FAX)
Tyna Young

Pioneer Standard
337 South-North Lake, #1000
Altamonte Springs, FL 32701
(1) 407-834-9090 (phone)
(1) 407-834-0865 (FAX)
Ron Davis

Pioneer Standard
674 South Military Trail
Deerfield Beach, FL 33442
(1) 305-428-8877 (phone)
(1) 305-481-2950 (FAX)
Edie Mittler

Distributors/Trading Companies (continued)

Georgia

Arrow/Schweber Electronics
4250 East River Green Parkway
Duluth, GA 30136
(1) 404-497-1300 (phone)
(1) 404-476-1493 (FAX)
Carolyn Grey

Pioneer Standard
4250 C Rivergreen Parkway
Duluth, GA 30136
(1) 404-623-1003 (phone)
(1) 404-623-0665 (FAX)
Jim Fettigar

Anthem Electronics, Inc.
3305 Breckenridge Blvd.
Suite 108
Duluth, GA 30136
(1) 404-813-0023 (phone)
(1) 404-931-3902 (FAX)
Mike Fowler

Illinois

Anthem Electronics, Inc.
1300 Remington Road, Suite A
Schaumburg, IL 60173
(1) 708-884-0200 (phone)
(1) 708-884-0480 (FAX)
John Emme

Arrow/Schweber Electronics
1140 West Thorndale Avenue
Itasca, IL 60143
(1) 708-250-0500 (phone)
(1) 708-250-0916 (FAX)
John Skurka

Indiana

Arrow/Schweber Electronics
7108 Lakeview Parkway,
West Drive
Indianapolis, IN 46268
(1) 317-299-2071 (phone)
(1) 317-299-2379 (FAX)
Sheila Randolph

Kansas

Arrow/Schweber Electronics - Capstone
9801 Legler Road
Lenexa, KS 66214
(1) 913-541-9542 (phone)
(1) 913-752-2612 (FAX)
Barbara Wightman

Pioneer Standard
15810 Gaither Drive
Gaithersburg, MD 20877
(1) 301-921-8407 (phone)
(1) 301-921-3852 (FAX)
Stacy Kolper

Maryland

Anthem Electronics, Inc.
7168 Columbia Gateway Drive,
Suite A
Columbia, MD 21046
(1) 410-995-6640 (phone)
(1) 410-290-9862 (FAX)
Maryanne Hossick

Pioneer Standard
9100 Gaither Road
Gaithersburg, MD 20877
(1) 301-921-0660 (phone)
(1) 301-921-4255 (FAX)
Stacie Dias

Arrow/Schweber Electronics
9800J Patuxent Woods Drive
Columbia, MD 21046
(1) 301-596-7800 (phone)
(1) 301-596-7821 (FAX)
Janine Franey

Massachusetts

Anthem Electonics, Inc.
200 Research Dr.
Wilmington, MA 01887
(1) 508-657-5170 (phone)
(1) 508-657-6008 (FAX)
Rose Lecointre

Arrow/Schweber Electronics
25 Upton Drive
Wilmington, MA 01887
(1) 508-658-0900 (phone)
(1) 508-694-1754 (FAX)
Stew Dick

Zeus Electronics, Inc.
25 Upton Dr.
Wilmington, MA 01887
(1) 508-658-4776 (phone)
(1) 508-694-2119 (FAX)
Eric Lund

Pioneer Standard
44 Hartwell Ave.
Lexington, MA 02173
(1) 617-861-9200 (phone)
(1) 617-863-1547 (FAX)
Steve Smith, Janet Barnard

Distributors/Trading Companies (continued)

Michigan

Arrow/Schweber Electronics
44720 Helm St.
Plymouth, MI 48172
(1) 313-462-2290 (phone)
(1) 313-462-2686 (FAX)
Linda Pendergrass

Minnesota

Arrow/Schweber Electronics
10120A West 76th St.
Eden Prairie, MN 55344
(1) 612-946-4800 (phone)
(1) 612-946-4835 (FAX)
Leslie Gerlach

Anthem Electronics, Inc.
7690 Golden Triangle Drive
Eden Prairie, MN 55344
(1) 612-944-5454 (phone)
(1) 612-944-3045 (FAX)
Barb Romanelli

Arrow/Schweber Electronics
10100 Viking Drive, Suite 100
Eden Prairie, MN 55344
(1) 612-941-5280 (phone)
(1) 612-941-9405 (FAX)
Sara Lindsay

New Jersey

Anthem Electronics, Inc.
26 Chapin Road, Unit K
Pine Brook, NJ 07058
(1) 201-227-7960 (phone)
(1) 201-227-9246 (FAX)
Pete Kahler, Helen Joy

Arrow/Schweber Electronics
4 East Stow Road, Unit 11
Marlton, NJ 08053
(1) 609-596-8000 (phone)
(1) 609-596-9632 (FAX)
Linda Bruckner

Arrow/Schweber Electronics
43 Route 46 East
Pine Brook, NJ 07058
(1) 201-227-7889 (phone)
(1) 201-227-2064 (FAX)

Arrow/Schweber Electronics
101 Crawfords Corner Road,
Room 1M511
Holmdel, NJ 07733-3030
(1) 908-949-4700 (phone)
(1) 908-949-4035 (FAX)
Jennifer Campbell

New York

Anthem Electronics, Inc.
47 Mall Drive
Commack, NY 11725
(1) 516-864-6600 (phone)
(1) 516-864-2244 (FAX)
Wendy Anselmo

Arrow/Schweber Electronics
20 Oser Avenue
Hauppauge, NY 11788
(1) 516-231-1000 (phone)
(1) 516-231-1072 (FAX)
John Dowling

Arrow/Schweber Electronics
3375 Brighton-Henrietta,
Townline Road
Rochester, NY 14623
(1) 716-427-0300 (phone)
(1) 716-427-0735 (FAX)
Krista Olafsen

Arrow/Schweber Electronics
25 Hub Drive
Melville, NY 11747
(1) 516-391-1300 (phone)
(1) 516-391-1707 (FAX)
Steve Mussmacher, Sue Newton

Zeus Electronics, Inc.
100 Midland Ave.
Port Chester, NY 10573
(1) 914-933-4200 (phone)
(1) 914-937-2553 (FAX)
Dan Tiesi

Distributors/Trading Companies (continued)

North Carolina

Arrow/Schweber Electronics
5240 Greens Dairy Road
Raleigh, NC 27604
(1) 919-876-3132 (phone)
(1) 919-878-9517 (FAX)
Marcie Flansburg

Pioneer Standard
2200 Gateway Centre Boulevard
Suite 215
Morrisville, NC 27560
(1) 919-460-1530 (phone)
(1) 919-460-1540 (FAX)
Karen Warkany

Ohio

Arrow/Schweber Electronics
6573 East Cochran Road
Solon, OH 44139
(1) 216-248-3990 (phone)
(1) 216-248-1106 (FAX)
Deb Cramer

Arrow/Schweber Electronics
8200 Washington Village Drive,
Suite A
Centerville, OH 45458
(1) 513-435-5563 (phone)
(1) 513-435-2049 (FAX)
Kate Spraley

Pioneer Standard
4433 Interpoint Blvd.
Dayton, OH 45424
(1) 513-236-9900 (phone)
(1) 513-236-8133 (FAX)
Ted Smith

Oklahoma

Arrow/Schweber Electronics
12111 East 51st Street, Suite 101
Tulsa, OK 74146
(1) 918-252-7537 (phone)
(1) 918-254-0917 (FAX)
Elaine Chambers

Oregon

Anthem Electronics, Inc
9090 S.W. Gemini Drive
Beaverton, OR 97005
(1) 503-643-1114 (phone)
(1) 503-626-7928 (FAX)
Nancy Howard

Almac/Arrow Electronics Corp.
1885 Northwest 169th Place
Beaverton, OR 97006-7312
(1) 503-629-8090 (phone)
(1) 503-645-0611 (FAX)
Melissa Shute

Pennsylvania

Anthem Electronics, Inc.
355 Business Center Drive,
Suite C
Horsham, PA 19044
(1) 215-443-5150 (phone)
(1) 215-675-9875 (FAX)
Ralph Eisenschmid

Arrow/Schweber Electronics
2681 Mosside Blvd.
Suite 204
Monroeville, PA 15146
(1) 412-856-9490 (phone)
(1) 412-856-9507 (FAX)
Tom Batson (Cleveland Area)

Pioneer Standard
500 Enterprise Rd.
Horsham, PA 19044
(1) 215-674-4000 (phone)
(1) 215-674-3107 (FAX)
Ginger Schorn

Texas

Anthem Electronics, Inc
651 North Plano Road, Suite 429
Richardson, TX 75081
(1) 214-238-7100 (phone)
(1) 214-238-0237 (FAX)
Walt Beroza

Arrow/Schweber Electronics
3220 Commander Drive
Carrollton, TX 75006
(1) 214-380-6464 (phone)
(1) 214-248-7208 (FAX)
Kristin Muckelroy

Arrow/Schweber Electronics
19416 Park Row, Suite 190
Houston, TX 77084
(1) 713-530-4700 (phone)
(1) 713-568-8518 (FAX)
Dave Salinas

Arrow/Schweber Electronics
11500 Metric Blvd., Suite 160
Austin, TX 78758
(1) 512-835-4180 (phone)
(1) 512-832-9875 (FAX)
Mary Alice Castello

Anthem Electronics, Inc.
14050 Summit Drive, Suite 119
Austin, TX 78728
(1) 512-388-0049 (phone)
(1) 512-388-0271 (FAX)
Debbie Hamm

Pioneer Standard
1826-D Kramer Lane
Austin, TX 78758
(1) 512-835-4000 (phone)
(1) 512-835-9829 (FAX)

Pioneer Standard
13765 Beta Road
Dallas, TX 75244
(1) 214-419-5500 (phone)
(1) 214-419-6419 (FAX)
Ted Smith

Zeus Electronics, Inc.
3220 Commander Dr.
Carrollton, TX 75006
(1) 214-380-4330 (phone)
(1) 214-447-2222 (FAX)
Charles Pierce

Distributors/Trading Companies (continued)

Utah

Anthem Electronics, Inc.
2440 So. 1070 West, Suite A
Salt Lake City, UT 84119
(1) 801-973-8555 (phone)
(1) 801-973-8909 (FAX)
Dianne Suggett

Arrow/Schweber Electronics
1946 West Parkway Boulevard
Salt Lake City, UT 84119
(1) 801-973-6913 (phone)
(1) 801-972-0200 (FAX)
Bill Entwistle

Washington

Almac Electronics Corporation (Hdqts.)
14360 SE Eastgate Way
Bellevue, WA 98007
(1) 206-643-9992 (phone)
(1) 206-643-9709 (FAX)
Keith Hanson

Anthem Electronics, Inc.
19017 120th Avenue North East
Suite 102
Bothell, WA 98011
(1) 206-483-1700 (phone)
(1) 206-486-0571 (FAX)
Jim Graham

Pioneer Standard
2800 156th Ave. S.E. Group
Suite 100
Bellevue, WA 98007
(1) 206-644-7500 (phone)
(1) 206-644-7300 (FAX)
Gary Pilon

Wisconsin

Arrow/Schweber Electronics
200 North Patrick Boulevard
Suite 100
Brookfield, WI 53045
(1) 414-792-0150 (phone)
(1) 414-792-0156 (FAX)
Brenda Guagliardo

Canada

Ontario

Arrow/Schweber Electronics
1093 Meyerside Drive
Mississauga, Ontario L5T 1M4
(1) 905-670-7769 (phone)
(1) 905-670-5009 (FAX)
Keith Maccrimmon

Arrow/Schweber Electronics
36 Antares Drive, Unit 100
Nepean, Ontario K2E 7W5
(1) 613-226-6903 (phone)
(1) 613-723-2018 (FAX)
Jordan Houghton

Quebec

Arrow/Schweber Electronics
1100 St. Regis Boulevard
Dorval, Quebec H9P 2T5
(1) 514-421-7411 (phone)
(1) 514-421-7430 (FAX)
Tony Meyer

British Columbia

Arrow/Schweber Electronics
8544 Baxter Place
Burnaby, B.C. V5A 4T8
(1) 604-421-2333 (phone)
Marti Garisto

Distributors/Trading Companies (continued)

Europe

ATD Electronica LDA
Quinta Grande
Lote 20-R/C DTO.
Alfragide (Norte)
2700 Amadora
Portugal
(351) 1-4714-182/575 (phone)
(351) 1-4715-886 (FAX)
attme!attmail!fax!35114715886
Rafael Viela

ATD Electronica SA
ALBASANS, 75
28037
Madrid
Spain
(34) 1304-1534 (phone)
(34) 1327-2778 (FAX)
attme!attmail!fax!3416616300
Manuel Garcia

Berendsen Electronics
Konowsgt. 8
P.O. Box 9376
0135 Oslo
Norway
(47) 22-67-6800(phone)
(47) 22-67-7380 (FAX)
attme!attmail!odinelec
Jon Arne Kjaerstad

Berendsen Electronics
Ralssintie 6
FIN-00720 Helsinki
Finland
(35)8-0351-5100 (phone)
IC & Optoelectronics Only

Berendsen Electronics
Hammarby Kajvag 14
S-12006 Stockholm
Sweden
(46) 08615 7402 (phone)
IC & Optoelectronics Only

Bytech Components Ltd.
12A Cedarwood
Chineham Business Park
Crockford Lane
Basingstoke, Hampshire
RG 24 OWD
England
(44) 256-707107 (phone)
(44) 256-707162 (FAX)
attme!attmail!bytech
Colin Street

C-88 AS
101 Kokkedal Industripark
DK-2980 Kokkedal
Denmark
(45) 42-244-888 (phone)
(45) 42-244-889 (FAX)
attme!attmail!mermaid
Arne Jappe
Gert Rasmussen

Compress
30, Rue de L'Esrerel
94633 RUNGIS CEDEX
France
(33) 1-46878020 (phone)
(33) 1-46866763 (FAX)
attme!attmail!compscib
Francoise Marx
Robert Fogliani

Datacomp AG
Silbernstrasse 10
CH-8953 Dietikon
Switzerland
(41) 1-7405140 (phone)
(41) 1-7413423 (FAX)
attme!attmail!dtcmp
Peter Furrer

EURODIS Texim
Avenue des Croix de Guerre 116
B-1120 Brussels
Belgium
(32) 2-2474969 (phone)
(32) 2-2158102 (FAX)
attme!attmail!inelco
Eric Grussen

EURODIS TME Components BV
Nijverheidsstraat 16
P.O. Box 172
NL-7480 AD Haaksbergen
The Netherlands
(31) 5427-33-333 (phone)
(31) 5427-33-888 (FAX)
attme!attmail!tmenlwim
Wim Terlingen

I&R Electronic
Ziherlova Ulica 2
61000 Ljubljana
Slovenia
(38) 661-222-007 (phone)
(38) 661-224-111 (FAX)
Aleksander Razmovski

Inter Muehendislik
Danismanlik Ve Ticaret A.S.
Kadikoey Hasircibasi CAD. No. 55
TR-81310 Istanbul
Turkey
(90) 216-3499400 (phone)
(90) 216-3382823 (FAX)
attme!attmail!fax!902163382823
Ecruement Yagmur

Distributors/Trading Companies (continued)

Europe (continued)

Jermyn GmbH
Im Dachsstueck 9
D-65549 Limburg
Germany
(49) 6431-5080 (phone)
(49) 6431-508289 (FAX)
attme!attmail!jermynd
Jochem Pelzer

Eurodis Bytech, Ltd.
12 A Cedarwood
Chineham Business Park
Crockford Lane
Basingstoke, Hants RG12 0WD
England
(44) 1256 707107 (phone)
(44) 1256 707162 (FAX)
Mark Dyne

Macro
Burnham Lane
Slough SL1 6LN
England
(44) 628-606000 (phone)
(44) 628-666873 (FAX)
attme!attmail!macrofpga
attme!attmail!macro1
Elaine Thomson

Nordcomp Finland OY
Asemakuja 2
SF-02770 ESPOO
Finland
(35) 8-0859-3699 (phone)
(35) 8-0859-3644 (FAX)
attme!attmail!nordcomp
Petri Jarvinen

Semiconductores Investigacion y Diseno SA
Edificio "Centro de Encuentros"
Isaac Newton, s/n
28760 Tres Cantos (Madrid)
Spain
(34) 1 8035052 (phone)
(34) 1 8039557 (FAX)
attme!attmail!fax!34917423681
Jose Maria/Insenser Farre
FPGA Products Only

Lasi Electronica SPA
Viale Fulvio Testi, 280
I-20126 Milan
Italy
(39) 2-661-431 (phone)
(39) 2-66-101385 (FAX)
attme!attmail!fax!39266101385
Roberto Pirotta

NC Nordcomp Sweden AB
P.O. Box 4115
Hemvaernsgaten 11
S-171 04 Solna
Sweden
(46) 8-985140 (phone)
(46) 8-7645451 (FAX)
attme!attmail!naxab
Ingvar Larsson

Vectronics, Ltd.
Maskit 6 St.
P.O. 2024
Herzlia B' 46120
Israel
(972) 9-556070 (phone)
(972) 9-556508 (FAX)

Steiner Electronic Vertrieb GmbH
Egererstrasse 18
A-3013 Tullnerbach
Austria
(43) 2233-553660 (phone)
(43) 2233-55360 (FAX)
attme!attmail!steiner
Karl Kadiera/Sven Hendrich

Distributors/Trading Companies (continued)

Japan

Lucent Semiconductor Marketing Ltd.
7-18 Higashi-Gotanda 2-chome
Shinagawa-ku, Tokyo 141
Japan
(81) 3-5421-1777 (phone)
(81) 3-5421-1785 (FAX)

TOMEN Electronics Corp.
1-1, Uchisaiwaicho 2-chome
Chiyoda-ku, Tokyo 100
Japan
(81) 3-3506-3650 (phone)
(81) 3-3506-3497 (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) 3-3779-7800 (FAX)

Mitsui & Co., Ltd
2-1 Ohtemachi 1-chome
Chiyoda-ku, Tokyo 100
Japan
(81) 3-3285-4067 (phone)
(81) 3-3285-9868 (FAX)

Sumisho Electronic Devices Corporation
Eiseniwamotocho Building
2-8-8 Iwamotocho
Chiyoda-ku, Tokyo 101
Japan
(81) 3-3863-8200 (phone)
(81) 3-3863-8211 (FAX)

Japanese Semiconductor Marketing Subdistributors

Ryosan Co, Ltd.
Sumitomo-seimei Bldg.
5-15, Higashikanda 2-chome
Chiyoda-ku, Tokyo 101
Japan
(81) 3-3862-2591 (phone)

Sanshin Electronics Co, Ltd.
4-12, Shiba 4-chome
Minato-ku, Tokyo 108
Japan
(81) 3-3453-5111 (phone)

Shiko Shoji Co, Ltd.
1-5, Meguro 1-chome
Meguro-ku, Tokyo 153
Japan
(81) 3-5721-2111 (phone)

Satori Electric Co, Ltd.
14-10, Shiba 1-chome
Minato-ku, Tokyo 105
Japan
(81) 3-3452-7171 (phone)

Mikasa Shoji Co, Ltd.
4-12, Shimamachi 2-chome
Chuo-ku, Osaka 540
Japan
(81) 6-946-3781 (phone)

K. K. Shimizu Syntec
7-1,Musashimachi, Kanazawa-shi
Ishikawa 920
Japan
(81) 762-23-2221 (phone)

Nakamatsu Shokai K.K.
3-4, Kajicho 2-chome
Chiyoda-ku, Tokyo 101
Japan
(81) 3251-9541 (phone)

K. K. Com
Asakaze 2-gokan Bldg.
16-9, Sotokanda 1-chome
Chiyoda-ku, Tokyo 101
Japan
(81) 3-3255-4571 (phone)

Nikko Telecommunications Co, Ltd.
Banzai Bldg.
31-19, Shiba 2-chome
Minato-ku, Tokyo 105
Japan
(81) 3-3451-2111 (phone)

Hagiwara Electric Co, Ltd.
3-3, Higashisakura 2-chome
Higashi-ku, Nagoya-shi 461
Japan
(81) 52-931-3511 (phone)

Numata Corporation
1-20, Minamihorie 4-chome
Nishi-ku, Osaka 550
Japan
(81) 6-532-2491 (phone)

Taiyo Denki Tsushin K.K.
Shobunkan Bldg.
49, Higashi Kataha-cho.
Higashi-ku, Nagoya 461
Japan
(81) 52-931-0231 (phone)

Distributors/Trading Companies (continued)

Asia/Pacific

Taiwan-Taipei

Jedcom Ltd.
12F, No. 232 Sec. 3
Cheng-Teh Road
Taipei, Taiwan
(886) 2-5856661 (phone)
(886) 2-5864736 (FAX)

Seraphim Engineering Co., Ltd.
6th Floor, No. 34-1
Chiu-Chuan Street
Taipei, Taiwan, R.O.C.
(886) 2-5935195 (phone)
(886) 2-5949424 (FAX)

Maxtek Technology Co Ltd.
3rd Floor No. 197 Sec. 4
Nanking E. Road
Taipei, Taiwan
(886) 2-7130209 (phone)
(886) 2-7126780 (FAX)

Singapore

Serial System Pte. Ltd.
11 Jalan Mesin #06-00
Standard Industrial Building
Singapore 368813
(65) 2800200 (phone)
(65) 2861812 (FAX)

Astina Electronics
(5) Pte. Ltd.
203B Henderson Road
#12-08 Henderson Industrial Park
Singapore 159548
(65) 2769997 (phone)
(65) 2769996 (FAX)

Westech Electronics Pte., Ltd.
34 Kaki Bukit Crescent
Kaki Bukit Techpark
Singapore 416623
(65) 7436355 (phone)
(65) 7431403 (FAX)

Korea

Douil Inc.
1303 KMIC Building
168-9 Yeomli-Dong
Mapo-Gu Seoul 121-090
Korea
(82) 2-7127071 (phone)
(82) 2-7180817 (FAX)

Excel Tech
#398-14 Seokyo-Dong
Mapo-Gu Seoul 121-210
Korea
(82) 2-3357823/4 (phone)
(82) 2-3357825 (FAX)

Australia

Zatek Australia Pty Ltd.
Level 4, 5 Belmore St.
Durwood NSW 2134
Sydney, Australia
(61) 2-97445711 (phone)
(61) 2-97445527 (FAX)

Hong Kong

Tektron Electronics (HK) Ltd.
1702 Bank Centre
636 Nathan Road
Kowloon, Hong Kong
(852) 23880629 (phone)
(852) 27805871 (FAX)

Notes

For additional information, contact your Microelectronics Group Account Manager or the following:

❑ Internet
http://www.lucent.com/micro/fpga

❑ Microelectronics Group
Lucent Technologies Inc.
555 Union Boulevard
Room 30L-15P-BA
Alientown, PA 18103
1-800-372-2447
FAX 1-610-712-4106
E-Mail docmaster@micro.lucent.com

In Canada, call:
1-800-553-2448
FAX 1-610-712-4106

❑ Microelectronics Group Asia/Pacific
Lucent Technologies Singapore Pte. Ltd.
77 Science Park Drive
#03-18 Cintech III
Singapore 118256
Tel. (65) 778 8833
FAX (65) 777 7495

❑ Microelectronics Group
Lucent Technologies Japan Ltd.
7-18, Higashi-Gotanda 2-chome
Shinagawa-ku, Tokyo 141
Japan
Tel. (81) 3 5421 1600
FAX (81) 3 5421 1700

For data requests in Europe:

❑ Microelectronics Group Dataline
Tel. (44) 1734 324 299
FAX (44) 1734 328 148

For technical inquiries in Europe:

❑ Central Europe: **(49) 89 95086 0** (Munich)
Northern Europe: **(44) 1344 865 900** (Bracknell UK)
France: **(33) 1 47 67 47 67** (Paris)
Southern Europe: **(39) 2 6601 1800** (Milan)
(34) 1 807 1700 (Madrid)

October 1996
MN96-028FPGA (Replaces MN95-001FPGA and/or MN95-056FPGA)

microelectronics group

Lucent Technologies
Bell Labs Innovations